

Dissertation zur Erlangung des Doktorgrades  
der Fakultät für Chemie und Pharmazie  
der Ludwig-Maximilians-Universität München

# **Visualization and exploration of next-generation proteomics data**

Eugenia Voytik  
(geb. Baranova)

aus

Gomel, Belarus

**2022**

## **Erklärung**

Diese Dissertation wurde im Sinne von §7 der Promotionsordnung vom 28. November 2011 von Herrn Professor Dr. Matthias Mann betreut.

## **Eidesstattliche Versicherung**

Diese Dissertation wurde eigenständig und ohne unerlaubte Hilfe erarbeitet.

München, den 04.08.2022

---

Eugenia Voytik

Dissertation eingereicht am 22.08.2022

1. Gutachter: Prof. Dr. Matthias Mann

2. Gutachter: Dr. Stefan Canzar

Mündliche Prüfung am 28.09.2022

## Summary

Over the last decades, the rapid advances in the field of mass spectrometry (MS)-based proteomics have turned it into one of the most powerful technologies available for studying biological systems. This technology has led to the identification and quantification of thousands of different proteins and their modifications under various biological conditions. The recent introduction of trapped ion mobility spectrometry in proteomics – exemplified by the timsTOF instrument and providing an additional dimension of ion separation - has further extended capabilities, making it even more high-throughput and sensitive. As a result, this ‘next-generation proteomics’ is generating huge amounts of data on a daily basis that provide information to study complex biological problems, such as mechanisms controlling cell signalling or cellular heterogeneity in health and disease. However, the visualization and further exploration of this big data has not kept pace with its generation, which has posed problems in critically validating and interpreting the data. I therefore address this challenge in my thesis.

Looking through the course of my PhD studies in chronological order, when I started working in the group there were no software tools to efficiently access and visualize timsTOF data. The slow and inconvenient access to the extremely large timsTOF proteomics data was at that time a major limitation and also a foundation for my further projects. The next step was therefore the joint development of a software tool called AlphaTims, which efficiently indexed the next-generation proteomics data and drastically accelerated the data access (Article 4). This project, like many of the following ones, was based on the development in our department of a novel open-source Python-based framework for efficient processing of large scale, high-resolution MS data sets called AlphaPept (Article 5). This framework has become an ‘ecosystem’ for proteomics software development, not only providing the necessary functionality, but also incorporating the basis of scientific software development standards, such as high-quality code, extensive documentation, automated testing, and continuous integration. As a next step, I co-developed a tool called AlphaMap to facilitate the visual inspection of the peptide-level proteomics data with post-translational modifications (PTMs) (Article 3). Finally, to simplify the validation of the next-generation proteomics data acquired on the timsTOF instrument, I developed AlphaViz, an open-source Python-based visualization tool that allows the user to examine the validity of peptide identification and quantification by visually comparing them to the signal presented in the raw timsTOF data (Article 2).

Deviating from standard scientific software development, during my PhD studies I also had the opportunity to participate in collaborative projects covering areas such as method

development and deep learning prediction of peptide properties. Early in my PhD I contributed to the implementation of a novel diaPASEF scanning mode on the timsTOF instrument, where we demonstrated up to 100% ion utilization for fragmentation and achieved deep proteome coverage of more than 7,000 proteins in only two-hour HeLa runs (Article 6). Two years later, we were able to refine this method, now quite favored in the field, by optimally positioning the quadrupole isolation windows and gaining 14% coverage of the peptide population and almost 60% for phosphopeptides (Article 7). The ability to acquire very large timsTOF data sets with collisional cross section (CCS) values, enabled us to investigate for the first time the general nature of CCS values for peptides. We then trained a deep learning model that predicts them with high accuracy (median deviation of 1.4%) (Article 8). We found that CCS values correlate with known physical peptide properties, such as mass and bulkiness, but have large variance depending on the specific context in the peptide sequence. A year after that, we introduced a new highly modular deep learning system called AlphaPeptDeep, which allows us to predict with very high confidence all sequence related peptide properties using the same system, and to easily build and train custom deep learning models for any project in just a few lines of code (Article 9).

Altogether, the work represented in this thesis focuses on the exploration of different aspects of 'next-generation proteomics' timsTOF data, ranging from the development of scientific software to access, process or visualize this new type of complex multidimensional MS proteomics data, through MS-proteomics method development and finally the application of emerging artificial intelligence technologies, such as deep learning, in proteomics.



## Table of Contents

|   |     |
|---|-----|
| Summary .....   | iii |
| Abbreviations .....   | vii |
| 1. Introduction .....   | 1   |
| 1.1. Mass spectrometry-based proteomics .....   | 1   |
| 1.1.1. Bottom-up proteomics .....   | 2   |
| 1.1.2. Sample preparation .....   | 3   |
| 1.1.3. Liquid chromatography – mass spectrometry (LC-MS) .....  | 5   |
| 1.1.4. Proteomics data analysis .....   | 7   |
| 1.2. Ion mobility spectrometry in proteomics .....  | 10  |
| 1.2.1. The TIMS-TOF instrument .....  | 11  |
| 1.2.2. PASEF principle .....  | 13  |
| 1.2.3. PASEF acquisition strategies .....   | 15  |
| 1.3. Post-translational modifications in MS-based proteomics .....  | 17  |
| 1.3.1. Phosphorylation .....  | 19  |
| 1.3.2. Protein phosphorylation in cell signaling .....  | 20  |
| 1.3.3. Phosphoproteomics and its challenges .....   | 21  |
| 1.4. Scientific software development and open science .....   | 23  |
| 1.5. Visualization in MS-based proteomics .....   | 25  |
| 1.5.1. Raw data visualization .....   | 27  |
| 1.5.2. Peptide and PTM visualization .....  | 30  |
| 2. Aims of the thesis .....   | 32  |
| 3. Publications .....   | 34  |
| 3.1. Article 1: A practical guide to interpreting and generating bottom-up proteomics data visualizations .....         | 34  |
| 3.2. Article 2: AlphaViz: Visualization and validation of critical proteomics data directly at the raw data level ..... | 53  |

|   |     |
|---|-----|
| 3.3. Article 3: AlphaMap: an open-source Python package for the visual annotation of proteomics data with sequence-specific knowledge ..... | 87  |
| 3.4. Article 4: AlphaTims: Indexing Trapped Ion Mobility Spectrometry–TOF Data for Fast and Easy Accession and Visualization.....           | 92  |
| 3.5. Article 5: AlphaPept, a modern and open framework for MS-based proteomics .....  | 104 |
| 3.6. Article 6: diaPASEF: parallel accumulation – serial fragmentation combined with data-independent acquisition .....                     | 129 |
| 3.7. Article 7: Rapid and in-depth coverage of the (phospho-)proteome with deep libraries and optimal window design for dia-PASEF .....     | 140 |
| 3.8. Article 8: Deep learning the collisional cross sections of the peptide universe from a million experimental values .....               | 166 |
| 3.9. Article 9: AlphaPeptDeep: A modular deep learning framework to predict peptide properties for proteomics .....                         | 179 |
| 4. Discussion .....   | 208 |
| 5. References .....   | 212 |
| Appendix .....  | 221 |
| Acknowledgements .....  | 234 |

## Abbreviations

|       |   |
|-------|---|
| ADP   | adenosine 5'-diphosphate                            |
| ATP   | adenosine 5'-triphosphate                           |
| BPI   | base peak chromatogram                              |
| CAA   | chloroacetamide                                     |
| CCS   | collisional cross section                           |
| CID   | collision-induced dissociation                      |
| CPU   | central processing unit                             |
| DDA   | data-dependent acquisition                          |
| DIA   | data-independent acquisition                        |
| DNA   | deoxyribonucleic acid                               |
| DOI   | digital object identifier                           |
| DTIMS | drift tube ion mobility spectrometry                |
| DTT   | dithiothreitol                                      |
| EGF   | epidermal growth factor                             |
| EGFR  | epidermal growth factor receptor                    |
| ESI   | electrospray ionization                             |
| ETD   | electron-transfer dissociation                      |
| FAIMS | field asymmetric waveform ion mobility spectrometry |
| FASP  | filter-aided sample preparation                     |
| FDR   | false discovery rate                                |
| GPU   | graphics processing unit                            |
| GUI   | graphical user interface                            |
| HCD   | higher-energy collisional dissociation              |
| HLA   | human leukocyte antigen                             |
| HPLC  | high-performance liquid chromatography              |
| IAA   | iodoacetamide                                       |
| IM    | ion mobility  |

|              |  |
|--------------|--|
| IMAC         | immobilized metal affinity chromatography    |
| LC           | liquid chromatography                        |
| MALDI        | matrix-assisted laser desorption ionization  |
| MOAC         | metal oxide affinity chromatography          |
| MS           | mass spectrometry                            |
| MS/MS or MS2 | tandem MS                                    |
| PASEF        | parallel accumulation – serial fragmentation |
| PRM          | parallel reaction monitoring                 |
| PSM          | peptide-spectrum match                       |
| PTM          | post-translational modification              |
| RAM          | random-access memory                         |
| RNA          | ribonucleic acid                             |
| RT           | retention time                               |
| SCX          | strong cation exchange                       |
| SDC          | sodium deoxycholate                          |
| SDS          | sodium dodecyl sulfate                       |
| TCEP         | tris(2-carboxyethyl)phosphine                |
| TIC          | total ion current                            |
| TIMS         | trapped ion mobility spectrometry            |
| TOF          | time-of-flight                               |
| TWIMS        | traveling wave ion mobility spectrometry     |
| XIC          | extracted ion chromatogram                   |

# 1. Introduction

## 1.1. Mass spectrometry-based proteomics

Proteins are key components of all living organisms. They provide structure, transport other molecules, catalyze reactions, transmit signals and, in fact, execute or at least participate in every biological process. Although all cells within an organism generally contain the same genotype throughout their life cycle, their phenotype changes over time and between cells, tissues and organs. These alterations occur as a result of gene regulation, the process of protein translation, protein modification and localization as well as interaction between proteins in protein complexes. For the analysis of proteins on a global level, the term proteomics was coined several decades ago by M. Wilkins to describe the entire complement of proteins expressed in a specific state of a cell population, a tissue, an organ or an organism (1). It was used in analogy to other terms, such as genomics and transcriptomics, that already exist in the field, to refer to the sequencing of the complete deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) repertoires, respectively.

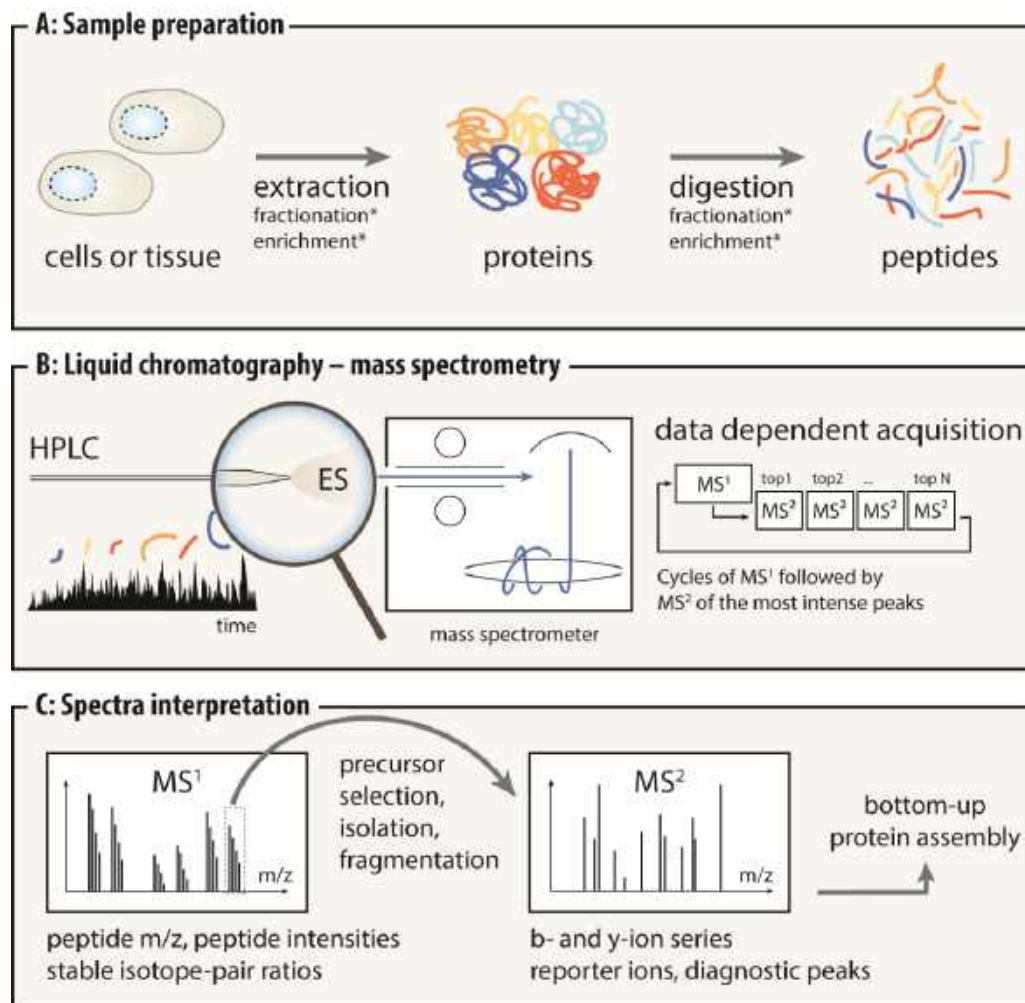
However, to compete with the speed and efficacy of analytical methods used for oligonucleotide sequences, accurate large-scale proteomics had to evolve from the qualitative analysis of a single isolated protein towards the robust high-throughput quantitative analysis of complex protein mixtures containing thousands of proteins. Note that proteomics is here taken to mean mass spectrometry (MS)-based proteomics and not large-scale antibody-based methods. Three milestone inventions played a major role even before and then in this transformation process (2, 3). Firstly, Joseph Thomson is considered the father of MS for his work in the discovery of the electron, which earned him the Nobel Prize in 1906, as well as for building the first mass spectrometer (3). The second was the discovery of the quadrupole and the three-dimensional ion trap by Wolfgang Paul who won a share of the physics Nobel Prize in 1989. His first device was the ancestor of many of the commercial mass spectrometers available today. The third was the almost simultaneous invention of two key soft ionization methods, electrospray ionization (ESI) by the team of John Fenn and matrix-assisted laser desorption ionization (MALDI) by two different groups. These two technologies were awarded part of the chemistry Nobel Prize in 2012 (4–6). They allowed biological macromolecules to be transferred from the liquid or solid matrix, respectively, to the gas phase for further analysis by mass spectrometry (MS). These advances have extended the application of MS to large molecules and opened up a new field to study biological systems and gradually made it the preferred method of proteomics analysis.

From these early days to the time when the complete proteomes of many species could be analyzed in a qualitative and quantitative manner (7), a huge number of discoveries and developments related to the various steps of the proteomics workflow had to take place. These ranged from the standardization and miniaturization of sample preparation protocols (8, 9), innovative columns and liquid chromatography systems (10), to new acquisition schemes (11–15), next-generation mass spectrometer types (16, 17) and software development (18–22).

In order to streamline the description of the methods that follow, I next introduce some important MS concepts and terminology that will be used in this thesis.

### 1.1.1. Bottom-up proteomics

MS-based proteomics can use different types of input material, ranging from cell lines, tissues and organs to entire microorganisms. First cells from the sample are lysed, proteins extracted and alkylated. Then they are enzymatically digested to peptides using sequence-specific enzymes such as trypsin, and are then further analyzed by a combination of analytical techniques (Fig. 1A). This approach, based on the digestion of proteins to peptides for further analysis, is referred to as “bottom-up” or “shotgun” proteomics, in contrast to the “top-down” protein-based approach, which aims at the analysis of entire proteins without prior digestion (23). The resulting peptide mixture is further separated by an aqueous/organic solvent gradient in the liquid chromatography (LC) step and ionized via ESI (Fig. 1B). Coordinated with the elution from the column, the mass spectrometer scans the entire mass range ( $MS^1$  level) every few seconds and, based on the preferred acquisition strategy, isolates and fragments only a list of pre-selected peptides (targeted approach) (24), the topN most intense precursors (data-dependent acquisition, or DDA) or all peptides falling within a certain  $m/z$  window (data-independent acquisition, or DIA) (25). This fragmentation process is called tandem MS (MS/MS). The information of the peptide masses together with their fragment masses is used for peptide identification and quantification by database searching (Fig. 1C). In the final step, the identified peptide sequences are assembled into a set of proteins while solving the protein inference problem (26). The three main steps of the classical bottom-up MS-based proteomics workflow, comprising sample preparation (A), LC-MS/MS analysis (B), and data analysis (C), shown in Figure 1 (27), are described in more detail in the following sections.



**Figure 1: Shotgun or bottom-up proteomics workflow. (A) Sample preparation.** In this step, proteins are extracted from cells or tissue and enzymatically digested into peptides. Additional enrichment and fractionation steps can be applied at the protein or peptide level to increase proteome coverage. **(B) Liquid chromatography – mass spectrometry.** Peptides are separated by a high-performance liquid chromatography (HPLC) system and ionized by electrospray (ESI) for subsequent mass spectrometry (MS) analysis. Here a typical topN data dependent acquisition scheme is depicted where a full MS scan (MS<sup>1</sup>) is followed by  $n$  MS<sup>2</sup> scans of the  $n$  most intense precursors at MS<sup>1</sup> level. **(C) Data analysis.** Information from the full MS and MS<sup>2</sup> spectra is used by proteomics workflows to search a database containing the sequence of all potential proteins in the sample. Figure by Hein et al. (27)

### 1.1.2. Sample preparation

As described earlier, bottom-up proteomics requires a specific sample preparation process that includes enzymatic digestion of proteins to short MS-accessible peptides and removal of any other agents that should not be introduced into the mass spectrometer. Various sample preparation protocols have been developed over the years, dependent on the type of samples (cell culture, tissue, organ, organism), the amount of sample material or even the type of biological questions to be answered. Regardless of the protocol used, the efficiency of protein extraction and isolation directly influences the quality of the subsequent MS analysis as well as the accuracy and

reliability of the results obtained. In the following, the main steps of sample preparation are described.

**Cell lysis.** The first step in efficient protein extraction is the disruption of cellular structures. Depending on the type of scientific question, the lysis of biological material can be performed either mechanically, e.g. by bead-milling, blending and grinding, physically, using heat or sonication, or chemically, by using various chemicals or enzymes (28). Usually native protein folding is undesirable and an additional denaturation step can be carried out to unfold all proteins and inhibit any enzymes to avoid modifications related to sample preparation or non-specific proteolysis. This includes the use of detergents such as sodium dodecyl sulfate (SDS) and sodium deoxycholate (SDC), emulsifiers or surfactants.

**Reduction and alkylation.** Next, the stable disulfide bonds of isolated proteins are reduced and alkylated to disrupt the disulfide bridges and prevent their possible reforming. Typical reducing agents in proteomics are dithiothreitol (DTT) or tris(2-carboxyethyl)phosphine (TCEP) (29), while the most commonly used alkylating agents are iodoacetamide (IAA) or chloroacetamide (CAA) (30).

**Digestion.** Different sequence-specific enzymes are employed for proteolytic digestion of proteins. Note that the choice of protease significantly affects the outcome of the experiment. The enzyme must be active in the presence of denaturants, highly sequence-specific and generate multiply-charged peptides of a certain average length. Considering all these aspects, trypsin is the most common protease in proteomics, often used in combination with the enzyme LysC. Trypsin cleaves C-terminally to lysine and arginine, whilst LysC – only to lysine residues (31). In special cases, chymotrypsin (cleaves C-terminally to aromatic residues), Asp-N, Lys-N, Lys-C, Arg-C, or Glu-C can be employed in shotgun proteomics to increase the overall protein sequence coverage or to generate peptides with different properties (32).

**Sample clean-up.** Before proceeding to the chromatographic separation, most protocols include a final clean-up step to remove all chemicals like detergents and salt remnants that could potentially damage the LC or MS setup as well as suppress ionization in ESI. This challenge has been successfully addressed by developing different techniques like 'Filter-Aided Sample Preparation' (FASP) or 'Stop and Go Extraction tips' (StageTips) that combine some or all of the afore-mentioned steps in a single reaction chamber (33–35). These approaches have greatly reduced the risk of loss or contamination of biological material, making it easier to automate the entire sample preparation step and have become routine in proteomics (7, 36).



In some specific cases, additional steps may be performed prior to the LC-MS analysis. Post-translational modification (PTM) studies typically involve additional enrichment steps to overcome sensitivity and complexity problems in the detection of low abundant and sub-stoichiometrically modified peptides (as will be discussed in Section 1.3 below) (37). In the case of deep proteomic measurements, fractionation techniques are often applied to reduce peptide sample complexity by dividing one peptide sample into several less complex ones (38). However, it is always necessary to be aware of the trade-offs between an increase in proteome depth and sequence coverage versus an increase in number of sample preparation steps, sample amount and measurement time.

### **1.1.3. Liquid chromatography – mass spectrometry (LC-MS)**

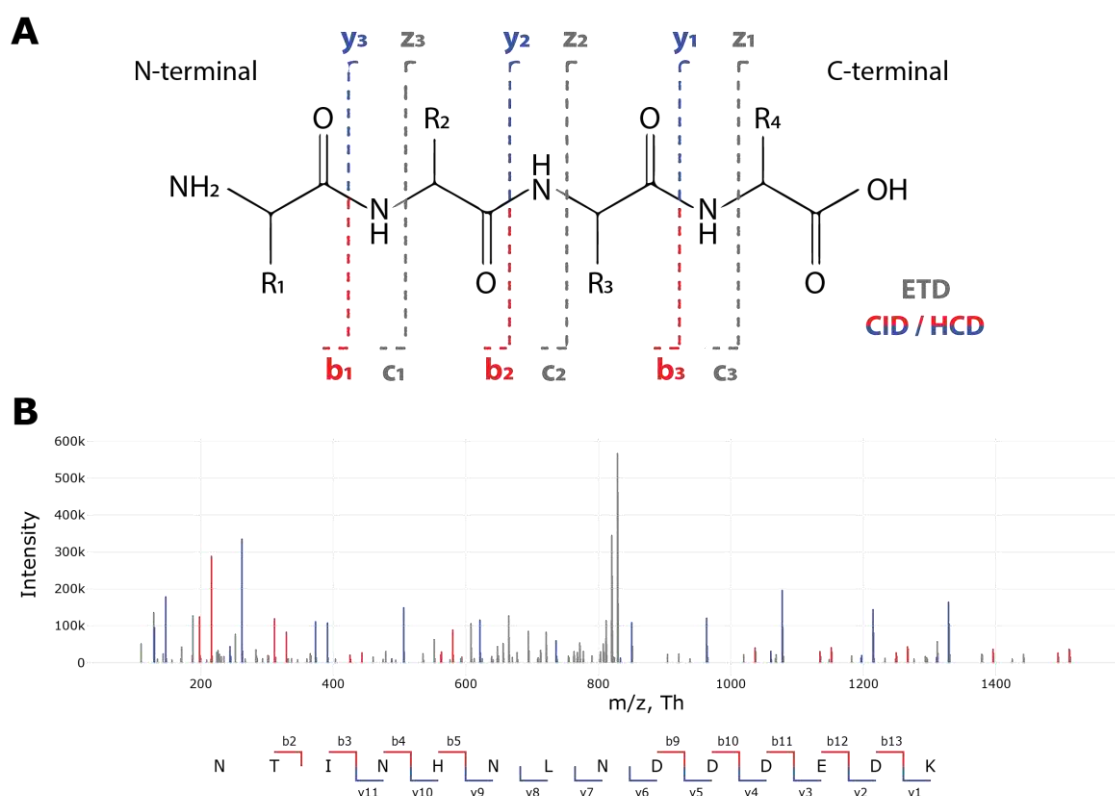
The separation of peptides in time and space is essential for high quality and reproducible results. This is particularly important for complex proteome samples, which exceed the scan capacities of even modern mass spectrometers (39). Historically, peptides were separated in the front end of the mass spectrometer using different coupled units (40). But nowadays, after digestion, the peptide mixture is further subjected to separation on columns of a liquid chromatography (LC) system coupled to the mass spectrometer.

**Liquid chromatography.** In reverse-phase chromatography in proteomics the separation of peptides is based on different hydrophobic interactions with a stationary phase, typically C18-silica phase. A gradient with a linearly increasing percentage of organic solvent, such as acetonitrile, in aqueous buffers gradually elutes peptides from the reversed-phase column. Using high-pressure pumps to ensure constant flow through the LC column, proteomics typically uses long columns with a small inner diameter, filled with small particle size to achieve better chromatographic resolution and lower numbers of co-eluting peptides concentrated into small volumes.

The EASY-nLC instrument of Thermo Fisher and the Evosep system were used for the projects in this thesis. The EASY-nLC system provides a well-established low-flow rate setup that enables optimal ionization of peptides and achieves high sensitivity in proteomics experiments (41). The recently introduced LC system called Evosep is beneficial for high throughput projects where short gradients are required (10).

**Mass spectrometry.** As the peptides elute from the chromatographic column, they are ionized via ESI and the resulting charged ions are passed through an ion transfer tube into the vacuum region of the MS instrument. The mass spectrometer continuously scans the peptide mass range and assigns masses, or more precisely mass-to-charge ratios ( $m/z$ ), and intensities to the eluting peptides. As peptide mass alone is not

sufficient for identification, a second MS step, called tandem MS, MS/MS or MS<sup>2</sup>, is employed. In this step the peptides selected for fragmentation are isolated and subsequently fragmented by collision with neutral gas molecules, such as nitrogen, helium or argon. Different peptide fragmentation methods are used in tandem MS, including collision-induced dissociation (CID), its variant higher-energy collisional dissociation (HCD) and electron-transfer dissociation (ETD) (42–44). All these dissociation strategies differ in the generation of distinctive fragmentation patterns in MS<sup>2</sup> scans. CID or HCD techniques almost exclusively result in the formation of b- (N-terminal part) and y-ions (C-terminal part) (45), whereas ETD predominantly produces c- and z-ions with a small number of y-ions (Fig. 2). Although HCD is the most widely used dissociation strategy in proteomics, the choice of using other fragmentation techniques may depend on the purpose of the experiment. For example, ETD is widely used to study labile PTMs and results in better fragmentation of long, multiply-charged or modified peptides (46).



**Figure 2: Fragmentation scheme. (A) Different fragmentation strategies applied to the peptide with four amino acids lead to the formation of different ion species.** CID and HCD dissociation techniques generate mainly b- and y-ions, whilst ETD generates c- and z-ions. Adapted from (45). **(B) Example of an HCD-type MS<sup>2</sup> spectrum.** A typical HCD fragmentation spectrum demonstrates a partial series of b-ions (red) and an almost complete series of y-ions (blue), which is a known characteristic of HCD. Below the spectrum, the identified ions are shown on the amino acid peptide sequence (Figure from Article 1).

#### 1.1.4. Proteomics data analysis

Nowadays, high-resolution MS-based proteomics produces an enormous amount of data. For instance, a single two-hour gradient run of a digested HeLa lysate contains around 90,000 spectra with a raw file size of several gigabytes. Therefore, efficient data analysis software has been developed to analyze and interpret these data. In our department, a freely available software package called MaxQuant was developed almost a decade and a half ago (22). Over the years, it has become a standard in the community due to its convenient all-in-one package solution and accurate quantification. However, in the era of open science, the need for transparency in research as well as opportunities for collaboration within the community have become an intrinsic part of software development (this will be discussed in Section 1.4). In Article 5, we show a new joint effort of our group to develop a new open and super-fast proteomics framework, called AlphaPept, maintaining high software standards with the integration of state-of-the-art machine learning and deep learning technologies. Based on the bioinformatics workflow employed in these software tools, the entire process of data analysis in proteomics can be divided into several parts (Fig. 3).

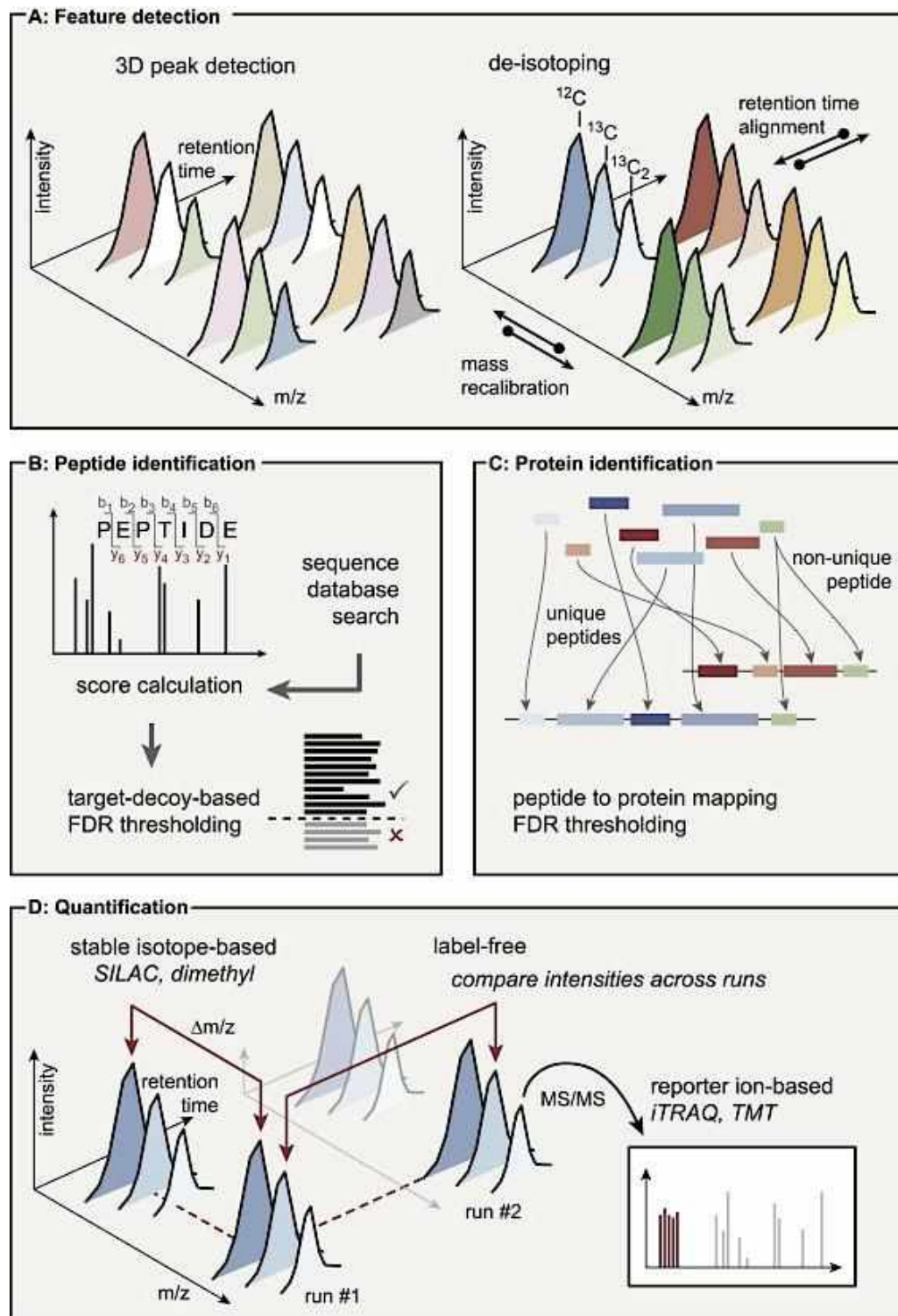
**Feature detection.** In the first step, the peptide features have to be detected in the full scans in a multidimensional space (retention time,  $m/z$ , intensity and optionally ion mobility as discussed in Section 1.2) and assembled into isotope patterns by deisotoping the spectrum (47). For each  $MS^1$  feature detected, the  $m/z$  and intensity of a possible peptide precursor are determined. To increase accuracy and achieve consistency across all measured data dimension, an additional recalibration step is suggested (48, 49).

**Peptide identification.** For peptide identification, known precursor masses derived from assembled isotope patterns are assigned to corresponding  $MS^2$  fragmentation spectra. There are several approaches to how this information can be used to determine the peptide sequence, mainly *de novo* sequencing or database searching. In *de novo* sequencing, the mass difference between all peaks in the  $MS^2$  spectrum is calculated and, if possible, assigned to the (un)modified amino acid (50). In principle, this should lead to a complete peptide sequence, even allowing the identification of new peptides and proteins not described in the available databases, but is still not sufficiently sensitive due to the effects of missing fragment ion peaks and spectral noise on accuracy (51). Conversely, database searching uses all known information about the precursor to compare this against theoretical spectra derived from *in-silico* digestion of a reference organism database containing all known or possibly be produced protein sequences

(26). However, some substantial improvements have already been made by introducing deep learning into *de novo* peptide sequencing (52, 53). Peptides are identified by scoring each measured fragmentation spectrum against all theoretical fragmentation spectra in the database within a specified peptide mass tolerance to find the highest scoring peptide-spectrum match (PSM). Various approaches have been developed to prevent false identifications, including the target-decoy approach and different machine learning algorithms (54).

**Protein assembly.** The next step is to solve the “protein inference problem” by assembling the identified peptides into proteins. This task is not trivial, as many peptides are non-unique and can be assigned to different proteins, especially in the case of proteoforms (26). As one possible solution, the concept of ‘protein grouping’ is introduced, which allows proteins to be assigned to the same group if they share one or more peptides and have no uniquely distinguishable peptides (55). Similar to the peptide identification level, false protein identifications should also be controlled at this level (56).

**Protein quantification.** The aim of every proteomics experiment is not only to identify proteins, but also to obtain information on the amount of proteins. Quantitative proteomics approaches can be divided into absolute and relative (57, 58). Absolute quantification uses different labelling techniques to measure the absolute amount of protein in a sample. In standard label-based methods, the intensity of differentially labeled peptides is compared within the same LC-MS/MS run at MS<sup>1</sup> level (stable isotope labeling with SILAC or dimethyl) or at MS<sup>2</sup> level (isobaric labeling with TMT or iTRAQ). In contrast to this, relative quantification uses no labeling and compares the relative amounts of protein between different samples to derive their relative concentration changes.



**Figure 3: Proteomics data analysis workflow.** (A) **Feature detection.** The detection of MS<sup>1</sup> peptide features in a three-dimensional  $m/z$ -retention time-intensity space and assembling them into isotope patterns. Additional mass and retention time recalibration steps are applied. (B) **Peptide identification by database search.** Experimental MS<sup>2</sup> spectra are scored against theoretical spectra from an *in-silico* digested sequence database. Using the target-decoy approach, true identifications are distinguished from false identifications at a defined false discovery rate (FDR) threshold. (C) **Protein identification.** By solving the ‘protein inference’ problem, peptides are assembled into proteins and additional FDR filtering is applied at this level to avoid protein false identifications. (D) **Protein quantification.** Quantification of peptides and proteins within the run based on stable-isotope labeling and across multiple runs based on the label-free quantification. Figure by Hein et al. (27).

### 1.2. Ion mobility spectrometry in proteomics

Ion mobility spectrometry (IMS) has advanced considerably in recent decades as an analytical technique for the analysis of ionized chemical substances on the basis of their velocity in the gas phase under the influence of an electric field. IMS, together with MS, traces its origins back to the late 1890s when scientists first studied the separation of charged particles in electric and magnetic fields (59). However, relative to the rapidly developing MS field, for quite a long time, IMS remained a stand-alone method with limited applications, e.g. for the detection of explosives, drugs, and chemical warfare agents (60). In the early 1960s, several groups showed in parallel the advantage of different configurations of hybrid IMS with MS analyzers, separating ions based on their ion mobility (IM), which takes into account their size, shape and charge, and  $m/z$  information (61–63). All this, together with many critical developments in MS, such as soft ionization, has extended the application of IMS and enabled the analysis of complex samples, which has found use in various fields, including proteomics (64–66).

Several IM technologies have been developed over the last decades, each having its own operating principle, design and already commercially released devices. According to the classification by May and McLean, all IM techniques can be categorized into three groups based on their underlying separation concepts, namely (i) time-dispersive methods, (ii) space-dispersive methods, and (iii) confinement, or trapping, and selective release methods (67). Time-dispersive IM methods generate an arrival time spectrum in which all ions drift along with the gas flow and include such methods as drift tube ion mobility spectrometry (DTIMS) and traveling wave ion mobility spectrometry (TWIMS), which have been commercialized by Agilent and Waters, respectively. Space-dispersive methods, which include field asymmetric waveform ion mobility spectrometry (FAIMS) and have been commercialized by Thermo and SCIEX, respectively, separate ions along different drift paths based on their IM differences. The latter group, which applies ion trapping followed by selective release, traps ions within a specific region and selectively releases them for further analysis based upon the differences in IM. It includes the trapped ion mobility spectrometry (TIMS) method recently introduced by Park and co-workers from Bruker Daltonics (68, 69).

The basic idea behind TIMS is the reverse concept of the classical DTIMS. Instead of driving ions through a stationary gas, TIMS holds ions stationary in a moving gas column. This allowed the size of the analyzer to be significantly reduced (down to 5-10 cm) as it has to be large enough to hold ions stationary. Since first publication, TIMS has become an efficient alternative to other IM techniques, providing high resolving power (up to  $R \sim 300$ ), duty cycle (100%), and efficiency ( $\sim 80\%$ ) (70). The small size of

the TIMS device and its rapid ion separation have opened up new perspectives when coupled to modern high-resolution mass analyzers, such as time-of-flight (TOF) instruments, which efficiently transmits ions and achieves mass accuracy from low to sub-ppm (71). Combined upfront with an LC system, the LC-TIMS-TOF instrument, nested within analytical time scales based on the separation speed, allows the separation of complex biological mixtures in 4 dimensions of separation, such as a retention time – ion mobility –  $m/z$  – intensity.

Almost all of the projects described in this thesis use completely or partially data acquired on the Bruker timsTOF Pro mass spectrometer using different acquisition strategies in the PASEF scanning mode. All of these aspects are therefore covered in the following chapters.

### **1.2.1. The TIMS-TOF instrument**

A hybrid analytical instrument employing both TIMS and TOF techniques called the timsTOF Pro has garnered much attention since its introduction by Bruker Daltonics (Fig. 4A). Since then, further advances have made the timsTOF popular due to its wide dynamic range, high sequence coverage and very high sensitivity, enabling it to reach even the single cell level (72).

After separating the peptides by LC and ionizing them via ESI, they enter the mass spectrometer. They are pushed through a glass capillary, are deflected by 90 degrees and focused using several focusing lenses. This prevents uncharged contaminants from entering the ion path and makes the instrument more robust. The long TIMS tunnel is divided into three separate sections (dual TIMS setup and transfer region) that are responsible for different processes (Fig. 4B). The trapping section accumulates and holds ions in an electrodynamic tunnel through which a constant gas flow ( $v_g$ ) is directed from the entrance funnel to the exit funnel and which is counteracted by an increasing electric field gradient ( $E$ ) along the tunnel. Based on these two forces, the ions entering the TIMS tunnel occupy a position in the flow in which the drag forces are counterbalanced by the electric ( $E$ ) forces separating the ion based on their IM values along the TIMS tunnel. For all peptide ions, IM values account for their size, shape and charge, and have an inverse correlation with collision cross sections (CCS). Since low-mobility ions with larger CCS need higher  $E$  forces to counterbalance their high drag forces, they end up closer to the exit, whilst high-mobility ions with smaller CCS are located closer to the tunnel entrance. After an accumulation time of 100 ms, all trapped ions are transferred in a single step to the TIMS analyzer through the transfer region within 1-3 ms. In the analyzer, a gradual linear decrease in the voltage gradient (TIMS

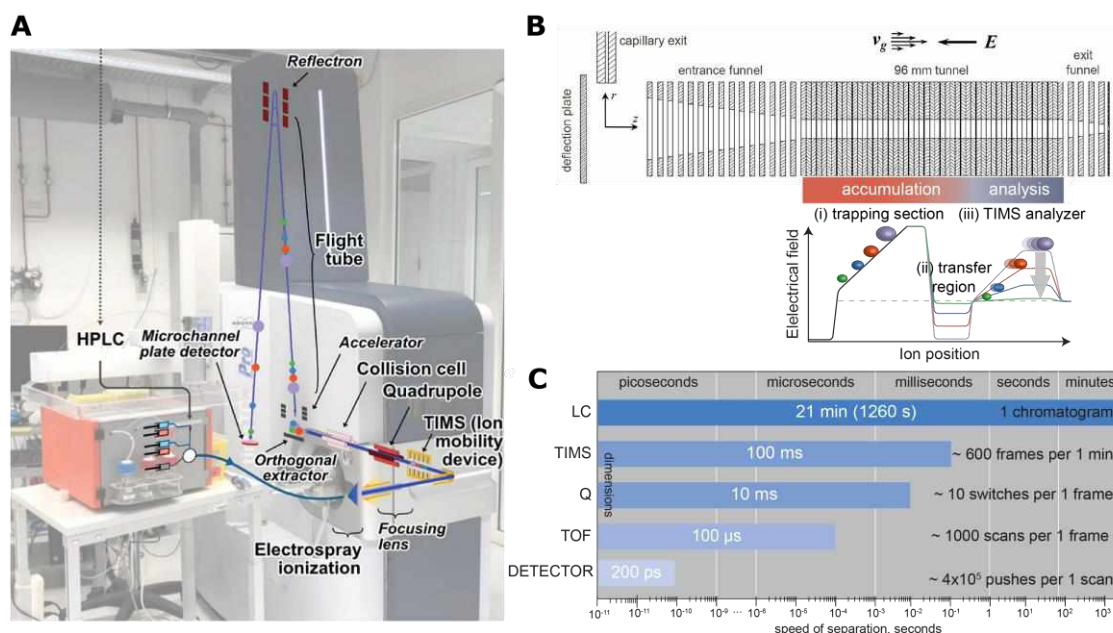
ramp time) releases ions as a function of their ion mobility within 100 ms, which is the so called TIMS 'frame'. In parallel, the trapping section starts filling up again with the next batch of ions. As a result, compared to the standard continuous acquisition mode, no ions need to be discarded with this approach, and it allows the duty cycle to be increased up to 100%.

At the exit of the dual TIMS device, ions pass through the ion transfer multipole into the quadrupole mass analyzer (73). As the name implies, it consists of four parallel cylindrical metal rods. A direct current and oscillating radio-frequency voltage are applied to the rods, which sets up the electric field in the quadrupole. This field forces ions traveling down the quadrupole between the rods to follow a trajectory that oscillates around the central axis. Ions with certain  $m/z$  values, called resonant ions, reach the detector, while other ions, or non-resonant ions, collide with the rods and become neutralized. The constant variation of the applied voltage tunes the quadrupole to different  $m/z$  values. After the filtering step all the ions are transferred to the collision cell. There, they collide with a low pressure of an inert gases, such as helium, argon or dinitrogen, causing them to break apart and form fragments.

At the last step, either intact peptides or fragment ions enter a second mass analyzer, which here is represented by a time-of-flight (TOF) analyzer. The basic principle of these analyzers is that they separate ions with different  $m/z$  by measuring the time it takes them to pass through a field-free region. The instrument first uses an electric field to accelerate the ions to the same potential. The accelerated ions fly orthogonally along the flight tube until they are reflected in the reflectron, which increases the flight time without increasing the length of the tube, while correcting for initial differences in kinetic energy. Finally, the ions are detected by the microchannel plate detector, where the measured time of flight depends on the  $m/z$  values of the ions. Lighter ions arrive before the heavier ones. The time during which the ions remain in flight is converted by the detector into an accurate arrival time.

When describing the composition of the LC-TIMS-Q-TOF hybrid instrument, it is important to remember that all analytical separation techniques used can only be combined due to their different time scales (Fig. 4C). Typically, the first separation of peptides occurs on the LC column in the range of minutes or hours with peak widths of several seconds, followed by accumulation and separation on the TIMS device and the quadrupole filtration within milliseconds. Further coupling with the fast TOF instrument is on the microsecond time scale, while detector operates even faster within picoseconds. All these dimensions exhibit an offset of one or more orders of magnitude in time, resulting in complete mass resolved MS<sup>1</sup> or MS<sup>2</sup> spectra.



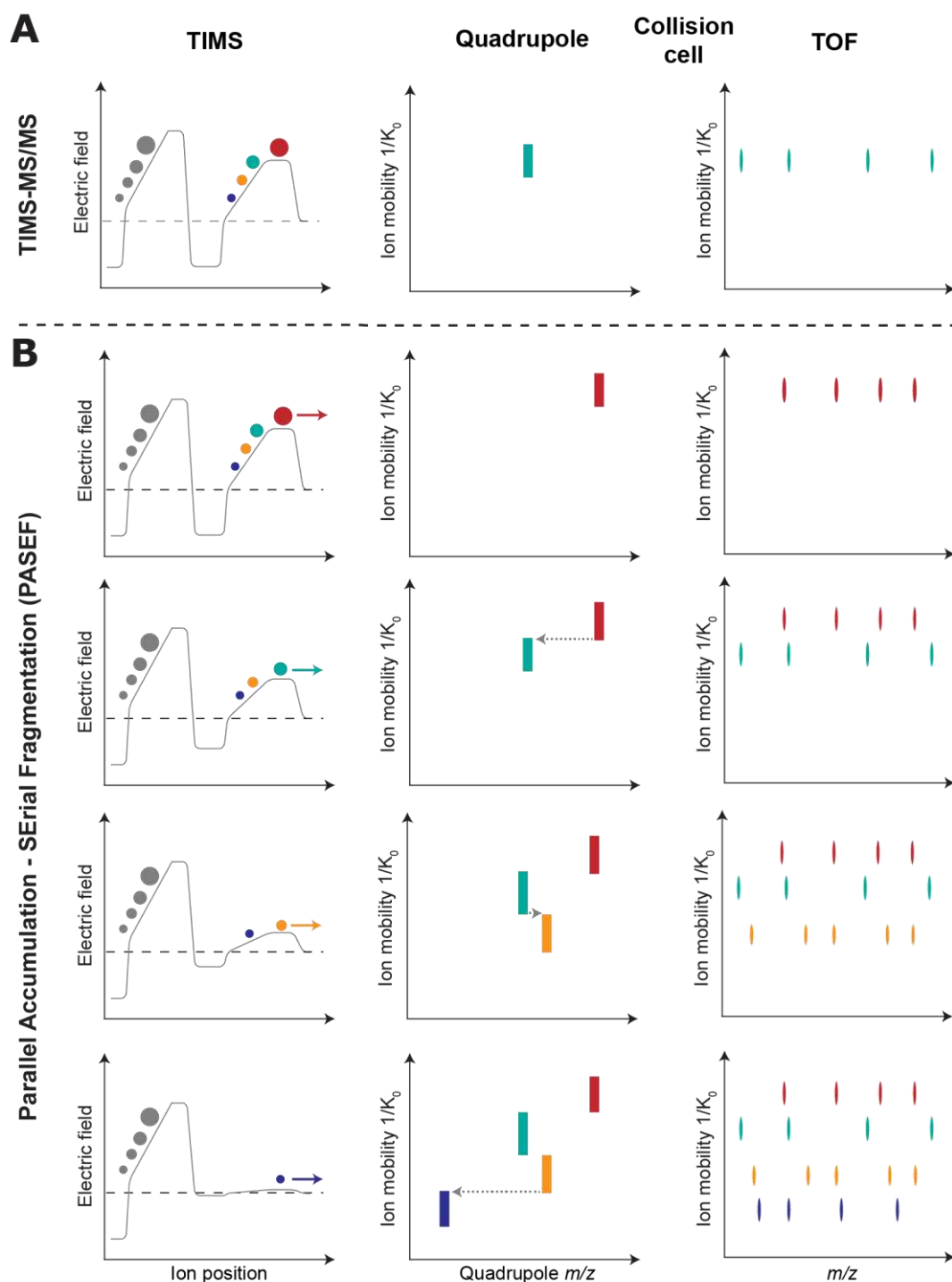


**Figure 4: Architecture of the timsTOF instrument. (A) Main components of the instrument.** High performance liquid chromatography (HPLC) separates the peptide mixture, which is ionized *via* electrospray. The peptide ions are then accumulated and separated in a TIMS device. After a quadrupole selection step, the ions are fragmented in a collision cell and finally analyzed by a TOF mass analyzer. Adapted from (74). **(B) Structure and principle of work of the dual TIMS device.** The entire TIMS tunnel consists of three parts: (i) the trap where ions are accumulated and trapped, (ii) the transfer region, and (iii) the analyzer which elutes trapped ions into the downstream mass analyzer by lowering the electrical field. Adapted from (70). **(C) Analytical time scales of the timsTOF pro instrument.** Each dimension of separation is approximated on the time scale in seconds. Adapted from (67).

### 1.2.2. PASEF principle

A new scanning mode on the timsTOF Pro instrument, termed Parallel Accumulation – Serial Fragmentation (PASEF), was recently introduced by our group and has become a valuable addition to the instrument for many workflows including proteomics (75–77).

In conventional MS/MS experiments on the TIMS-Q-TOF mass spectrometer, first the ions are accumulated and subsequent separated by IM. Downstream the quadrupole mass filter selects only one precursor ion from the ion beam for further analysis, whereas all other ions eluting from the TIMS device are discarded (Fig. 5A). In PASEF mode, the selection of precursor ions in the quadrupole occurs serially. Synchronized with the dual TIMS device elution, the quadrupole sequentially adjusts its position to select multiple precursor ions for further analysis within a single TIMS scan. These selected co-eluting precursor ions are further fragmented and their fragments can be exclusively distinguished by their different ion mobility positions even though they share the same elution time (Fig. 5B).



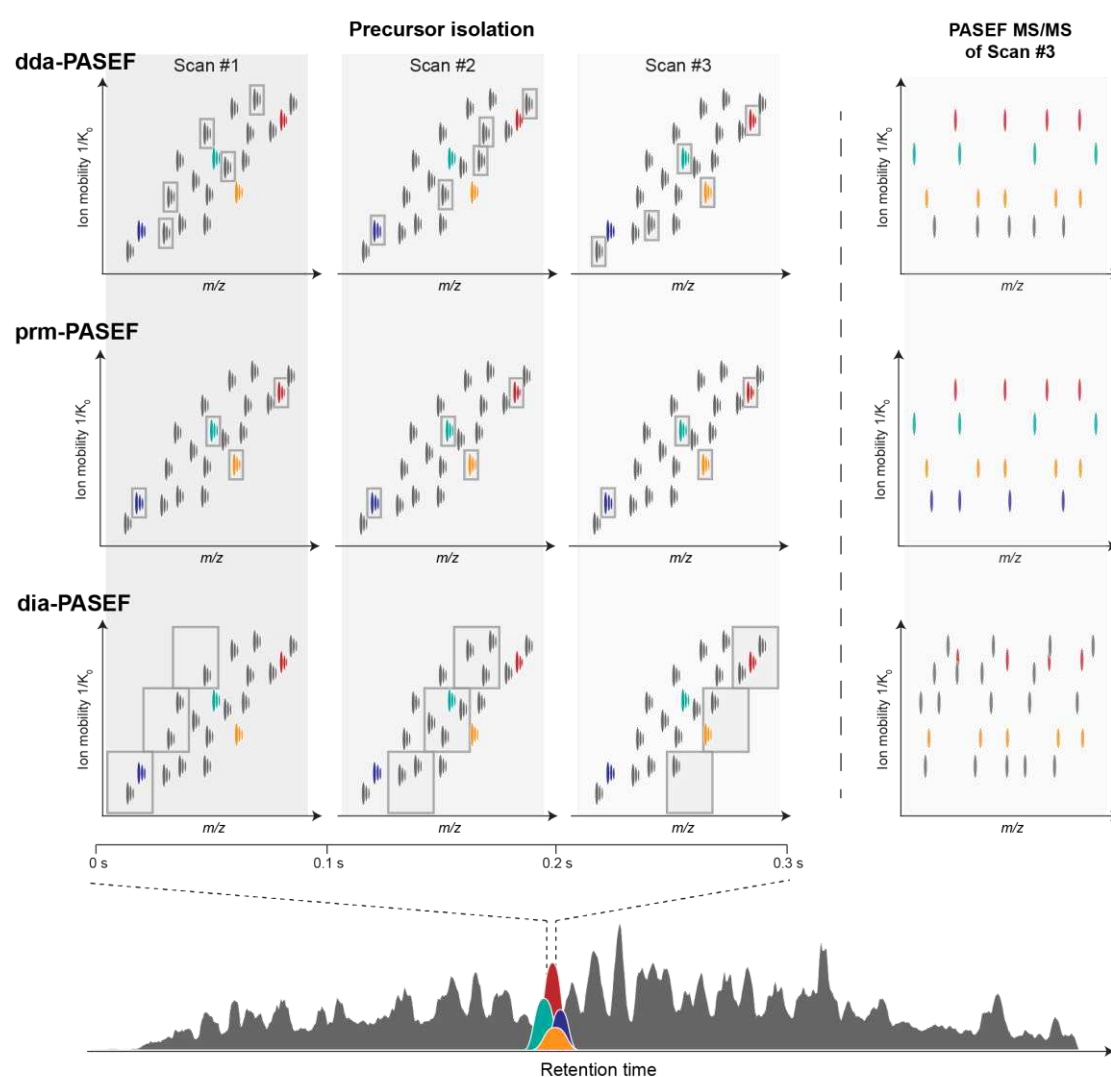
**Figure 5: The PASEF principle compared to the conventional TIMS-MS/MS operation mode. (A)** Selection by quadrupole of only one precursor from the TIMS scan in a conventional TIMS-MS/MS operation mode. All other precursor ions are discarded from further analysis. **(B).** In the PASEF scan mode the sequential rapid switching of the quadrupole allows multiple precursors with different  $m/z$  and IM values to be selected at the same retention time. Adapted from (78).

This implementation of the PASEF scan mode is made possible by the extremely fast switching time of the quadrupole position ( $< 1$  ms). Keeping in mind the time scale of the LC-TIMS-Q-TOF device (Fig. 4C), this allows to fit in the acquisition scheme the sequencing of up to fifteen precursors in each of PASEF scan (13). Combined with the

fast-sampling speed of the TOF analyzer, the instrument acquires complex proteomics samples with consistently high sensitivity, which is particularly important for proteomics studies with extremely low sample amount, such as single-cell analysis. Recently, we demonstrated the identification and quantification of 4000 protein groups with high reproducibility from just 1 ng of a HeLa sample (72).

### 1.2.3. PASEF acquisition strategies

Since it was first introduced, the PASEF principle has been successfully applied to three main acquisition schemes in proteomics: for data-dependent acquisition (dda-PASEF), for targeted approach (prm-PASEF), and for data-independent acquisition (dia-PASEF) (Fig. 6).



**Figure 6: Schematic of three main PASEF scan modes.** Coeluting peptides can be analyzed in three different modes: dda-PASEF (top panel), prm-PASEF (middle panel) and dia-PASEF (bottom panel). For each mode, three consecutive scans are shown within 100 ms each. The PASEF MS/MS scans are displayed for the last PASEF scan only. The quadrupole isolation windows appear in grey boxes. Adapted from (78).

### **dda-PASEF mode**

In the classical DDA approach, the mass spectrometer isolates and fragments the topN most abundant precursor ions from the MS<sup>1</sup> scan (79). This approach remains the most common MS acquisition strategy and has traditionally been employed for initial characterization of proteomics samples and discovery of new proteins. However, most implementations suffer from limited reproducibility of identification, a high number of missing values and a narrow dynamic range (80).

In the dda-PASEF mode, the masses of the individual precursors and their IM values are first determined in the full MS scan. The algorithm then finds the topN most abundant precursors and optimizes the quadrupole switching path for subsequent MS<sup>2</sup> scans. Depending on the  $m/z$  values of the precursor ions, the quadrupole isolation window varies from 2 to 3 Th, allowing to isolate at best only the monoisotopic peak of the target ion and (some of) its isotope peaks and to generate a high-quality fragmentation spectrum.

As discussed in the previous section, the PASEF scan mode by default significantly increases the number of precursors analyzed in a single experiment. Furthermore, the online PASEF precursor scheduling algorithm optimizes the quadrupole route for each dda-PASEF scan and aims to maximize the number of precursors per acquisition cycle that can be successfully identified and quantified (13). This real-time approach offers many advantages. Firstly, single-charged species can be easily removed from further analysis due to their characteristic positions in the  $m/z$  – IM plane. The ‘target intensity’ parameter included into the sequencing algorithm enables to achieve high proteomic depth by sequencing the low-abundant precursors repeatedly (several times to reach a certain intensity threshold) and aggregating their spectra in the postprocessing step to increase the signal-to-noise ratio.

### **prm-PASEF mode**

Targeted data acquisition strategies can be applied to obtain a predefined set of precursors with high reproducibility and specificity in complex biological samples. One of the main traditional targeted approaches is called parallel reaction monitoring, or PRM. In this method, a mass spectrometer monitors a list of predefined specific peptide precursors over an expected elution time window by acquiring MS<sup>2</sup> spectra (81). Although recent developments in targeted strategies on a global scale improved quantitative readout of relatively large groups of peptides, this method is still limited by the acquisition speed of the instrument and consequently the number of proteins (82).

Combining PRM with the PASEF scan mode overcomes some of the limitations of the targeted method due to the PASEF sequencing power. This can be exploited either to target more precursors without increasing cycle times or to use very fast separation methods, which is essential for clinical applications (83).

### **dia-PASEF mode**

In the standard DIA mode, the mass spectrometer sequentially isolates from every MS<sup>1</sup> scan not only individual precursor ions within narrow isolation windows, but a group of ions falling within a given mass range (e.g. 25 Th). Although this ensures that all precursors from this wider isolation window are fragmented at least once per cycle, the complexity of fragmentation spectra, consisting of fragments of many co-eluting precursors within the same window, still remains a challenge to disentangle.

In contrast to the standard DIA scheme, the dia-PASEF mode operates in the  $m/z$  – IM plane, where the isolation windows are defined in a two-dimensional space. Due to the fact that ion mobilities and masses are correlated, for peptide ions of a given charge state, we were able to program the quadrupole to efficiently isolate most of the precursor cloud along the IM elution (covered in Article 6). For this, in dia-PASEF mode, a full TIMS MS scan is acquired to determine the position of individual precursor ions, including their  $m/z$  and ion mobility values. The quadrupole mass isolation window then shifts from high  $m/z$  and high IM to low  $m/z$  and low IM (from upper right to lower left), as instructed by the method, to fully cover the ion cloud. In Figure 6, we demonstrate the dia-PASEF scheme, which consists of three dia-PASEF scans with equidistant isolation widths covering the precursor cloud in the three IM windows per dia-PASEF scan. This allowed us to achieve high sequence coverage and very high sensitivity for different biological samples. However, the original method still missed some regions of the precursor space and was not optimized for studies with non-standard peptides distribution in the  $m/z$  – IM plane, i.e. for phosphoproteomics studies. We have therefore recently developed optimal dia-PASEF methods with our Python tool `py_diAID`, described in Article 7, which allows variable isolation windows optimally positioned in two-dimensional space for nearly complete precursor coverage. Compared to the original 'high speed' dia-PASEF method, we gained 14% (84% vs. 98%) coverage of the unmodified peptide population and almost 60% (34% vs. 93%) for phosphopeptides.

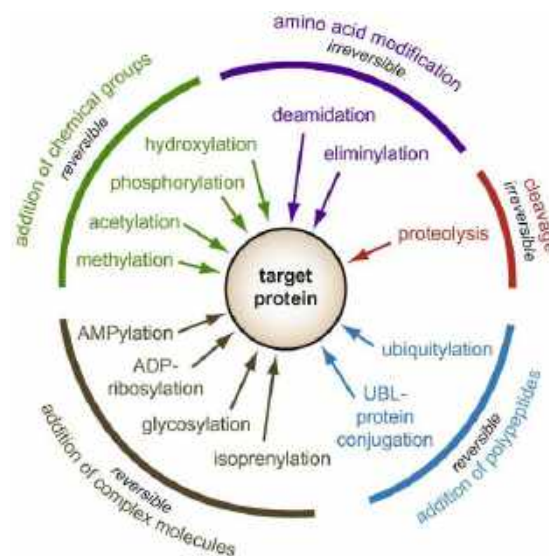
### **1.3. Post-translational modifications in MS-based proteomics**

Eukaryotic cells need to rapidly respond to a range of cell-intrinsic and cell-extrinsic cues. This is only possible through an extensively connected and tightly regulated complex signaling network that allows the integration of different stimuli. Post-

translational modifications (PTMs) are one of the essential mechanisms for controlling the entire cellular network by transducing signals coming from within the cell or the environment. PTMs increase the functional diversity of proteins by changing the biochemical properties, playing a key role in many cellular processes such as cellular differentiation, protein degradation, signaling and regulatory processes, regulation of gene expression and protein-protein interactions.

PTMs can be divided into reversible and irreversible ones (84). The reversible group includes (i) addition of chemical groups, such as methylation and phosphorylation, (ii) complex molecules, like some glycosylations or AMPylation, and (iii) polypeptides in case of ubiquitylation, while irreversible modifications, which only occur in one direction, include (iv) specific covalent modifications of the amino acid side chain, such as deamidation, and (v) proteolytic cleavage (Fig. 7). PTMs can occur on a single type of amino acid or in multiple amino acids and lead to changes in the chemical properties of the modified sites. More than 200 diverse types of PTMs are known to date, ranging from small chemical modifications (e.g. phosphorylation and acetylation) to the addition of complete proteins (e.g. ubiquitylation).

Phosphorylation is the best studied and one of the most common PTMs. I have worked with various phosphoproteomics datasets in many of the projects presented in this thesis. Therefore, I will primarily focus on this type of PTM, its role in cell signalling and the challenges we encounter studying it.

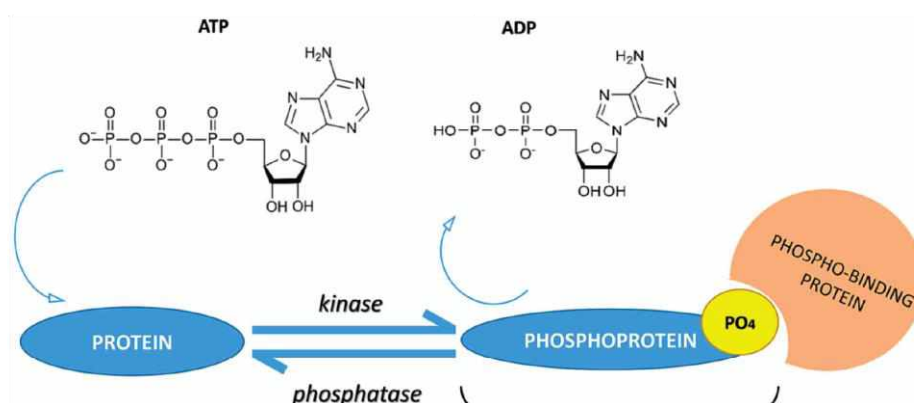


**Figure 7: Variety of post-translational modifications (PTMs).** The distinct PTM classes are colored differently. The type of modification, if they are reversible and examples of PTMs are indicated separately for each class. Adapted from (84).

### 1.3.1. Phosphorylation

Phosphorylation is a reversible PTM. It results from the transfer of the terminal phosphate group ( $\gamma\text{-PO}_3^{2-}$ ) from adenosine 5'-triphosphate (ATP) to the hydroxyl oxygen of certain amino acids (Fig. 8). Three amino acids, such as serine, threonine and tyrosine (Ser/Thr/Tyr), are mainly phosphorylated in cells and demonstrate a relative abundance of about 86%, 12% and 2% in normally growing cells, respectively (85). The phosphorylation changes the chemical properties of the amino acid from hydrophobic apolar to hydrophilic polar, which can lead to changes in protein structure and stability, protein–protein interactions, enzyme activation or subcellular localization.

The addition of a phosphate group to a substrate protein is carried out by enzymes called protein kinases, which are one of the largest gene families and account for about 2% of the entire human genome (86). More than 500 human kinases are known, mutations or dysregulations of which play a role in the progression of many human diseases, including cancer and neurological disorders. Therefore, to maintain a constant balance between phosphorylation and dephosphorylation, the phosphorylation process can be reversed through the activity of ~ 200 different phosphatases that transfer the phosphate group to adenosine 5'-diphosphate (ADP). Currently, about 300,000 phosphosites are recorded on the PhosphoSitePlus platform, providing information on experimentally observed PTMs of human and mouse proteins (87). But if we take into account the estimate that ~30% of all cellular proteins are phosphorylated on at least one residue, or the results of existing phosphorylation site prediction tools, then about 750,000 additional sites are likely to be phosphorylated (88, 89).



**Figure 8: Mechanism of reversible phosphorylation by kinase and phosphatase.** The protein receives the phosphate group as a result of ATP hydrolysis through the enzymatic activity of kinase. The reverse process is orchestrated by phosphatases through the transfer of the phosphor group to ADP. Adapted from (90).

### 1.3.2. Protein phosphorylation in cell signaling

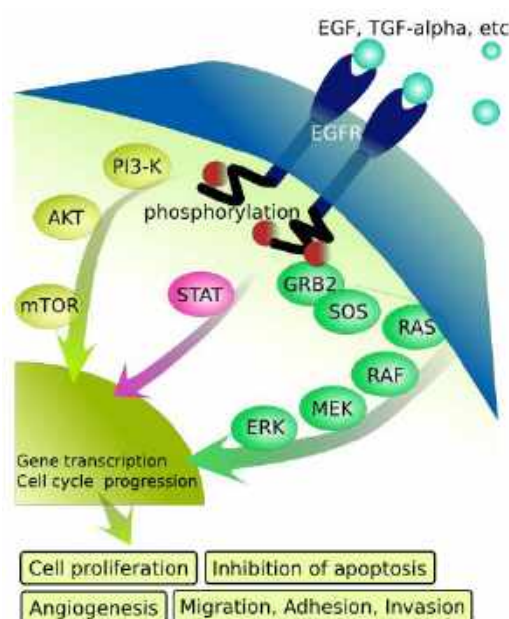
Protein phosphorylation plays a critical role in the regulation of many cellular processes, such as cell cycle, cell growth, apoptosis and countless signal transduction pathways. For a large subset of proteins, phosphorylation is tightly linked to protein activity and is a key mechanism of cell signalling. The conformational changes that can occur in proteins after phosphorylation may lead to several different outcomes. In some cases, they regulate the catalytic activity of proteins. One of the other possible outcomes is the recruitment by phosphorylated proteins of neighboring proteins with structurally conserved domains that specifically recognize and bind to different phosphomotifs. Both outcomes are essential for signal transduction. Signaling pathways can be constituted by kinases, ranging from tyrosine kinase receptors on the cell surface to downstream kinases, primarily serine/threonine kinases. In a nutshell, ligand binding at the cell surface triggers a phosphorylation cascade, with phosphorylation and activation of one protein stimulating the phosphorylation of another, amplifying the signal and transmitting it through the cell. The signal continues to propagate until it is turned off by the action of a phosphatase. To exemplify the cell signaling process, I have chosen one of the most important signaling pathways in mammalian cells, called the epidermal growth factor (EGF) pathway, which has also been investigated in the Articles 2 and 7.

The EGF signaling pathway acts through a series of different kinases, stimulating a whole signaling network associated with a large number of outcomes, such as cell proliferation, growth, migration, differentiation, and inhibition of apoptosis (Fig. 9) (91). The epidermal growth factor receptor (EGFR) is a transmembrane protein that is activated by binding of its specific ligands, including EGF and transforming growth factor  $\alpha$  (TGF $\alpha$ ). Upon activation, EGFR is converted from an inactive monomer to an active dimeric form. This dimerization stimulates the intrinsic intracellular tyrosine kinase domain, which auto phosphorylates tyrosine residues of the cytoplasmic EGFR domain. Activated EGFR is now able to bind various cytoplasmic adaptor proteins via tyrosine-specific binding domains, including src homology-2 (SH2) or phosphotyrosine binding (PTB) domains. One of these adaptor proteins is the growth factor receptor binding protein-2 (GRB2), which recruits the Son of sevenless homolog 1 (SOS-1) protein. SOS-1 moves in close proximity to members of the Ras family, from "Rat sarcoma virus", which is able of binding guanine nucleotides. SOS-1 activates the RAS protein by exchanging its bound guanosine 5'-diphosphate (GDP) for guanosine 5'-triphosphate (GTP). Activated RAS in turn activates the protein kinase (MAPK) cascade, in which the previous member phosphorylates and the next one in the following sequence: RAS – RAF kinase, named for Rapidly Accelerated Fibrosarcoma – mitogen-activated protein



kinase kinase (MEK) – extracellular-signal-regulated kinases (ERKs); ERKs act on various target molecules responsible for cell growth and proliferation. Another EGFR-activated signaling cascade, called PI3K-AKT-mTOR pathway, is responsible for controlling metabolism, proliferation, cell size and survival. EGFR can also directly activate transcription factors of the STAT family involved in processes such as immunity, cell division, cell death and tumor formation.

The EGFR signalling pathway demonstrates that a large number of signalling pathway control mechanisms are required in order to always maintain the correct level of signalling within cells. Any deviations from this control mainly lead to various disease states, e.g. aberrant EGFR signalling is common in a number of cancers and can correlate with tumor development and progression (92). Hence, the understanding of aspects of cellular signalling using phosphoproteomics can significantly help in explaining the biological behavior of disease cells and facilitate the search for a treatment.



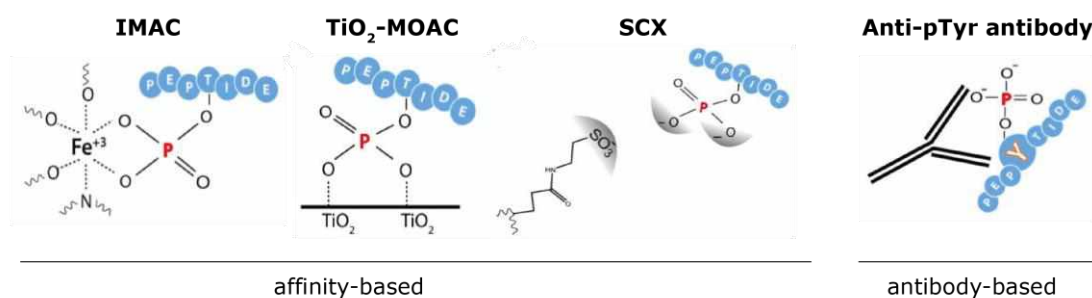
**Figure 9: Schematic representation of the EGF/EGFR signaling pathway.** Adapted from (<https://commons.wikimedia.org>).

### 1.3.3. Phosphoproteomics and its challenges

Many different analytical techniques have been developed over time to analyze groups of phosphoproteins in small targeted studies of various signalling pathways. However, due to the size and complexity of the signaling network, it is extremely important to investigate the dynamics of the phosphorylation on a global scale. Phosphoproteomics in principle allows the analysis of all phosphorylation events simultaneously, rather than looking at individual phosphorylated proteins. This is especially useful in cases where

deregulation of signaling is linked to a disease. To address this challenge, high-resolution mass spectrometry has become the primary method of choice for detecting and quantifying phosphorylation events on a proteome-wide scale in an unbiased manner. Recent developments in MS have greatly improved phosphoproteomics research, with tens of thousands of phosphorylation sites now being reported in a standard phosphoproteomics experiment (93).

Despite improvements in phosphoproteomics studies, the study of phosphorylation events using bottom-up MS is still challenging in many aspects (94). Firstly, phosphorylation is a transient and dynamic process, resulting in low abundance of phosphopeptides, which impedes their detection, especially in complex mixtures. This issue can be alleviated by employing prefractionation techniques and enrichment strategies prior to MS analysis. A plethora of such methods have already been introduced, including the frequently used affinity-based and antibody-based methods (Fig. 10) (95).



**Figure 10: Enrichment of phosphopeptides.** Affinity-based methods include (i) immobilized metal affinity chromatography (IMAC), (ii) metal oxide affinity chromatography (MOAC), and (iii) strong cation exchange (SCX) chromatography. In the IMAC and MOAC methods, positively charged metal ions, e.g. Fe(III), or metal oxides, e.g. titanium oxide, respectively, are immobilized with a solid phase on the chromatography column and bind the phosphopeptides. Separation and enrichment on negatively charged strong cation exchange (SCX) columns occurs based on peptide charge, with phosphopeptides enriched in earlier-eluted fractions. The antibody-based approach involves immunoprecipitation of phosphotyrosine peptides. Adapted from (95).

Another difficult challenge is the unambiguous localization of phosphosites, which is essential for understanding the role of phosphorylation events (96). The confident assignment of phosphorylation to a particular amino acid position on a modified sequence requires the presence of the corresponding fragment ions, which are not always present in MS<sup>2</sup> spectra. This can also be further complicated by the presence of multiple potential phosphosites. To address this problem, several computational algorithms have implemented a probability-based PTM localization score, e.g. the Andromeda PTM score, which reports a probability score for all sites (97). It is mainly determined by the presence and intensity of unmodified and phosphorylated fragments in the MS<sup>2</sup> spectra.

Finally, quantification in phosphorylation studies is based on single peptides and this remains a critical issue in the field, especially for DDA strategies, potentially resulting in a large number of missing values. Recent advances in the application of the DIA for rapid phosphoproteome profiling on the Orbitrap MS instrument have already enabled the quantification of over 13,000 phosphopeptides from HeLa cells using only a 15 minutes gradient (98). Further refinements and the use of the optimized dia-PASEF method on the timsTOF system, discussed in Article 7, doubled our numbers with less input material and provided a nearly complete data matrix including a high degree of confidence for positional phosphoisomers (99).

#### **1.4. Scientific software development and open science**

The development of any type of software pertains to the analysis, design, implementation, testing, deployment, and maintenance of software tools. All of these components play key roles in the success of any software development. However, in the scientific software development it is quite challenging to follow all software engineering practices employed in other fields due to the following characteristics of scientific projects (100). First of all, in most cases it is problematic to define all the requirements for the development of a software tool from the very beginning of a project. The research question tends to remain imprecise, and perceptions of the structure and functions of the software change as the research progresses. Moreover, very often the software implementation by itself is not the main purpose of the research. Secondly, because of the complexity of the topics, software in the scientific field is usually implemented by experts in the subject area being researched instead of trained software engineers. Finally, due to the lack of confidence in the success of the project and the reduced time for software development in scientific research, the focus tends to be on rapid code generation rather than architecture or project design, and explicit code documentation for further reuse or maintenance of tools after development are often missing.

To overcome the issues described, there are a number of practices that assist in achieving a better quality of the product in scientific development (101). Firstly, the choice of programming language is extremely important, as it must have a shallow learning curve for new developers, as well as overall high readability and versatility. Based on these criteria, Python is the main programming language in our department just like in many other research labs. Also, Python provides support for many scientific libraries, such as NumPy, SciPy, etc., making it easier for scientific software developers with different backgrounds to use already implemented complex algorithms for their own projects. These available community-proven packages make the code base more

reliable and maintainable, which allows the focus to be on the research questions rather than on the implementation details.

Furthermore, Open Science, as a set of practices that enhance openness, transparency, rigor, reproducibility and replicability of the scientific research, has recently become widespread in many scientific fields, including proteomics (102). This is partly due to the need to address the ‘crisis of reproducibility’ of published articles in all scientific fields, even in the natural sciences (103). An ‘open-source code’ base can benefit researchers by saving time and funding resources and by engaging the community in their research, which is especially important in rapidly evolving fields such as machine learning or visualization. Furthermore, public exposure motivates maintaining high standards of code quality, including comprehensive documentation and testing.

The natural combination of Python with Open Science empowers researches with a set of tools for a comfortable working environment. It allows to follow the standards of software engineering, i.e. to write self-explanatory code for data analysis and visualization, to store and share code and control its versions, and to apply authorship of the tools developed.

In this regard, Jupyter notebooks have become some of the most used tools for Python software development in various fields, including data science, machine learning, etc. (104). Jupyter notebooks can easily be adapted to the old concept of ‘literate programming’, introduced by Knuth in 1984, where code, analysis results, like visualizations, and static documentation are included in a single file (105). It additionally supports markdown syntax, which enables standard text formatting and the inclusion of complex elements such as images and formulas, providing additional information that simplifies the understanding of complex scientific concepts. Recent MS-based proteomics publications are also starting to provide analysis code written in Jupyter notebooks (106, 107).

Running on the local machine, Jupyter notebooks depend on the computer’s central or graphics processing units, CPU and GPU respectively, as well as the size of the random-access memory (RAM). Once the user begins to perform some complex manipulations, i.e. visualization of big data, this can quickly exceed the limits of the local machine. Community resources can be used to overcome these limitations. In particular, Google Colab, a free Jupyter notebook environment provided by Google, runs in the cloud and stores its notebooks on a connected Google Drive (108). This allows developing any code without depending on the computational power of the personal computer, and to

use freely available GPUs or even TPUs, tensor processing units built into Colab, to perform heavy computing tasks.

To provide access to the community, various resources such as source code, metadata, documentation, etc. can be hosted privately or publicly using Git and GitHub (109). GitHub is a free code hosting platform for software development and version control using Git. The GitHub repository enables to keep a history of all changes that have been made to code and text, making it possible to go back to earlier versions of the repository if needed. Developers and researchers alike can use GitHub as a dynamic and collaborative environment for continuous integration, often referred to as a social coding platform, for peer-reviewing, commenting or discussions.

Several archiving services, such as Zenodo (<https://zenodo.org>), are included in GitHub to apply authorship to the code. This allows the entire repository to be assigned a permanent DOI, a Digital Object Identifier, that can be included in literature information resources such as PubMed Central (110).

To make it easier for users to interact with published code and data locally and yet without additional installations, GitHub provides integration with online hosting solutions for Jupyter Notebooks such as Binder (<https://mybinder.org>). In this way, readers can execute code online without downloading any data or installing any software. Almost all of the aforementioned tools were used in the articles presented in this thesis.

### **1.5. Visualization in MS-based proteomics**

Visualization together with data processing and analysis are central and crucial components of all complex biological experiments, including all modern high-throughput MS-based proteomics experiments. As part of a complex proteomics pipeline, visualization assists in the interpretation of complex proteomics data and is key to communicate the results of complex experiments not only quantitatively, but also visually (111). Rapid advances in MS-based proteomics technology have prompted the development of new software tools in the field, including proteomics visualization. Currently, this proteomics visualization efforts can be divided into two main groups: (i) visualization functionalities integrated into proteomics data analysis tools (18, 112); (ii) independent visualization tools for different steps of the proteomics pipeline (113, 114).

Despite recent improvements in the area, there are still a number of challenges in proteomics data visualization. First of all, visualization is usually not a priority when developing scientific algorithms or creating new workflows for analyzing proteomics data. The release of visualization tools often occurs with a significant delay after the publication of the main workflows (22, 112). Another problem is the closed nature of

many popular proteomics tools. In this case, even recently published tools can rapidly become outdated, not allowing to extend functionality or add support for the latest visualization advances, such as interactivity or 'big data' visualization. Consequently, data analysis and visualization in proteomics often remain the exclusive ability of experts familiar with the data that have strong programming expertise.

To alleviate these issues, over the last decade the proteomics community has released a plethora of open-source visualization tools written in the programming languages R and Python (115, 116). In this thesis, I have extensively used Python to develop all visualization tools. In addition to its general advantages, described in the previous section, Python provides a large variety of well-documented and well-maintained visualization libraries. Some of the visualization packages I have regularly used for the projects presented in this thesis are Plotly, Bokeh, Datashader and Panel. The generation of interactive plots for exploratory data analysis, for instance in Bokeh (<https://bokeh.org/>) or Plotly (<https://plotly.com/>), allows on-demand access to data only, which is very important for high-throughput proteomics data, and includes various basic manipulation tools, like selection, zooming, saving, etc. In turn, the recently released Datashader library (<https://datashader.org>) efficiently handles big data visualization by rasterizing the data space like in the conventional histogram but using a two-dimensional (2D) space and color-coding the number of points per a 2D bin. This makes it much easier to distinguish patterns in big proteomics data without applying the usual workarounds, such as down sampling or reducing opacity. Another useful library is Panel (<https://panel.holoviz.org/>), which helps to combine data analysis and visualization approaches with modern web frameworks to create browser-based graphical user interfaces (GUIs) without any of the common technologies to develop a website or web application such as HTML or JavaScript. Due to the large amount of data acquired in proteomics experiments, much more biological insights can usually be extracted from the data than is presented in a single publication. Therefore, dedicated and easily built online resources or recently popular dashboards can help to solve this issue and provide additional access and visual inspection of data for all users. The popularity and usability of all the libraries mentioned is confirmed by the number of recently released visualization tools and resource pages for the field of proteomics (113, 117, 118).

The proteomics data visualization can be divided into several steps following the main phases of proteomics data analysis, which include visualization of (i) raw data at LC, MS<sup>1</sup> and MS<sup>2</sup> level; (ii) peptide identification with or without PTMs; (iii) quantitative information at protein, peptide and PTM level; (iv) multidimensional experimental designs; and (v) protein networks.

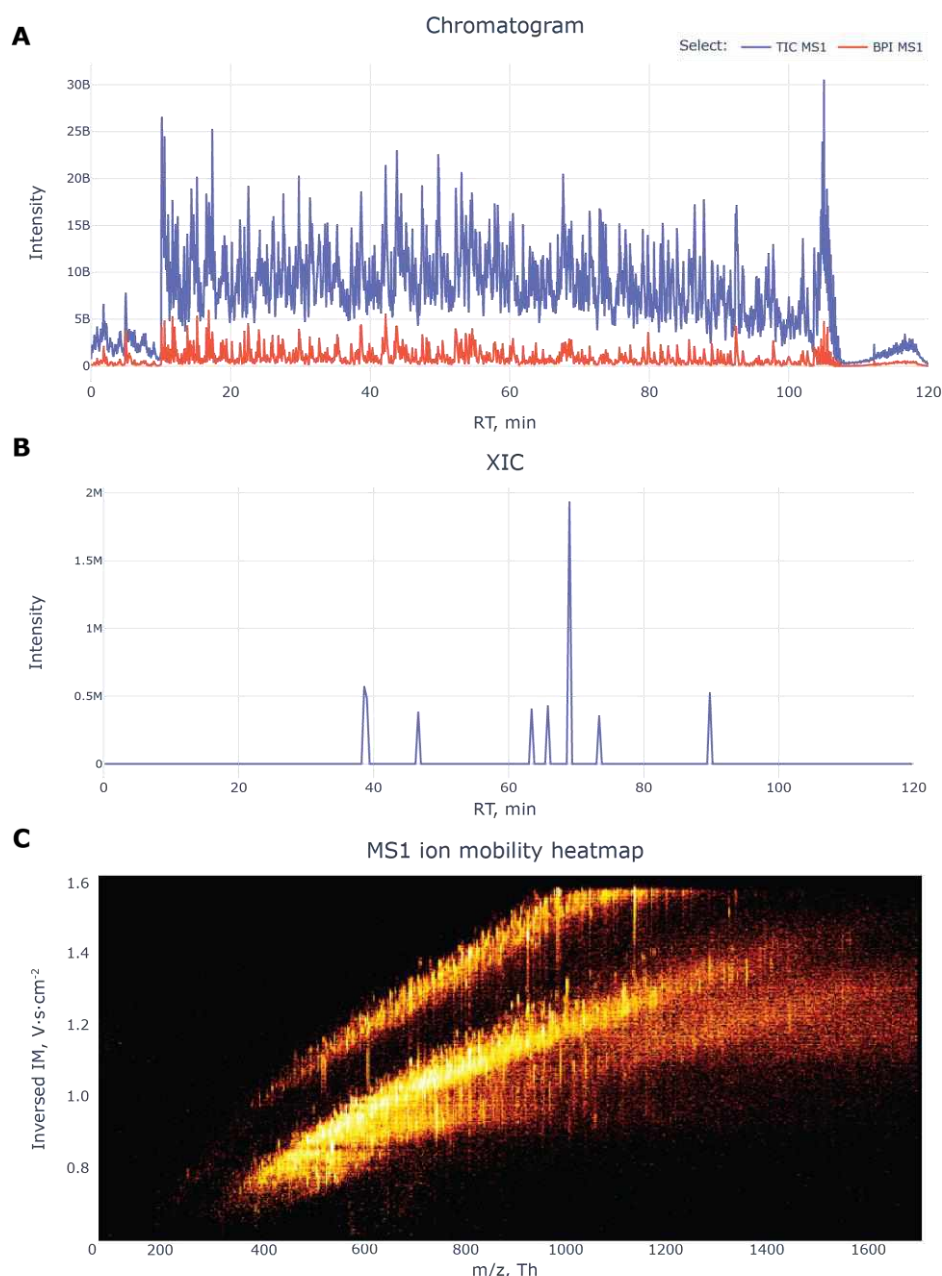
As all software tools presented in this thesis mainly cover visualization of raw data, as well as visualization of peptide identifications and PTMs, these topics will be covered in the following sub-chapters. More information on visualization of the other steps of the proteomics pipeline is contained in a recent review that I co-authored that discusses interpreting and generating bottom-up proteomics data visualizations (Article 1).

### 1.5.1. Raw data visualization

Raw data acquisition is an intermediate step in the proteomics pipeline between the experimental part in the laboratory and the data analysis part, and it is of utmost importance to ensure satisfactory quality of the raw data for further analysis. Data quality at this step is mainly assessed by visual inspection of the raw LC-MS data using various computational quality control tools, making it easier to reveal possible pitfalls in the samples or instrumentation setup (119). The software tools implemented in Articles 2 and 4 focus on the visualization of raw timsTOF data. Therefore, I will cover the standard important visualizations of raw MS data on precursor and fragment ion levels and how they can be interpreted and used to verify data quality.

**Precursor level.** Visualization at the level of intact peptide ions is the first step in checking the quality of sample data and helps to identify any LC or MS instrumentation issues. Firstly, the total ion chromatogram (TIC), which displays the summed intensity of all precursor ions detected over time, shows the number of precursor ions that reach the MS detector along the gradient (Fig. 12A, blue line). This step can already help to spot various LC-MS issues, such as mistakes in sample preparation (unexpected shape or low number of peaks, low intense peaks), unstable spray or MS failure (intensity drops) or poor peak separation (120). Visualizing another type of chromatogram, called base peak intensity (BPI) chromatogram and plotting the intensity of the most abundant precursor ions over time, can uncover other issues, such as sample overloading or contamination (Fig. 132A, red line). The quality of individual precursor ions can also be checked using extracted ion chromatograms (XICs) (Fig. 12B). In this case the raw data is sliced based on the mass and charge range within a given mass tolerance and its intensity is plotted against the retention time. This helps to assess the quality of the detected precursor features and the spread of contaminants in the sample.

To gain insight into the distribution of precursor ions along the gradient, precursor maps are widely used. In the case of timsTOF data, the signal intensity of the detected precursor is visualized in color on a heat map in the  $m/z$  - ion mobility plane for each time points and can be tracked over the entire retention time (Fig. 12C).



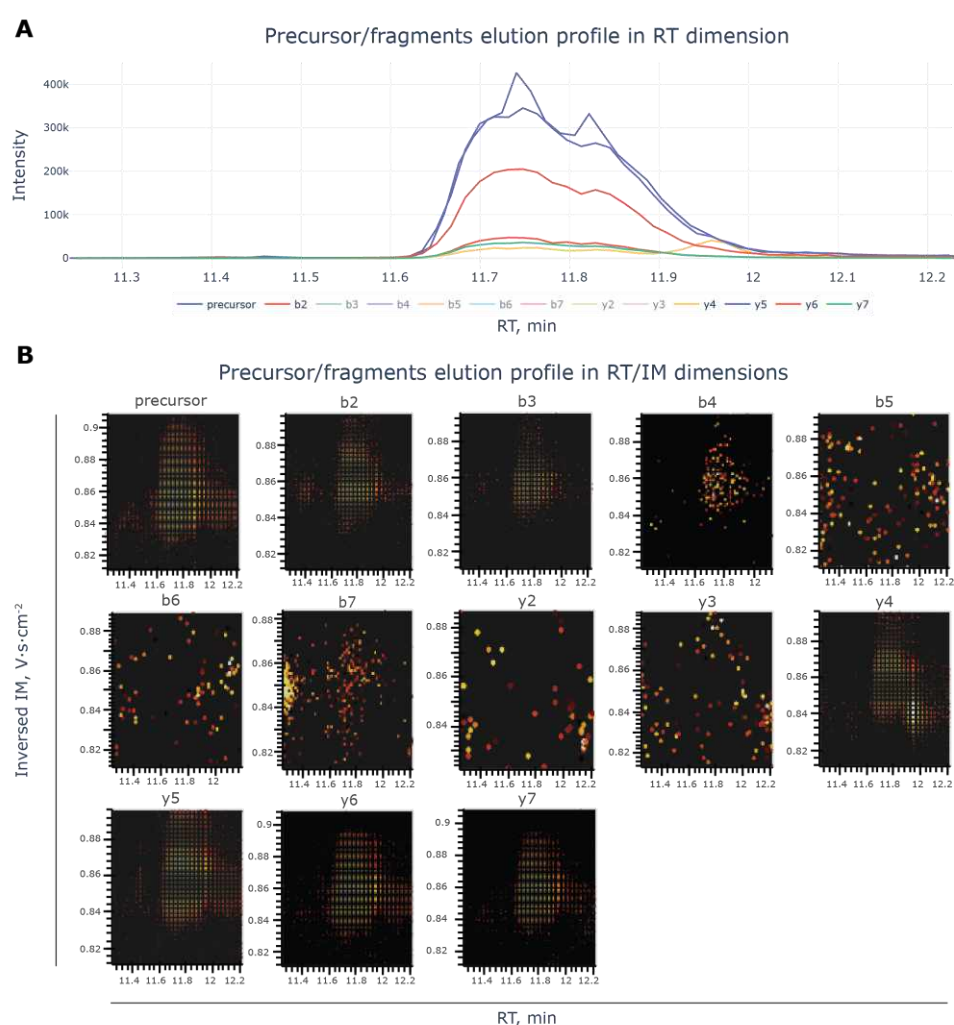
**Figure 12: Visualization of raw data at the precursor ion level. (A)** Total ion chromatogram (TIC) and base peak intensity (BPI) chromatogram of MS<sup>1</sup> data. **(B)** Extracted ion chromatogram (XIC) for the precursor ( $m/z = 457.9978$ ) with  $m/z$  tolerance of 5 ppm. **(C)** Two-dimensional MS<sup>1</sup> ion mobility heatmap of precursor intensities acquired on a timsTOF instrument at a single time point, demonstrating a correlation of  $m/z$  and ion mobility (Figure from Article 1).

**Fragment level.** Depending on the type of data acquisition strategy chosen, DDA or DIA, the information to be visualized at the fragment level is different. In the case of DDA data, it is essential to evaluate important MS<sup>2</sup> spectra and manually validate the identifications based on them. This can be done by plotting an MS<sup>2</sup> spectrum highlighting the identified fragment ions as shown in Figure 2. The pitfalls that can be revealed here are mainly related to fragmentation of the precursor ions, e.g. poor fragmentation



(intense precursor peak and few fragment ions) or co-fragmentation of several peptides (many additional fragments besides the identified ones are visible). The theoretical fragmentation spectrum (even with intensities predicted by deep learning) may be shown below the experimental spectrum as a mirrored spectrum. This helps to validate the identifications immediately showing which fragments are missing or (in)correctly identified in the experimental spectrum.

Due to the complexity of the MS<sup>2</sup> spectrum in DIA, it is more common to look at the elution profiles of the precursor and all its fragment ions in retention time or  $m/z$  – ion mobility dimensions to assess the quality of DIA identifications (Fig. 13). Ideally, they should have a sharp elution peak and high correlation between fragments. Misassignments are indicated by a shift of the fragment peaks or blending of additional peaks of individual fragments.



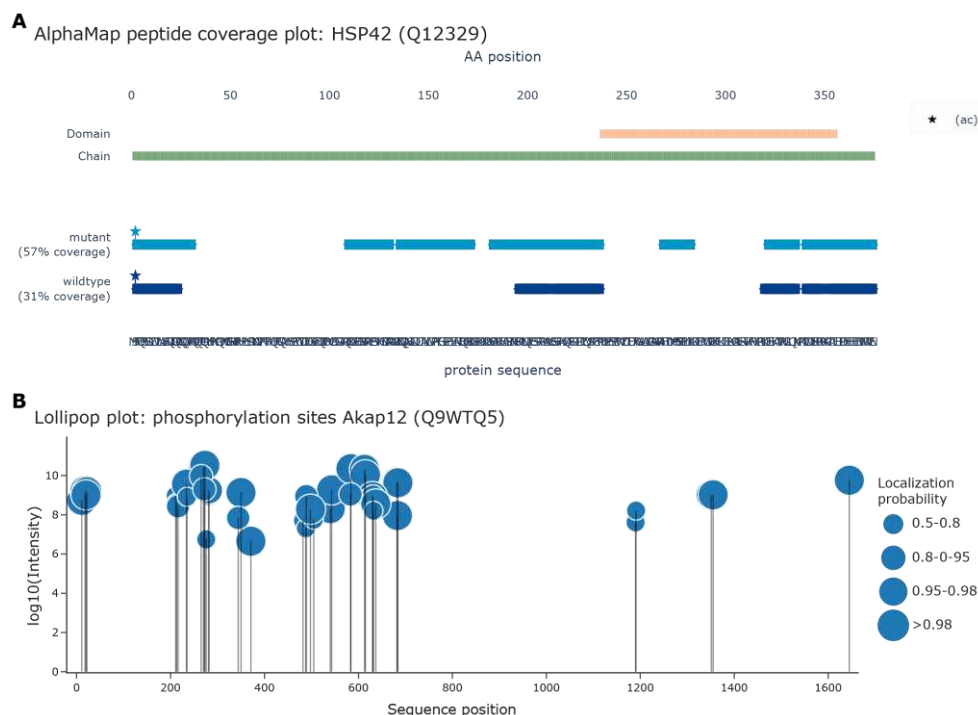
**Figure 13: Visualization of raw DIA data at the fragment ion level.** Elution profiles of the coeluted peptide precursor and its fragment ions in retention time (**A**) and retention time – ion mobility (**B**) dimensions (Figure from Article 1).

### 1.5.2. Peptide and PTM visualization

Identifications in bottom-up proteomics are always based on peptide rather than intact protein information. Therefore, visualization of the identified peptides and their PTMs aligned with known protein sequence information allows evaluation of protein coverage and is essential for downstream MS data exploration. Sequence coverage can be visualized along the protein sequence in a non-overlapping manner or collapsed into a single line per sample to avoid clutter, as in Article 3 (Fig. 14A). This makes it possible to assess differential sequence coverage across multiple samples or datasets acquired with different methods or analyzed by different software tools.

PTM studies can also benefit greatly from visualizations, where the position of PTMs, intensity or localization probability are shown for each modification site. This can either be done in a simple way where only PTM positions are shown (asterisk, Fig. 14A) or with a 'lollipop plot' showing the PTM site localization and intensity (Fig. 14B).

Peptide and PTM visualization can also benefit from the inclusion of sequence annotations available in public databases, such as UniProt or PhosphoSitePlus (121, 122), or any other useful information, i.e. the expected proteolytic cleavage sites. As described in Article 3, this helps researchers to inspect the peptide and PTM levels of a protein of interest in order to validate it in a biological and clinical context, i.e. by checking for possible sequence variations or unexpected anomalies.



**Figure 15: Peptide and PTM visualization. (A)** Peptide sequence visualization of mutant and wildtype samples with overlapping identified peptide collapsed into a single line, detected PTM (acetylation of protein

N-terminus) and external features (domain and chain). Figure created using AlphaMap (Article 3). **(B)** Lollipop plot visualizing phosphosites, their log<sub>10</sub> intensity and localization probability (bubble size) (Figure from Article 1).

## 2. Aims of the thesis

The aim of this thesis was to visually explore the newly emerging complex ‘next-generation proteomics’ data, which has been studied in our group since the first timsTOF instrument was introduced. Motivated by the lack of tools to effectively access, visualize and effectively process raw data, we set out to develop them for our own use and empower the community. This would enable exploration using different angles to understand more about the data itself, as well as to extract more information and biological insights from it.

Considering the computational perspective, we primarily sought to develop software tools enabling ourselves and, by extension, the entire proteomics community to explore the various phases of proteomics data analysis, such as fast data access, efficient data processing or comprehensive visualization as an essential step in understanding and validating the data. A red thread through all these computational projects was to use and build upon the various established scientific computing tools, in particular the Python universe. This common idea allowed for unified development, providing more support in the form of readily available and community-proven scientific libraries. In turn this allowed us to focus on in-house research questions instead of reimplementing existing algorithms. In line with the basic ideals of Python, I followed the concept of open-source software, making the tools and the source code freely available to the community. This motivates the community to use the tool, as well as to contribute ideas or their own implementations. Knowing that one does not need to implement an entire data analysis pipeline, but can simply update part of the code with some novel ideas, provides a good starting point even for a biological scientist and enables rapid progression of the field.

With these tools in hand or under development, I wanted to contribute to other aspects of data insight that remained challenging in the group. In terms of method development, the crucial point was to contribute to the implementation of the new scanning modes on the timsTOF instrument and to improve the existing ones. This helped to improve the sensitivity and to increase the throughput of the instrument, in addition to improved data quality.

In recent years, the field of proteomics has greatly benefited from the developments in machine learning and in particular deep learning. They are set to dramatically boosted the quality and reliability of proteomics workflows, as experimental results have to match predictions in a multidimensional data space. We applied this to the ‘next-generation

proteomics' data to understand the nature of collisional cross sections, as well as to be able to predict this dimension of separation to further improve validity.

All the knowledge gained was intended to also apply to the functional study of post-translational modifications, especially phosphorylation and its role in cell signalling.

In conclusion, the overall aim of my thesis was to enable the exploration and integration of several layers of information, including prior knowledge amassed by the community.

## 3. Publications

### 3.1. Article 1: A practical guide to interpreting and generating bottom-up proteomics data visualizations

Authors: Julia Patricia Schessner\*, Eugenia Voytik\*, Isabell Bludau

Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, 82152 Martinsried, Germany

\* These authors contributed equally

Published in *Proteomics* (2022).

Rapidly advancing mass spectrometry-based proteomics today allow in-depth analysis of highly complex proteomics mixtures. This results in the generation of a large amount of complex data that is often hard to interpret comprehensively. Adequate data visualization is a critical part of this proteomics data analysis process but is currently neglected. It could greatly assist in the interpretation of multidimensional proteomics data as well as in communicating the results of evermore complex experiments. However, due to the complexity of proteomics analysis, understanding the results is difficult for a broader audience, especially non-specialists, which slows down the dissemination of proteomics method.

In this review, we provide an overview of commonly used visualizations of the different steps of the proteomics pipeline. Covering the entire workflow, we describe the use cases and relevance of each visualization in proteomics, assisting researches with guidance as to which aspects are critical for interpretation and reporting. Moreover, an entire section of the review is devoted to how Python and various established open science tools can be used to transparently generate customized proteomics visualizations. To emphasize the importance of this and to help readers get started with their own visualizations, the code for generating all data figures from the review is provided on GitHub with all necessary documentation and examples. Finally, we also include a list of published Python libraries available for analysis and visualization of proteomics data.

I contributed to this review by helping to devise and writing the manuscript, as well as developing the Python codebase for figures.

Received: 22 September 2021 | Revised: 22 December 2021 | Accepted: 20 January 2022 | Accepted article online: 2 February 2022

DOI: 10.1002/pmic.202100103

**Proteomics**  
Proteomics and Systems Biology

## REVIEW

# A practical guide to interpreting and generating bottom-up proteomics data visualizations

Julia Patricia Schessner  | Eugenia Voytik  | Isabell Bludau 

Department of Proteomics and Signal Transduction, Max-Planck-Institute of Biochemistry, Planegg, Germany

## Correspondence

Julia Patricia Schessner, Department of Proteomics and Signal Transduction, Max-Planck-Institut für Biochemie, Abt. Mann, E03, Am Klopferspitz 18, 82152 Planegg, DE, Germany. Email: [schessner@biochem.mpg.de](mailto:schessner@biochem.mpg.de)

Julia Patricia Schessner and Eugenia Voytik contributed equally to this work.

## Funding information

Swiss National Science Foundation Postdoc.Mobility fellowship, Grant/Award Number: P400PB\_191046; Bayerisches Staatsministerium für Bildung und Kultus, Wissenschaft und Kunst, Grant/Award Number: Digimed Bayern; Max-Planck-Förderstiftung

**Abstract**

Mass-spectrometry based bottom-up proteomics is the main method to analyze proteomes comprehensively and the rapid evolution of instrumentation and data analysis has made the technology widely available. Data visualization is an integral part of the analysis process and it is crucial for the communication of results. This is a major challenge due to the immense complexity of MS data. In this review, we provide an overview of commonly used visualizations, starting with raw data of traditional and novel MS technologies, then basic peptide and protein level analyses, and finally visualization of highly complex datasets and networks. We specifically provide guidance on how to critically interpret and discuss the multitude of different proteomics data visualizations. Furthermore, we highlight Python-based libraries and other open science tools that can be applied for independent and transparent generation of customized visualizations. To further encourage programmatic data visualization, we provide the Python code used to generate all data figures in this review on GitHub (<https://github.com/MannLabs/ProteomicsVisualization>).

**KEYWORDS**

bottom-up proteomics, data visualization, open science, science communication

## 1 | INTRODUCTION

Mass spectrometry (MS)-based bottom-up proteomics allows comprehensive analysis of highly complex proteomes [1–6]. Thanks to recent technological advances that dramatically increased proteomic depth and throughput, MS technology is nowadays accessible to many non-expert labs either through core facilities or individual proteomics setups. Firstly, the field has witnessed a huge enhancement of instrumentation, exemplified by a new robust and high-throughput liquid

chromatography (LC) system [7] and new types of mass spectrometers allowing peptide separation by ion mobility [8–13]. Secondly, these advances were accompanied by the development of high-throughput data acquisition techniques [14–19] and a burst of computational methods for proteomics data analysis [20–24]. Facilitated by increasingly powerful computational hardware and programming backends, computational proteomics has evolved into an independent, multidisciplinary field, but now presents a new barrier to scientists lacking expertise either in proteomics or bioinformatics.

Adequate data visualization is crucial to interpret data and communicate results of evermore complex experiments [25, 26]. A variety of data analysis tools have integrated visualization functions to address this need [27–30], but visualization is usually not among the highest priorities in the development of novel data analysis workflows and is

**Abbreviations:** BPI, base peak intensity; DDA, data dependent acquisition; DIA, data independent acquisition; DOI, digital object identifier; FDR, false discovery rate; LC, liquid chromatography; MS, mass spectrometry; PC, principal component; PCA, principal component analysis; PTM, post-translational modification; TIC, total ion chromatogram; XIC, extracted ion chromatogram

This is an open access article under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *Proteomics* published by Wiley-VCH GmbH

*Proteomics* 2022, 22:2100103  
<https://doi.org/10.1002/pmic.202100103>

[www.proteomics-journal.com](http://www.proteomics-journal.com) | 1 of 18



often an afterthought. Consequently, data assessment, interpretation and visualization often remain exclusive abilities of experts familiar with the data and capable of handling it programmatically. This drastically slows down method dissemination and knowledge transfer to a broader audience from different research fields. Due to this required expertise, communication with non-experts in proteomics is often sub-optimal. While there are several reviews that either focus on stand-alone software tools [31, 32] or cover computational aspects of the visualization process by making an overview of available R libraries [33], they do not necessarily provide insight to non-experts in proteomics on why certain visualizations are important or how to interpret them.

In this review, we provide an overview of several common types of visualizations, focusing on their use and interpretation rather than the software. We also demonstrate how such visualization can be interactively created with Python, one of the most common programming languages in science that has a low threshold to learn and use. Following the main steps of proteomics data analysis, we first describe the visualization of raw data and peptide identification with a special focus on novel MS instrument types and data acquisition modes. Next, we cover the visualization of quantitative information on the level of proteins, peptides and post-translational modifications (PTMs). In light of the continuously increasing complexity of experimental designs, we also include strategies for visualizing multidimensional data and a primer on protein networks. For each visualization we describe its common use cases and relevance, what it shows, and what aspects of it are important for interpretation and reporting. In the final section, we describe how Python and community resources can be used to create and share customized data visualizations by utilizing both generic and specialized libraries. To make it easier for readers to adopt customized MS data visualization themselves, we provide fully documented Python code that was used to generate all data figures presented in this review on GitHub: <https://github.com/MannLabs/ProteomicsVisualization>. With this review we want to enable researchers working on interdisciplinary projects to (1) critically assess proteomics data visualizations in publications, (2) discuss effectively with experts, and ultimately (3) turn their own data into visualizations that optimally communicate their results.

## 2 | VISUALIZATION OF PROTEOMICS DATA

In brief, a standard MS-based bottom-up proteomics workflow can be described as follows (see fig. 1 in [6]). Proteins are enzymatically digested into short, MS-accessible peptides and separated using a LC setup that is directly coupled to a mass spectrometer (LC/MS setup). The MS then measures both intact peptide masses and the corresponding masses of peptide fragment ions that are generated on the fly, which is called tandem mass-spectrometry (LC-MS/MS setup). The resulting peptide and fragment ion spectra are then used to identify which peptides were present in the sample based on a reference proteome, commonly provided as species-specific protein FASTA file. With many

### Statement of significance

We review data visualizations used to evaluate and communicate bottom-up proteomics data. Critical aspects are explicitly explained by presenting concrete use-cases of raw and processed proteomics data. As practical guidance, we highlight publicly available Python-based tools and provide our own codebase for data visualizations that are presented herein. This should help the interdisciplinary use of bottom-up proteomics by ensuring a common ground for data communication and by enabling independent data exploration and visualization.

strategies available, identified peptides are then quantified and their information is aggregated to the protein level by protein inference. Strategies for peptide and protein quantification vary from absolute quantification within samples to relative quantification across samples. A more detailed introduction to bottom-up proteomics is available elsewhere [34]. In table 1 we provide an overview of the analysis steps, visualizations and most important pitfalls/best practices covered in this review. Many of the recommendations we make apply beyond the proteomics field and many statistical aspects are beautifully explained in the "Points of significance" series in Nature Methods.

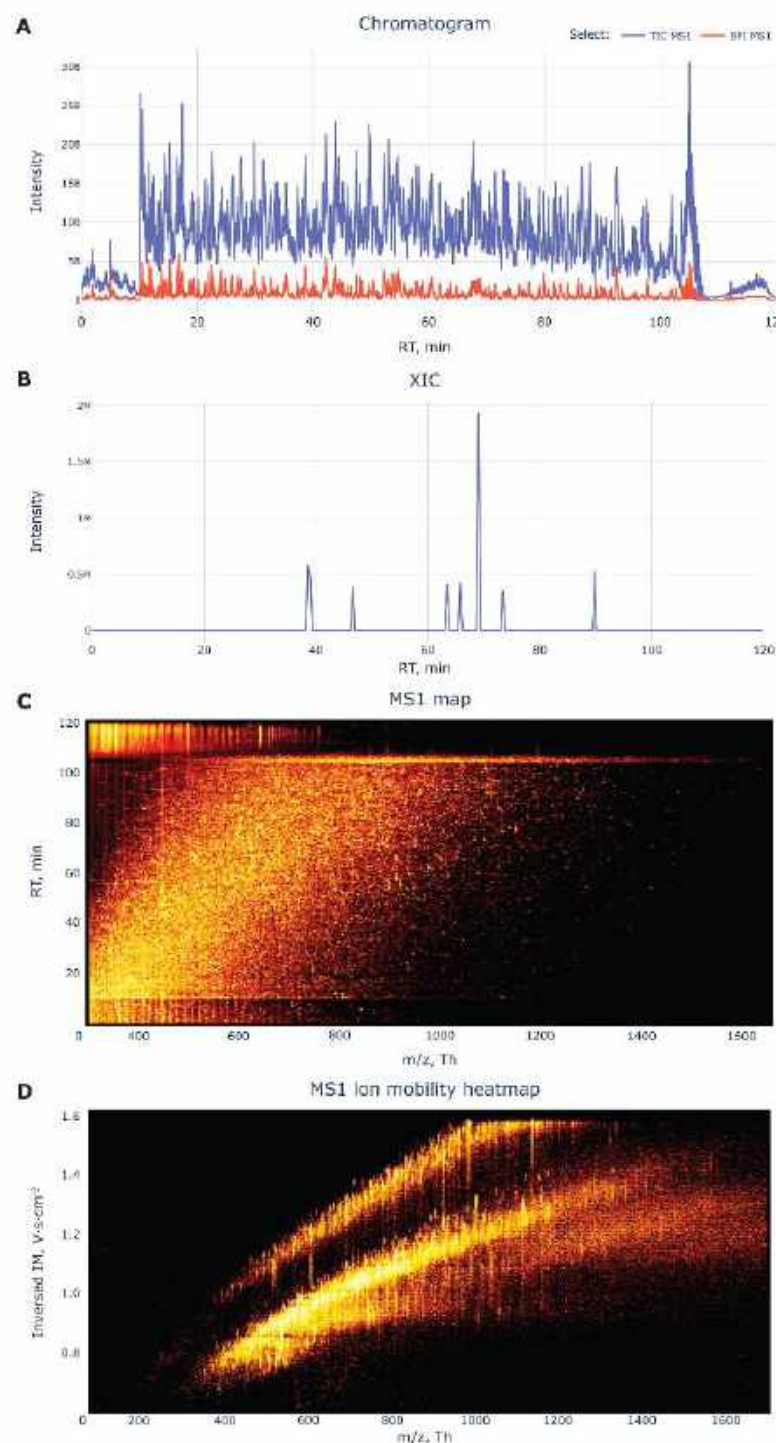
## 2.1 | Raw data visualization

At the heart of all proteomics projects is the raw data acquired by the MS [35] and unsatisfactory analysis results can often be traced back to low data quality. Evaluating the raw MS data quality is therefore a critical first step during data analysis, yet it is often neglected. Data quality is commonly assessed by visual exploration of the raw MS data, as it can reveal a variety of flaws of samples and instrumentation alike [31]. Alternately, various computational quality control methods are also available in the field and are extensively covered in literature [36]. In this section, we cover standard visualizations of raw MS data on precursor and fragment ion level and how to read them. For most of these visualizations either the MS vendors or the MS search software tools provide a graphical user interface. As one prominent option for visualizing data from public repositories we want to point out the PRIDE Inspector [37].

### 2.1.1 | Visualizations at the precursor level

**Ion chromatograms.** The first steps of data quality control should always include a performance assessment of the LC and the MS. This is commonly done by inspecting how many precursor ions reach the MS detector over time, visualized in the total ion chromatogram (TIC), showing the summed intensity of all detected precursor ions





**FIGURE 1** Visualization of proteomics data at the precursor level. (A-C) For these subFigures a dataset [62] from PXD012867 is used. (A) Total ion chromatogram (TIC) and base peak intensity (BPI) of MS1 data from 2 h nanoLC gradient measured on an Orbitrap based instrument. Low signal stretches in the first and last 10 min are due to loading time and LC flushing respectively. (B) Extracted ion chromatogram (XIC) for the analyte ( $m/z = 457.9978$ ) with 5 ppm  $m/z$  tolerance. (C) Two-dimensional MS1 map showing the intensity of observed precursor masses across the whole retention time. (D) Two-dimensional MS1 ion mobility heatmap of precursor intensities acquired on an ion mobility separating time-of-flight instrument at a single time point, demonstrating a correlation of  $m/z$  and ion mobility (PXD017703, [107]).

**TABLE 1** Overview of all visualizations presented in this review, including associated data, analysis steps, and pitfalls/recommendations

| Data                         | Analysis step  | Visualizations   | Figure     | Pitfalls/recommendations  |
|------------------------------|--|--|------------|---|
| MS1 raw data                 | Inspection of MS1 ion chromatograms to identify instrumentation and loading issues | Total ion chromatogram (TIC), Base peak intensity (BPI)    | 1A         | Compare to a high-quality reference chromatogram matched by instrument, gradient, and sample complexity.                    |
|                              | Analysis of individual elution profiles  | Extracted ion chromatogram (XIC)                           | 1B         | Mass range is critical: wide enough for mass errors, tight enough for specific selection.                                   |
|                              | Inspection of precursor maps to identify instrumentation issues                    | Two-dimensional precursor maps                             | 1C, 1D     | Compare to a high-quality reference map. Different dimensions can be displayed.   |
| MS2 raw data                 | Inspection of DDA peptide fragmentation  | (mirrored) MS2 spectra and sequence fragmentation          | 2A, 2B     | Number of fragments is crucial.   |
|                              | Inspection of DIA peptide fragment groups  | Two-dimensional or Three-dimensional elution profiles      | 2C, 2D     | Elution peak shape should be highly correlated across fragments.  |
| Peptide/PTM data             | Map identified peptides to protein sequence  | Non-overlapping traces                                     | 3A         | Missed cleavages and repeated fragmentation are apparent.   |
|                              | Map differential sequence coverage and external sequence features/PTMs             | Overlapping traces per condition + external traces         | 3B         | Missed cleavages are hidden in favor of differential coverage.  |
|                              | Map PTM positions and quantities to sequences                                      | Lollipop plot  | 3C         | Different quantitative measures can be shown on y-axis.   |
| Protein intensities          | Dynamic range and normalization  | Intensity histogram(s)                                     | 4A         | Replicates should have similar shape.   |
|                              | Proteome coverage  | Protein rank plot  | 4B         | Lower tail reveals depth limitation.  |
|                              | Proteome correlation and reproducibility   | Pairwise correlation plots and sample correlation heatmaps | 4C, 4D     | Use for small and high numbers of samples respectively.   |
| Two-condition comparisons    | Differential expression analysis by two-tailed tests                               | Volcano plots with square cutoffs/non-linear volcano lines | 4E, 4F     | Multiple hypothesis correction is mandatory. FDR and power (square cutoff) or $sD$ (non-linear cutoff) need to be reported. |
|                              | Enrichment analysis (e.g., by Fisher's exact test)                                 | Variable visualization depending on experiment complexity  | 4G         | The $p$ -value is the most important parameter to display if fewer visual channels are available.                           |
| Multidimensional experiments | Dimensionality reduction to display complex datasets                               | Two-dimensional projection of proteins                     | 5A, 5D, 5E | Algorithm determines topology (PCA/UMAP/tSNE).  |
|                              | Reproducibility by PCA   | PCA loadings plot  | 5B         | Replicates should cluster.  |
|                              | Variability contribution in PCA  | Bar chart with all PCs                                     | 5C         | Main discriminators of samples can be identified.   |
|                              | Cluster analysis to group proteins and/or samples                                  | Heatmap with marginal dendrograms                          | 5F         | Distance measure and clustering algorithm are key parameters, cutoffs are largely arbitrary.                                |
|                              | Display all features for a subset/summary of the data                              | Profile plot/parallel coordinates/radar plot               | 5G-I       | Selection depends on data types and visibility of the key result.   |
| Protein networks             | Display protein distances  | Weighted edge network                                      | 6A         | Avoid hairballs, by parsimonious selection of nodes and edges, use a deterministic layouting algorithm.                     |
|                              | Display hierarchical groups  | Hierarchical network                                       | 6B         | Depends on underlying grouping.   |
|                              | Display biological processes   | Semantic network   | 6C         | Indicate source for relationships.  |

against the retention time (blue line in Figure 1A). Problems that can be revealed inspecting the TIC are poor peak separation (very broad peaks), unstable spray or MS failure (intensity drops) and mistakes in sample preparation (low intensity, few peaks, unexpected overall shape) [38, 39]. Another major issue is saturation of the whole LC-MS system, for example, by overloading or contamination. This can be revealed by the base peak intensity (BPI) plot, which shows the intensity of the most abundant ions detected over time (red line in Figure 1A). If the system is saturated one can see plateaus in the BPI trace. It is generally advisable to have a reference TIC and BPI plot for the sample type and instrument setup used to be able to detect anomalies.

It can further be important to follow up on individual detected ions or groups of ions, to evaluate, for example, the spread of contaminants, the peak shape of quality control ions or the quality of identified peptide features. To this end, extracted ion chromatograms (XICs) are commonly used (Figure 1B). The desired mass and charge range is extracted from the raw data and its intensity is plotted against the retention time. In doing so it is important to set adequate boundaries to the mass range ( $m/z$  tolerance), accounting for mass errors and coeluting ions.

**Precursor maps.** To get an overview of the whole range of precursor masses detected along the retention time, a two-dimensional MS1 map can be used [40, 41]. It shows the intensity (color) of observed precursor masses (x-axis) across the chromatographic retention time (y-axis) as a heatmap (Figure 1C). Same as for the TIC, it is advisable to have a reference image for this to be able to see anomalies, as they could again hint at technical issues with the instrument.

Recent developments in MS instrumentation introduced ion mobility as an additional separation dimension [9, 11, 13], which should be evaluated in a similar way as the  $m/z$  dimension. Akin to the two-dimensional MS1 map, precursor signal intensities can be visualized in the ion mobility dimension against the  $m/z$  dimension (Figure 1D). This heatmap would be even more informative if it showed the intensity across all three dimensions (retention time, ion mobility and  $m/z$ ). While this is in principle possible, the resulting visualizations are hard to interpret intuitively and improving them is one of the remaining challenges in proteomics data visualization [42].

### 2.1.2 | Visualizations at the fragment level

The first principal step of aggregating raw MS spectra into proteomic data is the identification of analyzed peptide sequences. The two required elements for sequence identification are the measured peptide fragment (MS2) spectra and the sequence search space, both of which depend on the acquisition mode and to a lesser extent the quantification strategy used [43]. We cover label-free data-dependent acquisition (DDA) and data-independent acquisition (DIA) here.

**DDA.** In the classical DDA approach the MS instrument isolates and fragments individual selected peptide ions from the precursor scan (MS1), most commonly the top- $N$  most intense ones. The spectra are then searched against a sequence database that contains masses,

sometimes also intensities, of peptide fragments from *in silico* protein digestion and fragmentation [44–46].

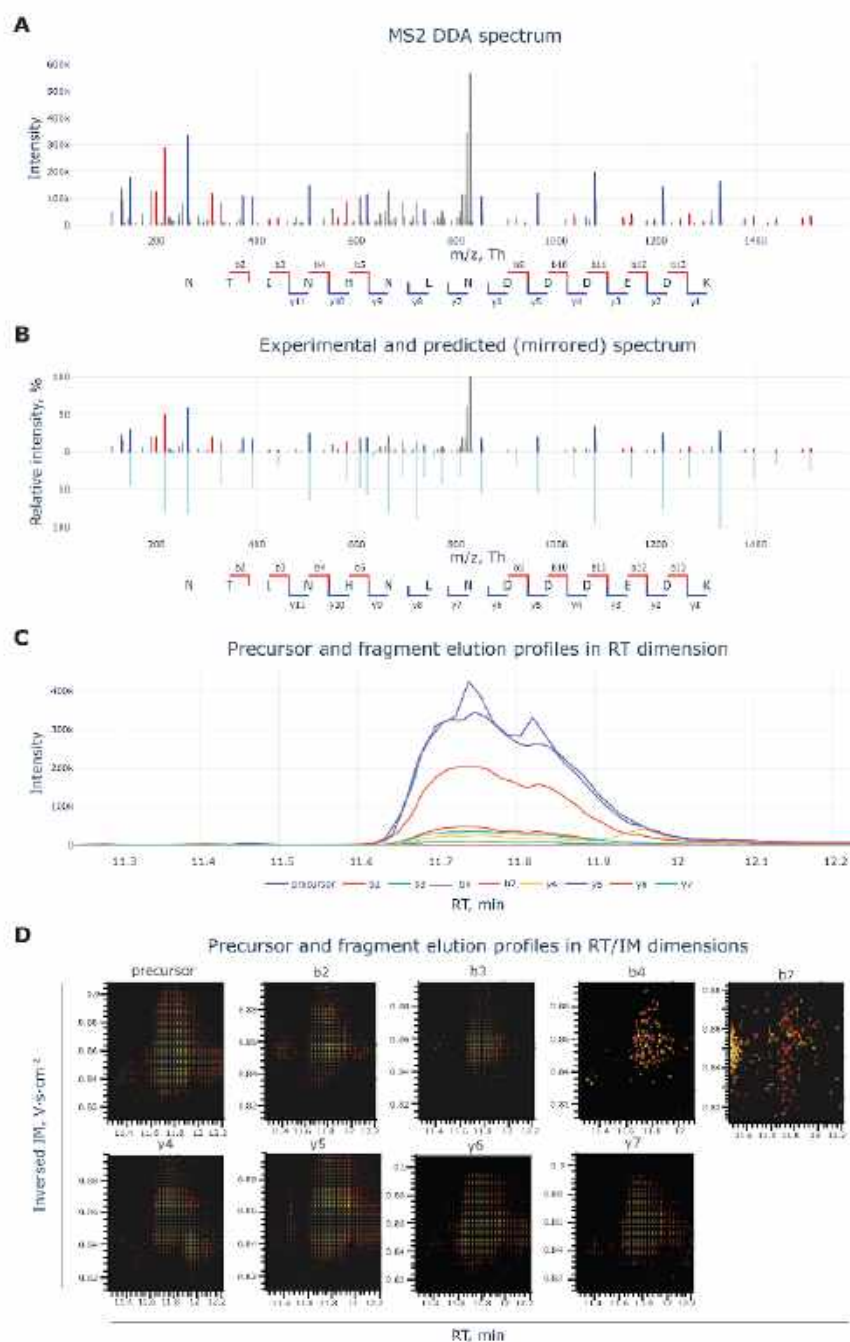
It can be important to manually evaluate the MS2 spectra and the identifications based on them, particularly when follow-up experiments hinge on a single or few proteins or even peptides. To do so, one can look at the individual MS2 spectra, highlighting the N-terminal and C-terminal fragment ions of the single selected precursor (Figure 2A). Underneath the spectrum itself, the sequence of the identified peptide and the position of identified N/C terminal fragment ions are indicated. Depending on the exact fragmentation method used, the peptide bond breaks at different positions, yielding different pairs of ions, most commonly b/y ions. Issues that can become apparent here are co-fragmentation of several peptides (many more fragments visible) or other isotopes of the same peptide (isotopic clusters for fragments), or poor fragmentation (very few ions and intense precursor peak). To check the quality of the peptide-spectrum-match against the library, mirrored spectra are commonly used (Figure 2B). Here the theoretical fragment masses are shown on a mirrored y-axis, which makes it immediately apparent which fragments are missing or should correctly be identified in the measured spectrum.

**DIA.** In DIA mode, instead of isolating a single precursor mass, mass ranges containing multiple precursors are isolated and fragmented for every MS1 scan, covering more precursors, but yielding more complex MS2 spectra. For a general introduction to DIA we suggest this review [49].

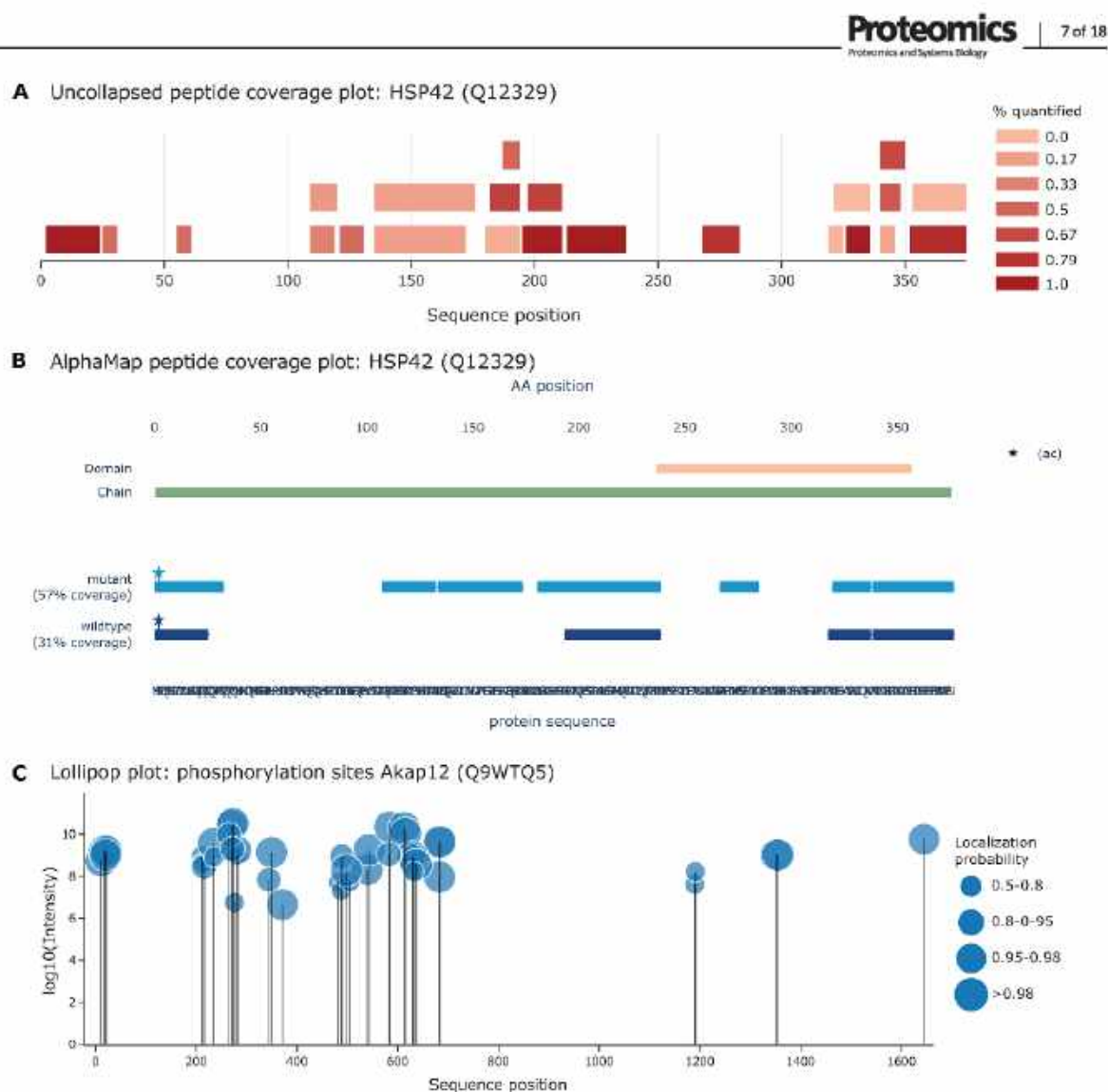
Due to the increased complexity, the simple MS2 spectrum visualizations lose most of their relevance and a spectral library containing only masses and intensities is no longer sufficient for identification. DIA libraries therefore additionally contain the retention time and if applicable the ion mobility of the precursor ions to narrow the search space at each time point [20, 48–50]. On top of the fragment masses, the exact coelution of fragments and their precursor is now crucial for scoring candidate identifications. To assess the quality of DIA identification, it is therefore most common to look at the elution profiles of all fragments associated with a specific precursor. Ideally, they should form a single sharp peak together with the precursor (Figure 2C). Indicators of peak misassignment would be peak shifts or blending additional peaks of individual fragments. Here, measuring ion mobility can lead to higher confidence, as fragments should correlate along this dimension as well. Both dimensions together can be visualized in heatmaps for the precursor and all its fragments in retention time and ion mobility space, colored by intensity (Figure 2D).

**Additional complexity.** Independent of the acquisition mode MS spectra can be complicated by peptide modifications, but the same visual techniques apply. Modifications can be either biologically generated PTMs (e.g., phosphorylation) [51, 52], artifacts introduced during sample preparation (e.g., oxidation) [53] or sample labelling techniques (e.g., TMT [54] or EASiTag [55]). Depending on the exact type, modifications lead to additional peaks for neutral losses or reporter ion series in MS2 spectra, or even require an additional level of fragmentation (MS3) to acquire additional fragments. To interpret these complex spectra more specialized background knowledge that goes beyond the scope of this review is required.





**FIGURE 2** Visualizations of proteomics data at the fragment ion level. (A) Peptide MS2 spectrum generated by data dependent acquisition (PXD012867, [62]). The peptide sequence is annotated with the identified b- and y-ions. (B) Mirrored MS2 spectrum showing the experimental (top) and predicted (bottom) spectra for the same peptide as in A, confirming the correct identification (PXD012867, [62]). (C-D) Coelution of a peptide precursor and its fragment ions acquired on an ion-mobility separating time-of-flight instrument (PXD017703, [107]). (C) Extracted ion chromatograms in the elution time window of precursor and fragments nicely overlap. (D) Heatmaps of ion intensities in ion-mobility and retention time dimensions provide additional information on coelution in the ion-mobility dimension.



**FIGURE 3 Peptide visualization.** (A) Figure displaying peptide coverage along the protein sequence, overlap between peptides and identification frequency (color scale) (PXD012867, [62]). (B) Figure displaying differential peptide coverage across sets of samples with overlapping peptides collapsed into a single trace, PTMs (here only n-terminal acetylation) and external features (PXD012867, [62]). Generated using [58]. (C) Lollipop plot displaying phosphosites, their intensity and localization probability (bubble size) (PXD010697, [77]).

## 2.2 | Peptide and PTM visualization

When moving from raw data to aggregated peptide and protein quantifications, it is important to point out again that all bottom-up proteomics data is based on the identification of peptides rather than intact proteins. Therefore, assessing the coverage of protein sequences with identified peptides provides essential information. Sequence coverage can for example be assessed using a Figure in the style of the PeptideAtlas [56] (Figure 3A). Here, all unmodified peptides are displayed in a non-overlapping way along the protein sequence and are colored by their identification frequency across samples. This representation

is well suited for assessing the reproducibility of peptide identification and to evaluate peptide overlaps caused by missed peptide cleavages. To evaluate differential sequence coverage between samples, overlapping peptides should be collapsed to a single line per sample to avoid clutter (Figure 3B).

If PTMs are measured, their position, intensity and localization probability can be visualized per modification site. If only the position needs to be visualized in the context of identified peptides, they can simply be added to these peptide views (start mark in Figure 3B). If a PTM's intensity and/or site probability are of interest a lollipop plot can be used (Figure 3C). These can for example be found on Phospho-

SitePlus [57]. Here, the size of the markers reflects the site probability and their vertical position reflects the intensity. For any of these visualizations it can be very informative to include additional annotation traces, for example, showing tryptic cleavage sites, and protein domains. This is for example possible using AlphaMap [58], which was also used here to create Figure 3B. With these visualizations in hand, various aspects of observed peptide and PTM signals associated with a protein of interest can be visualized and easily compared with data available in external databases. In doing so it is important to keep in mind that not all peptides are unique for just a single protein [59, 60].

### 2.3 | Protein quantity visualization and basic analysis

Aggregating peptide quantifications into protein quantifications is anything but a trivial task and highly depends on the inference strategy and quantification method used [61]. Agnostic to the quantification method, the assignment of peptides to proteins is not always uniquely possible, and therefore proteomics studies often talk about protein groups [59, 60]. These usually consist of any number of proteins that could be contained in the sample based on a set of shared non-unique, or “razor”, peptides identified. Most protein groups consist of genetically closely related proteins, like isoforms or paralogs. From here on out we will focus on the analysis of protein groups independent of the inference and quantification method used, but want to point out that each quantification method comes with individual parameters and visualizations used for quality control. All following visualizations can in principle also be applied on the peptide level, but are mostly used on the protein level. We will start with the evaluation of single condition samples and simple two-condition comparisons by the example of a knock-out versus wildtype experiment [62] and then move on to more complex experimental designs and protein networks in the following sections.

**Range and reproducibility.** Once protein groups are quantified the first thing to look at is the distribution of their intensities. This is frequently done using log-intensity histograms (Figure 4A) or boxplots. These can indicate if certain samples have different intensity distributions, which might necessitate normalization, or a significantly reduced depth. They can further be used to assess the distribution of certain protein categories relevant to the downstream analysis, like imputed values or reverse database hits as in Figure 4A.

The dynamic range of a dataset is another important parameter as the measurement of low abundant proteins is a major limitation in untargeted bottom-up proteomics. To display it, a protein rank plot can be used (Figure 4B). Depending on the quantification method and the downstream processing, the y-axis can represent either raw intensity units or estimates of absolute protein quantities (e.g., iBAQ [63], proteomic ruler [64]). In full proteome studies, the highest abundant proteins typically include cytoskeletal and ribosomal proteins and, depending on the proteomic depth, the lower tail includes, for example, signaling proteins and transcription factors.

Next, it is important to assess the reproducibility of replicate samples and the general similarity of samples to compare. For a limited number of samples, multi-scatter plots displaying all pairwise log-intensity distributions and their correlations can be used (Figure 4C). For larger numbers of samples, where a visualization of all sample pairs is no longer feasible, reproducibility can be assessed by a heatmap of correlation values (Figure 4D), or alternatively by principal component analysis (see next chapter).

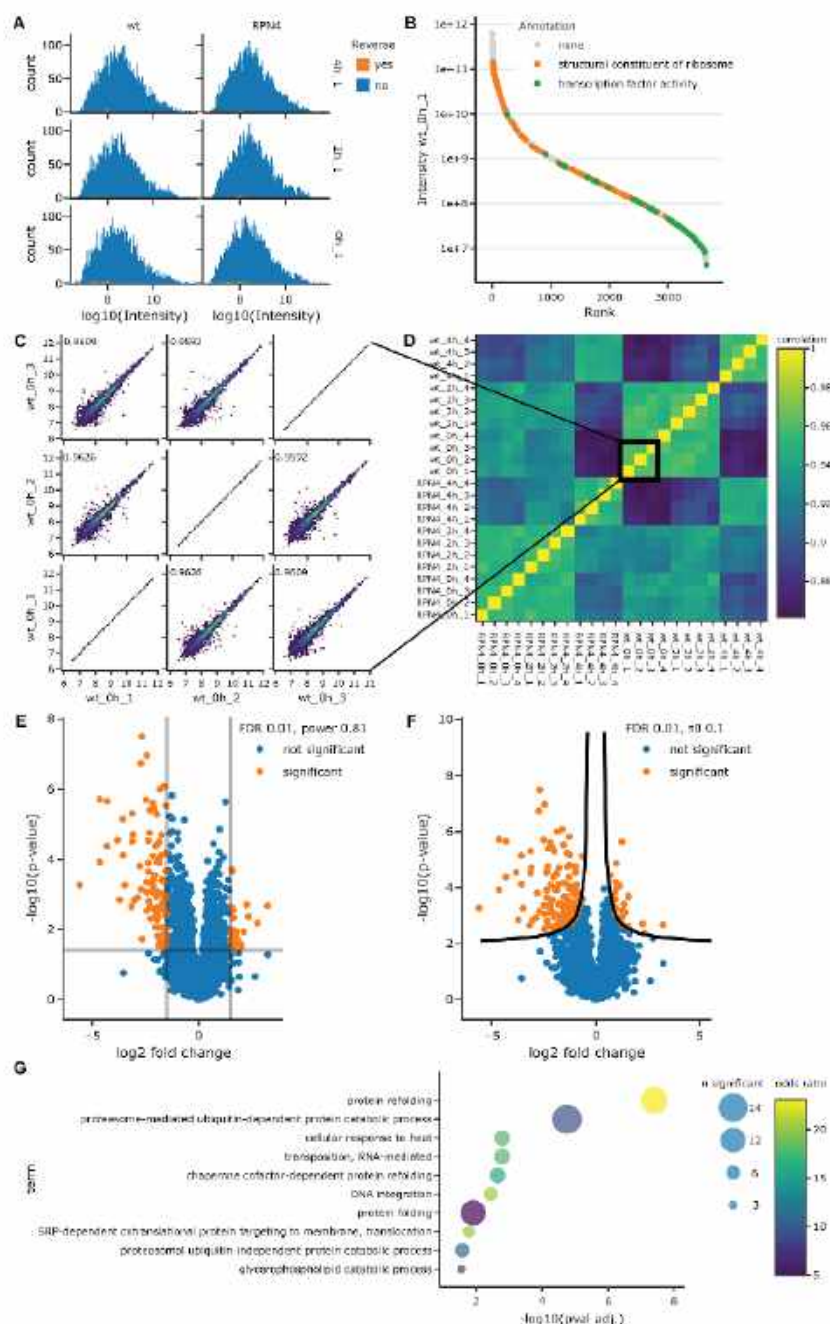
**Volcano plots.** The minimal comparative experiment spans two conditions with  $n$  biological replicates each. The standard analysis workflow for this is to perform multiple hypothesis corrected two-sample (Student's  $T$ -) tests [65, 66]. The multiple hypothesis correction is essential in any proteomics experiment, as  $p$ -values can be seemingly significant (i.e., very small) just by chance when making thousands of comparisons from the same dataset at once. Plotting the negative  $\log_{10}$  of the (corrected)  $p$ -value against the difference in log-space for each protein leads to the classical volcano plot (Figure 4E-F).

The thresholds for calling a protein differentially abundant can be determined by one of two methods: (1) square cutoffs for  $p$ -value and fold-change (Figure 4E), or (2) non-linear volcano lines (Figure 4F). (1) For square cutoffs, the horizontal threshold is selected based on a desired multiple hypothesis testing corrected  $p$ -value (or FDR). The vertical fold-change cutoff is set with regard to the experimental power, which is the probability of detecting an effect of a certain size, given it actually exists. When using square cutoffs, the power should always be indicated as in Figure 4E, regardless of whether a fixed power is used to calculate the fold-change cutoff or the other way around [67]. (2) For nonlinear volcano lines, an  $s_0$  parameter is set instead of a specific fold-change cutoff [68]. The  $s_0$  parameter is added as a constant to all standard deviations used in the  $t$ -tests and can roughly be interpreted as the assumed systematic error of the measurements, thereby setting a lower bound on the fold-change as a function of the measured standard deviation.

In both methods the boundaries on the fold-change ensure that the biological variability exceeds the numerical variability introduced by measurement noise or imperfect normalization. Both methods are valid if applied correctly, but yield slightly different hitlists and are both highly dependent on the arbitrarily selected parameters. It should also be kept in mind that either method still has a false discovery rate and protein groups can be on either side of the boundaries by mistake. The boundaries rather serve the purpose of generating a statistically sound list for further downstream analysis. Importantly, multiple hypothesis correction always has to be performed and documented. Usually this is done either by Benjamini-Hochberg correction or by performing a permutation test. For square cutoffs the y-axis usually shows the corrected  $p$ -value (not done here to ease comparison).

**Enrichment analysis.** One common analysis to do downstream of a volcano analysis is to look at overrepresentation of biologically relevant groups of proteins (e.g., biological pathways of cellular compartments) in the hitlist compared to the overall proteome (methods reviewed in [69]). This is usually done by a Fisher's exact test [70] or gene set enrichment analysis (GSEA, [71]) based on systematic annota-





**FIGURE 4** Dataset properties and two-condition comparisons. The data displayed in this Figure is taken from [62], where the principal comparison was drawn between wildtype and  $\Delta$ RPN4 budding yeast cells (PXDO12867). (A) Intensity histograms showing the distribution and number of protein groups are used to assess sample comparability. Hits from the reverse decoy database are annotated. (B) Protein rank plot from highest to lowest abundant proteins, illustrating the dynamic range. (C) Pairwise correlation plots demonstrate the biological and technical reproducibility. (D) Sample correlation matrix that is suitable to higher sample numbers than the pairwise correlation plot. It additionally illustrates sample grouping. (E, F) Volcano plots showing results of comparisons between two conditions, here between wildtype and  $\Delta$ RPN4 samples. Multiple hypothesis testing was done by permutation and the FDR was set to 0.01. (E) Square significance cutoffs with minimal  $\log_2$  fold change set to 1.5, which has a statistical power of 0.81. (F) Nonlinear volcano lines based on  $sD = 0.1$  adjusted  $p$ -value. (G) Enrichment analysis by Fisher's exact test for significant proteins from F. FDR = 5% after Benjamini-Hochberg correction. For all significant terms the corrected  $p$ -value, group size and the enrichment factor are displayed.

tions available, for example, through gene ontology [72,73]. Often this is done using online tools that use the whole theoretical proteome as background. However, bottom-up proteomics is not able to quantify all proteins and unidentified proteins should not be included in the background for an enrichment analysis [74]. Thus, only tools that can consider the specific background should be used (e.g., String [75] or Panther [76]). The three main values resulting from an enrichment analysis per candidate group are enrichment factor, group size and multiple hypothesis testing corrected *p*-value, which can be visualized together (Figure 4G). From this one could now draw biologically relevant conclusions, linking the prior difference between the compared samples to enriched sets of protein groups. If differential enrichment in several samples is displayed, the x-axis can be used to display the different samples and the size can be switched from group size to *p*-value. Perseus is a common tool to generate many of the aforementioned visualizations and to run most underlying analyses, including the enrichment analysis [29]. However, given the output of the statistical analysis almost any comprehensive visualization tool can create these Figures.

## 2.4 | Multi-conditional and multidimensional experimental designs

With increasing throughput, thanks to improvements in MS instrumentation, more complex experimental designs became practical. Common multi-conditional designs include time course experiments [77] and profiling experiments across subcellular compartments [78] or protein complex fractionation [79]. Two- and multi-conditional designs can further be combined into multidimensional experiments with each other (e.g., measuring subcellular profiles over time [80] or in different genetic backgrounds [78]) and with additional variables (e.g., demographic parameters in clinical sample cohorts [81]). In this section we use a comparative spatial proteomics dataset [78] for demonstration purposes.

**Dimensionality reduction.** While the full scope of a two-condition experiment can easily be displayed in two-dimensional, higher dimensional experiments require dimensionality reduction for visualization. Just selecting two dimensions can be useful if a direct comparison is needed, but this will always disregard biological variability added by other dimensions. This is problematic because it can mask correlated or orthogonal effects.

One universal tool to incorporate these effects into dimensionality reduction is PCA [82]: The data is usually scaled and log-transformed and then linearly transformed onto a new coordinate system, such that the first component describes the largest fraction of the overall data variability and successive components decreasingly less. This effectively aggregates a large fraction of the data variability into fewer dimensions. This serves three purposes: First, any number of dimensions can be reduced to the main PCs to visualize all proteins and their annotation groups in two-dimensional (Figure 5A). Second, the contribution of each original dimension to the PCs (loading plot) serves as quality control for sample grouping (Figure 5B), where tight clustering of replicates should be apparent. Third, the variability contributed by

each PC can inform on the independence of the acquired dimensions (Figure 5C). If many PCs have a similar contribution to the overall variability, this indicates independent underlying variables. In contrast, a single high variability PC often indicates that several of the underlying variables are at least partially dependent.

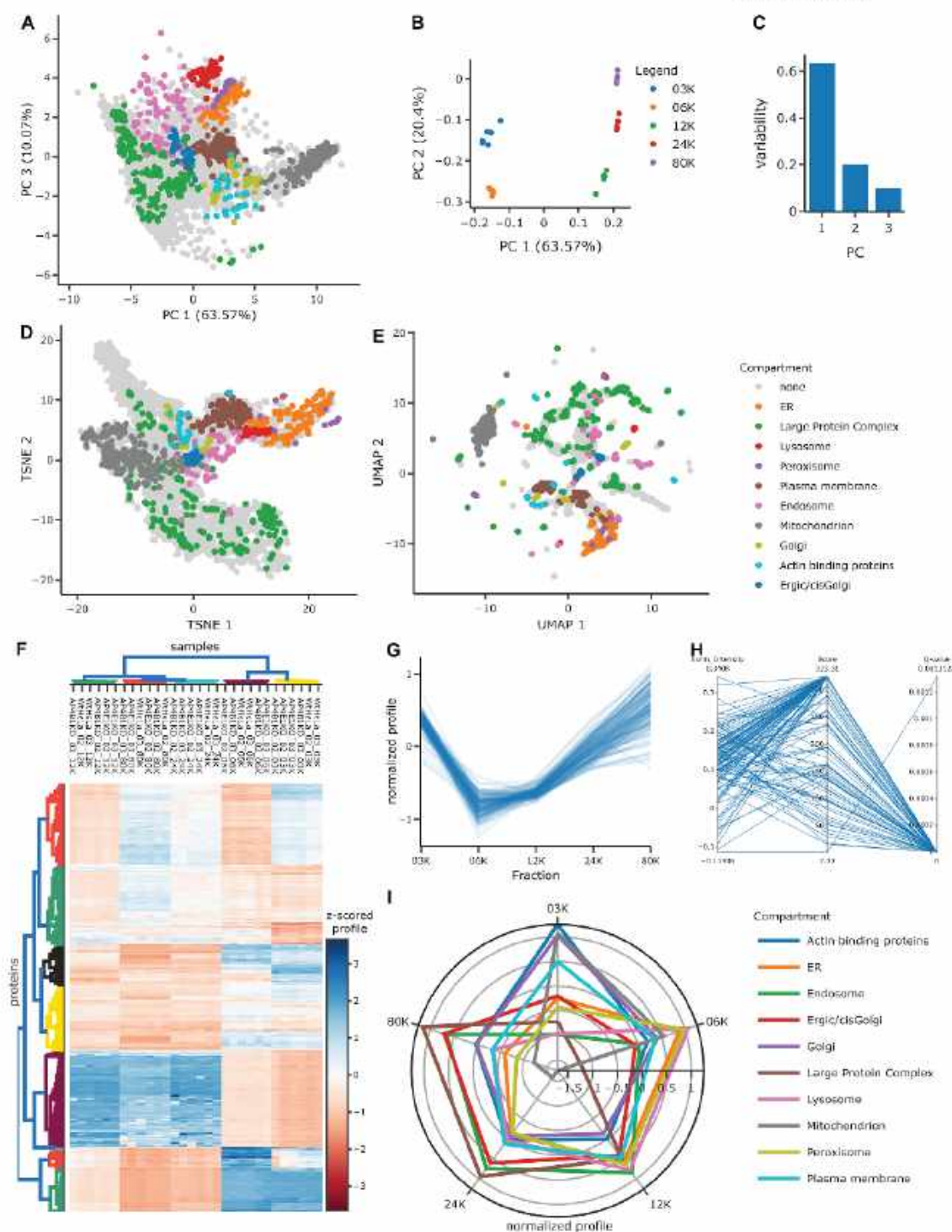
Other dimensionality reduction algorithms are tSNE [83] (Figure 5D) and UMAP [84] (Figure 5E). The major difference between PCA and tSNE/UMAP is that the latter performs non-linear transformations, whereby distances between individual proteins become incomparable. Their advantage is that they usually achieve visually more obvious separation of protein clusters in return and can provide performance benefits for two-dimensional clustering algorithms. In principle, these techniques can also be applied to a [sample x protein] rather than a [protein x sample] matrix to look at the data from a different perspective.

**Heatmaps.** A common visualization across different "omics" technologies are heatmaps with marginal dendrograms (Figure 5F). They can be used to understand the relations between samples and proteins alike. During the early stages of the analysis process heatmaps are often used similar to the PCA loadings plot to evaluate sample similarity. However, in contrast to the PCA plot they are based directly on the distance between the untransformed protein quantifications in each sample. Based on these distances a dendrogram is built, where branches of similar samples are grouped together. Additionally, it shows which groups of proteins follow a similar abundance pattern across samples, by building a vertical dendrogram across proteins in a similar fashion. The latter is particularly useful when it comes to a later stage of the analysis when proteins with specific behaviors of interest need to be grouped in order to form hypotheses about the underlying biology. Critical factors in creating and interpreting these heatmaps are the distance metric and clustering methods applied [85] to either axis and the normalization method that unifies the color scale across proteins. The distance is usually either euclidean distance or Pearson correlation. For normalization across samples z-scoring is often used.

**Visualizing individual dimensions.** The methods described above are most useful to display proteomic data across all measured dimensions. To show single dimensions (e.g., time course) or to combine proteomic data with other data types, different visualizations are better suited. The simplest way to display individual experimental dimensions is a line plot (Figure 5G), which works with continuous and categorical dimensions alike. Since showing the full proteomic scope would lead to clutter, we recommend either showing a relevant subset of proteins with thin lines, indicating density by opacity, or alternatively showing summary statistics. For some applications a radar plot might be preferred over a linear axis to ease interpretation (Figure 5I). Suitable applications include time course experiments along circadian cycles or biological slices of a bigger whole, for example, different organ tissues.

**Mixing data types.** If other data types (e.g., clinical parameters, additional "omics" data, quality parameters) are integrated with proteomic data, it is likely that none of the visualizations above can be applied. In that case one can turn to dimension plots having either parallel coordinates or categories. These have multiple parallel axes that can each represent a different data type with individual ranges.





**FIGURE 5** Visualization of multidimensional experimental designs. The data used for this Figure is a comparative spatial proteomics dataset from [78] (PXD010103). For organelle annotation marker proteins from [109] were used. (A-C) Dimensionality reduction by principal component analysis (PCA). (A) Projections onto PCs 1 and 3 show separation of protein groups into organelles. (B) Loading of PCs 1 and 2 with individual dimensions, that is, samples. Separation along PC1 is between  $\leq 6K$  and  $\geq 12K$  fractions, while PC2 separates fractions within each of these groups. As this represents the Eigenvectors of the PCA it is often represented with arrows instead of points. (C) Data variability explained by

Here, every line represents a dataset (e.g., protein or sample) and connects the data points across the parallel dimensions (Figure 5H). If all dimensions are categorical, the group sizes and membership combinations are displayed instead.

## 2.5 | Network representations of proteomic data

Many extensive proteomics studies, such as interactomics [86, 87], proteome profiling based [88, 104] or extensive clinical studies [89], focus on networks between proteins or could be mined for them. Any experiment that yields enough data to identify or quantify the physical or phenotypic relation between several pairs of proteins is sufficient to build a network, albeit of variable size. Since all networks are built from nodes and edges, many networks look similar at a first glance although they usually convey vastly different information. In proteomics, most often the nodes represent proteins and edges usually represent one of three types of information: physical interaction or proximity (interactomics), phenotypic similarity (profiling) or shared annotations (e.g., Gene Ontology). The most relevant distinctions made in graph theory are between weighted and unweighted networks and between directed and undirected networks. Additionally, the type of both nodes and edges can be homogenous throughout the network or not. For a general review of networks in biological systems see [47].

Different combinations of these characteristics give rise to three different types of networks often encountered in proteomics studies: (1) The most direct representation of measured relations between nodes are networks with homogenous node types and weighted edges (Figure 6A). Since a two-dimensional layout is often insufficient to convey edge properties accurately simply by length, additional visual channels like number of edges, color and thickness can be used. Groups of nodes are usually highlighted by color (e.g., query proteins vs interactors). (2) Akin to dendrograms, hierarchical networks (Figure 6B) convey information about the organization of proteins into groups. These networks are inherently directed, are often unweighted and generally have heterogeneous node types (e.g., protein complexes and proteins). (3) Incorporating extracted or annotated information about biological processes like protein regulation gives rise to semantic networks.

When reading or creating a network it is important to realize which type of network is used/required, what the main information behind nodes and edges is and how they are encoded in the visualization (see Fung et al. 2012 for more considerations). Depending on the degree of complexity and customization required, different tools can be used to create networks: Literature based interaction networks can be generated using STRING [76] and biological pathway graphs

are provided by Reactome [90]. For networks based on quantifications provided by a researcher, many tools are available, including Cytoscape [91] - a very extensive and expandable standalone software - and Perseus, although it only contains limited network functionalities [92]. For scientists with programming experience several options exist, including the Cytoscape API [93], Python libraries like NetworkX [94] and graphviz (<https://graphviz.readthedocs.io>), the R library network [95], or Igraph (<https://igraph.org>), which is available in both languages. A more specialized tool for clinical proteomics that aims to capture comprehensive prior knowledge is the clinical knowledge graph (CKG) [96].

## 3 | CUSTOM PROGRAMMATIC DATA VISUALIZATION

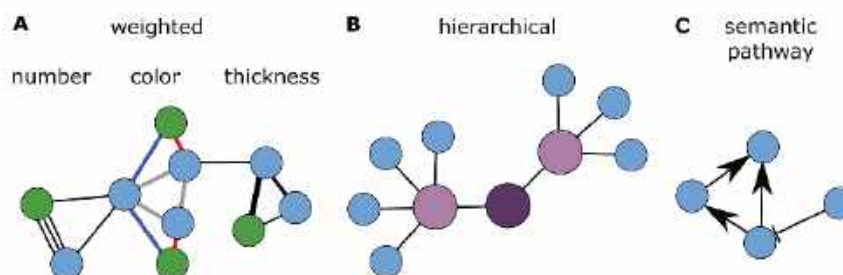
In the previous sections we have described several commonly used visualizations in the proteomics field, along with available software tools to create them. However, depending on the experimental design and specific focus of a study, it might still be challenging to find a fitting visualization in one of these tools. A scientist might want to create something entirely novel, or just customize the Figure beyond the capabilities of the tool that you are using. Besides these practical limitations, the data visualization process can also contribute to low transparency and reproducibility in scientific papers by use of closed source software and lack of documentation [97]. These challenges can be mastered by programming the visualizations oneself and sharing the code appropriately. Thus, in this section we describe how Python in combination with established open code/science tools can be used to generate customized proteomics visualizations transparently.

### 3.1 | Proteomics data visualization in Python

For this review we chose Python as a programming language, because it is widely known for its readability and versatility, as well as a shallow learning curve for new developers and a very active, supportive and collaborative community. The latter is particularly useful considering that "open code" and community engagement can benefit researchers by saving time and funding resources [98]. As a primer for proteomics visualization in R, we recommend [33]. Similar to R, Python already has a large variety of well-documented and well-maintained libraries for scientific computing [99]. Although Python has only been in widespread use in the computational proteomics field for roughly

individual PCs. Only the first three PCs are shown here, as they jointly cover > 90% of the data variability. (D) Projection onto non-linear tSNE dimensions. This has a similar density as the PCA, but different arrangement of organelles. (E) Projection onto non-linear UMAP dimensions. Although this shows the same dataset as A and D, clusters are a lot more visible because the local density is increased. (F) Heatmap with marginal dendrograms (complete linkage) of all organelle marker proteins. Samples are clustered by Pearson correlation, proteins by euclidean distance. (G) Line plot showing profiles along the subcellular dimension of all ER marker proteins. (H) Parallel coordinates plots can be used to relate proteomic data to other data dimensions that use different scales. Here, showing the identification score and q-value together with the normalized protein intensity in one sample (same proteins as in G). (I) Radar plot displaying average profiles per organelle marker group.





**FIGURE 6** Common network types encountered in proteomics studies. These are only schematics, in reality these networks are a lot more extensive and often turn into “hairballs” that are hard to read. (A) Weighted protein network with variable visual channels used to encode edge weights. Node color often shows query versus neighboring proteins. (B) Hierarchical network showing group membership of proteins. (C) Semantic pathway network describing biological processes.

**TABLE 2** Selection of open-source software libraries for proteomics data analysis and visualization in Python

| Library              | Description  |
|----------------------|--|
| pymzML [110,111]     | An mzML data parser for fast access and handling of the data with integrated data visualization.                     |
| Pyteomics [112,113]  | A framework for proteomics data analysis, supporting different data formats.   |
| pyOpenMS [114]       | A library for the analysis of proteomics and metabolomics data.  |
| multiplierz [115]    | A scriptable framework for access to manufacturers’ formats via mzAPI.   |
| PaDuA [116]          | A Python package optimized for the processing and analysis of quantified (phospho)proteomics data.                   |
| AlphaTims [117]      | A Python package for efficient accession and visualization of Bruker Tims TOF raw data.                              |
| AlphaMap [58]        | A Python package for the visual annotation of proteomics data on the peptide level with sequence specific knowledge. |
| spectrum_utils [118] | A Python package for processing and visualization of MS/MS spectra.  |

a decade, a number of libraries for MS data accession and specialized analysis tasks are already established (Table 2).

Similar to this data analysis stack, many data visualization libraries exist that are differently well suited for different purposes. Static plots in Python can be generated using Matplotlib [100] or Seaborn [101]. Both libraries are highly versatile, but Seaborn adds additional functionality on top of Matplotlib, for example, it offers more choices for plot styles and colors. Interactive plots are particularly useful for exploratory data analysis by providing data on demand and basic tools like zooming, selecting, rotating, and so on. These can be built in libraries such as Bokeh (docs.bokeh.org) and Plotly (https://plotly.com). Plotly is very popular in the scientific field due to the high number of unique visualizations, including three-dimensional and scientific use cases. Thus, we also used it throughout the code used to generate the Figures in this review.

One overall challenge of data visualization is how to efficiently handle big data. Big data is particularly challenging, because the simultaneous display of thousands of data points usually leads to occlusion of information (as can be seen in Figure 5A) and oftentimes misinterpretation. Common workarounds are down sampling, reduced opacity (as in Figure 5E), replacement by summary statistics (as in Figure 5F) and more. While these methods can often improve data display, the full data scope should always be evaluated and in many cases, it cannot be replaced. An easy way to visualize it without occlusion is offered by the Datashader library (https://datashader.org). It rasterizes the data space similar to a histogram, but in two-dimensional and encodes the number of points per two-dimensional bin by color (Figure 1C, Figure 3C). This facilitates quick visualization of patterns or structures in big data sets.

Due to the amount of data contained in most proteomic studies, there is usually more biological insights to be gained than can be described in a single publication. While uploading datasets to repositories is generally mandatory nowadays, data can be made even more accessible by providing a dedicated online resource or even an analysis service with embedded interactive visualizations. Python provides several libraries that integrate data analysis and visualization capabilities with modern web frameworks to create browser based graphical user interfaces, examples being Dash (https://dash.plotly.com), Streamlit (https://docs.streamlit.io) and Panel (https://panel.holoviz.org).

Using a combination of the scientific Python stack, the generalized visualization libraries and web engines, several visualization tools and resource pages for the proteomics field have already been created [46,58,96,102–104].

### 3.2 | Open science tools

To enable full accessibility, transparency and reusability of custom visualizations we briefly introduce several existing open science and open source principles and tools.

Firstly, it is important to fully document what any code is doing and to provide necessary context, akin to wet-lab protocols and documentation. A modern software development tool supporting this is Jupyter

(<https://jupyter.org>), which is compatible with Python, R and Julia. It integrates code, execution output (e.g., visualizations) and static documentation in a single interactive, but freezable file format. The documentation is written in the very simple markdown syntax, which allows standard text formatting and inclusion of complex elements like images and formulas. In recent MS-based proteomics publications, one already sees links to the study specific code provided in Jupyter [98,105]. Given a suitable Python environment and access to the data anybody can thereby reproduce results transparently. In case local hardware is limiting code execution, community resources can be used. Specifically, Google provides a free but powerful Jupyter notebook environment called Google Colab [106].

Secondly, it is important to share code publicly and since code usually continues evolving after publication it is crucial to transparently keep track of code versions, dependencies and contributions. The community standard tool for version control is Git, complemented by the public hosting service GitHub [107], which is free to use for scientific projects. Beyond sharing versioned code, it is also a social coding platform that enables community contributions like peer-review and ensures transparent attribution of code contributions to authors. For code that requires interactive execution, or creates interactive elements, GitHub provides integration online hosting solutions like Binder (<https://mybinder.org>). To create persistent and citable digital object identifiers (DOIs) for code repositories, Zenodo (<https://zenodo.org>) can be used directly from GitHub.

To give new developers an easy entry point and an example of what these tools can do, we applied them to the Python code we wrote to create the data visualizations in this review. The repository is hosted on the GitHub (<https://github.com/MannLabs/ProteomicsVisualization>), which includes a link to the hosted interactive version in Binder and installation instructions for a computational proteomics Python environment and a short guide on how to contribute custom visualizations for others to reuse.

## 4 | CONCLUSION

In this review we have summarized data visualizations specific to the proteomics field, from raw data to complex experimental designs. As this field is rapidly progressing and highly translational, we decided to not only cite existing tools for visualization, but to further provide guidance towards creating common data visualizations programmatically and interpreting them critically and correctly. As the options for experimental design are constantly evolving we could not cover all flavors of proteomics data visualization herein. It will be exciting to see how interactive web technologies and virtual reality will improve the way we visually explore proteomics data in the years to come, especially with regard to current limitations on three-dimensional visualization. Lastly, we want to encourage our readers to try out different visualization types and visual channels interactively for the data they have at hand and to view data visualization as a creative, yet crucial step of science and science communication.

## ACKNOWLEDGMENTS

This study was supported by The Max-Planck Society for Advancement of Science. Eugenia Voytik acknowledges funding by the Bavarian State Ministry of Health and Care through the research project DigiMed Bayern ([www.digimed-bayern.de](http://www.digimed-bayern.de)). IB acknowledges funding support from her Postdoc.Mobility fellowship granted by the Swiss National Science Foundation (P400PB\_191046). The authors would like to acknowledge all our colleagues at the Department for Proteomics and Signal Transduction, particularly the head of the department, Matthias Mann, and the head of the research group on systems biology of membrane trafficking, Georg Börner, who constantly support us in our work provided valuable feedback. Special thanks go to Alexandra Davies, a cell biologist and MS-expert, who helped us tailor this review to our target audience. Further valuable feedback was given to us by Sander Willems, Peter Treit and Vincent Albrecht.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS


Julia Patricia Schessner and Eugenia Voytik devised and wrote the manuscript. All authors contributed code and edited the manuscript.

## DATA AVAILABILITY STATEMENT

Proteomics data from the following ProteomeExchange repositories were reused to generate Figures in this study: PXD012867, PXD017703, PXD010697, PXD010103.

## ORCID

Julia Patricia Schessner  <https://orcid.org/0000-0003-3361-9830>

Eugenia Voytik  <https://orcid.org/0000-0003-4776-0771>

Isabell Bludau  <https://orcid.org/0000-0002-2601-238X>

## REFERENCES

1. Kelstrup, C. D., Jersie-Christensen, R. R., Batth, T. S., Arrey, T. N., Kuehn, A., Kellmann, M., & Olsen, J. V. (2014). Rapid and deep proteomes by faster sequencing on a benchtop quadrupole ultra-high-field Orbitrap mass spectrometer. *Journal of Proteome Research*, 13, 6187–6195.
2. Linscheid, N., Santos, A., Poulsen, P. C., Mills, R. W., Calloe, K., Leurs, U., Ye, J. Z., Stolte, C., Thomsen, M. B., Bentzen, B. H., Lundegaard, P. R., Olesen, M. S., Jensen, L. J., Olsen, J. V., & Lundby, A. (2021). Quantitative proteome comparison of human hearts with those of model organisms. *PLOS Biology*, 19(4), e3001144.
3. Michalski, A., Cox, J., & Mann, M. (2011). More than 100,000 detectable peptide species elute in single shotgun proteomics runs but the majority is inaccessible to data-dependent LC-MS/MS. *Journal of Proteome Research*, 10, 1785–1793.
4. Nagaraj, N., Wisniewski, J. R., Geiger, T., Cox, J., Kircher, M., Kelso, J., Pääbo, S., & Mann, M. (2011). Deep proteome and transcriptome mapping of a human cancer cell line. *Molecular Systems Biology*, 7(1), 548.
5. Müller, J. B., Geyer, P. E., Colaco, A. R., Treit, P. V., Strauss, M. T., Oroshi, M., Dall, S., Winter, S. V., Bader, J. M., Köhler, N., Theis, F., Santos, A., & Mann, M. (2020). The proteome landscape of the kingdoms of life. *Nature*, 582, 592–596.



6. Aebersold, R., & Mann, M. (2016). Mass-spectrometric exploration of proteome structure and function. *Nature*, 537, 347–355.
7. Bache, N., Geyer, P. E., Bekker-Jensen, D. B., Hoerning, O., Falkenby, L., Trait, P. V., Döll, S., Paron, I., Müller, J. B., Meier, F., Olsen, J. V., Vorm, O., & Mann, M. (2018). A novel LC system embeds analytes in pre-formed gradients for rapid, ultra-robust proteomics. *Molecular and Cellular Proteomics*, 17, 2284–2296.
8. Beck, S., Michalski, A., Raether, O., Lubeck, M., Kaspar, S., Goedecke, N., Baessmann, C., Hornburg, D., Meier, F., Paron, I., Kulak, N. A., Cox, J., & Mann, M. (2015). The impact II, a very high-resolution quadrupole time-of-flight instrument (QTOF) for deep shotgun proteomics. *Molecular and Cellular Proteomics*, 14, 2014–2029.
9. Buryakov, I. A., Krylov, E. V., Nazarov, E. G., & Rasulev, U. K. (1993). A new method of separation of multi-atomic ions by mobility at atmospheric pressure using a high-frequency amplitude-asymmetric strong electric field. *International Journal of Mass Spectrometry and Ion Processes*, 128(3), 143–148.
10. Hebert, A. S., Prasad, S., Belford, M. W., Bailey, D. J., McAlister, G. C., Abbatiello, S. E., Huguet, R., Wouters, E. R., Dunyach, J.-J., Brademan, D. R., Westphall, M. S., & Coon, J. J. (2018). Comprehensive single-shot proteomics with FAIMS on a hybrid orbitrap mass spectrometer. *Analytical Chemistry*, 90, 9529–9537.
11. Silveira, J. A., Michelmann, K., Ridgeway, M. E., & Park, M. A. (2016). Fundamentals of trapped ion mobility spectrometry part II: Fluid dynamics. *Journal of the American Society for Mass Spectrometry*, 27, 585–595.
12. Rodriguez-Suarez, E., Hughes, C., Cethings, L., Giles, K., Wildgoose, J., Stapels, M. E., Fadgen, K. J., Geromanos, S. P. C., Vissers, J., Elortza, F. I., & Langridge, J. (2013). An ion mobility assisted data independent LC-MS strategy for the analysis of complex biological samples. *Current Analytical Chemistry*, 9(2), 199–211.
13. Helm, D., Vissers, J. P. C., Hughes, C. J., Hahne, H., Ruprecht, B., Pachi, F., Grzyb, A., Richardson, K., Wildgoose, J., Maier, S. K., Marx, H., Wilhelm, M., Becher, I., Lemeier, S., Bantscheff, M., Langridge, J. I., & Kuster, B. (2014). Ion mobility tandem mass spectrometry enhances performance of bottom-up proteomics. *Molecular & Cellular Proteomics*, 13(12), 3709–3715.
14. Gillet, L. C., Navarro, P., Tate, S., Röst, H., Selevsek, N., Reiter, L., Bonner, R., Aebersold, R. (2012). Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: A new concept for consistent and accurate proteome analysis. *Molecular & Cellular Proteomics*, 11(6), O111.016717.
15. Meier, F., Geyer, P. E., Winter, S. V., Cox, J., & Mann, M. (2018). Box-Car acquisition method enables single-shot proteomics at a depth of 10,000 proteins in 100 minutes. *Nature Methods*, 15, 440–448.
16. Meier, F., Beck, S., Grassl, N., Lubeck, M., Park, M. A., Raether, O., & Mann, M. (2015). Parallel accumulation-serial fragmentation (PASEF): Multiplying sequencing speed and sensitivity by synchronized scans in a trapped ion mobility device. *Journal of Proteome Research*, 14, 5378–5387.
17. Geromanos, S. J., Vissers, J. P. C., Silva, J. C., Dorschel, C. A., Li, G.-Z., Gorenstein, M. V., Bateman, R. H., & Langridge, J. I. (2009). The detection, correlation, and comparison of peptide precursor and product ions from data independent LC-MS with data dependant LC-MS/MS. *Proteomics*, 9, 1683–1695.
18. Messner, C. B., Demichev, V., Wendisch, D., Michalick, L., White, M., Freiwald, A., Textoris-Taube, K., Vernardis, S. I., Egger, A. S., Kreidl, M., Ludwig, D., Kilian, C., Agostini, F., Zelezniak, A., Thibault, C., Pfeiffer, M., Hippenstiel, S., Hocke, A., von Kalle, C., ... Ralser, M. (2020). Ultra-high-throughput clinical proteomics reveals classifiers of COVID-19 infection. *Cell Systems*, 11, 11–24.e4.
19. Messner, C. B., Demichev, V., Bloomfield, N., Yu, J. S. L., White, M., Kreidl, M., Egger, A. S., Freiwald, A., Ivosev, G., Wasim, F., Zelezniak, A., Jürgens, L., Suttrop, N., Sander, L. E., Kurth, F., Lilley, K. S., Müllerer, M., Tate, S., & Ralser, M. (2021). Ultra-fast proteomics with Scanning SWATH. *Nature Biotechnology*, 39(7), 846–854.
20. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S., & Ralser, M. (2020). DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. *Nature Methods*, 17, 41–44.
21. Gessulat, S., Schmidt, T., Zolg, D. P., Samaras, P., Schnatbaum, K., Zerweck, J., Knaute, T., Rechenberger, J., Delanghe, B., Huhmer, A., Reimer, U., Ehrlich, H.-C., Aiche, S., Kuster, B., & Wilhelm, M. (2019). Prosit: Proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, 16, 509–518.
22. Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D., & Nesvizhskii, A. I. (2017). MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nature Methods*, 14, 513–520.
23. Tiwary, S., Levy, R., Gutenbrunner, P., Salinas Soto, F., Palaniappan, K. K., Deming, L., Berndt, M., Brant, A., Cimermanic, P., & Cox, J. (2019). High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis. *Nature Methods*, 16, 519–525.
24. Zhou, X.-X., Zeng, W.-F., Chi, H., Luo, C., Liu, C., Zhan, J., He, S.-M., & Zhang, Z. (2017). pDeep: Predicting MS/MS spectra of peptides with deep learning. *Analytical Chemistry*, 89, 12690–12697.
25. Gehlenborg, N., O'Donoghue, S. I., Baliga, N. S., Goesmann, A., Hibbs, M. A., Kitano, H., Kohlbacher, O., Neuweger, H., Schneider, R., Tenenbaum, D., & Gavin, A. C. (2010). Visualization of omics data for systems biology. *Nature Methods* 2010 7:3, 7, S56–S68.
26. Pavlopoulos, G. A., Secrier, M., Moschopoulos, C. N., Soldatos, T. G., Kossida, S., Aerts, J., Schneider, R., & Bago, P. G. (2011). Using graph theory to analyze biological networks. *BioData Mining*, 4, 1–27.
27. Adams, K. J., Pratt, B., Bose, N., Dubois, L. G., John-Williams, L. St., Perrott, K. M., Ky, K., Kapahl, P., Sharma, V., MacCoss, M. J., Moseley, M. A., Colton, C. A., MacLean, B. X., Schilling, B., Thompson, J. W., & Consortium, A. D. M. (2020). Skyline for small molecules: a unifying software package for quantitative metabolomics. *Journal of Proteome Research*, 19, 1447–1458.
28. Bruderer, R., Bernhardt, O. M., Gandhi, T., Miladinović, S. M., Cheng, L.-Y., Messner, S., Ehrenberger, T., Zanotelli, V., Butscheid, Y., Escher, C., Vittek, O., Rinner, O., & Reiter, L. (2015). Extending the limits of quantitative proteome profiling with data-independent acquisition and application to acetaminophen-treated three-dimensional liver microtissues. *Molecular & Cellular Proteomics*, 14(5), 1400–1410.
29. Yanova, S., Temu, T., Sinitcyn, P., Carlson, A., Hein, M. Y., Geiger, T., Mann, M., & Cox, J. (2016). The Perseus computational platform for comprehensive analysis of (prote)omics data. *Nature Methods*, 13, 731–740.
30. Yanova, S., Temu, T., Carlson, A., Sinitcyn, P., Mann, M., & Cox, J. (2015). Visualization of LC-MS/MS proteomics data in MaxQuant. *Proteomics*, 15, 1453–1456.
31. Oveland, E., Muth, T., Rapp, E., Martens, L., Berven, F. S., & Barsnes, H. (2015). Viewing the proteome: How to visualize proteomics data?. *PROTEOMICS*, 15, 1341–1355.
32. Perez-Riverol, Y., Wang, R., Hermjakob, H., Müller, M., Vesada, V., & Vizcaino, J. A. (2014). Open source libraries and frameworks for mass spectrometry based proteomics: A developer's perspective. *Biochimica et Biophysica Acta - Proteins and Proteomics*, 1844, 63–76.
33. Gatto, L., Breckels, L. M., Naake, T., & Gibb, S. (2015). Visualization of proteomics data using R and bioconductor. *Proteomics*, 15, 1375–1389.
34. Sinha, A., & Mann, M. (2020). A beginner's guide to mass spectrometry-based proteomics. *The Biochemist*, 42(5), 64–69.
35. Deutsch, E. W. (2012). File formats commonly used in mass spectrometry proteomics. *Molecular & Cellular Proteomics*, 11, 1612–1621.

36. Bittremieux, W., Valkenburg, D., Martens, L., & Laukens, K. (2017). Computational quality control tools for mass spectrometry proteomics. *PROTEOMICS*, 17(3-4), 1600159.
37. Perez-Riverol, Y., Xu, Q.-W., Wang, R., Uszkoreit, J., Griss, J., Sanchez, A., Reisinger, F., Csordas, A., Ternent, T., Del-Toro, N., Dianes, J. A., Eisenacher, M., Hermjakob, H., & Vizcaino, J. A. (2016). PRIDE inspector toolsuite: Moving toward a universal visualization tool for proteomics data standard formats and quality assessment of proteome exchange datasets. *Molecular & Cellular Proteomics: MCP*, 15, 305–317.
38. Noga, M., Sucharski, F., Suder, P., & Silberrring, J. (2007). A practical guide to nano-LC troubleshooting. *Journal of Separation Science*, 30, 2179–2189.
39. Rudnick, P. A., Clauser, K. R., Kilpatrick, L. E., Tchekhovskoi, D. v., Neta, P., Blonder, N., Billheimer, D. D., Blackman, R. K., Bunk, D. M., Cardasis, H. L., Ham, A. J. L., Jaffe, J. D., Kinsinger, C. R., Mesri, M., Neubert, T. A., Schilling, B., Tabb, D. L., Tegeler, T. J., Vega-Montoto, L., ... Stein, S. E. (2010). Performance metrics for liquid chromatography-tandem mass spectrometry systems in proteomics analyses. *Molecular & Cellular Proteomics*, 9, 225–241.
40. Cox, J., & Mann, M. (2008). MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, 26, 1367–1372.
41. Avtonomov, D. M., Raskind, A., & Nesvizhskii, A. I. (2016). BatMass: A Java software platform for LC-MS data visualization in proteomics and metabolomics. *Journal of Proteome Research*, 15, 2500–2509.
42. Meier, F., Park, M. A., & Mann, M. (2021). Trapped ion mobility spectrometry and parallel accumulation-serial fragmentation in proteomics. *Molecular & Cellular Proteomics*, 20, 5378–5387.
43. Ting, Y. S., Egerton, J. D., Payne, S. H., Kim, S., MacLean, B., Käll, L., Aebersold, R., Smith, R. D., Noble, W. S., & MacCoss, M. J. (2015). Peptide-centric proteome analysis: An alternative strategy for the analysis of tandem mass spectrometry data. *Molecular & Cellular Proteomics*, 14(9), 2301–2307.
44. Perkins, D. N., Pappin, D. J. C., Creasy, D. M., & Cottrell, J. S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. 3551–3567.
45. Cox, J., Neuhauser, N., Michalski, A., Scheltema, R. A., Olsen, J. v., & Mann, M. (2011). Andromeda: A peptide search engine integrated into the Maxquant environment. *Journal of Proteome Research*, 10, 1794–1805.
46. Strauss, M. T., Bludau, I., Zeng, W.-F., Voytk, E., Ammar, C., Schessner, J., Ilango, R., Gill, M., Meier, F., Willems, S., ... (2021). AlphaPept, a modern and open framework for MS-based proteomics. *bioRxiv*, 2021.07.23.453379. <https://www.biorxiv.org/content/10.1101/2021.07.23.453379>.
47. Koutrouli, M., Karatzas, E., Paez-Espino, D., & Pavlopoulos, G. A. (2020). A guide to conquer the biological network era using graph theory. *Frontiers in Bioengineering and Biotechnology*, 8. <https://doi.org/10.3389/fbioe.2020.00034>.
48. Hu, A., Noble, W. S., Wolf-Yadlin, A., Hu, A., Noble, W. S., & Wolf-Yadlin, A. (2016). Technical advances in proteomics: new developments in data-independent acquisition F1000Research, 5, 419.
49. Ludwig, C., Gillet, L., Rosenberger, G., Amon, S., Collins, B. C., & Aebersold, R. (2018). Data-independent acquisition-based SWATH-MS for quantitative proteomics: A tutorial. *Molecular Systems Biology*, 14(8), e8126.
50. Tsou, C.-C., Avtonomov, D., Larsen, B., Tucholska, M., Choi, H., Gingras, A.-C., & Nesvizhskii, A. I. (2015). DIA-Umpire: Comprehensive computational framework for data-independent acquisition proteomics. *Nature Methods* 2015 12:3, 12, 258–264.
51. Havilio, M., & Wool, A. (2007). Large-scale unrestricted identification of post-translation modifications using tandem mass spectrometry. *Analytical Chemistry*, 79, 1362–1368.
52. Boersema, P. J., Mohammed, S., & Heck, A. J. R. (2009). Phosphopeptide fragmentation and analysis by mass spectrometry. *Journal of Mass Spectrometry*, 44, 861–878.
53. Wiśniewski, J. R., Zettl, K., Pilch, M., Rysiewicz, B., & Sadok, I. (2020). "Shotgun" proteomic analyses without alkylation of cysteine. *Analytica Chimica Acta*, 1100, 131–137.
54. Thompson, A., Schäfer, J., Kuhn, K., Kienle, S., Schwarz, J., Schmidt, G., Neumann, T., & Hamon, C. (2003). Tandem mass tags: A novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Analytical Chemistry*, 75, 1895–1904.
55. Virreira Winter, S., Meier, F., Wichmann, C., Cox, J., Mann, M., & Meissner, F. (2018). EASI-tag enables accurate multiplexed and interference-free MS2-based proteome quantification. *Nature Methods*, 15, 527–530.
56. Desiere, F. (2006). The PeptideAtlas project. *Nucleic Acids Research*, 34(90001), D655–D658.
57. Hornbeck, P. V., Zhang, B., Murray, B., Kornhauser, J. M., Latham, V., & Skrzypek, E. (2015). PhosphoSitePlus, 2014: Mutations, PTMs and recalibrations. *Nucleic Acids Research*, 43(D1), D512–D520.
58. Voytk, E., Bludau, I., Willems, S., Hansen, F. M., Brunner, A.-D., Strauss, M. T., & Mann, M. (2021). AlphaMap: An open-source Python package for the visual annotation of proteomics data with sequence-specific knowledge. *Bioinformatics*, 38, 849–852.
59. Claassen, M. (2012). Inference and validation of protein identifications. *Molecular & Cellular Proteomics*, 11(11), 1097–1104.
60. Nesvizhskii, A. I., Keller, A., Kolker, E., & Aebersold, R. (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry*, 75, 4646–4658.
61. Matzke, M. M., Brown, J. N., Gritsenko, M. A., Metz, T. O., Pounds, J. G., Rodland, K. D., Shukla, A. K., Smith, R. D., Waters, K. M., McDermott, J. E., & Webb-Robertson, B.-J. (2013). A comparative analysis of computational approaches to relative protein quantification using peptide peak intensities in label-free LC-MS proteomics experiments. *Proteomics*, 13, 493–503.
62. Schmidt, R. M., Schessner, J. P., Borner, G. H. H., & Schuck, S. (2019). The proteasome biogenesis regulator Rpn4 cooperates with the unfolded protein response to promote ER stress resistance. *eLife*, 8. <https://doi.org/10.7554/eLife.43244>.
63. Schwanhäusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, Wei, & Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature*, 473(7347), 337–342.
64. Wiśniewski, J. R., Hein, M. Y., Cox, J., & Mann, M. (2014). A "Proteomic Ruler" for protein copy number and concentration estimation without spike-in standards. *Molecular & Cellular Proteomics*, 13(12), 3497–3506.
65. Krzywinski, M., & Altman, N. (2014). Comparing samples—part I. *Nature Methods*, 11(3), 215–216.
66. Krzywinski, M., & Altman, N. (2014). Comparing samples—part II. *Nature Methods*, 11(4), 355–356.
67. Krzywinski, M., & Altman, N. (2013). Power and sample size. *Nature Methods*, 10(12), 1139–1140.
68. Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98, 5116–5121.
69. Maleki, F., Ovens, K., Hogan, D. J., & Kuslik, A. J. (2020). Gene set analysis: Challenges, opportunities, and future research. *Frontiers in Genetics*, 11. <https://doi.org/10.3389/fgene.2020.00654>.
70. Fisher, R. A. (1922). On the interpretation of  $\chi^2$  from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 85(1), 87–94.
71. Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., & Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression pro-



- files. *Proceedings of the National Academy of Sciences*, 102, 15545–15550.
72. Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. (2000). Gene ontology: Tool for the unification of biology. The gene ontology consortium. *Nature Genetics*, 25, 25–29.
  73. The gene ontology resource: Enriching a GOld mine. (2021). *Nucleic Acids Research*, 49, D325–D334.
  74. Khatri, P., & Draghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 21(18), 3587–3595.
  75. Snel, B. (2000). STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Acids Research*, 28(18), 3442–3444.
  76. Mi, H., Ebert, D., Muruganujan, A., Mills, C., Albu, L. P., Mushayamaha, T., & Thomas, P. D. (2021). PANTHER version 16: A revised family classification, tree-based classification tool, enhancer regions and extensive API. *Nucleic Acids Research*, 49, D394–D403.
  77. Brüning, F., Noya, S. B., Bange, T., Koutsouli, S., Rudolph, J. D., Tyagarajan, S. K., Cox, J., Mann, M., Brown, S. A., & Robles, M. S. (2019). Sleep-wake cycles drive daily dynamics of synaptic phosphorylation. *Science*, 366(6462), <https://doi.org/10.1126/science.aaa3617>.
  78. Davies, A. K., Itzhak, D. N., Edgar, J. R., Archuleta, T. O. L., Hirst, J., Jackson, L. P., Robinson, M. S., & Borner, G. H. H. (2018). AP-4 vesicles contribute to spatial control of autophagy via RUSC-dependent peripheral delivery of ATG9A. *Nature Communications*, 9(1), <https://doi.org/10.1038/s41467-018-06172-7>.
  79. Bludau, I., Heusel, M., Frank, M., Rosenberger, G., Hafen, R., Banaei-Farahani, A., van Drogen, A., Collins, B. C., Gstaiger, M., & Aebersold, R. (2020). Complex-centric proteome profiling by SEC-SWATH-MS for the parallel detection of hundreds of protein complexes. *Nature Protocols*, 15(8), 2341–2386.
  80. Jean Beltran, P. M., Mathias, R. A., & Cristea, I. M. (2016). A portrait of the human organelle proteome in space and time during cytomegalovirus infection. *Cell Systems*, 3, 361–373.e6.
  81. Pangratz-Fuehrer, S., Genzel-Boroviczeny, O., Bodensohn, W., Eisenburger, N., Scharpenack, J., Geyer, P. E., Müller-Reif, J. B., van Hagen, N., Müller, A. M., Jensen, M. K., Klein, C., Mann, M., & Nussbaum, C. (2021). Cohort profile: the MUNICH preterm and term clinical study (MUNICH-PreTCI), a neonatal birth cohort with focus on prenatal and postnatal determinants of infant and childhood morbidity. *BMJ Open*, 11(6), e050652.
  82. Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(7), 498–520.
  83. Van Der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605. <http://jmlr.org/papers/v9/vandermaaten08a.html>.
  84. McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. <https://arxiv.org/abs/1802.03426>
  85. Do, J. H., & Choi, D.-K. (2008). Clustering approaches to identifying gene expression patterns from DNA microarray data. *Molecules and Cells*, 25, 279–288.
  86. Hein, M. Y., Hubner, N. C., Poser, I., Cox, J., Nagaraj, N., Toyoda, Y., Gak, I. A., Weisswange, I., Mansfeld, J., Buchholz, F., Hyman, A. A., & Mann, M. (2015). A human interactome in three quantitative dimensions organized by stoichiometries and abundances. *Cell*, 163, 712–723.
  87. Huttlin, E. L., Bruckner, R. J., Navarrete-Perea, J., Cannon, J. R., Baltier, K., Gebreab, F., Gygi, M. P., Thorbeck, A., Zarraga, G., Tam, S., Szpyt, J., Gassaway, B. M., Panov, A., Parzen, H., Fu, S., Golbazi, A., Maenpää, E., Stricker, K., Guha Thakurta, S., ..., Gygi, S. P. (2021). Dual proteome-scale networks reveal cell-specific remodeling of the human interactome. *Cell*, 184, 3022–3040.e28.
  88. Kirkwood, K. J., Ahmad, Y., Larance, M., & Lamond, A. I. (2013). Characterization of native protein complexes and protein isoform variation using size-fractionation-based quantitative proteomics. *Molecular & Cellular Proteomics*, 12, 3851–3873.
  89. Shi, Y., Ding, Y., Li, G., Wang, L., Osman, R. A., Sun, J., Qian, L., Zheng, G., & Zhang, G. (2021). Discovery of novel biomarkers for diagnosing and predicting the progression of multiple sclerosis using TMT-based quantitative proteomics. *Frontiers in Immunology*, 12, <https://doi.org/10.3389/fimmu.2021.700031>.
  90. Joshi-Tope, G. (2004). Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database issue), D428–D432.
  91. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11), 2498–2504.
  92. Rudolph, J. D., & Cox, J. (2019). A network module for the perseus software for computational proteomics facilitates proteome interaction graph analysis. *Journal of Proteome Research*, 18, 2052–2064.
  93. Otasek, D., Morris, J. H., Bouças, J., Pico, A. R., & Demchak, B. O. (2019). Cytoscape automation: Empowering workflow-based network analysis. *Genome Biology*, 20(1), <https://doi.org/10.1186/s13059-019-1758-4>.
  94. Hagberg, A., Swart, P., & Schult, D. (2008). Exploring network structure, dynamics, and function using networkx. <https://www.oost.gov/biblio/960616>
  95. Butts, C. T. (2008). Network: A package for managing relational data in R. *Journal of Statistical Software*, 24(2), 1–36.
  96. Santos, A., Colaço, A. R., Nielsen, A. B., Niu, L., Strauss, M., Geyer, P. E., Coscia, F., Albrechtsen, N. J. W., Mundt, F., Jensen, L. J., & Mann, M. (2022). A knowledge graph to interpret clinical proteomics data. *Nature Biotechnology*, <https://doi.org/10.1038/s41587-021-01145-6>.
  97. Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, 533, 452–454.
  98. Bittremieux, W., Adams, C., Laukens, K., Dorrestein, P. C., & Bandeira, N. (2021). Open science resources for the mass spectrometry-based analysis of SARS-CoV-2. *J. Proteome Res.*, 20, 1464–1475.
  99. Perez, F., Granger, B. E., & Hunter, J. D. (2011). Python: An ecosystem for scientific computing. *Computing in Science & Engineering*, 13(2), 13–21.
  100. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.
  101. Waskom, M., Botvinnik, O., O’Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., Augspurger, T., Halchenko, Y., Cole, J. B., Warmenhoven, J., Ruiter, J., Pye, C., Hoyer, S., Vanderplas, J., Villalba, S., Kunter, G., Quintero, E., Bachant, P., Martin, M., & Qaleh, A., .... (2017). *mwaskom/seaborn: v0.8.1* (September 2017). Zenodo. <https://doi.org/10.5281/zenodo.883859>
  102. Petras, D., Phelan, V. V., Acharya, D., Allen, A. E., Aron, A. T., Bandeira, N., Bowen, B. P., Belle-Oudry, D., Boecker, S., Cummings, D. A., Deutsch, J. M., Fahy, E., Garg, N., Gregor, R., Handelsman, J., Navarro-Hoyos, M., Jarmusch, A. K., Jarmusch, S. A., Louie, K., Maloney, K. N., Marty, M. T., Meijler, M. M., Mizrahi, I., Neve, R. L., Northen, T. R., Molina-Santiago, C., Panitchpakdi, M., Pullman, B., Puri, A. W., Schmid, R., .... (2021). GNPS Dashboard: Collaborative exploration of mass spectrometry data in the web browser. *Nature Methods*, <https://doi.org/10.1038/s41592-021-01339-5>.
  103. Hansen, F. M., Tanzer, M. C., Brüning, F., Bludau, I., Stafford, C., Schulman, B. A., Robles, M. S., Karayel, O., & Mann, M. (2021). Data-independent acquisition method for ubiquitinome analysis reveals regulation of circadian biology. *Nature Communications*, 12(1), <https://doi.org/10.1038/s41467-020-20509-1>.

104. Martín-Jaular, L., Nevo, N., Schessner, J. P., Tkach, M., Jouve, M., Dingli, F., Loew, D., Witwer, K. W., Ostrowski, M., Borner, G. H. H., & Théry, C. (2021). Unbiased proteomic profiling of host cell extracellular vesicle composition and dynamics upon HIV-1 infection. *The EMBO Journal*, 40(8), e105492.
105. Meier, F., Köhler, N. D., Brunner, A.-D., Wanka, J.-M. H., Voytik, E., Strauss, M. T., Thels, F. J., & Mann, M. (2021). Deep learning the collisional cross sections of the peptide universe from a million experimental values. *Nature Communications*, 12(1). <https://doi.org/10.1038/s41467-021-21352-8>.
106. Bisong, E. (2019). Google Colaboratory. In E. Bisong (Ed.), *Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners* (pp. 59–64). Apress. [https://doi.org/10.1007/978-1-4842-4470-8\\_7](https://doi.org/10.1007/978-1-4842-4470-8_7).
107. Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F., da Ve, F., Fufezan, C., Tennent, T., Eglén, S. J., Katz, D. S., Pollard, T. J., Koronolov, A., Flight, R. M., Blin, K., & Vizcaino, J. A. (2016). Ten simple rules for taking advantage of git and GitHub. *PLOS Computational Biology*, 12(7), e1004947.
108. Meier, F., Brunner, A.-D., Frank, M., Ha, A., Bludau, I., Voytik, E., Kaspar-Schoenefeld, S., Lubeck, M., Raether, O., Bache, N., Aebersold, R., Collins, B. C., Röst, H. L., & Mann, M. (2020). diaPASEF: Parallel accumulation-serial fragmentation combined with data-independent acquisition. *Nature Methods*, 17, 1229–1236.
109. Itzhak, D. N., Tyanova, S., Cox, J., & Borner, G. H. H. (2016). Global, quantitative and dynamic mapping of protein subcellular localization. *eLife*, 5, <https://doi.org/10.7554/eLife.16950>.
110. Bald, T., Barth, J., Niehues, A., Specht, M., Hippler, M., & Fufezan, C. (2012). pymzML-Python module for high-throughput bioinformatics on mass spectrometry data. *Bioinformatics*, 28(7), 1052–1053.
111. Kösters, M., Leufken, J., Schulze, S., Sugimoto, K., Klein, J., Zahedi, R. P., Hippler, M., Leidel, S. A., & Fufezan, C. (2018). pymzML v2.0: Introducing a highly compressed and seekable gzip format. *Bioinformatics*, 34, 2513–2514.
112. Goloborodko, A. A., Levitsky, L. I., Ivanov, M. V., & Gorshkov, M. V. (2013). Pyteomics—a Python framework for exploratory data analysis and rapid software prototyping in proteomics. *Journal of the American Society for Mass Spectrometry*, 24(2), 301–304.
113. Levitsky, L. I., Klein, J. A., Ivanov, M. V., & Gorshkov, M. V. (2019). Pyteomics 4.0: Five years of development of a Python proteomics framework. *Journal of Proteome Research*, 18, 709–714.
114. Röst, H. L., Schmitt, U., Aebersold, R., & Malmström, L. (2014). pyOpenMS: a Python-based interface to the OpenMS mass-spectrometry algorithm library. *Proteomics*, 14, 74–77.
115. Alexander, W. M., Ficarro, S. B., Adelmant, G., & Marto, J. A. (2017). multiplier v2.0: A Python-based ecosystem for shared access and analysis of native mass spectrometry data. *Proteomics*, 17(15–16), 1700091.
116. Ressa, A., Fitzpatrick, M., van den Toorn, H., Heck, A. J. R., & Altelaar, M. (2019). PaDuA: A Python library for high-throughput (phospho)proteomics data analysis. *J. Proteome Research*, 18, 576–584.
117. Willems, S., Voytik, E., Skowronek, P., Strauss, M. T., & Mann, M. (2021). AlphaTims: Indexing trapped ion mobility spectrometry-TOF data for fast and easy accession and visualization. *Molecular & Cellular Proteomics*, 20, 100149.
118. Bittremieux, W. (2020). spectrum\_utils: A python package for mass spectrometry data processing and visualization. *Analytical Chemistry*, 92(1), 659–661.

**How to cite this article:** Schessner, J. P., Voytik, E., & Bludau, I. (2022). A practical guide to interpreting and generating bottom-up proteomics data visualizations. *Proteomics*, 22, e2100103. <https://doi.org/10.1002/pmic.202100103>



## 3.2. Article 2: AlphaViz: Visualization and validation of critical proteomics data directly at the raw data level

Authors: **Eugenia Voytik**<sup>‡</sup>, Sander Willems<sup>‡</sup>, Patricia Skowronek<sup>‡</sup>, Maria C. Tanzer<sup>‡</sup>, Andreas-David Brunner<sup>‡</sup>, Wen-Feng Zeng<sup>‡</sup>, Marvin Thielert<sup>‡</sup>, Maximilian T. Strauss<sup>¶</sup>, Matthias Mann<sup>‡¶\*</sup>

<sup>‡</sup> Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, 82152 Martinsried, Germany

<sup>¶</sup> NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, 2200 Copenhagen, Denmark

Pre-print published online: *bioRxiv* (2022), doi: 10.1101/2022.07.12.499676v1.

In review in *Molecular & Cellular Proteomics*.

As the method of choice in proteomics, MS routinely identified and quantifies thousands of proteins and their (modified) peptides. However, only a small subset of these proteins of specific biological or clinical relevance, such as key proteins of signaling pathways or biomarker candidates, are typically subjected to in-depth downstream analysis. Unfortunately, expert evaluation of the underlying raw data of these proteins and their individual peptides rarely occurs or only occurs in one of several possible dimensions, preventing researchers from concentrating their investigation on the best possible biological candidates.

In this publication, we introduce a new open-source software tool called AlphaViz. It allows to superimpose the identifications found by common proteomics workflows on the raw data for easy validation of proteins by visualization of their (modified) peptides. This is the first visualization tool that takes advantage of recent developments in deep learning prediction of peptide properties to verify experimental versus predicted results. AlphaViz mainly focuses on four-dimensional ‘next generation proteomics’ timsTOF data and utilizes all available data dimensions for validation purposes, including the additional ion mobility dimension.

Using AlphaViz, we demonstrate how easy it is to evaluate critical proteins and their peptides reported by various proteomics search engines at the raw chromatographic, ion mobility, MS<sup>1</sup> and MS<sup>2</sup> levels. This helped to reveal likely false positives, despite their high search engine scores. Conversely, the deep learning prediction of various peptide properties, the so-called ‘predict mode’ in AlphaViz, demonstrates the retrieval of the raw data for peptides likely present in the raw data but not reported by the search engine. By applying AlphaViz to the phosphoproteomics study of the EGF signaling pathway, we (in)validated the presence of key signaling proteins, and were also able to explore specific signalling events of interest that were missed by the proteomics software through direct inspection of the raw data.

As the first author of this paper, I conceptualized and designed the study, wrote the manuscript, and analyzed the data. I also implemented the Python code of AlphaViz along with the development of its graphical user interface.

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

## **AlphaViz: Visualization and validation of critical proteomics data directly at the raw data level**

Eugenia Voytik‡, Patricia Skowronek‡, Wen-Feng Zeng‡, Maria C. Tanzer‡, Andreas-David Brunner‡, Marvin Thielert‡, Maximilian T. Strauss¶, Sander Willems‡, Matthias Mann‡¶\*

‡ Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

¶ NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark

\*Corresponding author. E-mail: [mmann@biochem.mpg.de](mailto:mmann@biochem.mpg.de)

**Running title: Visualization and validation of critical results in raw data**

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

#### ABBREVIATIONS

|              |  |
|--------------|--|
| BPI          | base peak intensity                          |
| CCS          | collisional cross section                    |
| DDA          | data-dependent acquisition                   |
| DIA          | data-independent acquisition                 |
| EGF          | epidermal growth factor                      |
| FDR          | false discovery rate                         |
| GOBP         | Gene Ontology Biological Process             |
| GUI          | graphical user interface                     |
| IM           | ion mobility                                 |
| IQR          | interquartile range                          |
| MS/MS or MS2 | tandem MS                                    |
| PASEF        | parallel accumulation – serial fragmentation |
| PEP          | posterior error probability                  |
| PTM          | post-translational modification              |
| PyPI         | Python Package Index                         |
| RT           | retention time                               |
| TIC          | total ion current                            |
| TIMS         | trapped ion mobility spectrometry            |
| TOF          | time-of-flight                               |
| XIC          | extracted ion chromatogram                   |

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

## ABSTRACT

Although current mass spectrometry (MS)-based proteomics identifies and quantifies thousands of proteins and (modified) peptides, only a minority of them are subjected to in-depth downstream analysis. With the advent of automated processing workflows, biologically or clinically important results within a study are rarely validated by visualization of the underlying raw information. Current tools are often not integrated into the overall analysis nor readily extendable with new approaches. To remedy this, we developed AlphaViz, an open-source Python package to superimpose output from common analysis workflows on the raw data for easy visualization and validation of protein and peptide identifications. AlphaViz takes advantage of recent breakthroughs in the deep learning-assisted prediction of experimental peptide properties to allow manual assessment of the expected versus measured peptide result. We focused on the visualization of the 4-dimensional data cuboid provided by Bruker TimsTOF instruments, where the ion mobility dimension, besides intensity and retention time, can be predicted and used for verification. We illustrate how AlphaViz can quickly validate or invalidate peptide identifications regardless of the score given to them by automated workflows. Furthermore, we provide a 'predict mode' that can locate peptides present in the raw data but not reported by the search engine. This is illustrated the recovery of missing values from experimental replicates. Applied to phosphoproteomics, we show how key signaling nodes can be validated to enhance confidence for downstream interpretation or follow-up experiments. AlphaViz follows standards for open-source software development and features an easy-to-install graphical user interface for end-users and a modular Python package for bioinformaticians. Validation of critical proteomics results should now become a standard feature in MS-based proteomics.

**Keywords:** data visualization; quality control; DIA-NN; AlphaPept; TimsTOF

## INTRODUCTION

Mass spectrometry (MS)-based proteomics has evolved into a powerful and widely used analytical technique for researchers in diverse biological and clinical fields (1, 2). The increased throughput of MS instruments has led to the identification and quantification of thousands of proteins and their (modified) peptides in many experimental settings. To ensure the quality of such experiments, many journals now require to follow specific guidelines prior to submission (3). However, automated analysis workflows typically present long lists of identified and quantified peptides and proteins used for downstream analysis by the investigator. Only a small subset, like key proteins of signaling pathways and biomarker candidates are chosen for biological follow-up experiments or additional validation by orthogonal assays. Unfortunately, the underlying raw data for these critical peptides or proteins are rarely assessed at all or only in few of several possible dimensions, which could prevent investigators from following up on the best study candidates.

Applying the famous proverb "One picture is worth ten thousand words" to proteomics, visualization may be the most obvious solution for validating identifications at the level of raw MS data (4, 5). Inspecting the actual spectra of particular peptides, such as those with post-translational modifications (PTMs) or those uniquely identifying a protein of interest, can reveal important information, in addition to that used by the search engine. Furthermore, the advent of ultra-high sensitivity LC-MS based workflows for the analysis of minute protein amounts down to the level of single cells is currently lacking raw data visualization tools for the inspection and validation of proteins of interest at the limit of detection (6–9). The identification of a peptide amino acid sequence is part and parcel of high-confidence spectral identification and traditionally entailed visual inspection and validation by the investigator. However, the ever-increasing acquisition speed of mass spectrometers and the complexity of state-of-the-art scan modes in large-scale proteomics experiments rendered this approach impractical when dealing with huge data sets.



In part, this is due to many challenges in proteomic data visualization. The visualization step usually ranks last in the development of scientific algorithms or the establishment of novel workflows for analyzing proteomics data. Visualization tools tend to become publicly available with a considerable delay after the publication of the main workflows (10, 11). Because of their closed nature, even recently published tools may rapidly become outdated if they fail to take advantage of current advances in visualization such as in interactive biological ‘big data’ visualization. In this regard, increasing established open source concepts can help to keep up with the rapid pace of computational developments, building on powerful collaborative packages, for instance those in the increasing popular Scientific Python environment (12–14). In our group, we have focused on the visualization of highly complex multi-dimensional data acquired on Bruker TimsTOF instruments, which includes the additional ion mobility dimension (15). The AlphaTims package, as well as the parallel OpenTIMS effort, allows ready access and visualization of raw data, which has not been practical due to the long accession times and absence of convenient data structures (16, 17).

A major development in MS-based proteomics in recent years has been the success of machine learning in predicting peptide properties including retention time, ion mobility and the intensities of fragments in the MS2 spectra (18, 19). As a result, all these properties could be used to validate the proposed peptide spectrum matches, and this has already been done for spectral intensities (20). We reasoned that combining data visualization with the benefits of deep learning predictions, such as fragment ion intensities, retention time or ion mobility predictions, could dramatically benefit the entire visualization and validation approach. As a particular example, the assignment of convoluted fragmentation patterns in Data Independent Acquisition (DIA) to peptide sequences is still an active area of research with major search engines such as DIA-NN or Spectronaut sometimes disagreeing on the identification or matching of particular peptides (21, 22). Clearly, visualization of co-eluting fragments in the context of predicted retention times and fragment intensities (‘in silico truths’) could help in establishing confidence in critical peptide identifications.

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Many of the currently existing visualization tools are proprietary and integrated into MS data analysis software pipelines by the MS manufacturer, such as Compass DataAnalysis (Bruker Daltonics), Freestyle and Xcalibur (Thermo Fisher Scientific), Sciex OS (Sciex) or by independent providers, such as Spectronaut and Skyline (22, 23). There are also standalone tools for visualization such as the PRIDE Inspector (24). However, in all these cases, these tools are difficult to reuse, extend, or integrate into existing workflows for example for novel multi-dimensional data, such as TIMS-TOF data.

Here, we developed a visualization tool with the following goals: It should allow (1) intuitive visualization of search engine results of the underlying raw data; (2) integration of *in silico* predictions by deep learning algorithms; (3) automation for end users through a graphical user interface or Jupyter Notebooks; (4) open-source accessibility and easy extendibility by bioinformaticians to incorporate new developments, for example interactivity, big data visualization and graph customization.

With these goals in mind, we developed AlphaViz, an open-source Python-based visualization tool that allows the user to explore identification and quantification confidence of peptides by visually comparing them to the signal presented in the unprocessed MS data. AlphaViz links identifications to the evidences of the raw data to assess their quality by using results from currently supported software tools, such as MaxQuant, AlphaPept and DIA-NN (10, 13, 21). It makes use of current advances in visualization, such as interactivity, “big data” visualization or real-time graph customization. The interactive plots included in AlphaViz provide the data on-demand in order not to overwhelm users, and include, for instance, zooming, selection and annotation. “Big data” capabilities make it possible to visualize millions of data points in a single graph in a browser. This enabled the visualization of MS heatmaps in AlphaViz, allowing to plot intensity of observed precursor masses across retention time and to visually assess MS peptide features in an enlarged view. In addition, customization of the plots, such as selection of a chart color scale or the size and format of the exported plots, enables researchers to easily create and extract illustrations of candidate proteins and peptides that are suitable for publication.



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

AlphaViz follows robust software development standards (high-quality code, extensive documentation, automated testing, and continuous integration) as a part of the AlphaPept 'ecosystem' (13).

## EXPERIMENTAL PROCEDURES

### Publicly Available MS Datasets

To demonstrate the use of AlphaViz for DDA data, we obtained raw data files of a fractionated HeLa library (fraction 1) generated with the 120-min gradient dda-PASEF method together with output of the MaxQuant software (v.1.6.1.13) from ProteomeXchange (data set PXD010012) (25). For the visualization of DIA data, we used a dataset previously acquired in our group: a 21-min gradient (60 samples per day) HeLa sample acquired on Evosep / timsTOF with the dia-PASEF method (data set PXD017703) (26). The results of DIA-NN analysis (v.1.7.15) of these data were taken from reference (27).

Additionally, we are presenting in-detail phosphoproteomics analyses. We used a recently published dataset where HeLa cells were stimulated with EGF or left untreated, enriched for phosphopeptides and acquired in three replicates each on a timsTOF Pro instrument with a 21-min gradient and an optimal phosphoproteomics dia-PASEF method (28). The copied output of DIA-NN analysis (v.1.8) was filtered for 1 % PTM q-value, collapsed with the Perseus plug-in and filtered for 75 % localization probability (28).

### Data Acquisition for the Predict Mode Measurements

To demonstrate the 'predict mode' of AlphaViz, we synthesized phosphorylation positional isomers of the Rab10 peptide FHTITTSYYR. These isomers were dissolved in solution A\* (0.1% TFA/2% ACN), and 125, 250, 500, 1250, 2500, and 5000 fmol of them were spiked into 50 fmol of bovine serum albumin. We measured the samples using a dia-PASEF method optimized for phosphoproteomics and 21 minutes Evosep gradients (60 samples per day method) combined with the timsTOF Pro (Bruker Daltonics) (28). The peptides were separated using an 8 cm x 150  $\mu$ m reverse-phase column packed with 1.5  $\mu$ m C<sub>18</sub>-beads (Pepsep) connected to a 10  $\mu$ m ID nano-electrospray emitter (Bruker Daltonics). Our dia-PASEF method covered an  $m/z$ -range from 400 to 1400 Da and an ion mobility range from 0.6 to 1.5 Vs cm<sup>-2</sup> with 12 dia-PASEF scans (cycle time: 1.38s). The collision energy depended on the ion mobility and changed from 60 eV at 1.5 Vs cm<sup>-2</sup> to 54 eV at 1.17 Vs cm<sup>-2</sup> to 25 eV at 0.85 Vs cm<sup>-2</sup>, and to 20 eV at 0.6 Vs cm<sup>-2</sup>.

### Design and Implementation

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

AlphaViz is written in Python, and its source code is freely available on GitHub (<https://github.com/MannLabs/alphaviz>) under the Apache license. The AlphaViz implementation combines a comfortable, reproducible and transparent working environment (Jupyter notebooks, GitHub, Binder, pytest) with a Python scientific stack consisting of highly optimized packages with elaborate testing, documentation and maintenance, allowing a focus on domain knowledge rather than implementation details (Fig. 1A). For data analysis in Python, we use NumPy for array manipulation, Pandas to handle tabular data, and Numba to speed up code execution with just-in-time code compilation. Furthermore, we use several open-source Python libraries for proteomics data analysis, such as AlphaTims to access Bruker '.d' files and to convert them to Hierarchical Data Format (HDF) for fast reuse (16), and Pyteomics to handle '.fasta' files (29). A set of well-established plotting libraries was used to generate all plots and a graphical user interface (GUI): (1) Bokeh, Plotly and Holoviews were used to build different types of interactive visualizations; (2) Datashader for fast visualization of large data sets; (3) Panel to implement a fully stand-alone GUI.

The AlphaViz implementation in the GitHub repository is organized into independent functional modules: (1) a 'data' folder with some necessary tables for performing calculations; (2) an 'io' module providing functionality for reading output files of proteomics data analysis programs; (3) a 'preprocessing' module that includes data preprocessing functionality; (4) a 'plotting' module containing all functions creating plots; (5) a 'utils' module including common utilities; (6) and a 'gui' module containing the entire implementation of the AlphaViz GUI. The helper units include: (1) a 'style' folder with files specifying the style of the dashboard elements; (2) an 'img' folder with logos and static images included in the GUI; (3) a 'docs' folder including a comprehensive GUI user guide. Besides the modular 'alphaviz' folder, the repository contains additional important information such as: (1) an 'nbs' folder with Jupyter Notebooks as tutorials for AlphaViz as a Python package usage; (2) the 'test' and 'test\_data' folders containing functions which test the functionality of all previously mentioned Python modules and the necessary test data for them; (3) a general .README file with details on installation, usage of the different AlphaViz modes (GUI or a Python package), contributions and much more; (4) a 'requirements' folder with specific dependencies; (5) all other folders, e.g. 'misc', 'docs', '.github' and 'release'



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

that are involved in a continuous integration pipeline with automatic testing, creation of GUI installers for all OSs, the release of the new versions on GitHub, PyPI (<https://pypi.org/project/alphaviz/>) and 'Read the Docs' (<https://alphaviz.readthedocs.io/en/latest/>).

#### **Modes**

Depending on users' programming skills, AlphaViz can be operated in two modes: a user-friendly browser-based GUI and a well-documented and tested module with Python functionalities.

The AlphaViz GUI has one-click installers provided on the GitHub page for Windows, macOS and Linux (<https://github.com/MannLabs/alphaviz#one-click-gui>). A comprehensive AlphaViz user guide is provided on GitHub.

AlphaViz can be installed from PyPI using the standard pip module. Compared to the GUI, this mode provides more flexibility for users with programming experience, allowing reuse of the plotting or data importing or preprocessing functions to reproduce the same analysis and visualization. To facilitate the use of AlphaViz as a Python package and to lower the entry barrier for users, we created Jupyter notebook tutorials separately for the different available pipelines: for DDA data analyzed with MaxQuant, for DIA data analyzed with DIA-NN, and for the targeted mode without any prior identification. The tutorials offer code to reproduce the results obtained in the GUI.

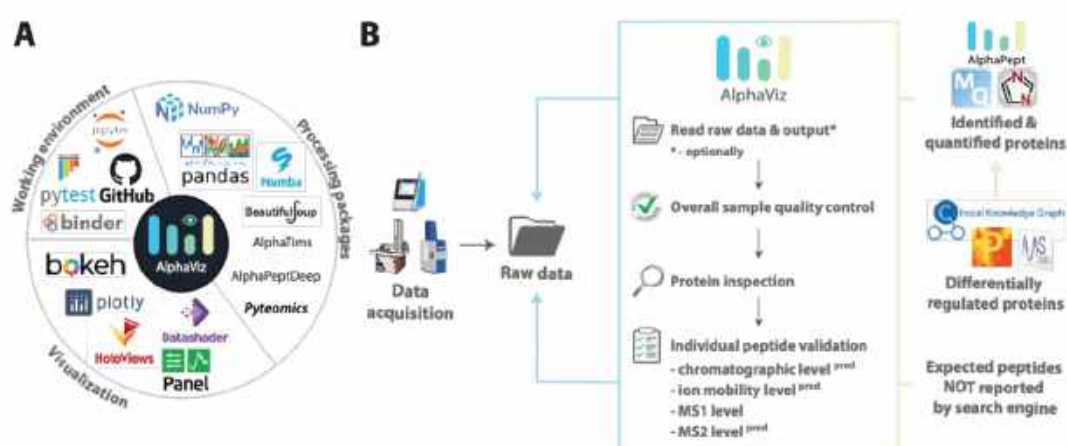
#### **Quality metrics**

We include the following statistical distributions of peptide data which should be checked in AlphaViz to ensure good data quality:

- for DDA data analyzed by MaxQuant (sixteen parameters): m/z, Charge, Length, Mass, 1/K0, CCS, K0 length, Missed cleavages, Andromeda score (peptide score), Intensity, Mass error [ppm], Mass error [Da], Uncalibrated mass error [ppm], Uncalibrated mass error [Da], Score (protein score), (EXP) # peptides (the number of experimentally found peptides);

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- for DDA data analyzed by AlphaPept (eighteen parameters): m/z, Charge, Mass, IM, Length, delta\_m, delta\_m\_ppm, fdr, prec\_offset\_ppm, prec\_offset\_raw, hits, hits\_b, hits\_y, n\_fragments\_matched, (EXP) # peptides, q\_value, score, score\_precursor;
- for DIA data analyzed by DIA-NN (seventeen parameters): m/z, Charge, Length, IM, CScore, Decoy.CScore, Decoy.Evidence, Evidence, Global.Q.Value, Q.Value, Quantity.Quality, Spectrum.Similarity, (EXP) # peptides, Global.PG.Q.Value, PG.Q.Value, PG.Quantity, Protein.Q.Value.



**Fig. 1. AlphaViz project dependencies and workflow.** *A*, Python libraries and other services used in AlphaViz. The Python libraries and services fall into three groups, those for (1) efficient working and test environment; (2) data preprocessing and handling; and (3) visualization and the graphical user interface. *B*, Overview of the AlphaViz workflow. First AlphaViz directly reads the raw data together with the results of the supported proteomics workflows, reporting identified and quantified proteins of interests i.e. differentially regulated proteins. The overall sample quality can then be assessed using various quality metrics as a basis for further evaluation. Next, the user can inspect the individual quality of the critical proteins as well their identified peptides through AlphaViz. This is done at different levels, such as LC, IM, MS1 and MS2 levels, which can also be predicted using the built-in deep learning models for comparison. The 'predict mode' also allows to retrieve the signals from the raw data for peptides of interest that were not reported by the search engine (see Results for further explanation).

## RESULTS

We developed AlphaViz to visually validate critical proteins and peptides at the raw data level. It currently supports timsTOF data acquired in data-dependent acquisition (DDA) or data-independent acquisition (DIA) mode. As detailed in the Experimental Procedures, AlphaViz is written in Python using various open-source libraries for data accession, analysis and visualization. To ensure that the tool can be used by a wide audience, AlphaViz is available on Windows, macOS, and Linux, in two different modes: a convenient graphical user interface (GUI) and as a well-documented and tested Python package.

AlphaViz works either with the output of proteomics software pipelines or only at the raw data level. Using the output results, it first enables the overall quality of a particular sample to be assessed, as a basis for further automated analysis. It then superimposes the identifications provided by common proteomics workflows, such as MaxQuant, AlphaPept, or DIA-NN, on the raw data signals.

For integrating *in silico* predictions of experimental peptide predictions from the (modified) sequences, we use our AlphaPeptDeep package that itself is built on the pDeep model (30–32). In contrast to pDeep, AlphaPeptDeep supports not only MS2 prediction but also retention time (RT) and collisional cross section (CCS) prediction for any peptide modification (33, 34). Furthermore, AlphaPeptDeep provides easy-to-use transfer learning functionalities that were also used in AlphaViz to fine-tune the experiment-specific RT predictions.

These readily available *in silico* predictions for any peptide sequence, enables a distinct ‘predicted mode’ whereby the calculated coordinates of a peptide sequence of interest are projected onto the raw data. A variety of its applications are shown below.

In the following, we employ several use cases or examples to describe the entire validation procedure for peptides of several specific proteins using DDA and DIA data, pinpointing unreliable peptides although they were highly scored by software analysis tools. We then show applications of peptide signals retrieved directly from the raw data based on the predicted or experimental properties of the peptides. Finally, we illustrate the use of AlphaViz to explore critical nodes in the phosphoproteome of the EGF signaling pathway.



### Visual validation of global parameters and individual peptides of critical proteins

Today, researchers typically at best inspect a few examples of all detected peptides, yet rarely the main biological or clinical hits that are the main results of the project. Furthermore, the overall quality of the proteomics dataset is often not examined at the level of MS results. This is partly because it may not be easy in the available software to assess these crucial parameters and results, especially for dda- or dia-PASEF data. Clearly, it would be desirable to be able to confirm at a global level of each LC-MS run that the proteomics data is free of major issues or biases so there is a solid basis for further evaluation. After that, verification should be done individually for each protein of particular interest at the peptide level. This will help to increase confidence in the protein identifications reported by the search engines or result in discarding the identification during the various quality checks as illustrated below.

In Figure 2A we exemplify the entire validation process applied to the DDA data, which originates from a HeLa sample acquired on a timsTOF instrument with a 120-min gradient in dda-PASEF mode and analyzed by MaxQuant (Fig. 2A, Experimental Procedures) (10). We first imported all raw data and MaxQuant results. AlphaViz then displays the overall quality metrics of the raw data to ensure the quality of the MS runs, which is shown for Fraction 1 as an example (Fig. 2B). In the total ion chromatogram (TIC) and base peak intensity (BPI) chromatogram the typical shape and overall high stable intensity level of the MS1 and MS2 TIC reveal no anomalies (Fig. 2B). The MS1 and MS2 BPI also indicate no major issues with saturation of the LC-MS system, such as overloading or contamination (Supplementary Fig. S1A). To dig deeper into the raw data quality, we suggest using AlphaTims, which quickly displays any desired slice of the billions of raw data points (16). Next, we selected six metrics available in AlphaViz to obtain an overview of all the peptides identified by MaxQuant, which revealed typical distributions for  $m/z$  values, peptide lengths, ion mobility values, and number of peptides per protein (Fig. 2B, Supplementary Fig. S1B, Experimental Procedures). However, a clear overall mass-shift is apparent. This is caused by AlphaViz using the raw data directly instead of re-calibrated values after a first database search. When inspecting individual peptides (see below), the re-calibrated mass measurements are used, with a user-definable tolerance, i.e. for visualizing extracted ion chromatogram (XIC) traces. Many peptide metrics are relative to an overall distribution and visualizing their position

respective to the raw data with AlphaViz allows context-specific interpretation. For instance, the Andromeda score (MaxQuant) at 1% FDR shows an interquartile range (IQR) between 47 and 100 with 379 outliers above 177, suggesting that values above 100 should have a very high probability to be correct.

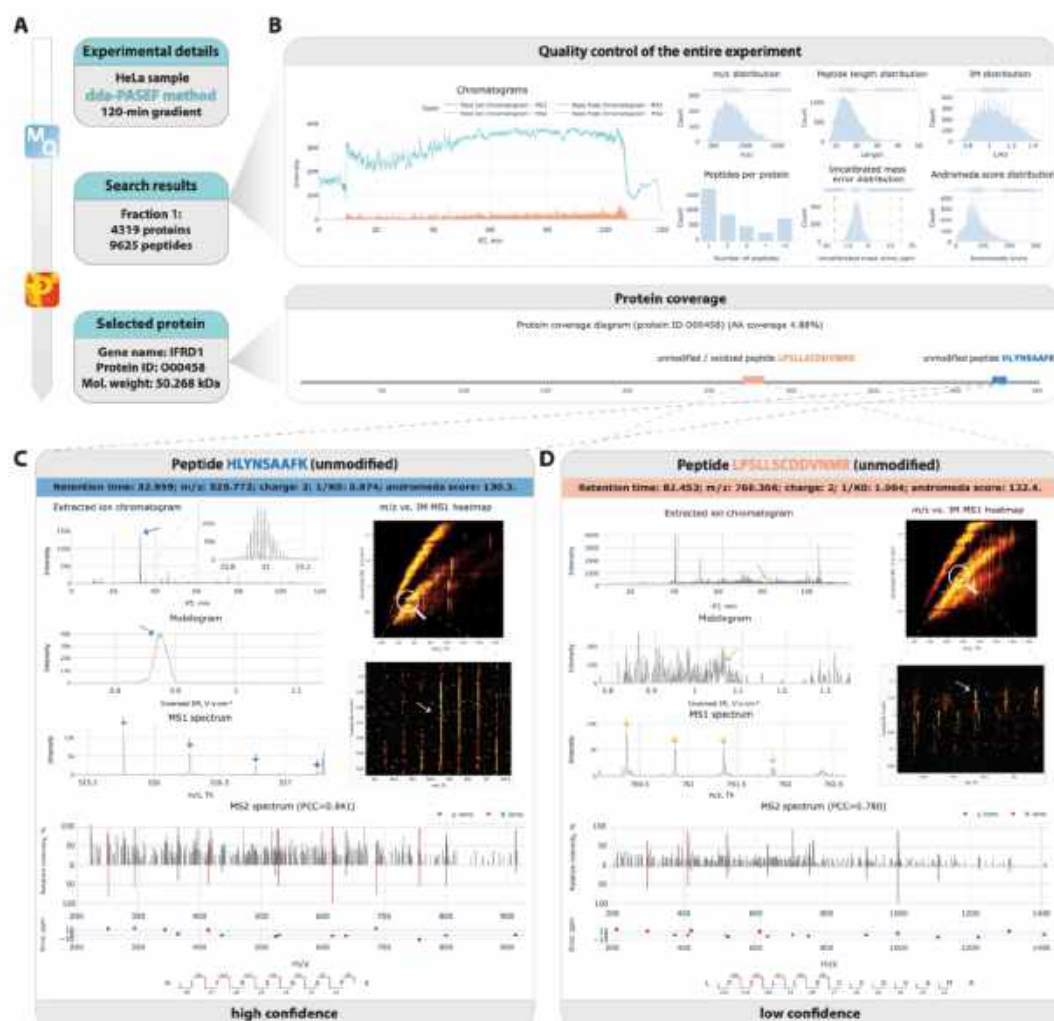
We next inspected peptides of interferon-related developmental regulator 1 protein (ID O00458) to represent a protein of particular biological importance that was identified with only a few peptides and a relatively low protein score of 35 which is in the first quartile of the distribution (Supplementary Fig. S1B). The protein q-value (probability to be wrongly identified) is only about  $10^{-4}$ , derived from the peptide posterior error probabilities (PEPs) of its two identified peptides, one of them also as an oxidized form. The Andromeda scores of the unmodified peptides are 130.3 and 132.4, well within the highest quartile with a PEP of less than 0.6%. Because of the discrepancy of these high peptide scores and the low protein score, we visualized the underlying raw peptide data in AlphaViz.

We first assessed the XIC of the unmodified peptide HLYNSAAFK ( $\pm 15$  ppm,  $\pm 0.05$  1/K0, Fig. 2C). This revealed a pronounced peak at the reported retention time of 32.96 min, close to the value of 34.82 min predicted by AlphaPeptDeep. Moreover, the peak shape was Gaussian with limited tailing. Similarly, the extracted ion mobilogram (ppm and retention time window of  $\pm 15$  ppm and  $\pm 30$  seconds) shows a narrow peak at the reported 1/K0 of 0.874, almost identical to 0.892 predicted by AlphaPeptDeep. This also illustrates the advantage of the additional ion mobility dimension to evaluate the quality of peptide identifications.

AlphaViz can also visualize the MS1 context from which the precursor was picked for sequencing, in this case revealing a well-defined feature in the m/z and ion mobility dimensions (the entire heatmap with the zoomed view in Fig. 2C). All fragment ions for this particular peptide are present in the MS2 spectrum with an average absolute mass error of 3.1 ppm. The spectrum predicted by deep learning in the mirrored spectrum has a similar intensity pattern as the measured one (Pearson correlation coefficient of 0.841, Fig. 2C, bottom panel). Although some peaks remain unidentified, most of the larger peaks are correctly annotated.



We then examined the second unmodified peptide LPSLLSCDDVNMR. Despite its similarly high score, and PEP of  $10^{-4}$ , its XIC was two orders of magnitude less intense and without a well-defined peak shape at the claimed retention time (82.45 min; predicted 83.28 min). Similarly, the extracted ion mobilogram also lacks the expected clear peak shape at 1.064 1/K0 (predicted 1.038 1/K0, Fig. 2D). In comparison to the previously analyzed peptide, it is apparent in the heatmap for the MS1 frame that the peptide of interest was picked in a crowded region, which could potentially lead to a chimeric MS2 spectrum. This goes along with its fuzzy MS1 feature in the  $m/z$  and ion mobility dimensions and a relatively large mass deviation of around 50 ppm (Fig. 1D). In addition, the MS1 spectrum reveals an isotope pattern with some interference from another precursor. However, when inspecting the MS2 spectrum, many ions from the b- and y-series were identified by MaxQuant with a mean mass error of 0.3 ppm and demonstrated a similar intensity pattern with the predicted mirrored spectrum (Pearson correlation coefficient of 0.780, Fig. 2D, bottom panel). This turned out to be the reason for the high Andromeda score of the peptide, which is based on the number of detected fragment ions. Nevertheless, both the low values of the overall absolute peak intensities in the MS2 spectrum (below 300) and the poor data quality in other above-mentioned dimensions suggest that this peptide is a false positive hit despite the high peptide score. Thus, we illustrate the use of AlphaViz to evaluate two identified peptides of the same protein reported by MaxQuant with similar scores, only one of whom should be considered as a reliable hit according to our analysis.



**Fig. 2. Validation pipeline in AlphaViz of two unmodified peptides of the same protein using timsTOF DDA data analyzed by MaxQuant.** *A, Workflow.* A fractionated 120-min HeLa sample, acquired with dda-PASEF and analyzed by MaxQuant (PXD010012) (25), was imported in AlphaViz. *B, Overall sample quality.* Chromatograms and additional quality metrics of fraction 1. Interferon-related developmental regulator 1 protein (ID O00458) was selected for further detailed exploration. The “Protein coverage” bottom panel shows the identified peptides in the sequence context of the protein (similar to, but less detailed than AlphaMap (35)). *C and D, The visualization of XIC, mobilogram, MS1 spectrum with overall and zoomed MS1 heatmaps together with the experimental and predicted MS2 spectrum.* *C, Peptide view.* Inspection of the unmodified peptide HLYNSAAFK reveals it to be a high confidence identification. *D, Peptide view.* The unmodified peptide LPSLLSCDDVNMR has low confidence.

### Visual validation of peptidoforms of the proteins of interest

In recent decades, MS has become the tool of choice for large-scale identification and quantitation of proteins and their post-translational modifications (PTMs) and the computational workflows for the analysis of DDA have matured. DIA analysis is comparatively newer and less established, especially when PTMs are being analyzed (28, 36–39). Although modern proteomics workflows report the localization of the identified PTMs and the associated probabilities, in our experience it is still necessary to manually validate the results for individual proteins and PTMs of critical importance. Compared to DDA data, validation of peptides at the raw level in the DIA pipeline should additionally include detailed inspection at the precursor retention time and, if applicable, ion mobility values. The values extracted from the library should then match the values in the raw data within the experimental error, especially the coelution of matched fragments and precursors.

Figure 3A presents the entire validation process of peptidoforms applied to a HeLa sample acquired on a timsTOF Pro instrument with a 21-min gradient in dia-PASEF mode and analyzed by DIA-NN (Experimental Procedures). We first imported the raw data file of the selected sample along with the DIA-NN output result into AlphaViz. The overall sample quality panel in AlphaViz was used to evaluate the overall quality of the selected sample (A5\_1\_2451) (Fig. 3B). The TICs and BPIs for MS1 and MS2 levels demonstrate typical shapes and overall high level of intensity without any visible anomalies (Fig. 3B, Supplementary Fig. S2A). As before, for further quality checks, such as verification of mass calibration or ion mobility stability, we suggest using AlphaTims (16). By selecting six out of seventeen available quality metrics, we observed typical distributions of important parameters, such as peptide  $m/z$ , ion mobility values and a preponderance of double- and triple-charged ions (Fig. 3B, Supplementary Fig. S2B, Experimental Procedures). Furthermore, a high number of peptides identifying each protein also serves as an important quality assurance. For our example, the peptide score distribution (Quantity.Quality score) in DIA-NN for each individual peptide at 1% FDR shows an IQR between 0.67 and 0.91, suggesting that scores above 0.91 (top 25% of the significant scores) should be correct with a high probability.

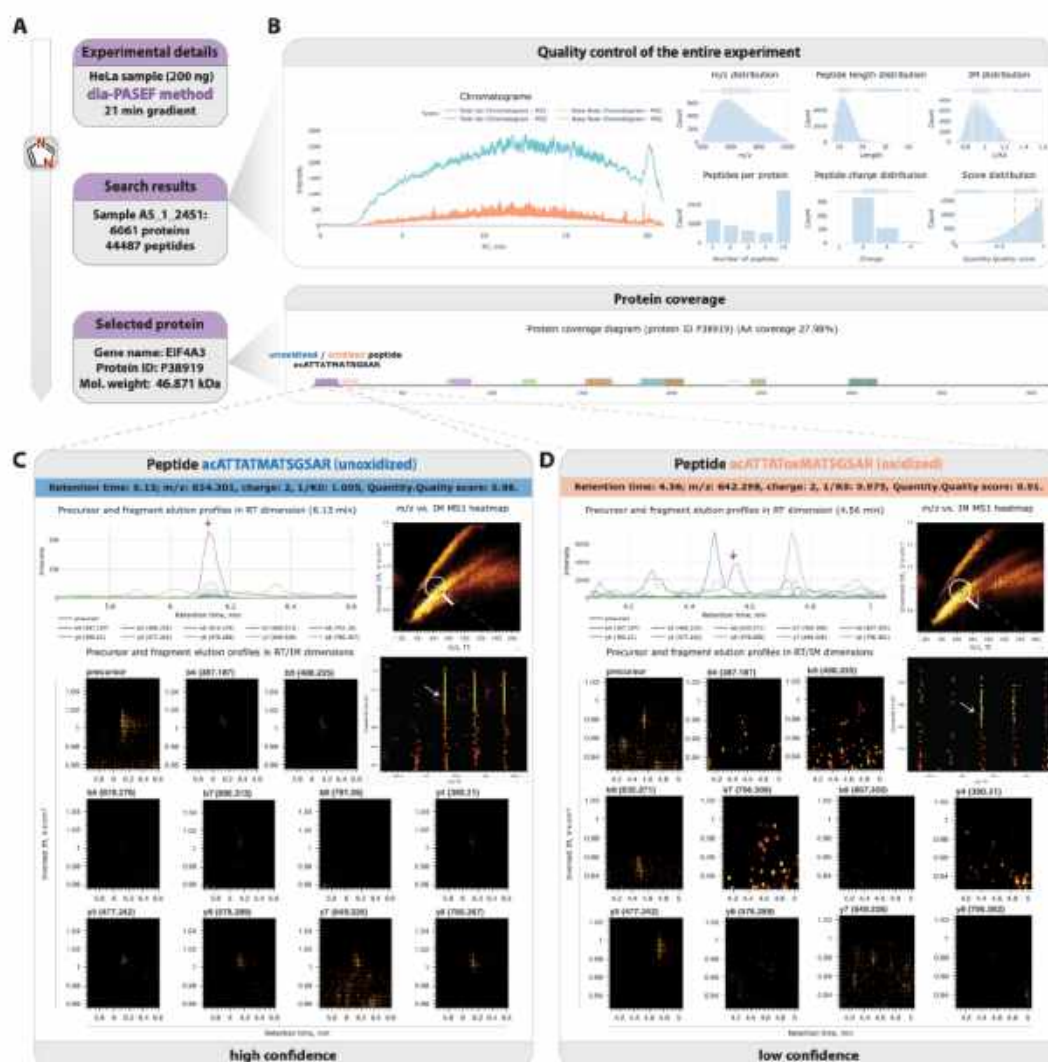


Satisfied with the overall sample quality check, we chose eukaryotic initiation factor 4A-III protein (ID P38919, q-values  $< 10^{-4}$ ) for further investigation because of its high sequence and PTM coverage including a total of 20 peptide variants that mapped to thirteen peptides with q-values  $< 5 \times 10^{-3}$ . Two of these peptidoforms were the unoxidized and oxidized forms of the N-terminally acetylated peptide ATTATMATSGSAR, which were reported with similar scores by DIA-NN of 0.98 and 0.91 respectively.

To investigate if both forms were actually present, we first assessed the MS1 heatmap of the unoxidized peptide ATTATMATSGSAR (Fig. 3C). The extracted position of the peptide on the  $m/z$  versus ion mobility MS1 heatmap revealed a well resolved feature (Fig. 3C). The elution profiles of its precursor with all fragment ions ( $\pm 30$  ppm,  $\pm 0.05$  1/K0,  $\pm 30$  sec) likewise demonstrated a sharp high-intensity precursor peak at 6.13 min (predicted 6.29 min), which coelutes with almost all of the main fragment ions. Taking advantage of ion mobility to ensure the presence of the MS signals, we also visualized the heatmaps for the precursor and each individual fragment in retention time and ion mobility dimensions colored by intensity. These heatmaps confirm the presence of the analyzed peptide.

Investigation of the MS1 heatmap of the oxidized form of the same peptide revealed a comparably well-defined MS1 feature (Fig. 3D). However, analysis of the elution profiles in the retention time dimension only showed a low-intensity precursor peak at 4.56 min (predicted 4.07 min) with only few and unaligned peaks. Conversely, the heatmaps for the precursor ion and its fragment ions confirm the absence of fragment ions signals within the expected retention time and ion mobility ranges. Correct identification of modified peptides in DIA data is a known challenge and is thought to be impeded by the presence of shared unmodified fragments of the base peptide in the DIA matching library (38).

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



**Fig. 3. Validation pipeline in AlphaViz of two peptide variants using timsTOF DIA data analyzed by DIA-NN. A, Workflow.** A 21-min HeLa sample, acquired with dia-PASEF and analyzed by DIA-NN (PXD017703), was imported into AlphaViz (26, 27). **B, Overall sample quality.** Chromatograms and additional quality metrics of fraction 1. Eukaryotic initiation factor 4A-III protein (ID P38919) was selected for further detailed exploration. The “Protein coverage” bottom panel shows the identified peptides in the sequence context of the protein. **C and D, Visualization of overall and zoomed-in MS1 heatmaps, precursor and fragments elution profiles in both retention time (line plots) and retention time and ion mobility (heatmaps) dimensions.** **C, Peptide view.** The unoxidized N-terminal acetylated peptide ATTATMATSGSAR shows high confidence. **D, Peptide view.** The oxidized peptidofrom of the same peptide demonstrates low confidence.

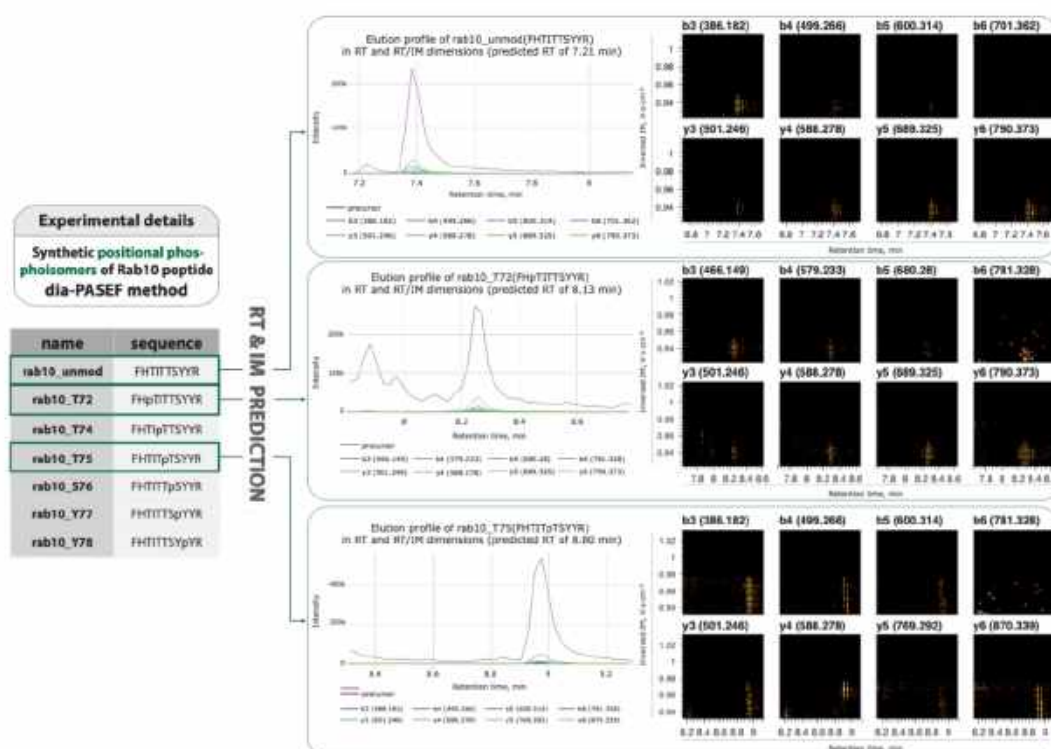


#### **Validation of peptides directly in DIA raw data with the ‘predict mode’ in AlphaViz**

The processing of DIA data is computationally challenging due to the high complexity of MS2 spectra containing fragments of multiple precursors from each single isolation window. Peptides with low signal intensities can be difficult to detect and easy to misinterpret. Conversely, quality measures of such peptides, provided as q-values, often fall below preset thresholds, resulting in “missing values”, even though the peptide is actually present. We hypothesized that by manually visualizing their signals in AlphaViz, they could still be extractable from the raw data. This is enabled by the integration of deep learning assisted prediction of retention time and ion mobility of AlphaPeptDeep in AlphaViz. We tested our hypothesis with two use cases: one regarding the detection of positional isoforms of a synthetic phosphopeptide and one regarding the retrieval of missing values.

We had previously identified the Rab10 protein as a clinically important substrate in Parkinson’s disease (40, 41). In the course of developing an assay to measure the phosphorylation site occupancy, we had synthesized positional phosphoisomers of the Rab10 peptide FHTITTSYYR (Fig. 4, left panel) (42). When analyzing these peptides by DIA software, they were not reported to be present (Experimental procedure). By predicting the retention time and ion mobility values for the different charges of two known phosphoisomers, their elution profiles were easily detected in the raw data of the two highest concentrations (Fig. 4, right panel, Supplementary Fig. S3). The XICs ( $\pm$  30 ppm,  $\pm$  0.05 1/K0,  $\pm$  30 sec) demonstrate clearly defined high-intense precursor peaks with coeluting b3-b6 and y3-y6 fragment ions. The presence of these fragment ion signals is further confirmed by heatmaps that take advantage of the additional ion mobility dimension. Note that there is a slight difference between the predicted and actually observed retention time, which is not unexpected given the estimated accuracy of the prediction (32). Given the co-elution behavior of the expected fragments in the AlphaViz, we confirmed the presence of the intended phosphoisomers and concluded that they were not detected in all samples because of low MS signals.

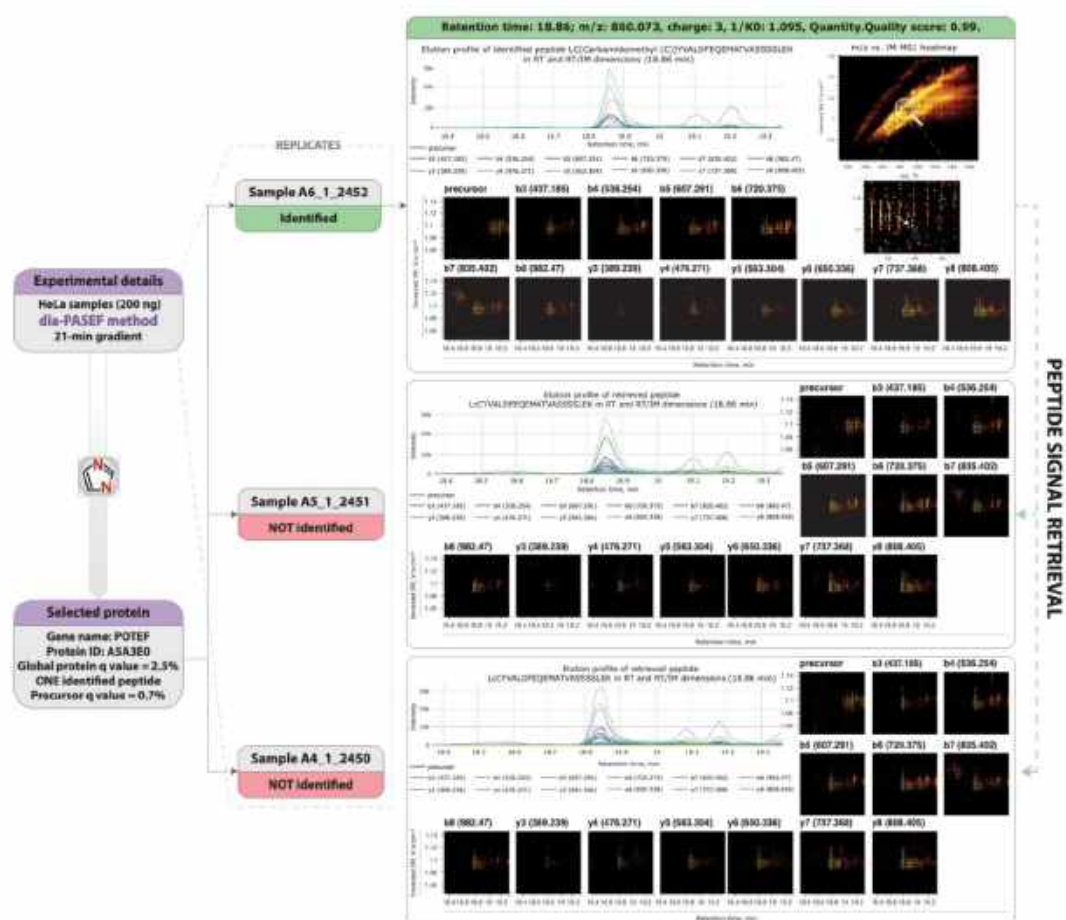
bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



**Fig. 4. Validation of the presence of synthetic phosphoisomers of Rab10 peptide in DIA raw data.** Left panel, Synthetic positional phospho-isomers of the Rab10 peptides. Right panel, Extracted peptide signals for the sequences with a green box in the left panels. Heat map of transitions from the raw data for the unmodified peptide and its two phosphoisomers.

To test if AlphaViz could retrieve seemingly missing values, we used the same HeLa DIA dataset as for the visual validation of peptide-forms above (Experimental Procedures). For illustration, we selected the cysteine-carboxylated peptide LCYVALDFEQEMATVASSSSLEK, which was the only one identifying the POTE ankyrin domain family member F protein (ID A5A3E0). It was only reported by DIA-NN in one of three technical replicate analyses (Sample A6\_1\_2452) but with a high peptide q-value of  $7 \times 10^{-3}$ . To assure that the protein is really present, we investigated the raw data signal of this peptide in AlphaViz (Fig. 5). For the replicate in which the peptide was identified, the position of the peptide on the MS1 heatmap is in a crowded part of the ion cloud, but the zoomed-in view revealed no interfering peptides (Fig. 5). The XICs and heatmaps for the peptide and fragments (b3-b8 and y3-y8 fragment ion series) confirm the presence of the peptide. Taking into account the information about the detected peptide, such as its retention

time, charge, and ion mobility, we were able to retrieve this peptide signal in the other two replicates where this peptide had not been reported. Interestingly, we found high quality signals for this peptide comparable to the first sample in both remaining copies, suggesting that improvements to the software could in the future lead to even higher data completeness.



**Fig. 5. Validation of peptide signal presence in all replicates of an experiment.** Left part, *Experimental details*. Three replicates of a HeLa analysis on a timsTOF pro instrument analyzed by DIA-NN (PXD017703) (26, 27). The POTE ankyrin domain family member F protein (ID ASA3E0) with only a single cysteine-carboxylated peptide LCYVALDFEQEMATVASSSSLEK identified in one replicate (Sample A6\_1\_2452) was selected for further investigation. Right part, *Extracted peptide signals*. Raw data extracted for the identified peptide (top). Based on the known peptide sequence, the raw signal of this peptide was successfully extracted from the two remaining samples in which the peptide was not initially identified.



### Data quality assessment and discovery of EGF signaling events using AlphaViz

Studies on post-translational modifications (PTMs) by their nature rely on the identification and quantification of single peptides and are especially affected by poor peak qualities and missing values, the two major challenges AlphaViz tries to overcome. We investigated signalling events activated along the well-studied epidermal growth factor (EGF) signaling pathway. Binding of EGF to its receptor EGFR induces a signalling cascade mediated by phosphorylation leading to cellular proliferation, differentiation and survival (43). We used a recently published dataset where Hela cells were stimulated with EGF or left untreated, acquired in three replicates each on a timsTOF pro instrument with a 21-min gradient in dia-PASEF mode and analyzed with DIA-NN (Experimental Procedures) (28). When filtered for 100% valid values in each condition, DIA-NN detected 1,403 phosphosites as significantly upregulated, of which 56 were localized on proteins known to be part of the EGFR signaling pathway (according to Gene Ontology Biological Process (GOBP) (44). To evaluate the data quality of regulated phosphosites, we picked significantly upregulated phosphosites with DIA-NN scores  $> 0.7$  ( $FDR < 0.05$ ). The majority of regulated phosphosites with higher DIA-NN scores showed well correlating elution profiles of precursor and fragment ions, for example the peptide carrying the phosphorylation on S642 of RAF1 (Fig. 6A, green in Fig. 6B). However, others demonstrated poor data quality (red in Fig. 6A). This also affected phosphorylation on proteins known to be associated with the EGFR signaling pathway that are presumably correct. For example, the CBL (Casitas B-lineage Lymphoma) protein, an E3 ligase known to ubiquitylate EGFR showed poor quality elution profiles for its peptides phosphorylated on S619 and S667 despite a maximum DIA-NN score between replicates of 0.92 and 0.94 respectively (red in Fig. 6B) (45). Specifically, AlphaViz only retrieved an elution peak for the precursor but none of the expected fragments co-eluted. This was the case for a number of peptides in the EGFR pathway (red in Fig. 6A). We assume that the neural network in DIA-NN scored the presence of the peptide in these cases mainly based on the precursor. While this may be justified in these cases, it would be problematic without supporting biological a priori information. We hope that this observation will initiate improvement to software tools – for instance it could be reported that matching was only based on MS1 level. In any case, we recommend to employ AlphaViz for data quality checks before extensive follow-up experiments.

A second challenge are missing values, especially in PTM studies. Although this problem is much reduced in DIA compared to DDA data, it still occurs frequently especially as sample size grows (39, 46–48). This is due to the complexity of spectra, the low abundance of modified peptides and the technical variability. Unfortunately, it has been impossible or extremely laborious to manually check raw data for specific spectra or elution groups of modified peptides that were not reported by the proteomics workflows. The predict mode in AlphaViz addresses this issue. It only requires the peptide sequence, peptide charge and type and localization of its modification to predict its retention time, ion mobility and fragment intensities.

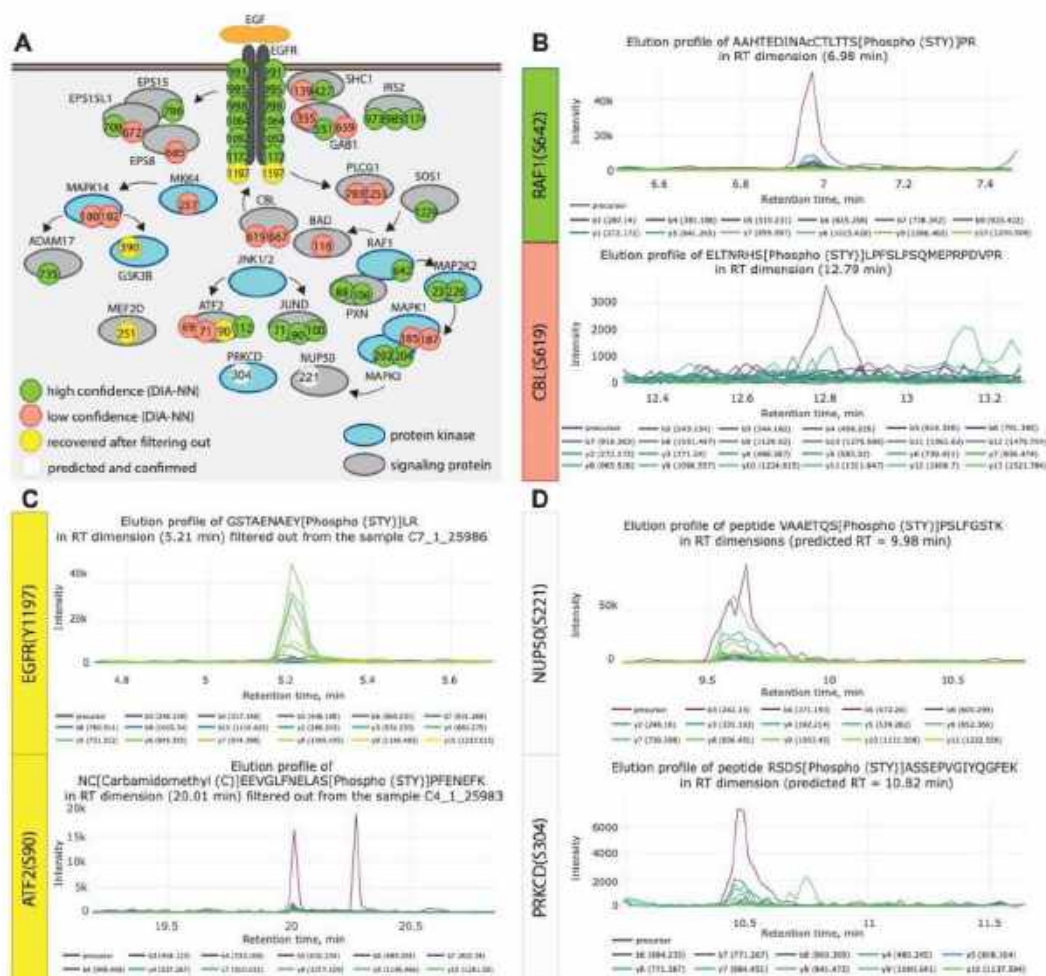
We first investigated functional phosphorylation events on proteins involved in EGF signalling that showed increased intensities upon EGF treatment, but were lost due to filtering the dataset for 100 % valid values in at least one condition. In most of these cases, the data quality of phosphopeptides in replicates where the DIA software did not report intensities was comparable to the respective replicates with reported intensities. This affected phosphorylation events on proteins along the whole EGF signalling pathway starting with the EGFR receptor itself (Y1197), kinases regulating downstream signalling like GSK3B (T390) and phosphorylation events activating transcription factors like MEK2 (S218) and ATF2 (S90) (Fig. 6C, Supplementary Fig. S4). These examples are clearly false negatives of the computational pipeline and they prove the potential of our tool to recover biologically correct regulatory sites. In the case of novel sites, AlphaViz could have prevented them from being discarded because of data incompleteness.

Besides these reported regulatory phosphosites, the predict mode also provides the possibility to look for phosphorylation events in the raw data that have not been identified by the proteomics software at all. In these cases, AlphaViz uses the peptide sequence, PTM localization and charge state of modified peptides of interest to retrieve the corresponding locations in the raw data. To illustrate, in the EGF dataset, we would have expected increased phosphorylation of the nuclear pore complex protein 50 (NUP50) on position S221, which is mediated by the extracellular signal-regulated kinases (ERK) downstream of the EGF receptor (49), but no such peptide was reported. Remarkably, elution group profiles at the predicted retention time and ion mobility were of good quality and confirmed the presence of this phosphopeptide in the sample (Fig. 6D). This was also the case in a second example, relating to EGF-induced activation of the



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

protein kinase C delta (PRKCD), a kinase regulating cell-adhesion upon EGF stimulation, which leads to its autophosphorylation at S304 (50, 51). Hence, the predict mode allows us to efficiently investigate specific signalling events of interest that were missed by MS software tools through direct inspection of the raw data.



**Fig. 6. Investigation of EGF-induced phosphorylation events in AlphaViz.** A, Scheme of significantly upregulated phosphosites investigated in AlphaViz. Based on visual inspection, we divided the reported phosphosites into four main groups: high confidence (green), low confidence (red), recovered after filtering (yellow), and predicted and confirmed (white). B, Elution profiles of two phosphorylation sites with high (S642 of RAF1) and low (S619 of CBL) confidence. C, Elution profiles of two phosphorylation sites (Y1197 of EGFR and S90 of ATF2) recovered after filtering out. D, Elution profiles of two phosphorylation sites (S221 of NUP50 and S304 of PRKCD) not reported by DIA-NN but found in the data using predict mode.

## DISCUSSION

Increasingly automated and capable proteomics processing workflows provide researchers with an easy way to summarize the results of proteomics experiments with statistical confidence measures. However, expert evaluation of individual peptides and proteins is lost along the way. To remedy this, we have developed AlphaViz, a Python-based software package for easy visual validation of critical identifications. Like other members of the AlphaPept ecosystem, it adheres to modern and robust software development principles and it is available to community as a Python module and the GUI for end users.

Future implementations of AlphaViz will include the support for more MS platforms, especially Orbitrap instruments, as well as the integration of results from other used proteomics software packages. Furthermore, due to the well-documented and tested open-source code, AlphaViz is easily extendable by bioinformaticians who want to integrate the latest cutting-edge ideas, as already demonstrated by AlphaPept and AlphaTims (13, 16). Additionally, directly linking protein candidates from fully automated downstream analysis packages like the clinical knowledge graph will further strengthen the link between raw data and biological insight (14). Since the visualization capabilities of AlphaViz are only limited by data structure, it can also be used for the in-depth inspection of lipidomics and metabolomics data.

Here we have demonstrated how AlphaViz can quickly give the researcher confidence in identified, critical peptides by inspection of the search results with the raw chromatographic, ion mobility, MS1 and MS2 levels. Conversely, visualization strongly suggests that some peptides are likely false positives despite of their high search engine scores. Furthermore, AlphaViz makes use of the revolution in deep learning enabled prediction of experimental peptide properties from the identified amino acid sequence. This feature is the basis for the ‘predict mode’, in which we retrieve the raw data for peptides that were not reported by the automated workflow, but were potentially present in the data. This may allow the rescue of low-level signals that are biologically expected to be present or are present in some but not all replicates. In phosphoproteomics of the EGF signaling pathway, we showed how this can help to validate the presence of reported

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

signal nodes, and to avoid extensive follow-up experiments for novel phosphorylation sites whose raw data make them unlikely to be true.

In conclusion, we believe researchers will profit from the minimal time investment to visually check their critical peptides and proteins, potentially saving the community and themselves from futile follow up work. This is particularly true of very surprising and biologically unexpected results that then fail to be reproduced by the wider community. In this context, journals could encourage or mandate the inclusion of such extra data and visualizations for the critical peptides or peptidoforms that form the basis of the new hypotheses, helping to address the 'crisis of reproducibility' (52, 53).



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

## ACKNOWLEDGEMENTS

We thank our colleagues in the Department of Proteomics and Signal Transduction (Max Planck Institute of Biochemistry). We are grateful to Stephan Uebel and Stephan Pettera from the core facility and to Özge Karayel for providing experimental data on phosphoisomers. We are thankful to Maria Wahle and Corazon Ericka Mae Itang for testing and providing critical feedback on AlphaViz, and to Medini Steger for great help in editing the manuscript and user manual.

## FUNDING AND ADDITIONAL INFORMATION

This study was supported by the Bavarian State Ministry of Health and Care through the research project DigiMed Bayern ([www.digimed-bayern.de](http://www.digimed-bayern.de)), by the Max-Planck Society for Advancement of Science and by the Deutsche Forschungsgemeinschaft (DFG) project 'Chemical proteomics inside us' (grant 412136960). M.T.S. is supported financially by the Novo Nordisk Foundation (Grant agreement NNF14CC0001).

## DATA AVAILABILITY

The source code and user guide are available under Apache 2.0 license and can be found on GitHub at <https://github.com/MannLabs/alphaviz>. The mass spectrometry proteomics data have been deposited to the ProteomeXchange Consortium *via* the PRIDE partner repository (54) with the dataset identifier PXD034223.

## AUTHOR CONTRIBUTIONS

E.V. and M.M. conceptualized and designed the study; P.S. performed the experiments; E.V. implemented the Python code and GUI; S.W., W-F.Z., and M.T.S. reviewed and contributed the code; E.V., S.W., P.S., M.C.T., and M.M. analyzed the data; S.W., A.-D.B., P.S., and M.T. provided valuable ideas for the concept and visualization in AlphaViz; E.V., S.W., P.S., M.C.T., A.-D.B., and M.M. wrote the manuscript; S.W. and M.M. coordinated and supervised the study.

## CONFLICT OF INTEREST

The authors declare no conflicts of interest with the contents of this article.

## SUPPLEMENTAL DATA

This article contains supplemental data.

## REFERENCES

1. Aebersold, R., and Mann, M. (2016) Mass-spectrometric exploration of proteome structure and function. *Nature* 537, 347–355
2. Larance, M., and Lamond, A. I. (2015) Multidimensional proteomics for cell biology. *Nature Reviews Molecular Cell Biology* 2015 16:5 16, 269–280
3. Required Manuscript Content and Publication Guidelines: Molecular & Cellular Proteomics
4. Oveland, E., Muth, T., Rapp, E., Martens, L., Berven, F. S., and Barsnes, H. (2015) Viewing the proteome: How to visualize proteomics data? *PROTEOMICS* 15, 1341–1355
5. Schessner, J. P., Voytik, E., and Bludau, I. (2022) A practical guide to interpreting and generating bottom-up proteomics data visualizations. *PROTEOMICS*, 2100103
6. Brunner, A.-D., Thielert, M., Vasilopoulou, C., Ammar, C., Coscia, F., Mund, A., Hoerning, O. B., Bache, N., Apalategui, A., Lubeck, M., Richter, S., Fischer, D. S., Raether, O., Park, M. A., Meier, F., Theis, F. J., and Mann, M. (2022) Ultra-high sensitivity mass spectrometry quantifies single-cell proteome changes upon perturbation. *Molecular Systems Biology* 18, e10798
7. Gebreyesus, S. T., Siyal, A. A., Kitata, R. B., Chen, E. S. W., Enkhbayar, B., Angata, T., Lin, K. I., Chen, Y. J., and Tu, H. L. (2022) Streamlined single-cell proteomics by an integrated microfluidic chip and data-independent acquisition mass spectrometry. *Nature Communications* 2022 13:1 13, 1–13
8. Williams, S. M., Liyu, A. v., Tsai, C. F., Moore, R. J., Orton, D. J., Chrisler, W. B., Gaffrey, M. J., Liu, T., Smith, R. D., Kelly, R. T., Pasa-Tolic, L., and Zhu, Y. (2020) Automated Coupling of Nanodroplet Sample Preparation with Liquid Chromatography-Mass Spectrometry for High-Throughput Single-Cell Proteomics. *Analytical Chemistry* 92, 10588–10596
9. Mund, A., Coscia, F., Kriston, A., Hollandi, R., Kovács, F., Brunner, A.-D., Migh, E., Schweizer, L., Santos, A., Bzorek, M., Naimy, S., Rahbek-Gjerdum, L. M., Dyring-Andersen, B., Bulkescher, J., Lukas, C., Eckert, M. A., Lengyel, E., Gnann, C., Lundberg, E., Horvath, P., and Mann, M. (2022) Deep Visual Proteomics defines single-cell identity and heterogeneity. *Nature Biotechnology* 2022, 1–10
10. Cox, J., and Mann, M. (2008) MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat Biotechnol* 26, 1367–1372
11. Tyanova, S., Temu, T., Carlson, A., Sinitcyn, P., Mann, M., and Cox, J. (2015) Visualization of LC-MS/MS proteomics data in MaxQuant. *Proteomics* 15, 1453–1456
12. Bittremieux, W., Adams, C., Laukens, K., Dorrestein, P. C., and Bandeira, N. (2021) Open Science Resources for the Mass Spectrometry-Based Analysis of SARS-CoV-2. *J. Proteome Res.* 20, 1464–1475
13. Strauss, M. T., Bludau, I., Zeng, W.-F., Voytik, E., Ammar, C., Schessner, J., Ilango, R., Gill, M., Meier, F., Willems, S., and Mann, M. (2021) AlphaPept, a modern and open framework for MS-based proteomics. *bioRxiv*, 2021.07.23.453379
14. Santos, A., Colaço, A. R., Nielsen, A. B., Niu, L., Strauss, M., Geyer, P. E., Coscia, F., Albrechtsen, N. J. W., Mundt, F., Jensen, L. J., and Mann, M. (2022) A knowledge graph to interpret clinical proteomics data. *Nature Biotechnology* 2022, 1–11
15. Meier, F., Park, M. A., and Mann, M. (2021) Trapped ion mobility spectrometry and parallel accumulation–serial fragmentation in proteomics. *Molecular and Cellular Proteomics* 20, 100138



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

16. Willems, S., Voytik, E., Skowronek, P., Strauss, M. T., and Mann, M. (2021) AlphaTims: Indexing Trapped Ion Mobility Spectrometry-TOF Data for Fast and Easy Accession and Visualization. *Mol Cell Proteomics* 20, 100149
17. Łacki, M. K., Startek, M. P., Brehmer, S., Distler, U., and Tenzer, S. (2021) OpenTIMS, TimsPy, and TimsR: Open and Easy Access to timsTOF Raw Data. *Journal of Proteome Research* 20, 2122–2129
18. Wen, B., Zeng, W. F., Liao, Y., Shi, Z., Savage, S. R., Jiang, W., and Zhang, B. (2020) Deep Learning in Proteomics. *PROTEOMICS* 20, 1900335
19. Mann, M., Kumar, C., Zeng, W. F., and Strauss, M. T. (2021) Artificial intelligence for proteomics and biomarker discovery. *Cell Systems* 12, 759–770
20. Schmidt, T., Samaras, P., Dorfer, V., Panse, C., Kockmann, T., Bichmann, L., van Puyvelde, B., Perez-Riverol, Y., Deutsch, E. W., Kuster, B., and Wilhelm, M. (2021) Universal Spectrum Explorer: A Standalone (Web-)Application for Cross-Resource Spectrum Comparison. *Journal of Proteome Research* 20, 3388–3394
21. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S., and Ralser, M. (2019) DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. *Nature Methods* 17:1 17, 41–44
22. Bruderer, R., Bernhardt, O. M., Gandhi, T., Miladinović, S. M., Cheng, L.-Y., Messner, S., Ehrenberger, T., Zanotelli, V., Butscheid, Y., Escher, C., Vitek, O., Rinner, O., and Reiter, L. (2015) Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues. *Molecular & Cellular Proteomics : MCP* 14, 1400
23. MacLean, B., Tomazela, D. M., Shulman, N., Chambers, M., Finney, G. L., Frewen, B., Kern, R., Tabb, D. L., Liebler, D. C., and MacCoss, M. J. (2010) Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics* 26, 966–968
24. Perez-Riverol, Y., Xu, Q.-W., Wang, R., Uszkoreit, J., Griss, J., Sanchez, A., Reisinger, F., Csordas, A., Ternent, T., Del-Toro, N., Dianes, J. A., Eisenacher, M., Hermjakob, H., and Vizcaino, J. A. (2016) PRIDE Inspector Toolsuite: Moving Toward a Universal Visualization Tool for Proteomics Data Standard Formats and Quality Assessment of ProteomeXchange Datasets. *Mol Cell Proteomics* 15, 305–317
25. Meier, F., Brunner, A. D., Koch, S., Koch, H., Lubeck, M., Krause, M., Goedecke, N., Decker, J., Kosinski, T., Park, M. A., Bache, N., Hoerning, O., Cox, J., Räther, O., and Mann, M. (2018) Online Parallel Accumulation-Serial Fragmentation (PASEF) with a Novel Trapped Ion Mobility Mass Spectrometer. *Molecular & Cellular Proteomics : MCP* 17, 2534–2545
26. Meier, F., Brunner, A. D., Frank, M., Ha, A., Bludau, I., Voytik, E., Kaspar-Schoenefeld, S., Lubeck, M., Räther, O., Bache, N., Aebersold, R., Collins, B. C., Röst, H. L., and Mann, M. (2020) diaPASEF: parallel accumulation-serial fragmentation combined with data-independent acquisition. *Nat Methods* 17, 1229–1236
27. Demichev, V., Yu, F., Teo, G. C., Szyrwił, L., Rosenberger, G. A., Decker, J., Kaspar-Schoenefeld, S., Lilley, K. S., Müller, M., Nesvizhskii, A. I., and Ralser, M. (2021) High sensitivity dia-PASEF proteomics with DIA-NN and FragPipe. *bioRxiv*, 2021.03.08.434385
28. Skowronek, P., Thielert, M., Voytik, E., Tanzer, M. C., Hansen, F. M., Willems, S., Karayel, O., Brunner, A.-D., Meier, F., and Mann, M. (2022) Rapid and in-depth coverage of the

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- (phospho-)proteome with deep libraries and optimal window design for dia-PASEF. *bioRxiv*, 2022.05.31.494163
29. Goloborodko, A. A., Levitsky, L. I., Ivanov, M. v. and Gorshkov, M. v (2013) Pyteomics— a Python Framework for Exploratory Data Analysis and Rapid Software Prototyping in Proteomics. *J. Am. Soc. Mass Spectrom.* 24, 301–304
  30. Zeng, W. F., Zhou, X. X., Zhou, W. J., Chi, H., Zhan, J., and He, S. M. (2019) MS/MS Spectrum prediction for modified peptides using pDeep2 Trained by Transfer Learning. *Analytical Chemistry* 91, 9724–9731
  31. Tarn, C., and Zeng, W. F. (2021) PDeep3: Toward More Accurate Spectrum Prediction with Fast Few-Shot Learning. *Analytical Chemistry* 93, 5815–5822
  32. <https://github.com/MannLabs/alphapeptdeep>
  33. Müller, J. B., Geyer, P. E., Colaço, A. R., Treit, P. v, Strauss, M. T., Oroshi, M., Doll, S., Virreira Winter, S., Bader, J. M., Köhler, N., Theis, F., Santos, A., and Mann, M. (2020) The proteome landscape of the kingdoms of life. *Nature* 582, 592–596
  34. Meier, F., Köhler, N. D., Brunner, A. D., Wanka, J. M. H., Voytik, E., Strauss, M. T., Theis, F. J., and Mann, M. (2021) Deep learning the collisional cross sections of the peptide universe from a million experimental values. *Nature Communications* 2021 12:1 12, 1–12
  35. Voytik, E., Bludau, I., Willems, S., Hansen, F. M., Brunner, A.-D., Strauss, M. T., and Mann, M. (2022) AlphaMap: an open-source Python package for the visual annotation of proteomics data with sequence-specific knowledge. *Bioinformatics* 38, 849–852
  36. Rosenberger, G., Liu, Y., Röst, H. L., Ludwig, C., Buil, A., Bensimon, A., Soste, M., Spector, T. D., Dermizakis, E. T., Collins, B. C., Malmström, L., and Aebersold, R. (2017) Inference and quantification of peptidofoms in large sample cohorts by SWATH-MS. *Nature Biotechnology* 2017 35:8 35, 781–788
  37. Gillet, L. C., Navarro, P., Tate, S., Röst, H., Selevsek, N., Reiter, L., Bonner, R., and Aebersold, R. (2012) Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis. *Mol Cell Proteomics* 11,
  38. Bekker-Jensen, D. B., Bernhardt, O. M., Høgrebe, A., Martinez-Val, A., Verbeke, L., Gandhi, T., Kelstrup, C. D., Reiter, L., and Olsen, J. v. (2020) Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. *Nature Communications* 2020 11:1 11, 1–12
  39. Tanzer, M. C., Bludau, I., Stafford, C. A., Hornung, V., and Mann, M. (2021) Phosphoproteome profiling uncovers a key role for CDKs in TNF signaling. *Nat Commun* 12,
  40. Steger, M., Tonelli, F., Ito, G., Davies, P., Trost, M., Vetter, M., Wachter, S., Lorentzen, E., Duddy, G., Wilson, S., Baptista, M. A. S., Fiske, B. K., Fell, M. J., Morrow, J. A., Reith, A. D., Alessi, D. R., and Mann, M. (2016) Phosphoproteomics reveals that Parkinson’s disease kinase LRRK2 regulates a subset of Rab GTPases. *Elife* 5,
  41. Steger, M., Diez, F., Dhekne, H. S., Lis, P., Nirujogi, R. S., Karayel, O., Tonelli, F., Martinez, T. N., Lorentzen, E., Pfeffer, S. R., Alessi, D. R., and Mann, M. (2017) Systematic proteomic analysis of LRRK2-mediated rab GTPase phosphorylation establishes a connection to ciliogenesis. *Elife* 6,
  42. Karayel, Ö., Tonelli, F., Winter, S. V., Geyer, P. E., Fan, Y., Sammler, E. M., Alessi, D. R., Steger, M., and Mann, M. (2020) Accurate MS-based Rab10 Phosphorylation



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.12.499676>; this version posted July 13, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- Stoichiometry Determination as Readout for LRRK2 Activity in Parkinson's Disease. *Molecular & Cellular Proteomics : MCP* 19, 1546
43. Herbst, R. S. (2004) Review of epidermal growth factor receptor biology. *International Journal of Radiation Oncology\*Biophysics* 59, S21–S26
44. Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000) Gene Ontology: tool for the unification of biology. *Nat Genet* 25, 25
45. de Melker, A. A., van der Horst, G., Calafat, J., Jansen, H., and Borst, J. (2001) c-Cbl ubiquitinates the EGF receptor at the plasma membrane and remains receptor associated throughout the endocytic route. *J Cell Sci* 114, 2167–2178
46. Bekker-Jensen, D. B., Bernhardt, O. M., Hogrebe, A., Martinez-Val, A., Verbeke, L., Gandhi, T., Kelstrup, C. D., Reiter, L., and Olsen, J. v. (2020) Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. *Nature Communications* 2020 11:1 11, 1–12
47. Hansen, F. M., Tanzer, M. C., Brüning, F., Bludau, I., Stafford, C., Schulman, B. A., Robles, M. S., Karayel, O., and Mann, M. (2021) Data-independent acquisition method for ubiquitinome analysis reveals regulation of circadian biology. *Nat Commun* 12,
48. Steger, M., Demichev, V., Backman, M., Ohmayer, U., Ihmor, P., Müller, S., Ralser, M., and Daub, H. (2021) Time-resolved in vivo ubiquitinome profiling by DIA-MS reveals USP7 targets on a proteome-wide scale. *Nature Communications* 2021 12:1 12, 1–13
49. Kosako, H., Yamaguchi, N., Aranami, C., Ushiyama, M., Kose, S., Imamoto, N., Taniguchi, H., Nishida, E., and Hattori, S. (2009) Phosphoproteomics reveals new ERK MAP kinase targets and links ERK to nucleoporin-mediated nuclear transport. *Nat Struct Mol Biol* 16, 1026–1035
50. Rybin, V. O., Guo, J., Harleton, E., Feinmark, S. J., and Steinberg, S. F. (2009) Regulatory autophosphorylation sites on protein kinase C-delta at threonine-141 and threonine-295. *Biochemistry* 48, 4642–4651
51. Singh, R. K., Tapia-Santos, A., Bebee, T. W., and Chandler, D. S. (2009) Conserved sequences in the final intron of MDM2 are essential for the regulation of alternative splicing of MDM2 in response to stress. *Exp Cell Res* 315, 3419–3432
52. Baker, M. (2016) 1,500 scientists lift the lid on reproducibility. *Nature* 533, 452–454
53. Crüwell, S., van Doorn, J., Etz, A., Makel, M. C., Moshontz, H., Niebaum, J. C., Orben, A., Parsons, S., and Schulte-Mecklenbeck, M. (2019) Seven easy steps to open science: An annotated reading list. *Zeitschrift für Psychologie* 227, 237
54. Perez-Riverol, Y., Bai, J., Bandla, C., García-Seisdedos, D., Hewapathirana, S., Kamatchinathan, S., Kundu, D. J., Prakash, A., Frericks-Zipper, A., Eisenacher, M., Walzer, M., Wang, S., Brazma, A., and Vizcaíno, J. A. (2022) The PRIDE database resources in 2022: a hub for mass spectrometry-based proteomics evidences. *Nucleic Acids Research* 50, D543–D552

### 3.3. Article 3: AlphaMap: an open-source Python package for the visual annotation of proteomics data with sequence-specific knowledge

Authors: **Eugenia Voytik**<sup>1,\*</sup>, Isabell Bludau<sup>1,\*</sup>, Sander Willems<sup>1</sup>, Fynn M. Hansen<sup>1</sup>, Andreas-David Brunner<sup>1</sup>, Maximilian T. Strauss<sup>1</sup>, Matthias Mann<sup>1,2,#</sup>

<sup>1</sup> Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, 82152 Martinsried, Germany

<sup>2</sup> NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, 2200 Copenhagen, Denmark

\* These authors contributed equally

Published in *Bioinformatics* (2021).

Although bottom-up MS proteomics routinely reports thousands of proteins, it is conceptually based on the analysis of peptides rather than intact proteins. Therefore, evaluating the sequence coverage of individual proteins by identified peptides with respect to specific protein domains and comparing observed and previously annotated post-translational modifications is an important part of the downstream MS data exploration and is carried out on a regular basis. However, the ability to integrate and visualize experimental data at the peptide and PTM level together with curated protein sequence information across multiple datasets while supporting different state-of-the-art proteomics data analysis software frameworks is still missing in the proteomics field.

To remedy this, we developed AlphaMap, a Python package that facilitates the visual exploration of proteomics data at the peptide level, while additionally integrating protein sequence information based on proteolytic cleavage sites and annotations from the UniProt database. We described the use of AlphaMap in a variety of applications, from optimizing sample processing, through technical comparisons, to validating candidate in a biological or clinical context. Since its publication, it has already become a regularly used tool in our group (Article 7). Furthermore, the easily extendible modular design of AlphaMap has made it possible to further integrate the mapping of peptides and PTMs to three-dimensional protein structures predicted by AlphaFold (123).

My contribution to this paper was to implement the core AlphaMap functions and the graphical user interface, as well as to help write the manuscript.



## Sequence analysis

# AlphaMap: an open-source Python package for the visual annotation of proteomics data with sequence-specific knowledge

Eugenia Voytik<sup>1,†</sup>, Isabell Bludau<sup>1,†</sup>, Sander Willems<sup>1</sup>, Fynn M. Hansen<sup>1</sup>, Andreas-David Brunner<sup>1</sup>, Maximilian T. Strauss<sup>1</sup> and Matthias Mann <sup>1,2,\*</sup>

<sup>1</sup>Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, 82152 Martinsried, Germany and

<sup>2</sup>Department of Clinical Proteomics, NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, 2200 Copenhagen, Denmark

\*To whom correspondence should be addressed.

<sup>†</sup>The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Olga Vitek

Received on July 30, 2021; revised on September 2, 2021; editorial decision on September 13, 2021; accepted on September 22, 2021

## Abstract

**Summary:** Integrating experimental information across proteomic datasets with the wealth of publicly available sequence annotations is a crucial part in many proteomic studies that currently lacks an automated analysis platform. Here, we present AlphaMap, a Python package that facilitates the visual exploration of peptide-level proteomics data. Identified peptides and post-translational modifications in proteomic datasets are mapped to their corresponding protein sequence and visualized together with prior knowledge from UniProt and with expected proteolytic cleavage sites. The functionality of AlphaMap can be accessed via an intuitive graphical user interface or—more flexibly—as a Python package that allows its integration into common analysis workflows for data visualization. AlphaMap produces publication-quality illustrations and can easily be customized to address a given research question.

**Availability and implementation:** AlphaMap is implemented in Python and released under an Apache license. The source code and one-click installers are freely available at <https://github.com/MannLabs/alphamap>.

**Contact:** [mmann@biochem.mpg.de](mailto:mmann@biochem.mpg.de)

**Supplementary information:** [Supplementary data](#) are available at *Bioinformatics* online.

## 1 Introduction

Bottom-up mass spectrometry (MS) has become the leading technology for identifying and quantifying proteomes (Aebersold and Mann, 2003, 2016; Müller *et al.*, 2020). Since peptides rather than intact proteins are measured, visualizing identified peptides and post-translational modifications (PTMs) together with known protein sequence information is an important aspect of downstream MS data exploration. However, the ability to easily integrate and visualize experimental data together with already known sequence annotations is an unmet need in the proteomics community. Although established visualization platforms provide manual visualization of a single experimental sample or dataset at a time (Omasits *et al.*, 2014), there is a lack of tools that support state-of-the-art data analysis software frameworks and that can visualize experimental sequence coverage across multiple samples or datasets in combination with available sequence annotations mined from UniProt, the standard knowledgebase for protein information (Bateman, 2019). To make this wealth of information easily accessible to proteomics

researchers, we developed AlphaMap, a Python package that facilitates the visual exploration of peptide-level proteomics data.

## 2 The AlphaMap computational framework

In line with other recently developed software tools from our lab (Strauss *et al.*, 2021; Willems *et al.*, 2021), we implemented AlphaMap in pure Python because of its clear, easy to understand syntax and the availability of excellent supporting scientific libraries. To read fasta files, we leverage the Pyteomics Python package (Goloborodko *et al.*, 2013; Levitsky *et al.*, 2019). Plotly is a well-established plotting library that we use for generating AlphaMap's sequence visualization (Plotly Technologies Inc., 2015), allowing flexible customization and great user interactivity. To enable easy access to the AlphaMap functionality with a low barrier of entry, a stand-alone graphical user interface (GUI) was implemented using the Panel library (Rudiger *et al.*, 2021). AlphaMap can be launched either as a browser-based GUI after simple local installation or as a



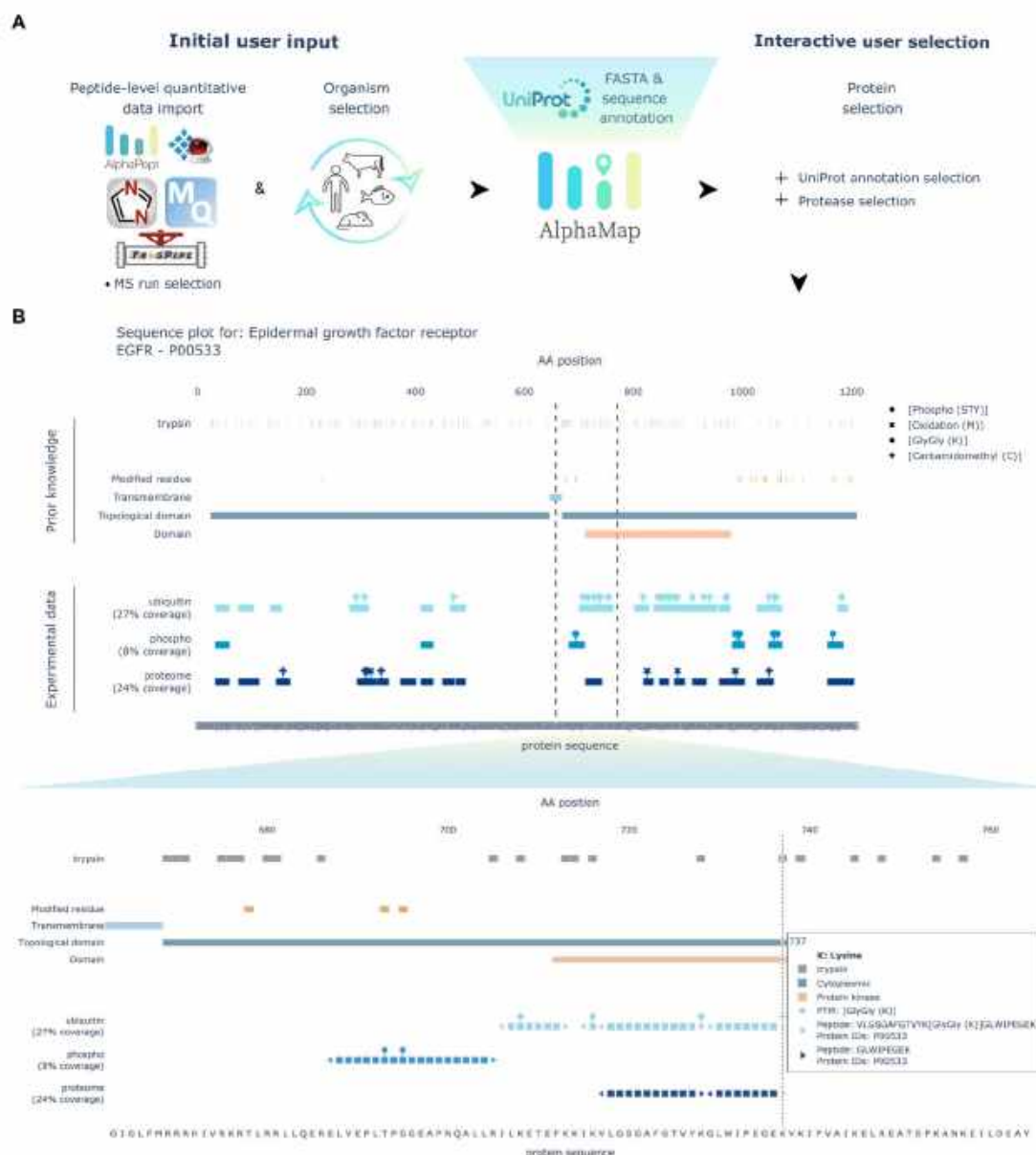
standard Python module installed via PyPI (Python Software Foundation, n.d.) or directly from its GitHub repository.

In line with the AlphaPept ecosystem (Strauss *et al.*, 2021), we make the AlphaMap code openly available on GitHub, using its many supporting features for unit and system testing via GitHub actions. For code development, we adopted the concept of 'iterate programming' (Knuth, 1984), which combines the algorithmic code with readable documentation and testing. Using the nbdev package, the codebase can directly be inspected in well documented Jupyter Notebooks, from which the code is automatically extracted (Kluyver *et al.*, 2016). We envision that these design principles will encourage

the broader community to integrate AlphaMap in their own data analysis and visualization workflows with the possibility to easily adopt the code according to specific needs.

### 3 Overview of the AlphaMap workflow

AlphaMap uses peptide-level proteomics data as input. It currently supports the direct import of data processed by MaxQuant (Cox and Mann, 2008), Spectronaut (Bruderer *et al.*, 2015), DIA-NN (Demichev *et al.*, 2020), FragPipe (Kong *et al.*, 2017) and our



**Fig. 1.** (A) Overview of the AlphaMap workflow from MS data upload to the interactive sequence visualization. (B) Exemplary sequence visualization for epidermal growth factor receptor (EGFR). A zoom-in on a selected sequence region, indicated by dashed lines, is provided at the lower part of the panel

recently introduced AlphaPept framework (Strauss *et al.*, 2021). In contrast to Protter (Omasits *et al.*, 2014), users can select multiple independent datasets for co-visualization. These could either have been processed by the same or with different MS analysis tools. It is also possible to select only a single sample, or a subset of samples of a given input file for individual sequence visualization. In addition to the peptide-level data generated from LC-MS analysis, AlphaMap leverages a plethora of manually curated sequence-specific protein level information available from UniProt. Fasta files and UniProt sequence annotations are readily accessible in AlphaMap for the 13 most popular UniProt organisms as well as for SARS-CoV and SARS-CoV-2. Functionality to enable the integration of additional organisms is further available as part of our Python package. Finally, the user can select the different layers of information that should be displayed in the interactive sequence representation, including selected protease cleavage sites and UniProt sequence annotations. Figure 1A shows a schematic overview of the AlphaMap workflow. Detailed instructions for its installation and usage are further provided in the [supplementary user guide](#). In addition to interactive sequence visualization of a user-selected protein, AlphaMap provides individual links to external databases and tools for further sequence evaluation in UniProt (Bateman, 2019), PhosphoSitePlus (Hornbeck *et al.*, 2015), Protter (Omasits *et al.*, 2014), PDB (Berman *et al.*, 2000) and Peptide Atlas (Desiere *et al.*, 2006).

#### 4 Application of AlphaMap to investigate full proteome and PTM data

Figure 1B shows the sequence visualization of the peptides and PTMs identified for the epidermal growth factor receptor (EGFR) in human A549-ACE2 cells that were infected with SARS-CoV-2 or SARS-CoV (an exemplary viral protein detected in this dataset is visualized in the [Supplementary Material](#)) (Stukalov *et al.*, 2021). We show three independent experimental traces: one for full proteome data, one for phospho-enriched peptides and one for ubiquitin-enriched peptides. The proteome data indicates a homogeneous coverage across the entire protein sequence. As expected, phosphorylation and ubiquitination are limited to the C-terminal region of the protein, which is annotated to be exposed to the cytosol. In addition, the kinase domain of EGFR is highly ubiquitinated in our dataset, whereas the surrounding cytosolic regions are phosphorylated. Interestingly, AlphaMap reports that most of our observed phosphorylation sites have been previously identified, whereas none of the identified ubiquitination sites are annotated in UniProt. Please note that unmodified peptides are also observed in both the phospho- and ubiquitin-enriched samples due to the imperfect selectivity of enrichment protocols.

Beyond the uses highlighted here, we envision AlphaMap to facilitate data analysis and interpretation for a variety of different applications:

- **Candidate validation:** AlphaMap can be used to assess the sequence coverage of identified biomarker candidates (or other proteins of interest) to evaluate possible sequence variations or unexpected anomalies on the basis of readily available sequence information.
- **Preparation of panels for publication:** Sequence visualizations from AlphaMap can directly highlight the precise MS derived information about proteins of interest in biological or clinical projects.
- **Technical comparisons:** AlphaMap can be used to evaluate sequence coverage between different data acquisition strategies such as data-dependent and data-independent acquisition, alternative instrument platforms or software tools.
- **Optimization of sample processing:** Visualization of protein cleavage sites for different proteases can help to optimize sample

processing with the goal to achieve a more complete sequence coverage.

#### 5 Conclusion

AlphaMap offers an interactive GUI and a Python package for visualizing peptide-level bottom-up proteomics data on the basis of individual protein sequences, including information of curated UniProt sequence annotations and expected proteolytic cleavage sites. We expect that future developments by us and the community will extend the variety of available annotations in AlphaMap, for example by including prior knowledge of sequence conservation or predicted functional domains. In addition, we will integrate quantitative information and differential analysis results into the AlphaMap sequence representations. We envision that AlphaMap will assist MS-based proteomics researchers in inspecting peptide- and PTM-level data, thereby providing valuable information in the process of candidate validation in biological and clinical context.

#### Author contributions

L.B. conceptualized the project and together with E.V. and M.M. wrote the manuscript with contributions from all authors. L.B. and E.V. implemented the core AlphaMap functions. E.V. implemented the GUI. S.W. provided important help with the AlphaMap installers. F.M.H. and A.-D.B. provided valuable ideas for the concept and visualization in AlphaMap and F.M.H. further contributed by rigorous testing. M.T.S. designed the general AlphaPept ecosystem and assisted with the nbdev environment. M.M. supervised the study and provided critical feedback on all aspects of the presented work.

#### Funding

This study was supported by The Max-Planck Society for Advancement of Science and by the Bavarian State Ministry of Health and Care through the research project DigiMed Bayern ([www.digimed-bayern.de](http://www.digimed-bayern.de), G64b-A1070-2018/131-2 DMB-1805-0008). L.B. acknowledges funding support from her Postdoc.Mobility fellowship granted by the Swiss National Science Foundation [P400PB\_191046].

*Conflict of Interest:* none declared.

#### Acknowledgements

The authors thank Julia Schessner, Barbara Steigenberger, Jakob Bader and Sophia Madler for testing and providing critical feedback on AlphaMap. They are grateful to Ozge Karayel and Maria C. Tanzer for valuable discussions and for providing experimental data.

#### References

- Aebersold, R. and Mann, M. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198–207.
- Aebersold, R. and Mann, M. (2016) Mass-spectrometric exploration of protein structure and function. *Nature*, **537**, 347–355.
- Bateman, A. UniProt Consortium. (2019) UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, **47**, D506–D515.
- Berman, H.M. *et al.* (2000) The protein data bank. In *Nucleic Acids Res.*, **28**, 235–242.
- Bruderer, R. *et al.* (2015) Extending the limits of quantitative proteome profiling with data-independent acquisition and application to acetaminophen-treated three-dimensional liver microtissues. *Mol. Cell. Proteomics*, **14**, 1400–1410.
- Cox, J. and Mann, M. (2008) MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.*, **26**, 1367–1372.
- Demichev, V. *et al.* (2020) DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. *Nat. Methods*, **17**, 41–44.

- Desiere, F. *et al.* (2006) The PeptideAtlas project. *Nucleic Acids Res.*, **34**, D655–D658.
- Goloborodko, A.A. *et al.* (2013) Pyteomics – a python framework for exploratory data analysis and rapid software prototyping in proteomics. *J. Am. Soc. Mass Spectrometry*, **24**, 301–304.
- Hornbeck, P.V. *et al.* (2015) PhosphoSitePlus, 2014: mutations, PTMs and recalibrations. *Nucleic Acids Res.*, **43**, D512–D520.
- Kluyver, T. *et al.* (2016) Jupyter Notebooks—a publishing format for reproducible computational workflows. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas – Proceedings of the 20th International Conference on Electronic Publishing, ELFUB 2016*, Göttingen, Germany, pp. 87–90.
- Knuth, D.E. (1984) Literate programming. *Comput. J.*, **27**, 97–111.
- Kong, A.T. *et al.* (2017) MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nat. Methods*, **14**, 513–520.
- Levitsky, L.I. *et al.* (2019) Pyteomics 4.0: five years of development of a Python proteomics framework. *J. Proteome Res.*, **18**, 709–714.
- Müller, J.B. *et al.* (2020) The proteome landscape of the kingdoms of life. *Nature*, **582**, 592–596.
- Omasits, U. *et al.* (2014) Protter: interactive protein feature visualization and integration with experimental proteomic data. *Bioinformatics*, **30**, 884–886.
- Plotly Technologies Inc. (2015) *plotly*. Montréal, QC. <https://plot.ly> (27 September 2021, date last accessed).
- Python Software Foundation. (n.d.) Python Package Index – PyPI. <https://pypi.org/> (27 September 2021, date last accessed).
- Rudiger, P. *et al.* (2021) *holoviz/panel: Version 0.11.3*. doi: 10.5281/ZENODO.4692827.
- Strauss, M.T. *et al.* (2021) AlphaPept, a modern and open framework for MS-based proteomics. *BioRxiv*, 2021.07.23.453379, doi:10.1101/2021.07.23.453379.
- Stukalov, A. *et al.* (2021) Multilevel proteomics reveals host perturbations by SARS-CoV-2 and SARS-CoV. *Nature*, **594**, 246–252.
- Willems, S. *et al.* (2021) AlphaTims: indexing trapped ion mobility spectrometry – time of flight data for fast and easy accession and visualization. *Mol. Cell. Proteomics*, 100149.

### **3.4. Article 4: AlphaTims: Indexing Trapped Ion Mobility Spectrometry–TOF Data for Fast and Easy Accession and Visualization**

Authors: Sander Willems<sup>1</sup>, Eugenia Voytik<sup>1</sup>, Patricia Skowronek<sup>1</sup>, Maximilian T. Strauss<sup>1,2</sup>, Matthias Mann<sup>1,3,\*</sup>

<sup>1</sup> Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

<sup>2</sup> OmicEra Diagnostics GmbH, Planegg, Germany

<sup>3</sup> NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark

Published in *Molecular & Cellular Proteomics* (2021).

Over the years, technical advances in MS field have enormously increased the amount and complexity of the data acquired. This is particularly relevant for complex biological samples, e.g. in proteomics, lipidomics or metabolomics, which greatly benefit from separation in more than just the mass dimension. In particular, the recent coupling of MS with trapped ion mobility spectrometry (TIMS) has led to the emergence of a new ion mobility dimension and has demonstrated many advantages, especially for MS-based proteomics samples. However, the fast data accession of complex multidimensional timsTOF data has proven to be a challenge in visualizing and understanding the data.

In this publication we introduced an open-source Python package termed AlphaTims. By indexing sparse four-dimensional timsTOF data of billions of detector events at the scale of seconds, AlphaTims subsequently allows access to data along any available dimension in a sub-second accession time on standard hardware. The fast data access, combined with easy adaptability and extendibility due to detailed documentation and an open code base makes AlphaTims an important software tool that is already widely used in the proteomics community (124).

For this project, I was involved in the formal analysis and the project's investigation, the implementation of the code base and the writing of the manuscript.





# AlphaTims: Indexing Trapped Ion Mobility Spectrometry–TOF Data for Fast and Easy Accession and Visualization

Sander Willems<sup>1</sup>, Eugenia Voytik<sup>1</sup>, Patricia Skowronek<sup>1</sup>, Maximilian T. Strauss<sup>1,2</sup>, and Matthias Mann<sup>1,3,\*</sup>

High-resolution MS-based proteomics generates large amounts of data, even in the standard LC–tandem MS configuration. Adding an ion mobility dimension vastly increases the acquired data volume, challenging both analytical processing pipelines and especially data exploration by scientists. This has necessitated data aggregation, effectively discarding much of the information present in these rich datasets. Taking trapped ion mobility spectrometry (TIMS) on a quadrupole TOF (Q-TOF) platform as an example, we developed an efficient indexing scheme that represents all data points as detector arrival times on scales of minutes (LC), milliseconds (TIMS), and microseconds (TOF). In our open-source AlphaTims package, data are indexed, accessed, and visualized by a combination of tools of the scientific Python ecosystem. We interpret unprocessed data as a sparse four-dimensional matrix and use just-in-time compilation to machine code with Numba, accelerating our computational procedures by several orders of magnitude while keeping to familiar indexing and slicing notations. For samples with more than six billion detector events, a modern laptop can load and index raw data in about a minute. Loading is even faster when AlphaTims has already saved indexed data in an HDF5 file, a portable scientific standard used in extremely large-scale data acquisition. Subsequently, data accession along any dimension and interactive visualization happens in milliseconds. We have found AlphaTims to be a key enabling tool to explore high-dimensional LC–TIMS–Q–TOF data and have made it freely available as an open-source Python package with a stand-alone graphical user interface at <https://github.com/MannLabs/alphatims> or as part of the AlphaPept ‘ecosystem’.

The increasing amounts and complexity of data present a fundamental challenge of data accession in different scientific fields. MS, a leading analytical method in clinical and (bio)

chemical research, is no exception. This issue is compounded when coupling MS with other techniques such as LC and ion mobility spectrometry (1), which allow separating analytes efficiently in scientific domains such as proteomics, lipidomics, and metabolomics (2–4). In our laboratory, this is exemplified by TOF mass analyzers and trapped ion mobility spectrometry (TIMS) (5–7). Typically, analytes are first separated throughout LC gradient times of several minutes or hours. After ionization, they enter a TIMS tunnel where they are trapped and separated in approximately 100 ms. This step discretizes continuous LC separation into ion packets with undistinguishable chromatographic retention time values, and this smallest unit of LC separation is defined as a frame. After TIMS separation, a quadrupole (Q) usually provides selection for tandem MS (MS/MS) before ions reach the TOF accelerator. Ion packets are then sent orthogonally into the TOF analyzer at regular intervals of about 100  $\mu$ s by an electrodynamic pusher. As mentioned previously, such a pusher event discretizes continuous TIMS separation into ion packets with undistinguishable ion mobility ( $1/K_0$ ), and this smallest unit of TIMS separation is defined as a scan. Finally, a detector at the end of the TOF accelerator discretizes continuous ion arrival times into TOF peaks of a few hundred picoseconds wide. This combination of analytical techniques, in brief LC–TIMS–Q–TOF, has received much attention since the introduction of the timsTOF Pro instrument (Bruker Daltonics).

The parallel accumulation–serial fragmentation (PASEF) method synchronizes ion mobility separation with Q selection, combining high-throughput with high sensitivity in both data-dependent acquisition and data-independent acquisition (DIA) (5, 8). Despite its very high data-acquisition rate, the full mass resolution is maintained in the MS or MS/MS mode by coupling the high-resolution TOF mass analyzer to a GHz detector. This rapid detection rate in combination with high sensitivity often leads to billions of detector events per

From the <sup>1</sup>Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany; <sup>2</sup>OmicEra Diagnostics GmbH, Planegg, Germany; <sup>3</sup>Faculty of Health Sciences, NNF Center for Protein Research, University of Copenhagen, Copenhagen, Denmark

\* For correspondence: Matthias Mann, [mmann@biochem.mpg.de](mailto:mmann@biochem.mpg.de).



Mol Cell Proteomics (2021) 20 100149 1

© 2021 THE AUTHORS. Published by Elsevier Inc on behalf of American Society for Biochemistry and Molecular Biology. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.1016/j.mcpro.2021.100149>



## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

sample. While the actual measurements are intensity values of ion species, the exact time of a detector event can be directly converted to the TOF  $m/z$ ,  $Q$   $m/z$ , ion mobility, and chromatographic retention time values.

As a consequence of the resulting large data size, the accession and further visualization of LC-TIMS-Q-TOF data have proven to be challenging and slow in practice. During the last years, the single solution in the field was provided by the manufacturer's closed-source library, integrated into Bruker's proprietary software Compass DataAnalysis. To achieve reasonable data size and access times, this involved preprocessing steps, including data binning. However, this requires choosing parameters such as bin sizes somewhat arbitrarily and, in general, conceals the actual measurements. Consequently, the results depend on this preprocessing, and validation at the level of raw data is impractical.

Very recently, this led to parallel developments tackling some of these issues. The notable examples are OpenTIMS (9), an open-source C++ library with bindings for the Python and R languages to read Bruker data, and MSFragger in combination with IonQuant, which allows to identify and quantify proteins rapidly without the need to preprocess raw data (10). However, these tools were developed using specific applications in mind. We reasoned that fast and generic accession in arbitrary dimensions of the data would need to be optimized for speed, usability, and extensibility. This combination would enable community-driven developments to tackle current bottlenecks such as novel implementations of feature-finding algorithms, retrieval of extracted ion chromatograms (XICs) for DIA analysis, or fast interactive data visualization of raw MS data.

Here, we present AlphaTims, a user-friendly software tool, that drastically accelerates accession and visualization of raw LC-TIMS-Q-TOF data compared with the vendor's software. It provides an indexing procedure in such a way that the unprocessed data are interpreted as a sparse four-dimensional matrix. This matrix is specifically designed for LC-TIMS-Q-TOF data, allowing fast retrieval of arbitrary data slices along all of the available dimensions in milliseconds. It is implemented in pure Python with only a few dependencies to make it readable, flexible, and lightweight. This makes it easily adoptable and adaptable by the community. At the same time, it matches the performance of programs written in the C programming language, by using the popular packages NumPy for array manipulation and Numba for just-in-time (JIT) compilation to machine code (11, 12). AlphaTims can save an indexed dataset as a single portable high-performance hierarchical data format (HDF5) file (13), which has proven its efficiency and extensibility in various scientific fields and has also been used in MS-based proteomics before (14–16). This further accelerates data access and allows us to store arbitrary metadata and downstream processing results. We then use Datashader, an optimized rendering Python package to plot millions of data points on standard hardware (17), in

combination with Panel and Bokeh (Python packages to build user-friendly dashboards to access and visualize data) to extend the usability of AlphaTims to a broader audience regardless of computational expertise. AlphaTims is a modular tool that is also a part of the AlphaPept (18) (<https://github.com/MannLabs/alphapept>) 'ecosystem' developed in our department, which provides tools for the different facets of MS-based computational proteomics. It can be used as a fully stand-alone graphical user interface (GUI), command-line interface (CLI), or Python module for Windows, macOS, and Linux and is freely available under an Apache license at <https://github.com/MannLabs/alphatims>.

## EXPERIMENTAL PROCEDURES

## Sample Preparation

Human cervical cancer cells (HeLa, S3, and ATCC) were cultured in Dulbecco's modified Eagle's medium with 10% fetal bovine serum, 20 mM glutamine, and 1% penicillin-streptomycin (all Life Technologies Ltd). The cells were collected using centrifugation, washed with PBS, flash-frozen in liquid nitrogen, and stored at  $-80^{\circ}\text{C}$ .

Following the in-StageTip protocol (19), cell lysis, reduction, and alkylation with chloroacetamide were carried out simultaneously in a lysis buffer (PreOmics). The resultant dried peptides were reconstituted in double-distilled water comprising 2 vol% acetonitrile and 0.1 vol% TFA to a concentration of 200 ng/ $\mu\text{l}$  and further diluted with double-distilled water containing 0.1 vol% formic acid. The manufacturer's instructions were followed to load approximately 50 ng or 200 ng peptides onto Evotips (EvoSep).

## LC

Purified tryptic digests were separated with either a predefined '200 samples per day' (SPD) method (6-min gradient time, 50 ng peptides) or a predefined 60 SPD method (21-min gradient time, 200 ng peptides) on an EvoSep One LC system (EvoSep) (20). A fused silica 10- $\mu\text{m}$  ID emitter (Bruker Daltonics) was placed inside a nanoelectrospray source (CaptiveSpray source, Bruker Daltonics). For the 200 SPD method, the emitter was connected to a 4-cm  $\times$  150- $\mu\text{m}$  reverse-phase column, packed with 3- $\mu\text{m}$   $\text{C}_{18}$  beads, and for the 60 SPD method, to an 8-cm  $\times$  150- $\mu\text{m}$  reverse-phase column, packed with 1.5- $\mu\text{m}$   $\text{C}_{18}$  beads (PepSep). Mobile phases were water and acetonitrile, buffered with 0.1% formic acid.

In addition, 400-ng peptides were separated over a 120-min gradient time on a 50-cm in-house reverse-phase column with an inner diameter of 75  $\mu\text{m}$ , packed with 1.9- $\mu\text{m}$   $\text{C}_{18}$  beads (Dr Maisch ReproSil-Pur AQ) and a laser-pulled electrospray emitter. The column was heated to  $60^{\circ}\text{C}$  in an oven compartment. The binary LC system consisted water as buffer A and acetonitrile/water (80%/20%, v/v) as buffer B, both buffers containing 0.1% formic acid (Easy-nLC 1200, Thermo Scientific). The gradients started with a buffer B concentration of 3%. In 95 min, the buffer B concentration was increased to 30%, in 5 min to 60%, and in 5 min to 95%. A buffer B concentration of 95% was held for 5 min before decreasing to 5% in 5 min and re-equilibrating for further 5 min. All steps of the gradients were performed at a flow rate of 300  $\text{nl min}^{-1}$ .

## MS

LC was coupled online to a TIMS Q-TOF instrument (timsTOF Pro, Bruker Daltonics) with ddaPASEF and diaPASEF (7, 8) via a CaptiveSpray nano-electrospray ion source. For both acquisition modes, the



## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

ion mobility dimension was calibrated with three Agilent ESI-L Tuning Mix ions ( $m/z$ ,  $1/K_0$ : 622.0289 Th, 0.9848 Vs  $\text{cm}^{-2}$ ; 922.0097 Th, 1.1895 Vs  $\text{cm}^{-2}$ ; 1221.9906 Th, 1.3820 Vs  $\text{cm}^{-2}$ ). Furthermore, the collision energy was decreased linearly from 59 eV at  $1/K_0 = 1.6$  Vs  $\text{cm}^{-2}$  to 20 eV at  $1/K_0 = 0.6$  Vs  $\text{cm}^{-2}$ .

For the ddaPASEF method, each topN acquisition cycle consisted four PASEF MS/MS frames for the 200 SPD and 60 SPD methods and ten PASEF MS/MS frames for the 120-min gradient time. The accumulation and ramp times were set to 100 ms. Singly charged precursors were excluded from fragmentation using a polygon filter in the ( $m/z$ ,  $1/K_0$ ) plane. Furthermore, all precursors that reached the target value of 20,000 were excluded for 0.4 min. Precursors were isolated using a Q window of 2 Th for  $m/z < 700$  and 3 Th for  $m/z > 700$ . For diaPASEF, we used the 'high-speed' method ( $m/z$  range: 400–1000 Th,  $1/K_0$  range: 0.6–1.6 Vs  $\text{cm}^{-2}$ , diaPASEF windows:  $8 \times 25$  Th), as described (8).

A seventh sample was acquired with identical settings as the 60 SPD ddaPASEF method. To intentionally introduce anomalies, the TOF was calibrated with an offset of 1 Da, and the air supply through the CaptiveSpray nano-electrospray source filter was blocked between minute 12 and 13.

## AlphaTims Development

The AlphaTims source code is freely available on GitHub (<https://github.com/MannLabs/alphatims>) under an Apache license. The Python code (alphatims folder) is divided into two core modules: `bruker.py` provides the TimsTOF class and all functions to create, index, and access objects from this class, whereas the `utils.py` module provides generic utilities for logging, compilation, parallelization, and I/O. Three additional modules implement all functionality for plotting, GUI, and the CLI.

In addition to the core Python code, the GitHub repository includes much introductory and background information. This includes (1) an extensive README for navigation, installation, and usage instructions, (2) a Jupyter Notebook folder (nbs) with a Python tutorial and a performance notebook to reproduce all timings as presented in this article, (3) a documentation folder (docs) to create all documentation for the Bruker, utils, and plotting modules hosted on <https://alphatims.readthedocs.io>, (4) a miscellaneous folder (misc) facilitating manual creation of new GUI releases and Python Package Index (PyPI) releases on <https://pypi.org/project/alphatims>, (5) a .github folder to perform continuous integration including testing and automatic releasing of new versions, and (6) a requirement folder to handle all dependencies.

AlphaTims is developed in pure Python and only has seven core dependencies: (1) h5py to handle HDF5 files, (2) Numba for JIT compilation, (3) Pandas for tabular results, (4) pyzstd for generic decompression of Bruker binary data, and (5–7) tqdm, psutil, and click for CLI support. All plotting capabilities and the GUI are enabled by four additional packages: (1) Bokeh for visualizations and the dashboard, (2) hvPlot to connect Pandas DataFrames with Bokeh, (3) Datashader for fast rendering of visualizations, and (4) selenium for browser support. As an alternative to  $m/z$  and  $1/K_0$  estimation, we also provide the option to retrieve calibrated values with Bruker libraries on Windows and Linux machines. Additional requirement files exist purely for legacy code and to facilitate development with dependencies such as, for example, PyInstaller to create the stand-alone GUI or twine to release new versions on PyPI.

## Computational System

All development and testing of AlphaTims was done on a MacBook Pro (13-inch, 2020) with a 2.3 GHz Quad-Core Intel Core i7 processor, 32 GB 3733 MHz LPDDR4X memory, and 2 TB Flash storage running macOS Catalina version 10.15.7. Functionality on Linux and Windows

was tested through continuous integration on default GitHub virtual machines running Ubuntu 20.04 and Windows Server 2019 (<https://docs.github.com/en/actions/using-github-hosted-runners/about-github-hosted-runners>).

## RESULTS AND DISCUSSION

To better explain the indexing procedure at the heart of AlphaTims, we shortly summarize the data structures used in the vendor's software in their TIMS data format (tdf). A '.d folder' contains two primary files to store raw LC-TIMS-Q-TOF data acquired with the timsTOF Pro (Bruker Daltonics) (Fig. 1A). The first of these is the analysis.tdf file, an ordinary SQLite database, that contains all metadata from the acquisition. It furthermore stores summarized information for each individual frame (ion packet with the same retention time values) and, if applicable, at which scans (ion packet with the same ion mobility values) the Q isolation window was changed. The second file, analysis.tdf\_bin, contains all raw detector events and their intensity values as compressed binary data.

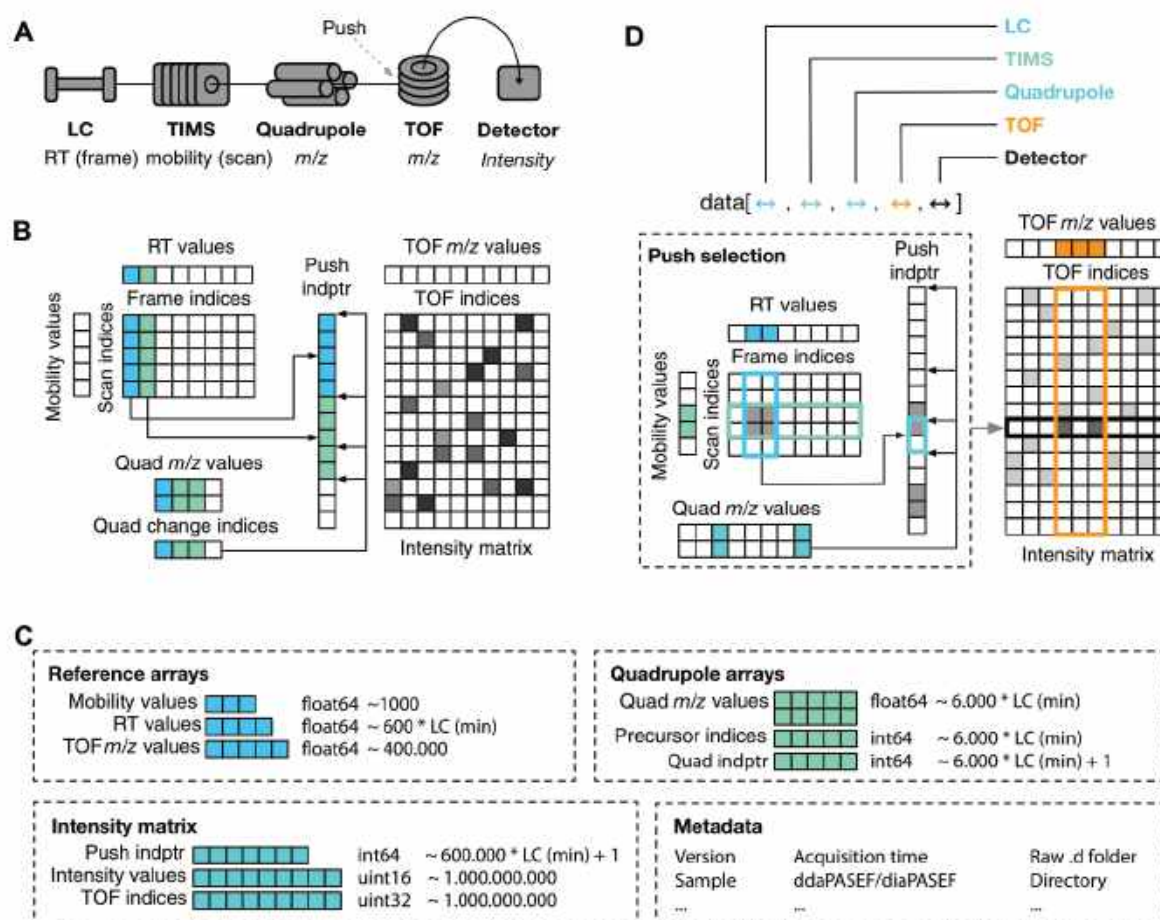
## Indexing Procedure and Performance

AlphaTims represents relevant data from a '.d folder' in multiple NumPy arrays. First, it decompresses the binary analysis.tdf\_bin file to read all detector events and corresponding intensity values. While Bruker stores detector events and intensity values in a single homogeneous array, AlphaTims separates them into three distinct arrays. In the first, the (nonzero) intensity values of all detector events are stored in order of their acquisition time. A second array of equal length then stores their TOF indices as offsets for each individual pusher event. To indicate when pusher events happened, AlphaTims defines a third dense array that stores the number of detector events that are registered per pusher event. By taking the cumulative sum of this latter array, pointers are created to indicate the start and end indices of individual pusher events in the two former arrays. Together, these three arrays unambiguously define a compressed sparse row matrix (21) with indices of pusher events as rows, TOF indices as columns, and intensity values as values (Fig. 1B).

Next, AlphaTims retrieves the unique number of frames, scans, and TOF indices from the analysis.tdf SQL database, and from an array containing all retention time values. On Windows and Linux, arrays with ion mobility and TOF  $m/z$  values are retrieved from Bruker libraries that are integrated into AlphaTims. These Bruker libraries are unavailable on macOS; however, as a work-around, we provide an estimation of these values based on the start values and end values as provided in the analysis.tdf SQL database. As there are typically 600 frames per minute, 1000 scans per frame, and 400,000 detector events per pusher event, the size of these three arrays is neglectable compared with the total number of detector events that frequently surpasses a billion.



## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data



**FIG. 1. Schematic of AlphaTims' indexing and data accession.** A, data dimensions: the timsTOF instrument acquires detector events after separation and selection in four different dimensions. After passing through the LC, TIMS, and quadrupole, an ion beam enters the TOF accelerator where a pusher event (synchronized with the LC, TIMS, and quadrupole) sends ions in an orthogonal direction toward the detector. LC, trapped ion mobility spectrometry (TIMS), and TOF coordinates can be represented as discrete indices (frame, scan, and TOF indices) or as continuous values (retention time [RT], ion mobility, and TOF  $m/z$  values). B, indexing procedure: AlphaTims uses several arrays to store LC-TIMS-Q-TOF data. First, the intensity values are stored in a compressed sparse row matrix (intensity matrix) with TOF indices as columns and indices of pusher events as rows (push index pointers/indptr). Each unique pusher event corresponds to a unique combination of a frame and scan index, according to the formula  $push_i = scan_n + frame_m \cdot \#scans$ . Note that the scan-frame matrix presented here is purely a visual aid and is not stored explicitly, as the unique relationship between frame, scan, and push indices makes this redundant. An additional sparse array stores the push indices where the quadrupole settings are changed (quad change indices). For instance, in the first frame (blue), the quadrupole is not changed, whereas it is changed once the second frame (green) starts and another time within this frame (e.g., diaPASEF with two windows per frame). An array of equal length denotes which  $m/z$  values (lower and upper bounds) are selected with the quadrupole at each of these indices. C, array storage: owing to the indexing, AlphaTims only needs to store a few arrays of variable size (each square represents an order of magnitude). The reference arrays containing mobility, retention time, and TOF  $m/z$  values take between a thousand and one million elements. While the quadrupole arrays are mostly dependent on the LC gradient length (in minutes), these arrays are generally also less than one million elements. The largest arrays are those that represent the sparse intensity matrix: push indptr, intensity values, and TOF indices, with the latter two arrays frequently containing billions of elements. Finally, a few bytes are used to store relevant metadata. D, accession procedure: data accession with AlphaTims can be performed in any dimension. This can be done by providing ranges of interest either as indices or as values. In case of the latter, LC, TIMS, and TOF values are always converted to the closest index by fast binary searches in their corresponding arrays. All of the selected LC and TIMS indices are then converted to push indices by the formula  $push_i = scan_n + frame_m \cdot \#scans$ . Because the quadrupole  $m/z$  array is not ordered, a linear pass over all quadrupole  $m/z$  values is required to determine which quadrupole index pointers are valid, and only those that overlap with the previously selected push indices are retained. For each individually selected push index, a binary search retrieves all TOF indices that satisfy the requested TOF range. Finally, all selected detector events are filtered with a single pass over their corresponding intensity values to obtain the final set of detector events that satisfies the multidimensional range of interest.



## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

Finally, another sparse array is created to indicate at which push indices the Q settings change. In ddaPASEF, this happens on average ten times per frame to select different precursors. In diaPASEF, this depends on the acquisition scheme and desired cycle time. Typically, each frame of a recurring diaPASEF acquisition cycle is split up into eight window groups that all have different Q settings. This array of Q change indices is accompanied by two other arrays of equal length. The first of these is two-dimensional and defines the lower and upper Q  $m/z$  values selected by the Q. The second defines the precursor index. For DIA, the precursor indices are equal to the diaPASEF window group.

AlphaTims collects all these arrays, together with global and frame-specific metadata from the analysis.tdf file, and stores this as an `alphatims.bruker.TimsTOF` object into working memory (Fig. 1C). Because a single detector event takes up 6 bytes (an `UInt32` for the TOF index and an `UInt16` for the intensity) and their respective arrays generally dwarf all others, the required working memory (in gigabytes) is roughly equal to six times the number of detector events (in billions). The `alphatims.bruker.TimsTOF` object acts as a fully indexed sparse four-dimensional matrix with associated metadata.

To facilitate fast reuse of this object and avoid recreation of the indices, it can be stored on disk as a portable HDF5 file with Python's `h5py` package. This is possible on all operating systems, but TOF  $m/z$  and ion mobility values of HDF5 files created on macOS can differ from Windows and Linux owing to the availability of the Bruker libraries, as mentioned above. By default, the HDF5 file size is equal to the required working memory, but compression can be used to decrease this roughly two-fold. While compression slows down loading and saving of HDF5 files approximately from 2 to 10 times, an AlphaTims object in working memory is always decompressed and interactive accession is thus unaffected. These (de)compressed HDF5 files can always be (de)compressed and resaved, making them ideal for file transfer or archiving. A major benefit of such file transfer is that HDF5 files created on Windows or Linux can be transferred to macOS, thereby utilizing the  $m/z$  and ion mobility values from the Bruker libraries on all operating systems instead of requiring the aforementioned estimation. Note that not all HDF5 formats are interchangeable with the HDF5 format of AlphaTims. This is primarily because these formats were developed in the past as more general community standards for arbitrary MS data and therefore explicitly store (meta)data per individual spectrum. In contrast, AlphaTims HDF5 files are very efficient as we can assume they contain homogenous LC-ion mobility spectrometry-Q-TOF data that are stored in only a few arrays with a single set of indices and metadata.

To assess the performance of AlphaTims' indexing procedure, we acquired HeLa samples with gradients of 6, 21, and 120 min in both ddaPASEF and diaPASEF modes (Experimental Procedures). At the shortest time dimension, a single pusher event could record almost 400,000 TOF

detection events in an  $m/z$  range of 100 to 1700 Th. Separation in the TIMS tunnel lasted 100 ms and is composed of 1000 of these pusher events, covering a  $1/K_0$  range of 0.6 to 1.6 Vs  $\text{cm}^{-2}$ . Up to 240 billion events could thus have been recorded per minute; however, in practice, no run acquired more than 0.03% of these potential detector events, and the data can be considered sparse (Fig. 2).

On a laptop (Experimental Procedures), reading all detector events into working memory and indexing them took AlphaTims less than a second for the smallest run and less than 90 s even for the largest run with 6.4 billion detector events. In contrast, opening any of these runs with Bruker's Compass DataAnalysis software (v5.3) required at least double the time on a Windows desktop with overall better specifications. To speed up data import even further and allow modification or addition of downstream results, AlphaTims also allows exporting the indexed data as a portable HDF5 file, which only takes seconds. When these HDF5 files are imported, no decompression and indexing is required, making them roughly three times faster to load than raw Bruker '.d folders'. While reading '.d folders' with AlphaTims benefits from multiple CPUs to speed up decompression, loading from HDF5 files is only limited by disk reading speed. Regardless, the required time to load or save either a '.d folder' or HDF5 file is approximately linear in function of the number of detector events and independent of LC gradient or acquisition scheme.

Currently, reading and indexing data is done after acquisition. Given that these steps take only a fraction of the time it takes to acquire the data, we hypothesize that it would also be possible to index data that are being acquired in real time. This would only require to know the TOF and TIMS dimensions upfront, which are parameters that indeed are determined before acquisition. All other arrays are sorted in function of time and can thus easily be created in real time with dynamic buffer arrays. Such live indexing would not require storage of unindexed data and avoids wasting acquisition time on samples with poor quality.

## Accession Procedure and Performance

Once data are imported and indexed, an `alphatims.bruker.TimsTOF` object can be accessed in all dimensions with traditional Python slices or 'fancy index slicing' from NumPy (12) (Fig. 1D). The order of the dimensions in such an object is equal to the order of their respective components in the `timsTOF Pro`: LC, TIMS, Q, TOF, and detector. Typically, the user defines a range of interest that is translated into a slice with a single index or by a (start and stop) tuple. When decimal values are provided for the LC, TIMS, or TOF dimension instead of indices, AlphaTims always assumes them to represent retention time, ion mobility, or TOF  $m/z$  values. By default, these are converted to the closest integers representing frame, scan, or TOF indices by looking them up in their appropriate arrays with a fast binary search. In the case of Q



## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

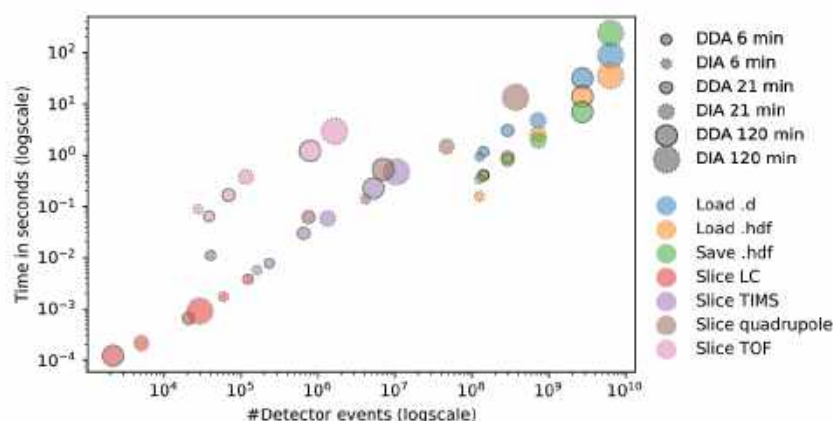


FIG. 2. **Time performance of AlphaTims.** Different HeLa samples were acquired in both ddaPASEF (full outline) and diaPASEF (dotted outline) with gradient lengths of 6, 21, and 120 min (Experimental Procedures). When a raw Bruker '.d' folder is read, AlphaTims needs to decompress, import, and index all detector events (blue). Once this is performed, the indexed dataset can be saved as an HDF5 file (green). When an HDF5 file is read instead of a raw Bruker '.d' folder, no decompression or indexing is required (orange). Multiple detector events of each run were retrieved by slicing each dimension individually. The retrieved detector events correspond to an LC slice with  $100 \leq \text{retention time (s)} < 100.5$  (red), a TIMS slice with  $\text{scan index} = 450$  (purple), a quadrupole slice with  $700.0 \leq \text{quad } m/z \text{ value} < 710.0$  (brown), and a TOF slice with  $621.9 \leq \text{TOF } m/z \text{ value} < 622.1$  (pink). All timings were obtained with Python `timeit` function for robust and reproducible results that were averaged over at least seven repeats. See <https://github.com/MannLabs/alphatims/blob/master/nbs/performance.ipynb> for exact numbers. TIMS, trapped ion mobility spectrometry.

$m/z$  values, precursor indices, or intensities, no translation is necessary.

Once a multidimensional slice of interest is defined, AlphaTims first selects all the possible push indices that satisfy the LC and TIMS dimensions and converts these to push indices with the formula  $\text{push}_i = \text{scan}_i + \text{frame}_i \cdot \text{\#scans}$ . As these push indices are ordered, they are located in the Q change index array in a single iteration. Only those push indices with a valid Q  $m/z$  value are selected, and for each of them, appropriate TOF indices are retrieved from the sparse intensity matrix. As the TOF indices are ordered per individual pusher event, a binary search quickly retrieves all TOF indices that satisfy the requested TOF slice. Finally, it is checked which of all the selected detector events have an intensity value that satisfies the detector slice. The results are then returned as a Pandas (<http://pandas.sf.net>) DataFrame whose columns describe all indices and values, or—if desired—as a NumPy array with indices of detector events.

For each of the six HeLa samples (Experimental Procedures), we tested four different slices: an LC slice with retention time values between 100 and 100.5 s, a TIMS slice with a scan index of 450 providing all mass spectra at the corresponding ion mobility, a Q slice with only fragments from a precursor range between 700 and 710 Th, and finally, a TOF slice with  $m/z$  values between 621.9 and 622.1 (Fig. 2). As expected, samples with longer gradients, and thus more detector events, also yield more detector events when sliced in the TIMS and TOF dimensions. While this is also true for the Q

dimension, the effect of being a ddaPASEF or diaPASEF method is stronger than the gradient length in these examples. This is not surprising because the Q selected just 2 or 3 Th in ddaPASEF, whereas the selected windows in diaPASEF were always 25 Th.

Next, we evaluated the time that was needed to access all of the previous data slices with AlphaTims. Owing to the indexing structure, the index of any pusher event can be converted to a frame and scan index with a simple linear formula and vice versa (Fig. 1D). As such, it can be expected that accession in these dimensions should be very fast as no actual searching is involved. Indeed, even retrieving five million detector events with slicing in the LC or TIMS dimension is carried out in just 0.2 s (Fig. 2). Moreover, the time required to slice in these dimensions only depends on the number of detector events that are retrieved and only indirectly on the gradient length or acquisition scheme. Slicing in the Q dimension is very similar. While slightly slower than the LC or TIMS dimensions, there is a comparable linear dependency for the required slicing time that is purely a function of the number of detector events that are retrieved. This slowdown is due to additional filtering of Q change indices from the sparse array. As this Q index pointer array itself is very sparse (on average, 1% nonzero elements when compared with the number of pusher events), the impact of this additional filtering is small. However, slicing in the TOF dimension is roughly an order of magnitude slower than slicing in any other dimension, primarily caused by the fact that every pusher event needs to be filtered individually, as the TOF

## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

dimension is indexed per pusher event. When TOF slicing is combined with other dimensions, fewer selected pusher events are selected, which makes even this slowest step instantaneous to the user. As the time required for TOF slicing is still linearly dependent only on the number of retrieved detector events, AlphaTims is very scalable even to long gradients, very complex samples, and data acquisition schemes.

## Using AlphaTims

AlphaTims is freely available as an open-source Python package with an Apache license on Windows, macOS, and Linux. To enable the usage for a wide audience regardless of computational background, it can be operated in any of the three following modes: a stand-alone GUI, a stand-alone CLI, or directly as a Python module.

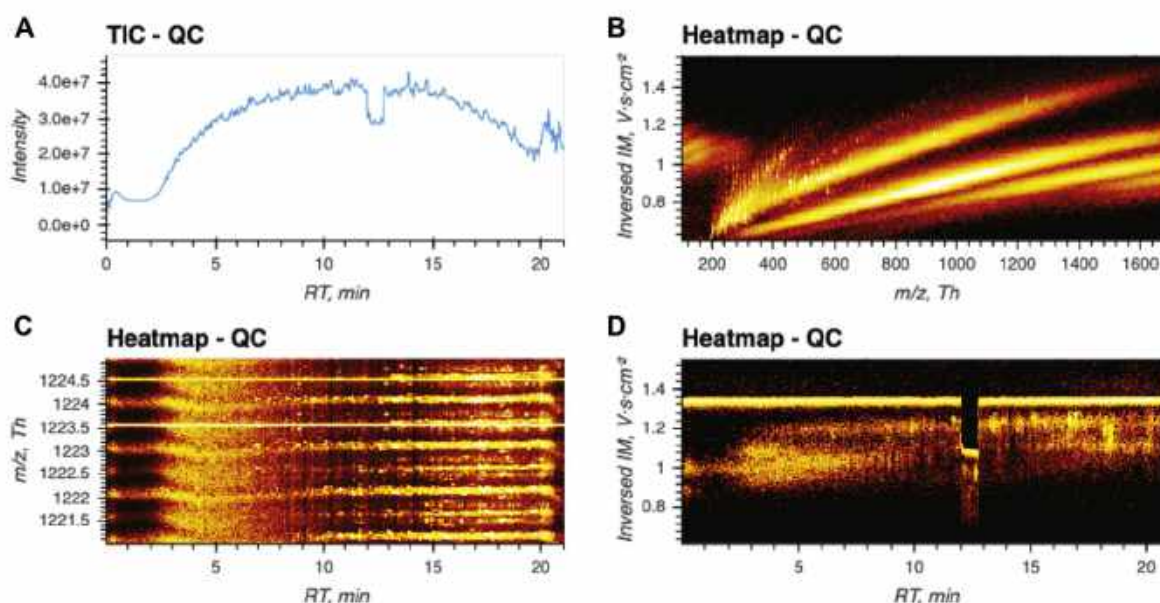
## GUI Mode

A simple installer for the AlphaTims GUI can be downloaded from our GitHub page, requiring just a few mouse clicks. Both the installation and usage of AlphaTims have been made as intuitive as possible, but a comprehensive GUI manual is also available with in-depth step-by-step explanations and screenshots.

The GUI allows interactive exploration of unprocessed LC-TIMS-Q-TOF data conveniently in browsers such as Google

Chrome or Mozilla Firefox. It was programmed in pure Python and uses only a few libraries of Python's Holoviz visualization ecosystem. These include Holoviews itself and Bokeh to visualize different plots such as the total ion current (TIC), Datashader for fast rendering of these plots, and Panel to combine the plots with control widgets into an interactive dashboard ([Experimental Procedures](#)). With the control widgets, the user can slice the data simultaneously in multiple dimensions as described previously ([Accession Procedure and Performance](#)). The selected coordinates can then be projected on either a single axis to show mass spectra, ion mobilograms, or XICs or on multiple axes to create heatmaps in the LC, TIMS, and TOF dimensions.

Having reduced the visualization of LC-TIMS-Q-TOF to a fast and straightforward task, it can be incorporated in a wide variety of practical applications. In the following text, we demonstrate this on the example of visual quality control. For this purpose, we intentionally acquired a sample with a few anomalies (including a large offset of the mass scale and temporary pressure change in the CaptiveSpray source) to see if we could indeed quickly detect any issues. There were 0.7 billion detector events in this 21-min ddaPASEF run. The data could be imported with a single mouse click, and the TIC was visible within 10 s of opening the AlphaTims GUI. This immediately revealed an anomaly, namely the drop in ion



**FIG. 3. Quality control (QC) with the AlphaTims graphical user interface.** A, total ion current: after importing a sample, the total ion current (TIC) is immediately available without requiring any additional user input. In this case, a clear drop in intensity between minute 12 and 13 is visible. B, relation between ion mobility and  $m/z$  values: by selecting the first 100 frames, the expected relation between  $m/z$  and ion mobility values of different charge states becomes clear. C, TOF calibration: by resetting the frames and adjusting the TOF selection and plot axis widgets, the expected  $m/z$  value of a calibrant spray is visualized throughout the whole gradient. The expected value of 1222.0 Th is not present, but, instead, a value of 1223.5 Th is displayed. D, ion mobility spectrometry stability: when the TOF selection is narrowed to  $1223.5 \pm 0.1$  Th and the y-axis is changed to  $1/K_0$  values, a discontinuity in ion mobility is detected between minute 12 and 13.



### AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

current between minute 12 and 13 that we had engineered beforehand (Fig. 3A). Without having done any processing at all, the user is forewarned about unreliable intensity values in that region. We then used the frame widget to select the first 100 frames and projected intensity values on the TOF and TIMS dimensions, showing the expected relation for  $m/z$  and ion mobility values of differently charged precursors (Fig. 3B). As an important quality metric, the user can assess the stability of added calibrant ions ( $1222.0\text{ Th}$ ,  $1.38\text{ Vs cm}^{-2}$ ), which is expected to be continuously present throughout the whole run. By resetting the selected frames to the whole range and modifying just two values of the TOF widget, we selected all ions in the  $m/z$  region between  $1221.0$  and  $1225.0\text{ Th}$ . By adjusting the heatmap axes to show chromatographic retention time values on the x-axis and  $m/z$  values on the y-axis, we expect to see a continuous signal throughout the whole gradient for the calibrant spray with an  $m/z$  value of  $1222.0\text{ Th}$ . However, there is a continuous and steady signal for an  $m/z$  value of  $1223.5\text{ Th}$  instead, accompanied by a less-intense isotope at  $1224.5\text{ Th}$  (Fig. 3C). Based on these observations, we deduce that the TOF  $m/z$  values are greatly miscalibrated (as intended for this sample) and that the reported  $m/z$  values are too unreliable for further analysis. Next, we changed the y-axis of the heatmap to show the ion mobility values and inspect the detected ion at  $1223.5 \pm 0.1\text{ Th}$  during the complete LC gradient. This clearly revealed another issue between minute 12 and 13. Normally, the ion mobility value of the calibrant spray should remain constant at a value of  $1.38\text{ Vs cm}^{-2}$ , but in this case, the apparent value drops to  $1.1\text{ Vs cm}^{-2}$  for a full minute (as a result of the purposely altered gas flow) (Fig. 3D). This coincides with the previously detected drop in the TIC, meaning that not only the intensity but also the other coordinates are unreliable in this timeframe. Thus, a brief assessment of the data in less than 30 s with just a few user inputs already detected and pinpointed the main issues with data quality. Other quality assessments to analyze, for example, fragmentation efficiency of ddaPASEF samples or positioning of Q selections in diaPASEF samples do not require much more effort and quickly become routine even for inexperienced users.

#### CLI Mode

Although it is very easy to use, AlphaTims' GUI requires manual input for visualization. For users who wish to automate repetitive tasks, the AlphaTims CLI provides the same functionality as the GUI. Instead of manually updating control widgets, all settings and values can be provided to the command-line either directly or with a simple script. As there is no need to display an interactive dashboard, this mode is even faster and more versatile than the GUI. More complex data slices can be selected than with the GUI, while all results can still be exported. This includes visualizations in png, or html format, csv tables with selected ion coordinates, and

alternative formats of the whole sample such as portable HDF5 files and mascot generic format files. All of these commands and their options are fully documented in the CLI, and a brief tutorial is available on GitHub.

#### Python Mode

Although the CLI is more flexible than the GUI, it is impossible for us to implement all the imaginable use cases of AlphaTims. Instead, we also make it available as a Python module and leave it to the end user to implement any additional functionality or incorporate it into other Python projects. AlphaTims can be installed from PyPi as a Python module with the standard pip module of Python 3.8. There is both a lightweight version available with just a few dependencies that purely focuses on data indexing and accession and an extended version with more dependencies that includes the complete visualization library as used for the GUI and CLI.

Enabling AlphaTims in other Python scripts or Jupyter notebooks requires a single line of code that imports the module. Some convenience functions enable logging or set the number of available threads for multithreading and ensure transparent, reproducible, and efficient usage of AlphaTims. All functions of AlphaTims are implemented in pure Python and fully documented to facilitate flexibility, readability, and usability. However, functions that are computationally intensive have been decorated with Numba to use JIT compilation to machine code. This enables a performance similar to the fastest low-level languages such as C.

Importing and indexing data is carried out with a single command that returns an `alphatims.brucker.TimsTOF` object, which can be treated as a four-dimensional matrix. Inspired by the slicing approach in NumPy, one of the fundamental Python libraries for scientific computing, AlphaTims provides slicing in multiple dimensions simultaneously as described previously (Accession Procedure and Performance). As a result, AlphaTims data slices can take advantage of the vast amount of Python packages that act on Pandas DataFrames as well.

To demonstrate the basic usage of AlphaTims in Python, we have provided a brief Jupyter Notebook tutorial on GitHub (<https://github.com/MannLabs/alphatims/blob/master/nbs/tutorial.ipynb>). This notebook explains how to set up AlphaTims and enable logging for transparent and reproducible data analysis, import samples and export indexed HDF5 files for faster reanalysis, select individual data points and data slices, and visualize data to create similar plots as with the GUI or CLI. The final part of the tutorial includes an example to show how AlphaTims can be used to investigate a specific peptide in diaPASEF data based on a spectral library created with, for instance, AlphaPept, Skyline, or Spectronaut (18, 22, 23).

The above example illustrates a use case of AlphaTims in Jupyter Notebooks that have become a standard in modern data science (Fig. 4). AlphaTims and Bruker diaPASEF data



## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

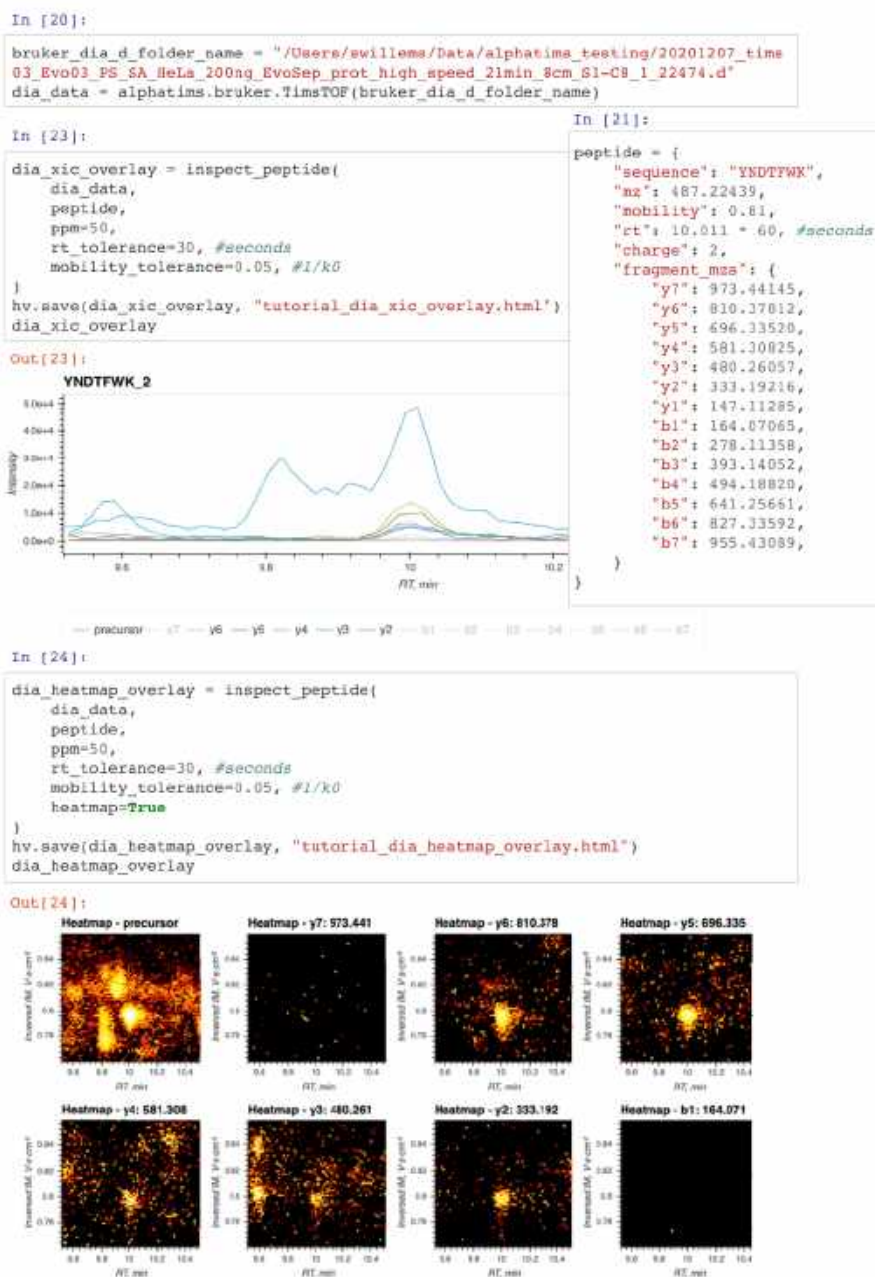


FIG. 4. A section of a Jupyter Notebook using AlphaTims as a Python module. Jupyter Notebooks allow to structure and execute Python code in individual cells. In the last part of the AlphaTims tutorial, data from a diaPASEF sample is imported (cell "in [20]"). The same sample was also acquired in ddaPASEF, and a spectral library was generated with AlphaPept. Relevant coordinates of the peptide YNDTFWK were retrieved from this spectral library and defined in the tutorial (cell "in [21]"). A function 'inspect\_peptide' was defined (cell "in [22]", see AlphaTims' Python tutorial at <https://github.com/MannLabs/alphatims/blob/master/nbs/tutorial.ipynb>), allowing to visualize extracted ion chromatograms (XICs) for the doubly charged precursor and all fragments of this peptide (cells "in [23]" and "out [23]"). Based on these XICs, some interference seems to be present for the precursor signal of this peptide. However, when the precursor and fragments of this peptide are visualized as a heatmap in both the LC and TIMS dimensions, it becomes clear that this interference is fully resolved in the TIMS dimension (cell "in [24]" and "out [24]"). TIMS, trapped ion mobility spectrometry.

## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

are first imported, and then, all coordinates of both the precursor and all fragments of a specific peptide are defined. With a simple custom Python function, all detector events that match these coordinates within a certain tolerance can be retrieved and visualized in an interactive plot. Traditionally, such an interactive plot represents only the XICs of the selected precursor and its fragments, but this ignores the TIMS dimension. In contrast, with AlphaTims in this Jupyter Notebook, we can easily provide heatmaps in both the LC and TIMS dimensions for the precursor and all fragments, thereby illustrating the benefit of using TIMS data for peak capacity and interference removal. Using this extra information allows us to manually verify that the peptide of the spectral library is both quantitatively and qualitatively present in the diaPASEF data as well.

## CONCLUSION

The composition of a wide variety of (bio)chemical samples can be determined with LC-TIMS-Q-TOF, which acquires the intensity values of ions with billions of detector events that are convertible to chromatographic retention time, ion mobility,  $Q$   $m/z$ , and TOF  $m/z$  values. Although there are several tools that use these data for specialized applications, a generic software tool that is optimized for speed, usability, and extensibility—thereby enabling community-driven developments—was lacking.

AlphaTims indexes unprocessed data in mere seconds, thereby making it equivalent to a sparse four-dimensional matrix. This allows to subsequently access the unprocessed data in milliseconds, regardless of the original complexity of the dataset. Owing to this fast accession, AlphaTims also requires only milliseconds to provide interactive data visualizations along any dimension, including XICs, ion mobilitygrams, mass spectra, TICs, or two-dimensional heatmaps. AlphaTims is easy to install and use on all major operating systems, without requiring any computational expertise. It can be used as a stand-alone GUI, CLI, or Python module and includes extensive help in the form of a README file, test data, a Python tutorial, CLI manual, and a GUI manual. It is a fully open-source package with a minimal number of dependencies and is freely available under an Apache license at <https://github.com/MannLabs/alphatims>.

Owing to the documented and freely available code base, AlphaTims can easily be integrated in other community projects. As an example, we are already actively integrating it in accelerated DIA workflows and AlphaViz, a new software tool in the AlphaPept 'ecosystem' that visualizes identified peptides within raw data. Furthermore, we also envision to expand the AlphaTims source code and include for instance other vendors, a low-memory mode with optimized usage of HDF5 files, a multisample mode to directly compare different runs, or even on-the-fly indexing of data that are being generated in real time.

## DATA AVAILABILITY

AlphaTims is a fully open-source package and is freely available with an Apache license at <https://github.com/MannLabs/alphatims>. The results in this article were obtained with AlphaTims, version 0.2.8. The mass spectrometry proteomics data have been deposited to the ProteomeXchange Consortium via the PRIDE (24) partner repository with the dataset identifier PXD027359.

**Acknowledgments**—We would like to thank Sven Brehmer and Sascha Winter from Bruker Daltonics for explaining the binary layout of analysis.tdf\_bin files. Additional feedback from Nagarjuna Nagaraj and other Bruker Daltonics colleagues is also much appreciated. Laura Sanchez and colleagues (University of California, Santa Cruz) provided very constructive feedback on the bioRxiv article. Finally, we are grateful for the feedback and support from within our own department, in particular Marvin Thielert, Andreas Brunner, Florian Meier, Igor Paron, Sophia Steigerwald, and all members of the bioinformatics team and interest group.

**Funding and additional information**—This study was supported by The Max-Planck Society for Advancement of Science and by the Bavarian State Ministry of Health and Care through the research project DigiMed Bayern ([www.digimed-bayern.de](http://www.digimed-bayern.de)).

**Author contributions**—S. W. and M. M. conceptualization; S. W. and E. V. formal analysis; S. W., E. V., P. S., M. T. S., and M. M. investigation; S. W., E. V., P. S., M. T. S., and M. M. writing – original draft.

**Conflict of interest**—M. M. is an indirect investor in Evosep. All other authors declare that they have no conflicts of interest with the contents of this article.

**Abbreviations**—The abbreviations used are: CLI, command-line interface; DIA, data-independent acquisition; GUI, graphical user interface; JIT, just-in-time; MS/MS, tandem MS; PASEF, parallel accumulation–serial fragmentation; PyPi, Python Package Index; Q, quadrupole; SPD, samples per day; tdf, TIMS data format; TIC, total ion current; TIMS, trapped ion mobility spectrometry; XIC, extracted ion chromatogram.

Received July 29, 2021, and in revised form, September 9, 2021. Published, MCPRO Papers in Press, September 17, 2021, <https://doi.org/10.1016/j.mcpro.2021.100149>

## REFERENCES

1. Gabelica, V., Shvartsburg, A. A., Alfonso, C., Barran, P., Benesch, J. L. P., Bleiholder, C., Bowers, M. T., Bilbao, A., Bush, M. F., Campbell, J. L., Campuzano, I. D. G., Causon, T., Clowers, B. H., Creaser, C. S., De Pauw, E., et al. (2019) Recommendations for reporting ion mobility mass spectrometry measurements. *Mass Spectrom. Rev.* **38**, 291–320.
2. Ridgeway, M. E., Lubeck, M., Jordans, J., Mann, M., and Park, M. A. (2018) Trapped ion mobility spectrometry: A short review. *Int. J. Mass Spectrom.* **425**, 22–35.



## AlphaTims: Indexing, accessing and visualizing TIMS-TOF data

3. Vasilopoulou, C. G., Sulek, K., Brunner, A. D., Meitel, N. S., Schweiger-Hufnagel, U., Meyer, S. W., Barsch, A., Mann, M., and Meier, F. (2020) Trapped ion mobility spectrometry and PASEF enable in-depth lipidomics from minimal sample amounts. *Nat. Commun.* **11**, 331
4. Luo, M.-D., Zhou, Z.-W., and Zhu, Z.-J. (2020) The application of ion mobility-mass spectrometry in untargeted metabolomics: From separation to identification. *J. Anal. Test.* **4**, 163–174
5. Beck, S., Michalski, A., Raether, O., Lubeck, M., Kaspar, S., Goedecke, N., Baessmann, C., Homburg, D., Meier, F., Paron, I., Kulak, N. A., Cox, J., and Mann, M. (2015) The impact II, a very high-resolution quadrupole time-of-flight instrument (QTOF) for deep shotgun proteomics. *Mol. Cell. Proteomics* **14**, 2014–2029
6. Fernandez-Lima, F., Kaplan, D. A., Suetering, J., and Park, M. A. (2011) Gas-phase separation using a trapped ion mobility spectrometer. *Int. J. Ion Mob. Spectrom.* **14**, 93–98
7. Meier, F., Brunner, A. D., Koch, S., Koch, H., Lubeck, M., Krause, M., Goedecke, N., Decker, J., Kosinski, T., Park, M. A., Bache, N., Hoerning, O., Cox, J., R  ther, O., and Mann, M. (2018) Online parallel accumulation-serial fragmentation (PASEF) with a novel trapped ion mobility mass spectrometer. *Mol. Cell. Proteomics* **17**, 2534–2545
8. Meier, F., Brunner, A. D., Frank, M., Ha, A., Bludau, I., Voytik, E., Kaspar-Schoenefeld, S., Lubeck, M., Raether, O., Bache, N., Aebbersold, R., Collins, B. C., R  st, H. L., and Mann, M. (2020) diaPASEF: Parallel accumulation-serial fragmentation combined with data-independent acquisition. *Nat. Methods* **17**, 1229–1236
9. Ł  cki, M. K., Startek, M. P., Brehmer, S., Distler, U., and Tenzer, S. (2021) OpenTIMS, TimsPy, and TimsR: Open and easy access to timsTOF raw data. *J. Proteome Res.* **20**, 2122–2129
10. Yu, F., Haynes, S. E., Teo, G. C., Avtonomov, D. M., Polasky, D. A., and Nesvizhskii, A. I. (2020) Fast quantitative analysis of timsTOF PASEF data with MSFragger and IonQuant. *Mol. Cell. Proteomics* **19**, 1575–1585
11. Lam, S. K., Pitrou, A., and SeibertNumba, S. (2015) A LLVM-based Python JIT compiler. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*. ACM Press, Times Square, New York City
12. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., et al. (2020) Array programming with NumPy. *Nature* **585**, 357–362
13. Folk, M., Heber, G., Koziol, Q., Pourmal, E., and Robinson, D. (2011) An overview of the HDF5 technology suite and its applications. In: *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases - AD '11*. ACM Press, Times Square, New York City
14. Wilhelm, M., Kirchner, M., Steen, J. A., and Steen, H. (2012) mz5: Space- and time-efficient storage of mass spectrometry data sets. *Mol. Cell. Proteomics* **11**, 0111.011379
15. Askenazi, M., Ben Hamidane, H., and Graumann, J. (2017) The arc of mass spectrometry exchange formats is long, but it bends toward HDF5. *Mass Spectrom. Rev.* **36**, 668–673
16. Bhamber, R. S., Jankevics, A., Deutsch, E. W., Jones, A. R., and Dowsey, A. W. (2020) mzMLb: A future-proof raw mass spectrometry data format based on standards-compliant mzML and optimized for speed and storage requirements. *J. Proteome Res.* **20**, 172–183
17. Cottam, J. A., Lumsdaine, A., and Wang, P. (2013) Abstract rendering: Out-of-core rendering for information visualization. *Visualization and Data Analysis 2014* Wong, P. C., Kao, D. L., Hao, M. C., Chen, C., eds (9017), SPIE, Bellingham, WA
18. [preprint] Strauss, M. T., Bludau, I., Zeng, W.-F., Voytik, E., Ammar, C., Schessner, J., Ilango, R., Gill, M., Meier, F., Willems, S., and Mann, M. (2021) AlphaPept, a modern and open framework for MS-based proteomics. *bioRxiv*. <https://doi.org/10.1101/2021.07.23.453379>
19. Kulak, N. A., Pichler, G., Paron, I., Nagaraj, N., and Mann, M. (2014) Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nat. Methods* **11**, 319–324
20. Bache, N., Geyer, P. E., Bekker-Jensen, D. B., Hoerning, O., Falkenby, L., Treit, P. V., Doll, S., Paron, I., M  ller, J. B., Meier, F., Olsen, J. V., Vorm, O., and Mann, M. (2018) A novel LC system embeds analytes in pre-formed gradients for rapid, ultra-robust proteomics. *Mol. Cell. Proteomics* **17**, 2284–2296
21. Eisenstat, S. C., Gursky, M. C., Schultz, M. H., and Sherman, A. H. (1982) Yale sparse matrix package I: The symmetric codes. *Int. J. Numer. Methods Eng.* **18**, 1145–1151
22. Egerton, J. D., MacLean, B., Johnson, R., Xuan, Y., and MacCoss, M. J. (2015) Multiplexed peptide analysis using data-independent acquisition and Skyline. *Nat. Protoc.* **10**, 887–903
23. Muntel, J., Gandhi, T., Verbeke, L., Bernhardt, Q. M., Treiber, T., Bruderer, R., and Reiter, L. (2019) Surpassing 10 000 identified and quantified proteins in a single run by optimizing current LC-MS instrumentation and data analysis strategy. *Mol. Omics* **15**, 348–360
24. Perez-Riverol, Y., Csordas, A., Bai, J., Bernal-Linares, M., Hewapathirana, S., Kundu, D. J., Inuganti, A., Griss, J., Mayer, G., Eisenacher, M., P  rez, E., Uszkoreit, J., Pfeuffer, J., Sachsenberg, T., Yilmaz, S., et al. (2019) The PRIDE database and related tools and resources in 2019: Improving support for quantification data. *Nucleic Acids Res.* **47**, D442–D450

### 3.5. Article 5: AlphaPept, a modern and open framework for MS-based proteomics

Authors: Maximilian T. Strauss<sup>†‡</sup>, Isabell Bludau<sup>¶</sup>, Wen-Feng Zeng<sup>¶</sup>, **Eugenia Voytik<sup>¶</sup>**, Constantin Ammar<sup>¶</sup>, Julia Schessner<sup>¶</sup>, Rajesh Ilango<sup>‡</sup>, Michelle Gill<sup>‡</sup>, Florian Meier<sup>¶,§</sup>, Sander Willems<sup>¶</sup>, Matthias Mann<sup>†\*,\*\*</sup>

<sup>†</sup> Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

<sup>\*\*</sup> NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark

<sup>‡</sup> Nvidia Corporation, Santa Clara, CA, USA

<sup>†‡</sup> OmicEra Diagnostics GmbH, Planegg, Germany

<sup>§</sup> Functional Proteomics, Jena University Hospital, Jena, Germany

Pre-print published online: *bioRxiv* (2021), doi: 10.1101/2021.07.23.453379v1.

Driven by the ever-increasing amount of raw data and its complexity in MS-based proteomics, computational proteomics has rapidly evolved into an independent interdisciplinary field. To process raw MS data and derive the identification and quantification of peptides and proteins, a large variety of various proteomic frameworks and algorithms, ranging from commercial, closed-source to freely available, open source, are now available. However, the complexity of the analysis and the lack of transparency for many closed-source software tools present a major barrier for needed further developments such as improvements in execution speed, integrating new developments and generally preventing scientists to contribute directly.

To address this challenge, we developed AlphaPept, a Python-based open-source framework for efficient and transparent processing of large amounts of high-resolution MS data. We achieved hundredfold increase in speed by using Numba for just-in-time machine code compilation on CPUs and GPUs, while retaining the clear syntax and fast development speed inherent in Python. Adopting the recent implementation of the ‘literate programming’ concept, the AlphaPept code base is implemented in Jupyter Notebooks, providing extensive documentation on the complex algorithmic background. Furthermore, we followed solid software engineering principles as embodied on GitHub, such as continuous integration, deployment and extensive automated testing. AlphaPept provides a platform for researchers with novel algorithmic ideas to test or integrate their functionality easily and in a transparent and efficient way, without having to re-create the entire pipeline, which is especially important for the rapidly developing field of machine and deep learning.

In this project, I contributed to the file importing functionality, performed testing and helped write the manuscript.



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

## AlphaPept, a modern and open framework for MS-based proteomics

Maximilian T. Strauss<sup>1,†,‡</sup>, Isabell Bludau<sup>1</sup>, Wen-Feng Zeng<sup>1</sup>, Eugenia Voytik<sup>1</sup>, Constantin Ammar<sup>1</sup>, Julia Schessner<sup>1</sup>, Rajesh Ilango<sup>2</sup>, Michelle Gill<sup>2</sup>, Florian Meier<sup>1,§</sup>, Sander Willems<sup>1</sup>, Matthias Mann<sup>1,\*,\*\*</sup>

<sup>1</sup> Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

<sup>\*\*</sup> NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark

<sup>2</sup> Nvidia Corporation, Santa Clara, CA, USA

<sup>‡</sup> OmicEra Diagnostics GmbH, Planegg, Germany

<sup>§</sup> Functional Proteomics, Jena University Hospital, Jena, Germany

\* To whom correspondence should be addressed; mstrauss@biochem.mpg.de or mmann@biochem.mpg.de

### ABSTRACT

In common with other omics technologies, mass spectrometry (MS)-based proteomics produces ever-increasing amounts of raw data, making their efficient analysis a principal challenge. There is a plethora of different computational tools that process the raw MS data and derive peptide and protein identification and quantification. During the last decade, there has been dramatic progress in computer science and software engineering, including collaboration tools that have transformed research and industry. To leverage these advances, we developed AlphaPept, a Python-based open-source framework for efficient processing of large high-resolution MS data sets. Using Numba for just-in-time machine code compilation on CPU and GPU, we achieve hundred-fold speed improvements while maintaining clear syntax and rapid development speed. AlphaPept uses the Python scientific stack of highly optimized packages, reducing the code base to domain-specific tasks while providing access to the latest advances in machine learning. We provide an easy on-ramp for community validation and contributions through the concept of literate programming, implemented in Jupyter Notebooks of the different modules. A framework for continuous integration, testing, and benchmarking enforces solid software engineering principles. Large datasets can rapidly be processed as shown by the analysis of hundreds of cellular proteomes in minutes per file, many-fold faster than the data acquisition. The AlphaPept framework can be used to build automated processing pipelines using efficient HDF5 based file formats, web-serving functionality and compatibility with downstream analysis tools. Easy access for end-users is provided by one-click installation of the graphical user interface, for advanced users via a modular Python library, and for developers via a fully open GitHub repository.

## INTRODUCTION

Increasingly large data sets, combined with exponentially increasing computational power and algorithmic advances, are transforming every aspect of science. This is accompanied and enabled by developments in open and transparent science. The open-source community has been a particular success, starting as a fringe movement to a recognized standard for software development, whose value is embraced and adapted even by the largest technology companies. Public exposure supports high code quality through scrutiny by developers from diverse backgrounds, while increasingly sophisticated collaboration mechanisms allow rapid and robust development cycles. The most advanced machine and deep learning research, for example, builds on open-source projects and datasets and is itself open-source. These laudable developments reflect the core ideas of science and present great opportunities in the ever more important computational fields.

In mass spectrometry (MS)-based proteomics, algorithms and computational frameworks have been a cornerstone in interpreting the data, resulting in a large variety of different proteomic software packages and algorithms, ranging from commercial, freely available to open source, exemplified by and reviewed in (Välikangas, Suomi, and Elo 2017; Chen et al. 2020). Typical computational workflows comprise the detection of chromatographic features, peptide spectrum matching, all the way through protein inference and quantification (Nesvizhskii, Vitek, and Aebersold 2007; Zhang et al. 2020). Advances in (MS)-based proteomics are also being accelerated through the sharing of datasets, such as publicly available data on the Proteome Exchange repository (Vizcaino et al. 2014; Deutsch et al. 2017).

Prompted by the developments in the Python scientific environment and in collaborative development tools, we developed AlphaPept, a Python-based open-source framework for efficient processing of large amounts of high-resolution MS data. Our main design goals were accessibility, analysis speed, and robustness of the code and the results. Accessibility refers to the idea of facilitating the contribution of algorithmic ideas for (MS)-based proteomics, which is today typically limited to bioinformatics experts. We decided on Python because its clear, easy-to-understand syntax, and because the excellent supporting scientific libraries make it easier for developers from different backgrounds to contribute to and implement new ideas. Using community-tested packages makes the codebase more maintainable and robust, allowing us to focus on domain knowledge instead of implementation details. We furthermore adopted a recent implementation of ‘literate programming’ (Knuth 1984), in which code and documentation are intertwined. Using the nbdev package, the codebase is connected to extensive documentation in Jupyter Notebooks in a way that immediately explains the algorithmic background, making it easier to understand the underlying principles and documenting design decisions for others (Kluyver et al. 2016). With the help of the Numba package for just-in-time compilation (JIT) of Python code (Lam, Pitrou, and Seibert 2015), AlphaPept achieves extremely fast computation times. Furthermore, we implemented robust design principles of software engineering on GitHub, such as continuous integration, deployment and extensive automated validation.

Depending on the user, AlphaPept can be employed in multiple ways. A ‘one-click’ installer can be freely downloaded for Windows, providing a web server-based graphical user interface (GUI) and a command line interface; A Python library that allows re-use and modification of its functionality in custom code, including in Jupyter Notebooks that have become a standard in data science and finally, in a scalable cloud environment.

In the remainder of the paper, we describe the functionality of AlphaPept on the basis of nbdev notebooks, such as feature finding, peptide identification and protein quantification. We demonstrate



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

the capabilities of AlphaPept on small- and large-scale datasets. Finally, we demonstrate how AlphaPept can be utilized as a proteomic workflow management system and how it can be integrated with downstream analysis tools such as Perseus or the Clinical Knowledge Graph (CKG), (Santos et al. 2020; Tyanova et al. 2016) and we provide an outlook on novel functionality to be incorporated soon.

## RESULTS

*Overview of AlphaPept architecture* - Academic software development is often highly innovative but is rarely undertaken with dedicated funding or long term personnel stability. Such constraints have successfully been mitigated by collaborative software engineering approaches and the collective efforts of volunteers. This is exemplified in state of the art open-source projects such as NumPy (Harris et al. 2020) and scikit-learn (Pedregosa et al. 2011). This paradigm has also been taken over by relatively recent and highly popular deep learning frameworks like Google's Tensorflow (Martin Abadi et al. 2015) and Facebook's PyTorch (Paszke et al. 2019) and is thought to lead to increased code quality due to community exposure and a large testing audience. Inspired by these developments, AlphaPept implements robust design principles of software engineering on GitHub, such as continuous testing and integration. For instance, code contributions can be made via pull requests which are automatically validated. By making the code publicly available and providing a stringent testing environment, we hope to encourage contribution and testing from a diverse background while maintaining very high code quality.

Organization in notebooks with nbdev allows us to collect documentation, code and tests in one place. This enables us to automatically generate the documentation, extract production code and test functionality by executing the notebooks. Furthermore, we extend the notion of unit and system testing by including real world data sets on which the overall improvement of newly implemented functionality is routinely evaluated. To continuously monitor system performance, summary statistics are automatically uploaded to a database where they are visualized in a dashboard.

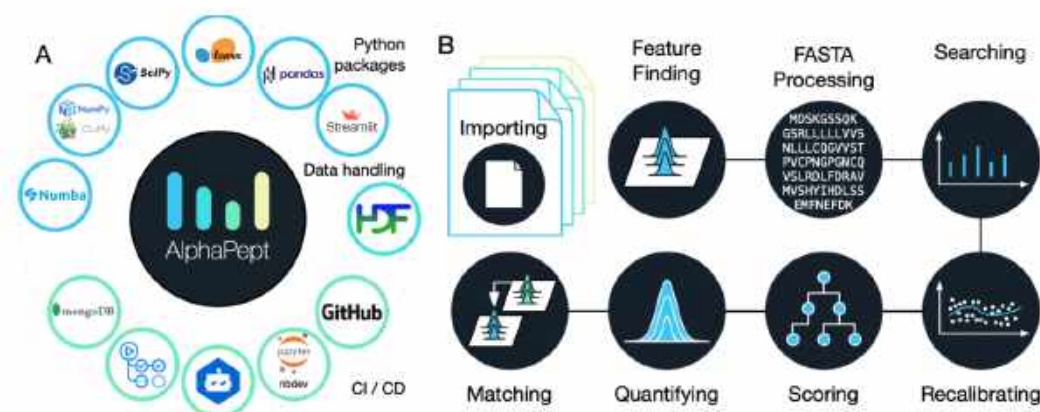
The advantages of high-level languages generally come at the price of execution speed, especially for Python. As a result, this expressive language is often only used as a thin wrapper on C++ libraries. In AlphaPept, we make use of the Numba project (Lam, Pitrou, and Seibert 2015), which allows us to compile our Python algorithms directly with the industry-standard LLVM compiler (backend to most C++ compilers and supercomputing languages such as Julia). This allows us to speed up our code by orders of magnitude without losing the benefits of the intuitive Python syntax. Furthermore, AlphaPept readily parallelizes computationally intensive parts of the underlying algorithms on multiple CPU cores or – if available - Graphical Processor Units (GPUs) for further performance gains.

As far as possible, AlphaPept uses the standard, but powerful packages of the Python data analysis universe, namely NumPy for numerical calculations, pandas for spreadsheet-like data structures and scikit-learn for machine learning (Fig. 1A). Furthermore, we chose the binary, high performance HDF5 file format, which is used across scientific areas, including 'big data' projects (see below). All these packages are platform-independent, allowing deployment of AlphaPept on Windows, Mac and Linux computers, including cloud environments.

An integral feature of AlphaPept development are Jupyter notebooks, which have become ubiquitous in scientific computing. Using the nbdev package, each part of the MS-based proteomics workflow is modularized into a separate notebook. This allows extensive documentation of the underlying

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

algorithmic production code, which is automatically extracted from and synchronized with the notebooks. Furthermore, the notebooks capture the background information of each part of the computational proteomics workflow, making it much easier to understand the underlying principles. We have found this to be an excellent way of developing software, which brings together the typical cycle of exploration in notebooks with the production of a robust and tested code base. Figure 1B shows an overview of the steps in the analysis of a typical proteomics experiment in AlphaPept corresponding to the notebooks. These separate processing steps will be discussed in turn in the sections below.



**Figure 1: AlphaPept 'ecosystem' and Modules**

**A** AlphaPept relies on multiple community-tested packages. We use highly optimized libraries such as Numba, NumPy, CuPy, scikit-learn, SciPy and pandas to achieve performant code. As GUI, we provide a browser-based application built on streamlit. For data handling, the HDF5 file technology is used. The repository itself is hosted on GitHub, the core code is documented in Jupyter Notebooks using the nbdev package. To ensure maintainability, packages are continuously monitored for updates via dependabot. New code is automatically validated using GitHub actions and summary statistics (timing, identifications and quantifications) are uploaded to a mongoDB database and visualized. **B** All algorithmic code of AlphaPept is organized in Jupyter Notebooks. For the key processing steps in the pipeline, such as importing raw data, Feature Finding, FASTA processing, Searching, Recalibrating, Scoring, Quantifying and Matching, there are individual notebooks with background information and the code.

*Highly efficient and platform-independent MS data access* – MS-based proteomics or metabolomics generates complex data types of MS1 level features, variable length MS2 data and mappings between them. Furthermore, data production rates are rapidly increasing, making robust and fast access a central requirement. The different MS vendors have their own file formats, which may be highly optimized but are meant to be accessed by their own software. We therefore faced the task of extracting the raw data into an equally efficient but vendor-neutral format that could be accessed rapidly.

First, AlphaPept needs to convert vendor specific raw files. For Thermo files we created a cross-platform Python application programming interface (API) that can directly read .RAW MS data (pyRawFileReader, Fig. 2a). It uses PythonNET for accessing Thermo's RawFileReader .NET library (Zeng, Wen-Feng 2021, 1), obviating the need for Thermo's proprietary MSFileReader. For Windows, PythonNET is available by default as a part of Windows' .NET Framework. For Linux and MacOS, PythonNET requires the open-source Mono library. Although our solution uses stacked APIs, loading the spectra of a Thermo .RAW file of 1.6 Gb into RAM takes only about one minute



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

which can be speeded up even more by parallel file processing. Access to Bruker's timsTOF raw data is also directly handled from our Python code, in this case through a wrapper to the external `timsdata.dll` C/C++ library, both made available by Bruker. In parallel with this publication, we provide AlphaTims, a highly efficient package to access large ion mobility time-of-flight data through Python slicing syntax and with ultra-fast access times (<https://github.com/MannLabs/alphatims>).

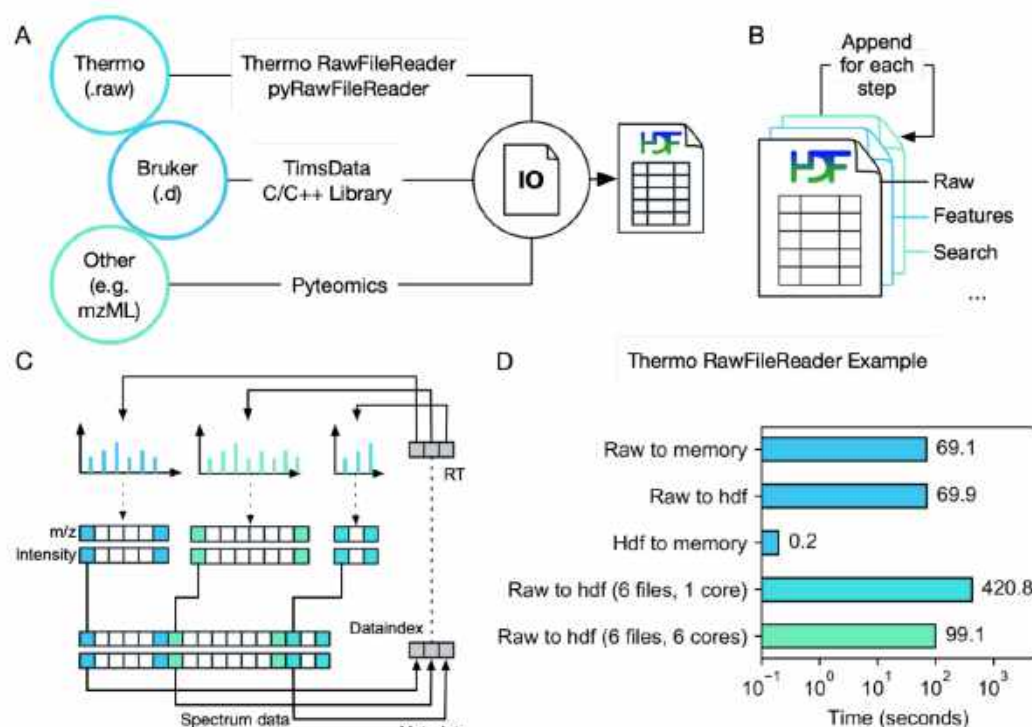
To accommodate raw data acquired through other vendors, we use Pyteomics (Goloborodko et al. 2013; Levitsky et al. 2019). This package allows reading mzML and other standard MS data formats with Python. Thus, by first converting raw data with external software such as e.g. MSConvert (Adusumilli and Mallick 2017), AlphaPept also provides a generic framework for all vendors.

As a storage technology, we chose HDF5 (Hierarchical Data Format 5), a standard originally developed for synchrotron and other extremely large scale experimental data sets, that has now become popular in a wide range of scientific fields (Folk et al. 2011). HDF5 has many benefits such as independence of operating systems, arbitrary file size, extremely fast accession and a transparent, flexible data structure. The latter is achieved by organizing HDF5 files in groups and subgroups, each containing arrays of arbitrary size and metadata which describes these arrays and (sub)groups. In the last few years, it is also becoming more popular in the field of MS (Wilhelm et al. 2012, 5). AlphaPept adopts the HDF5 technology via the Python's `h5py` package (Collette 2013).

As an additional design choice we also store intermediate processing results in the HDF5 container, so that individual processing steps can be performed in a modular way and from different computers. This enables researchers to quickly implement and validate new ideas within the downstream processing pipeline. Thus, for each new sample, AlphaPept creates a new `.ms_data.hdf` file and for each step in the workflow, the file is extended by a new group (Fig. 2b). In this way, the `.ms_data.hdf` file ensures full portability, transparency and reproducibility while being fast to access and with minimal storage requirements. For example, the 1.6 Gb Thermo file mentioned above is converted to a HDF5 file of 200 MB, all of which can be accessed in a total of 0.2 s (Fig. 2D).

We next provide functionality for MS data pre-processing, such as centroiding and extraction of the *n*-most abundant fragments, should this not already have happened in the vendor software. MS1 and MS2 scans form the two major subgroups in the HDF5 file. As HDF5 files are not optimized for lists of arrays with variable length, we convert the many individual spectra into a defined number of arrays, each containing a single data type, but concatenating all spectra. These arrays are organized in two sets: Spectrum metadata (spectrum number, precursor *m/z*, RT, etc), where each array position corresponds to one spectrum; and spectrum data, where each array position corresponds to a single *m/z*-intensity pair. To unambiguously match the spectrum datapoints to their metadata, an index array is created. It is part of the first set of arrays and contains a pointer to the position of the first data pair for each spectrum within the second set. The position of the last pair does not need to be stored as it is implied by the start position of the next spectrum. Thereby, all *m/z* values and intensities for each spectrum can easily be extracted with simple base Python slicing, while fixing the number arrays contained in the hdf container. Loading data from HDF5 to RAM takes less than a second, effectively speeding up data accession more than 300-fold compared to loading the RAW file (Fig. 2d).

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



**Figure 2: Highly efficient and platform-independent MS data access**

A MS data from different vendors is imported to an HDF5 container for fast and platform-independent data access. To read Thermo data, we provide a Python application programming interface. Bruker data is accessed via Bruker's proprietary DLL. Additionally, generic data can be imported using the Pyteomics package. **B** The output of each processing step is appended to the HDF5, allowing processing in a modular way. **C** To efficiently store MS spectra, multiple spectra of variable length are concatenated, and start indices are saved in a lookup table. **D** HDF5 Accessing times. Loading data from HDF5 into memory takes less than 1s for a typical 2h full proteome analysis of a HeLa sample acquired on a Thermo Orbitrap mass spectrometer.

*Extracting isotope features* – Having stored the MS peaks from all mass spectra in an efficient data structure, we next determine isotope patterns over chromatographic elution profiles. This computationally intensive task is crucial for subsequent peptide identification and quantification. MaxQuant (Cox and Mann 2008) introduced the use of graphs for feature finding, which was then improved upon by the Dinosaur tools (Teleman et al. 2016) and we also decided to follow this elegant approach.

In the first step - called hill building – centroided peaks from adjacent scans are connected. As there are millions of centroids, our first implementations using pure Python took several minutes of computing time. We subsequently refactored the graph problem and parallelized it for CPUs using Numba and CuPy for GPUs, resulting in a 300-fold speed up (about 1s on GPU). Since not every user has access to GPUs, AlphaPept employs dedicated Python 'decorators', a metaprogramming technique allowing a part of the program to modify its another part at compile time to transparently switch between parallelized CPU, GPU and pure Python operation.

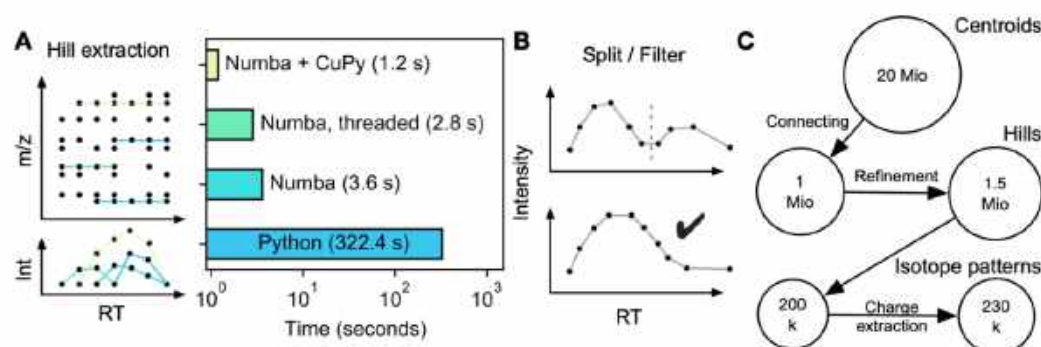


bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

In more detail, AlphaPept refines hills by first splitting them in case they have local minima indicating two chromatographic elution peaks (Fig. 3B). Additionally, hills are removed whose elution profiles do not conform to minimal criteria, like minimal length and the existence of local minima. To efficiently connect hills, we compute summary statistics such as weighted average  $m/z$  value and a bootstrap estimate of its precision. Hills within retention time boundaries are grouped into pre-isotope patterns. To correctly separate co-eluting features, we generate seeds, which we extend in elution time and check for consistency with a given charge state, similarity in elution profile and for conformity with peptide isotope abundance properties via the averagine model (Senko, Beu, and McLafferty 1995). This results in a feature (here a possible peptide precursor mass), which is described by a table.

Feature finding on the Bruker timsTOF involves ion mobility as an additional dimension. Currently, this functionality is provided by a Bruker component, which we linked into our workflow via a Python wrapper, and is the only part that is not natively included as Python code in AlphaPept. Instead, this wrapper uses Python's subprocess module, which can integrate other tools into AlphaPept just as easily.

For a typical proteomics experiment performed on an Orbitrap instrument, Figure 3C provides an overview of the number of data points from MS peaks to the final list of isotope patterns. Note that AlphaPept can perform feature finding separately for each file as soon as it is acquired (described below). Furthermore, although described here for MS1 precursors, the AlphaPept feature finder is equally suited to MS2 data that occur in parallel reaction monitoring (PRM) or DLA acquisition modes.



**Figure 3: Extracting isotope features**

**A** Individual MS peaks of similar masses are connected over the retention time using a graph approach, resulting in 'hills'. Using a native Python implementation, hill extraction takes several minutes. Numba, parallelization on CPUs or GPUs reduces hill extraction to seconds. **B** Extracted hills are refined by splitting at local minima and only allowing well-formed elution profiles. **C** Starting with 20 million points for a typical Thermo HeLa shotgun proteomics file, these are connected to approximately one million hills, which increased to 1.5 million after hill splitting and filtering. Subsequent processing results in 200,000 pre-isotope patterns that ultimately yield 230,000 isotope patterns due to assignment to specific charge states.

*Peptide spectrum matching* – The heart of a proteomics search engine is the matching of msms spectra to peptides in a protein sequence database. AlphaPept parses FASTA files containing protein sequences and descriptions, 'digests' them into peptides and calculates fragment masses according to user specified rules and amino acid modifications (Fig 3D). We again use HDF5 files, which enables



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

efficient storage of fragment series despite their varying lengths. Generation of this database only happens once per project and only takes minutes for typical organisms and modifications. From a FASTA file of the human proteome, typically five million ‘in silico’ spectra of fragment masses are generated. In case no enzyme cleavage rules are specified or for open search with wide precursor mass tolerances, the fragments are instead generated on the fly to avoid excessive file sizes.

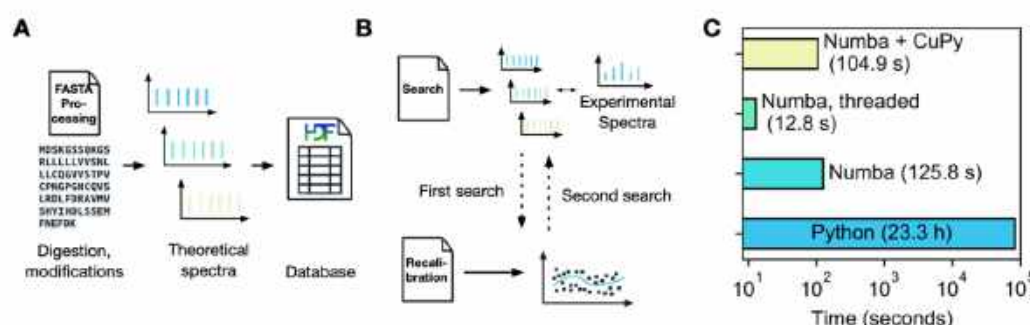
To achieve maximum speed, AlphaPept employs a very rapid fragment counting step to determine initial peptide spectrum matches (PSMs). As this step only involves addition and subtraction of elements in numerical arrays, the machine code produced by Numba is very efficient and easily parallelized. This leaves a much smaller number of peptides that have at least a minimum number of fragment matches to the experimental spectrum. (This is similar to the Morpheus score (Wenger and Coon 2013), which also computes the fraction of msms signals accounted for by the match.) For the human proteome and mass measurement accuracy of parts per million, the initial millions of comparisons are decreased to a maximum of top-n remaining candidates per msms spectrum (typically 10). This enables more computationally expensive scoring in a second step. Different scores can be implemented in AlphaPept, and by default we chose the widely used X!Tandem score (Craig and Beavis 2003). Note that the sole function of this score is to rank the PSMs, whereas statistical significance is determined by counting reverse database hits and by machine learning (see below).

We perform a first search for the purpose of recalibrating the mass scale as a function of elution time (Fig. 4B). Here, we use weighted nearest neighbor regression instead of binning by retention time (explained in the accompanying Jupyter Notebook). The k-nearest neighbors regressor that we selected allows non-linear grouping in several dimensions simultaneously (retention time and mass scale in the case of Orbitrap data and additionally ion mobility in the case of timsTOF data).

Having recalibrated the data, the main search is performed with an adapted precursor tolerance. We furthermore calculate the matched ion intensity, matched ions, neutral loss matches for further use and reporting together with charge, retention time and other data.

To demonstrate the speed up achieved by our architecture and the performance decorator, we timed illustrative examples (Fig. 4C). On a HeLa cell line proteome acquired in a single run, comparing 260k spectra to 5 million database entries, the computing time in pure Python was about 23 h. This decreased to 126 s when employing Numba (> 500x improvement), to 105 s when using Numba with CuPy on GPU and further to 13 s on multi-threaded CPU (see companion Figure Notebook). The GPU acceleration is not larger because the code is already very efficient on CPU and some workflow tasks are memory bound instead of computationally bound. Improved memory management on GPU could further decrease GPU computational time. In any case, AlphaPept reduces the PSM matching step to an insignificant part total computation time.

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



**Figure 4 Database search**

**A** The FASTA processing notebook contains functionality to calculate fragment masses from FASTA files which are saved in an HDF5 container for subsequent searches. **B** Initially, a first search is performed, and masses are subsequently recalibrated. Based on this recalibration, a second search with more stringent boundaries is performed. **C** Using the decorator strategy, the search can be drastically speeded up, from 23 h in a pure Python implementation to seconds with Numba and CuPy.

*Machine learning based scoring and FDR estimation* - Assessing the confidence of PSMs requires a scoring metric that separates true (correctly identified) from false (wrongly identified) targets in the database. Multiple defined features are calculated by the AlphaPept search engine and used in a score to rank the targets. A nonsense database of pseudo-reversed sequences where the terminal amino acid remains unchanged (de Godoy et al. 2008) is used to directly estimate the False Discovery Rate (FDR) by counting reverse hits. Score thresholds subsequently decide which targets should be considered identified. To further validate this approach and to ensure accurate FDR estimation across different development stages in AlphaPept, our GitHub testing routine includes an empirical two species FDR test based on an ‘entrapment strategy’ (Muntel et al. 2019).

In recent years, machine learning has gained increasing momentum in science in general, but also in its specific applications to MS data analysis. One of the first of these was the combination of multiple scoring metrics to a combined discriminant score that best separates high scoring targets from decoys. This was initially integrated into PSM scoring through an external reference dataset to train the classifier (Keller et al. 2002). The widely used Percolator approach subsequently employed a semi-supervised learning approach that was trained directly on the dataset itself (Käll et al. 2007). This automatically adapts the ML model to the experimental data and along with other MS analysis tools (MacLean et al. 2010; Röst et al. 2014; Teleman et al. 2015; Fondrie and Noble 2021; Rosenberger et al. 2017) we also employ semi-supervised learning for PSM scoring in AlphaPept.

The AlphaPept scoring module falls into five parts: (1) feature extraction for all candidate PSMs, (2) selection of a candidate subset, (3) training of a machine learning classifier, (4) scoring of all candidate PSMs and (5) FDR estimation by a target-decoy approach (Fig. 4A). Most features for scoring the candidate PSMs are directly extracted from the search results, such as the number of b- and y-ion hits and the matched ion intensity fraction. Some additional features are subsequently determined, including the sequence length and the number of missed cleavages. After feature extraction, a subset of candidate PSMs is selected with an initial 1% FDR threshold based only on the X!Tandem score (Fig. 4B). Together with an equal number of randomly selected decoys, this creates a balanced dataset for machine learning. This is split into training and test sets (20% vs. 80%) and provides the input of a ML classifier. We chose a standard scikit-learn random forest classifier as it performed similarly to XGBoost with fewer dependencies on other packages. We first identify



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

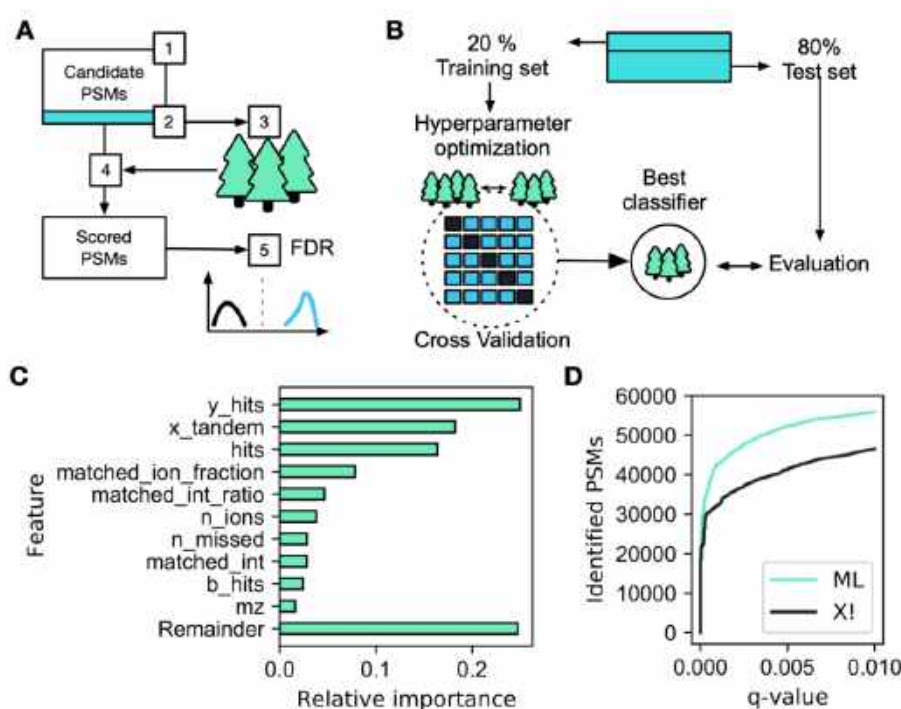
optimal hyper-parameters for the classifier with a grid-search via five-fold cross-validation. The resulting best classifier optimally separates target from decoy PSMs on the test set. Applying the trained classifier to the entire set of candidate PSMs yields discriminant scores that are used to estimate q-values based on the classical target-decoy competition approach.

The contribution of different features to the discriminant score for an exemplary tryptic HeLa sample is shown in Figure 4C. Interestingly, for our data, the number of matched y-ions alone outperforms the basic search engine score and most of the top-ranking features are related to the number of matched ions and their intensity. The ML algorithm markedly improved the separation of targets vs decoys, retrieving a larger number of PSMs at every q-value (Fig. 4D). ML-based scoring in AlphaPept improved identification rates by 15% at a 1% FDR at the PSMs level, in line with previous efforts (Käll et al. 2007). AlphaPept allows ready substitution of the underlying PSM score and machine learning algorithms. Furthermore, additional features to describe the PSMs are readily integrated, such as ion mobility or predicted fragment intensities. We envision that this kind of flexibility will enable continuous integration of improved workflows as well as novel ML techniques into AlphaPept.

Once a set of PSMs at a defined FDR is identified, protein groups are determined via the razor protein approach (Nesvizhskii and Aebersold 2005). Here, peptides that could potentially map to multiple unique proteins are assigned to the protein group that already has most peptide evidence. We determine protein-level q-values by selecting the best scoring precursor per protein, followed by FDR estimation by target-decoy competition similar to the peptide level (Nesvizhskii 2010; Savitski et al. 2015; The et al. 2016; Gupta and Pevzner 2009). Finally, we validated the scoring and FDR estimation in AlphaPept with the entrapment strategy mentioned above, by analyzing a HeLa sample with a mixed species library, containing targets and decoys derived from both a human FASTA and a FASTA from *Arabidopsis thaliana*. This revealed that AlphaPept provides accurate q-value estimates, reporting approximately the same number of *Arabidopsis thaliana* proteins as decoy proteins at 1% protein FDR.



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



**Figure 5: Machine learning-based scoring and FDR estimation**

**A** We train a Random Forest (RF) classifier on a subset of candidate PSMs to distinguish targets from decoys based on PSMs characteristics. A semi-supervised machine learning model is applied with the following steps: (1) extraction of all candidate PSM scores, (2) selection of a PSM subset for machine learning, (3) training of a RF classifier, and (4) application of the trained classifier to the full set of PSM candidates. Finally, the probability of the RF prediction is used as a score for subsequent FDR control (5). **B** Training of the classifier (step 4 in panel A) follows a train-test split scheme where only a fraction of the candidate subset is used for training. Using stringent cross-validation, multiple hyperparameters are tested to achieve optimal RF performance. The best classifier is benchmarked against the remaining test set. **C** Example feature importance for an Orbitrap test set, where the number of y-ion hits is the highest contributing factor to the model. Note that the RF algorithm can utilize any database identification score such as the X!Tandem score chosen here, which is the second most important feature. See the *AlphaPept workflow and files* Notebook for an explanation of features. **D** Optimized identification with the ML score. Compared to the X!Tandem score alone, the ML optimization identified about 15% more PSMs for the same q-value.

*Label-free quantification* - The ultimate goal of a proteomics experiment is to derive functional insights or assess biomarkers from quantitative changes at the protein level, to which peptide identifications are only means to an end. Algorithmically this quantification step entails either the determination of isotope ratios in the same scans (for instance SILAC, TMT or EASI-tag ratios) or the somewhat more challenging problem of first integrating peaks and then deriving quantitative ratios across samples (label-free quantification), which we focus on here. We initially adapted the MaxLFQ pipeline for label-free quantitative proteomics data (Cox et al. 2014). The first task is to determine normalization factors for each run as different LC MS/MS runs need to be compared – potentially spaced over many months in which instrument performance may vary – and as total loading amounts likewise vary for instance due to pipetting errors. The basic assumption is that the majority of peptides are not differentially abundant between different samples. This allows deriving

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

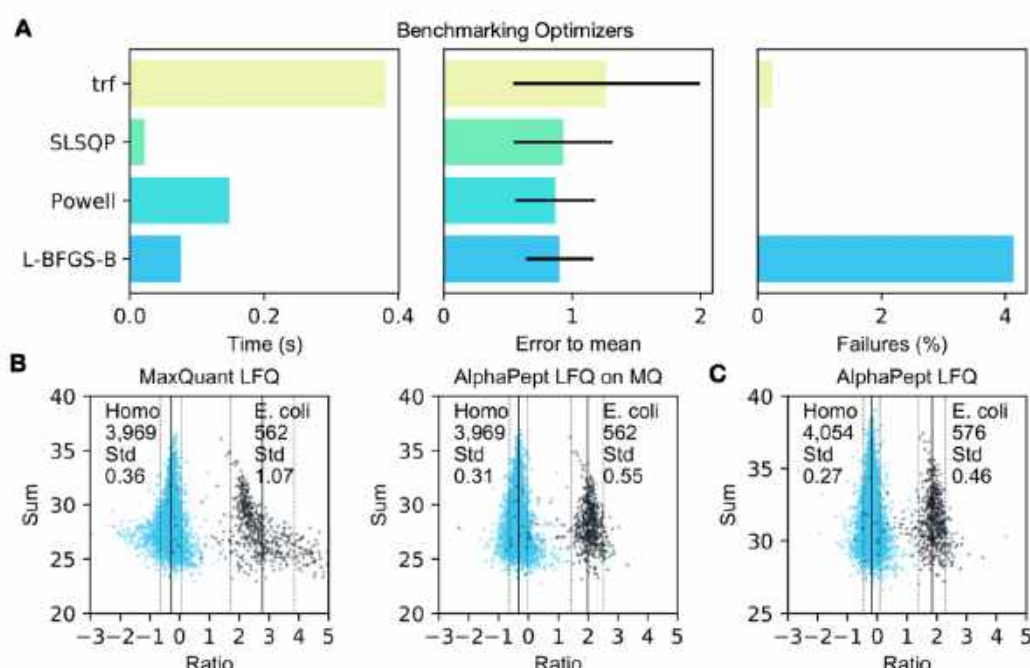
the run-specific normalization factors by minimizing the between-sample log peptide ratios (Cox et al. 2014) (Note that this assumption is not always valid and can be restricted to certain protein classes.). In a second step, adjusted intensities are derived for each protein, such that protein intensities between different MS runs can be compared. To this end we derive the median peptide fold changes that maximize consistency with the peptide evidence.

The normalization, as well as protein intensity profile construction, are quadratic minimization problems of the normalization factors or the intensities, respectively. Such minimization problems can be solved in various ways but one fundamental challenge is that these algorithms have a time complexity of  $O(n^2)$ , meaning that the computation time increases quadratically with the number of comparisons. One strategy to overcome this limitation is to only perform minimization on a subset of all possible pairs (termed 'FastLFQ') (Cox et al. 2014). Despite this, the computation time of the underlying solver will determine the overall runtime and accounts for the long run times on very large datasets. However, a variety of very efficient solvers that are based on different algorithms are contained in the Python SciPy package (SciPy 1.0 Contributors et al. 2020). To test these approaches, we created an *in silico* test dataset with a known ground truth (see Quantification Notebook). Comparing different solvers using our benchmarking set uncovered dramatic differences in precision, runtime and success rate (Fig. 6A). Among the better performing algorithms were the least-squares solvers that were previously used. The *Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS-B), *Sequential Least Squares Programming* (SLSQP) and *Powell* algorithms were particularly fast and robust solutions being up to 16x quicker than the Trust Region Reflective algorithm (trf) from the default least-squares solver. More remarkably, they were able to optimize much better to our known ground truth. Of all four tested optimizers, the mean error of trf was, on average 24% worse. Being able to readily switch between different solvers provided by SciPy allows us to fall back on other solvers if the default solver fails, i.e. AlphaPept will switch from L-BFGS-B to Powell if the solution does not converge.

We compared our method to MaxLFQ in a quantitative two-species benchmarking dataset, in which *E. coli* proteins change their abundance by a factor of six between conditions, while human proteins do not change (Meier et al. 2018). To specifically assess the benefits of the new optimization strategy, we first tested the algorithm directly on the MaxQuant output (see companion Notebook for Figure 6). Both approaches clearly separated human and *E. coli* proteins, however, the standard deviation was smaller when applying the AlphaPept optimization algorithm, which also has fewer outlier quantifications (Fig 6B), supporting the analysis of the *in-silico* test set. Comparing results of the complete workflow with AlphaPept on the same files further improved identifications and quantifications.



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



**Figure 6: Algorithm selection and performance of label-free quantification**

**A** Timings of different, highly optimized solvers from the SciPy ecosystem, to extract optimal protein intensity ratios in AlphaPept. Solvers showed drastic differences in speed, closeness to ‘ground truth’, and proportion of successful optimizations on *in-silico* test data. Based on these tests, AlphaPept employs a hybrid optimization strategy that uses L-BFGS-B and Powell for optimized performance, robustness and speed. **B** Comparing the AlphaPept LFQ solver on MaxQuant output data demonstrates similar separation in mixed-species datasets with smaller standard deviations. **C** Applying AlphaPept directly on the same dataset further improves identifications and quantification accuracy.

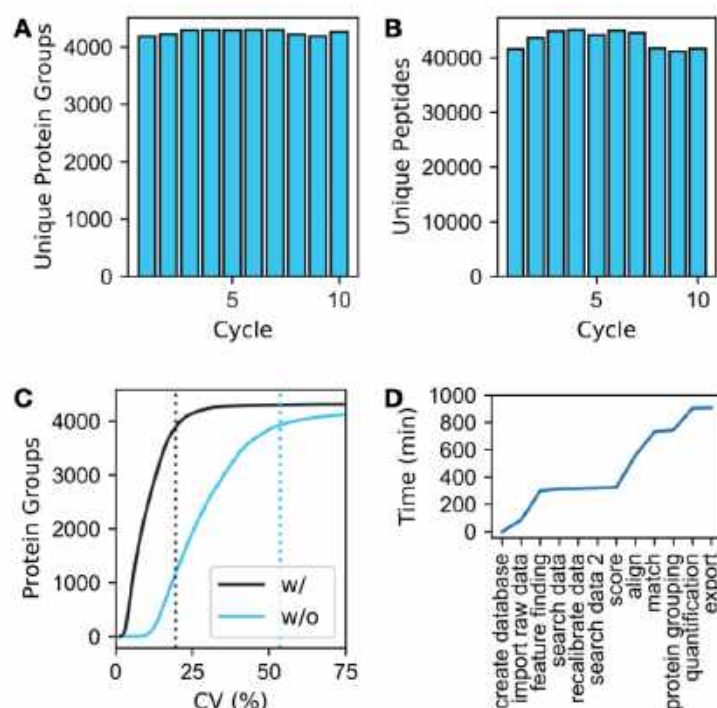
*Match-between-runs (MBR) and dataset alignment* – We implemented functionality to transfer the identifications of MS1 features to unidentified MS1 features of other runs (*match-between-runs*). First, we align multiple datasets on top of each other by applying a global offset in retention time, mass and – where applicable – ion mobility. To determine offsets for all runs, we first compare all possible pairs of runs and calculate the median offset from one dataset to another based on the precursors that were identified in both. As these offsets are linear combinations of each other, i.e., the offset from dataset A to dataset C should be the offset from dataset A to B and B to C; this becomes an overdetermined equation system, which we solve by a weighted linear regression model with the number of shared precursors as weights.

After dataset alignment, we group precursors of multiple runs and determine their expected properties as well as their probability density and create a library of precursors. Next, we take the unidentified MS1 features from each run and extract the closest match from the library of precursors. Finally, as we know the probability density of each feature, we can calculate the Mahalanobis distance from each identification transfer and use this as a probability estimate to assess the likelihood that a match is correct. Further information about the alignment and matching algorithm can be found in the Matching notebook.



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

**Benchmarking AlphaPept on large data sets** – A prime goal of the AlphaPept effort is robustness and speed. To showcase the usability of AlphaPept for large scale studies we re-analyzed 200 HeLa proteomes from a recently published long-term performance test (Bian et al. 2020). To confirm comparable identification performance in the initial analysis, which was done with MaxQuant, we evaluated the number of uniquely identified protein groups and PSMs per group. This yielded a median of 4277 unique protein groups and 43,872 unique peptides per experimentally defined group, as expected. Next, we compared the protein level quantification. The median coefficient of variation without our Python maxLFQ implementation was 27.1% and 9.2% after LFQ optimization. For 90% of protein groups, CVs were below 20% with LFQ optimization and below 54% without. Investigation of each computational task revealed that a large part is spent on importing raw data and feature finding. Searching and scoring are highly optimized and contribute only a small fraction of the overall computing time. Operations across files such as LFQ alignment and matching again make up a large part of computation time.



**Figure 7: Benchmarking AlphaPept on 200 HeLa proteomes**

A total of 200 DDA HeLa cell proteomes – the 10 cycle long term performance test from Kuster and coworkers (181 Gbyte) (Bian et al. 2020) – was analyzed by AlphaPept. **A** Identification performance at the protein group level. **B** Identification performance at the peptide level. **C** Quantification performance with or without MaxLFQ optimization. For 90% of protein groups, CVs are below 20% and 54%, respectively. **D** Timing of the AlphaPept computational pipeline. Search through scoring are highly optimized and contribute little to overall computation time.

**Continuous validation on standard datasets** – Our current continuous integration pipeline uses a range of data sets typical for MS workflows. These include standard single shot runs, such as HeLa

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

quality control (QC) runs, as well as recently published studies. For every addition to the main branch of the code base, AlphaPept reanalyzes these files fully automatically, allowing extensive systems checks. Additionally, these checks can be manually triggered at any time and therefore enable swift validation of proposed code changes prior to submitting pull-requests. This makes comparing studies that were analyzed with different software versions much more transparent. To further increase this idea of transparent performance tracking, we automatically upload summary statistics, such as runtime, number of proteins and number of features for each run to a database and visualize these metrics in a dashboard (Extended methods). Table 1 shows example tracking metrics from the database.

| Version | Test file     | Processing time (min) | Number of features | Number of peptides |
|---------|---------------|-----------------------|--------------------|--------------------|
| 0.2.8   | HeLa Orbitrap | 19                    | 218792             | 41777              |
| 0.2.8   | HeLa timsTOF  | 102                   | 231545             | 54058              |
| 0.2.9   | HeLa Orbitrap | 19                    | 218780             | 41939              |
| 0.2.9   | HeLa timsTOF  | 113                   | 231545             | 66776              |
| 0.2.10  | HeLa Orbitrap | 19                    | 218779             | 41949              |
| ...     | ...           | ...                   | ...                | ...                |
| 0.3.25  | HeLa timsTOF  | 105                   | 664992             | 76217              |
| 0.3.26  | HeLa Orbitrap | 18                    | 260709             | 53522              |
| 0.3.26  | HeLa timsTOF  | 88                    | 664992             | 77464              |
| 0.3.27  | HeLa Orbitrap | 21                    | 260622             | 54283              |
| 0.3.27  | HeLa timsTOF  | 89                    | 664992             | 77162              |

**Table 1: Example performance tracking metrics for different AlphaPept versions extracted from the database.**

*AlphaPept user interface and server* – A central element for any software tool is ease of use for the end user. In the most basic setup, this is determined by the accessibility of the GUI. Following recent trends, we decided on server-based technology for AlphaPept. In a basic setup, the web interface is called by connecting to a local server instance on the user's laptop or local workstation (Fig. 8A) via a browser. For more demanding pipelines, AlphaPept can be run on a powerful processing PC and be accessed from multiple other devices. This makes access to AlphaPept platform independent, including mobile devices.

Adding server functionality typically comes at the cost of maintaining a dedicated API and infrastructure. For AlphaPept we make use of a very recent but already very popular Python package called streamlit ([www.streamlit.com](http://www.streamlit.com)), which was developed to facilitate the sharing of machine learning models. By only adding one additional Python package, we have access to a powerful and responsive server infrastructure. Here, the web interface serves merely as an input wrapper to gather the required settings and display results and starts the AlphaPept processing in the background.

*AlphaPept workflow management system* – Importantly, the server-based user interface extends the processing functionality of AlphaPept from only processing individual experiments to a continuous processing and monitoring framework. The core processing function of AlphaPept accepts a dictionary-type document to process an experiment, with defined parameters per setting. To store these settings, we chose YAML, a standard human-readable data-serialization language, resulting in



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

files of only a few kilobytes in size. This ensures that they can be modified programmatically and easily checked with common editors.

The settings structure is used by the AlphaPept GUI to build a folder-based workflow management system. It creates three folders in the user folder ('Queue', 'Failed', and 'Finished') and monitors them for new data. When defining a new experiment within the GUI, a settings YAML file is created in the Queue folder, and the core function will start processing. This allows defining multiple experiments, which will then be processed one after another. YAML files of processed runs will be moved to the 'Finished' or 'Failed' folder (Fig. 8B).

We chose this folder-based processing queue as this allows manual inspection of the processing queue by simply checking the files in the folders. Furthermore, computational alterations of the processing queue are straightforward by writing custom scripts that copy settings files generated elsewhere to the queue folder. AlphaPept has a file watcher module that can monitor folders for new raw files and automatically add them to the processing queue immediately after acquisition is finished. Its modular structure can easily be extended with custom code for integration into larger processing environments with database-based queuing systems. Refer to the interface notebook, which calls the wrapper function and allows customization of the pipeline.

*Visualization of results and continuous processing* – For visualization of tabular or summary statistics results, our streamlit application utilizes the 'Finished' folder structure where it stores readily accessible summary information of previously processed files (Fig. 8C). AlphaPept has a History tab that compiles these previous results to show performance over time or across analyzed MS runs (Fig. 8D). Here, the user can choose to plot various summary statistics such as identified proteins or peptides as well as chromatographic information such as peak width or peak tailing. As a particular use case, this provides a standard interface which allows instant QC run evaluation in combination with the file watcher.

To inspect an individual experiment, AlphaPept's browser interface can also plot identification and quantification summary information. Furthermore, basic data analysis functions such as volcano or scatter plots and Principal Component Analysis (PCA) are provided. This is based on streamlit and scikit-learn functionality and can therefore be readily extended. AlphaPept exports the analysis results (quantified proteins and peptides) in tabular format to the specified results path so that it can be readily used for other downstream processing tools such as Perseus (Tyanova et al. 2016) or the recently introduced CKG (Santos et al. 2020).

*AlphaPept deployment and integration* – The utility of a computational tool critically depends on how well it can be integrated into existing workflows. To enable maximum flexibility and to address all major use cases, AlphaPept offers multiple ways to install and integrate it.

First, we provide a one-click installer solution that is packaged for a standard Windows system obviating additional installation routines. It provides a straightforward interface to the web-based GUI. We chose Windows for the one-click solution as it is the base OS for the vendor-provided acquisition and analysis software and most users. The one-click installation also has a command-line interface (CLI) for integration into data pipelines.

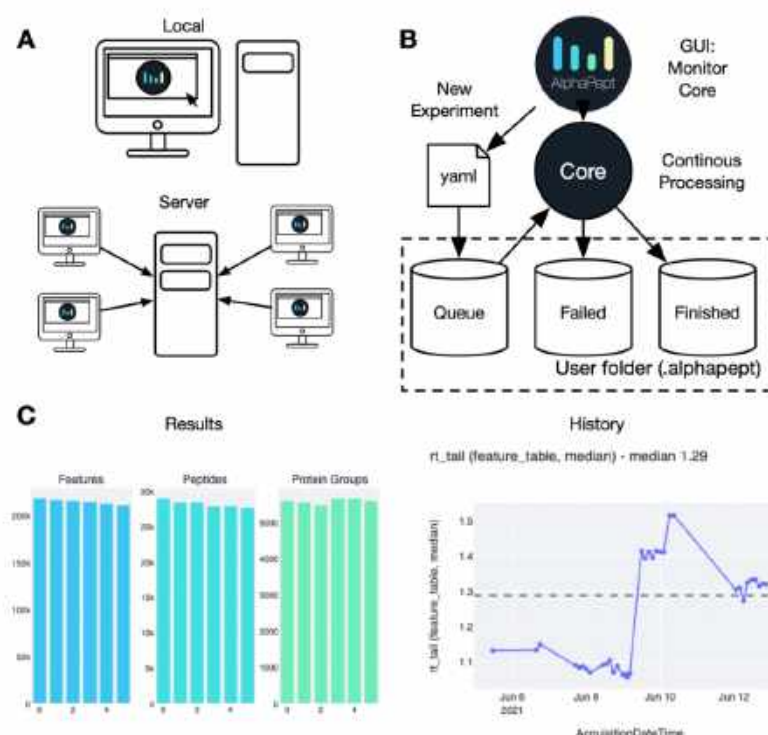
Next, AlphaPept can be used as a module in the same way as other Python packages. This requires setting up a Python environment to run the tool, which also contains all the functionality of the previously described CLI and GUI. Compared to the Windows one-click installer, the Python module extends the compatibility to other operating systems. While Python code is in principle cross-platform, some third-party packages can be platform bound, such as the Bruker feature finder or



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

DLLs required to read proprietary file types. The modular nature of the AlphaPept file system allows to preprocess files and continue the analysis on a different system (e.g., feature finding and file conversion on a Windows acquisition PC and processing on a Mac system).

Finally, the Python module makes the individual functions available to any Python program. This is particularly useful to integrate only parts of a workflow in a script or to optimize an individual workflow step. Besides the nbdev notebooks that contain the AlphaPept core code, we provide several sandboxing Jupyter Notebooks that show how individual workflow steps can be called and modified. In this way, AlphaPept allows the creation of completely customized workflows.



**Figure 8: AlphaPept user interface, workflow management, deploying and integrating**

**A** The AlphaPept GUI is based on a server architecture that can be installed on a workstation and used locally. Additionally, it can be installed on a server and accessed remotely from multiple workstations in the network. **B** AlphaPept processing pipeline. The AlphaPept GUI creates three folders for its processing system. New experiments are defined within the interface and saved as YAML files in the Queue folder with automatically triggered processing. **C** Example plots from the History and Results Tab in AlphaPept: Overview of the number of features, peptides and protein groups per injected sample (left panel). Graphing retention time tailing as a function of acquisition date, as an illustration of using AlphaPept for quality assurance.

*AlphaPept processing times* – To give the reader an impression of typical processing timings for each of these deployment variants, we ran AlphaPept on various hardware for several use cases: laptop, office PC, workstation and cloud (Table 1). AlphaPept can be readily employed with cloud providers such as Amazon Web Services. We tested our default testing pipeline (see timing table below) on

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

two different Amazon EC2 instances (t3a.2xlarge: 0.42 Eur/h and t3.xlarge: 0.22 Eur/h), an incurred computational costs of 0.22 and 3.82 Euros for one 120 min Orbitrap HeLa file and 8 timsTOF files, respectively, when processed in a European location. Computational costs can be further improved by choosing resource-optimized hardware or buying compute power in advance.

For a typical proteomics laboratory, we envision AlphaPept running in continuous mode, to automatically process all new files. This allows continuous feedback about experiments while drastically speeding up computation when subsequently combining multiple processed files into experiments and experiments into an overall study, because the computational steps that do not change (e.g., raw conversion, database generation or feature finding) can be reused. To illustrate this, the test set with 8 Bruker files from PXD010012 takes 194 minutes on a Workstation with preprocessing and 23 minutes when using preprocessed files.

|                                 |              | <b>Laptop</b><br>Macbook Pro<br>macOS Big Sur<br>i9 2.3 GHz x 8<br>32 Gb RAM | <b>Office Pc</b><br>Optiplex 7080<br>Windows 10<br>i9 3.7 GHz x10<br>64 Gb RAM | <b>Workstation</b><br>Custom<br>Windows 10<br>i9 3.5 GHz x12<br>128 Gb RAM | <b>Cloud I</b><br>AWS (t3a.2xlarge)<br>Windows Server<br>EPIC 2.2 GHz x4<br>32 Gb RAM | <b>Cloud II</b><br>AWS (t3.xlarge)<br>Windows Server<br>XEON 2.4 GHz x2<br>16 Gb RAM |
|---------------------------------|--------------|--|--|--|---|--|
| IRT Sample*<br>(Thermo)         | Full         | 1  | 1  | 2  | 3   | 2  |
| HeLa 120 min<br>(Thermo)        | Full         | 23   | 16   | 19   | 40  | 41   |
|                                 | Preprocessed | 6  | 4  | 5  | 11  | 12   |
| PXD006109 - 6<br>files (Thermo) | Full         | 36   | 17   | 21   | 46  | 73   |
|                                 | Preprocessed | 30   | 8  | s  | 18  | 24   |
| IRT Sample<br>(Bruker)          | Full         | **   | 1  | 2  | 3   | 2  |
| HeLa 120 min<br>(Bruker)        | Full         | **   | 57   | 111  | 131   | 399  |
|                                 | Preprocessed | 6  | 6  | 7  | 16  | 19   |
| PXD010012 - 8<br>files (Bruker) | Full         | **   | 242  | 194  | 790   | 893  |
|                                 | Preprocessed | 62   | 24   | 23   | 85  | 132  |

**Table 2: Running times of AlphaPept for various hardware (timings in minutes)**

\* IRT = low complexity mixture of peptides (internal retention time standard)

\*\* to process Bruker files on Mac Os X, we preprocessed them on Windows

Being able to import AlphaPept as a Python package also lowers the entry barrier of proteomics analysis workflows for individual researchers and laboratories with little computational infrastructure, as it makes it compatible with platforms like Google Colab, a free cloud-based infrastructure built on top of Jupyter notebooks with GPUs. This allows processing without having to set up software on specialized hardware and allows direct modification of the underlying algorithms. We provide an explanatory notebook for running a workflow on Google Colab, including a 120 min HeLa example file that has been converted on the Windows acquisition computer. This also highlights how the modular HDF5 file format allows us to move the MS data between operating systems.

## DISCUSSION

Here we have introduced AlphaPept, a computational proteomics framework where the relevant algorithms are written in Python itself, rather than Python being used only as a scripting layer on top of compiled code. This architectural choice allows the user to inspect and even modify the code and



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

enables seamless integration with the tools of the increasingly powerful and popular Python scientific ecosystem. The major drawback of such an approach would have been the slow execution speed of pure Python, however extensive use of the Numba just in time compiler – on multiple CPUs or a GPU – makes AlphaPept exceptionally fast, as we have shown in this manuscript. Together with the use of recently developed browser-based deployment, AlphaPept covers the full range of potential users from novice users to systems administrators wishing to build large cloud pipelines.

A related and important design objective of AlphaPept was to enable a diverse user community and invite community participation in its further development. To ensure quality, reproducibility and stability, we implemented a large suite of mechanisms from unit through end-to-end tests via automatic deployment tools. This in turn allows us to streamline the integration of community contributions after rigorous assessment. Furthermore, GitHub provides state-of-the-art tools and mechanisms to allow the effective collaboration of diverse and dispersed developer communities.

Currently, AlphaPept provides functionality for DDA proteomics but we are in the process of enabling analysis of DIA data, ultra-fast access to and visualization of ion mobility data (AlphaTims, <https://github.com/MannLabs/alphatims>), deep learning for predicted peptide properties and improved quantification, all made possible by its modular design.

One of the large goals of AlphaPept is to ‘democratize’ access to computational proteomics. To this end, besides implementation in Python, we adopted the ‘literate programming’ paradigm which integrates documentation and code. We adopted the nbdev package, providing both beginner and expert computational proteomics researchers with an easy and interactive ‘on ramp’. In our case this takes the form of currently 12 Jupyter notebooks dealing with all the major sub tasks of the entire computational pipeline from database creation, raw data import all the way to the final report of the results. We imagine that students and researchers with novel algorithmic ideas can use this paradigm to add their functionality in a transparent and efficient manner, without having to re-create the entire pipeline. This could especially enable increasingly powerful machine learning and deep learning technologies to be integrated into computational proteomics (Torun et al. 2021; Wen et al. 2020; Meyer 2021).

### Acknowledgements

We thank Sven Brehmer, Wiebke Timm, Konstantin Schwarze and Sebastian Wehner from Bruker Daltonik for providing support with the feature finder for Bruker data. Further, we thank Andreas Brunner, Igor Paron, Patricia Skowronek and Mario Oroshi for providing sample files and descriptions and feedback on the QC pipeline. Xie-Xuan Zhou contributed to discussions and testing. We are grateful of the feedback, testing and support from our group members and colleagues at OmicEra Diagnostics GmbH for testing.

### Abbreviations

**API** (application programming interface), **CLI** (command-line interface), **DDA** (data-dependent acquisition), **DIA** (data-independent acquisition), **FDR** (false discovery rate), **GPU** (graphical processor unit), **GUI** (graphical user interface), **HDF5** (hierarchical data format 5), **JIT** (just-in-time), **L-BFGS-B** (Broyden–Fletcher–Goldfarb–Shanno), **ML** (machine learning), **MS** (mass spectrometry), **MS/MS** (tandem mass spectrometry), **PASEF** (Parallel Accumulation–Serial Fragmentation), **PRM** (parallel reaction monitoring), **PSM** (peptide spectrum match), **QC** (quality control), **RF** (random forest), **SLSQP** (sequential least squares programming), **trf** (Trust Region Reflective algorithm).



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

#### Keywords

Mass spectrometry, Python, open-source, proteomics, open-source, search algorithm, proteome informatics

#### Software and Data availability

AlphaPept is fully open-source and is freely available under an Apache license at <https://github.com/MannLabs/alphapept>. All data is available on GitHub or the Max-Planck datashare as test data. Each notebook / file contains respective download links for the files used. The results in this manuscript were obtained with AlphaPept version 0.3.26 if not otherwise indicated.

#### Author contributions

MM and MTS conceived the core idea of the AlphaPept framework and MM wrote the first iteration of the search algorithm. MTS wrote the Thermo feature finder, quantification and downstream processing modules, code structure and user interface. EV contributed file importing functionality. IB extended the scoring functionality with ML and FDR control. SW added HDF file handling, revised the general code structure and added performance functions. WFZ and CA contributed and improved quantification. JS critically reviewed testing and documentation. RI and MG contributed to GPU support and code acceleration. All authors contributed ideas, performed testing and wrote the manuscript.

### EXTENDED METHODS

#### Notebook availability.

All notebooks are available in the repository on GitHub. The documentation created based on the notebooks is available here: <https://mannlabs.github.io/alphapept/>. Additional information about code not covered in the Notebooks presented here can be found in the Documentation ([https://mannlabs.github.io/alphapept/additional\\_code.html](https://mannlabs.github.io/alphapept/additional_code.html)).

A cloud hosted Notebook with an example data file is provided at the free Google Colab site: [https://colab.research.google.com/drive/163LTlyzBCDgyCkSJiikbmsnny\\_EiQ7SG?usp=sharing](https://colab.research.google.com/drive/163LTlyzBCDgyCkSJiikbmsnny_EiQ7SG?usp=sharing)

#### MongoDB Dashboard

The continuous integration pipeline has the action "Performance test pyinstaller". This action freezes the current Python environment into an executable and runs the test files. The results of these tests are uploaded to a noSQL database (MongoDB) for the tested version number. Key performance metrics are visualized in charts here: <https://charts.mongodb.com/charts-alphapept-itfxv/public/dashboards/5f671dcf-bcd6-4d90-8494-8c7f724b727b>

*timsTOF and Orbitrap HeLa samples* – The test files comprise representative single run analyses of complex proteome samples. Human HeLa cancer cells were lysed in reduction and alkylation buffer with chloroacetamide as previously described (Kulak et al. 2014), and proteins were enzymatically digested with LysC and trypsin. The resulting peptides were de-salted and purified on styrenedivinylbenzene reversed-phase sulfonate (SDB-RPS) StageTips before injection into an EASY nLC 1200 nanoflow chromatography system (Thermo Scientific). The samples were loaded on a 50 cm x 75 µm column packed in-house with 1.9 µm C<sub>18</sub> beads and fitted with a laser-pulled emitter tip. Separation was performed during 120 min with a binary gradient at a flow rate of 300 nL/min. The LC system was coupled online to either a quadrupole Orbitrap (Thermo Scientific Orbitrap Exploris 480) or a trapped ion mobility – quadrupole time-of-flight (Bruker timsTOF Pro 2) mass spectrometer. Data were acquired with standard data-dependent top15 (Orbitrap) and PASEF methods (timsTOF), respectively.

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

*timsTOF and Orbitrap iRT samples* – 11 iRT peptides (<https://biognosys.com/product/irt-kit/>) were separated via a 5.6 min Evosep gradient (200 “samples per day”) yielding test data with low complexity, that facilitated quick testing of computational functionality. An Evosep One liquid chromatography system (Evosep) was coupled online with a trapped ion mobility spectrometry (TIMS) quadrupole time-of-flight (TOF) mass spectrometer (timsTOF pro, Bruker Daltonics). iRT standards (Biognosys) were loaded onto Evotips according to the manufacturers’ instructions and separated with a 4 cm x 150  $\mu$ m reverse-phase column with 3  $\mu$ m C<sub>18</sub>-beads (Pepsep). The analytical column was connected with a zero-dead volume emitter (10  $\mu$ m) placed in a nano-electrospray ion source (CaptiveSpray source, Bruker Daltonics). Mobil phase A contained 0.1 vol% formic acid and water and mobil phase B of 0.1 vol% formic acid and acetonitrile. The sample was acquired with the dda-PASEF acquisition mode. Each topN acquisition mode contained four PASEF MS/MS scans and the accumulation and ramp time were both 100 ms. Only multiply charged precursors over the intensity threshold of 2500 arbitrary units (a.u.) and within a  $m/z$ -range of 100 – 1700 were subjected to fragmentation. Peptides that reached the target intensity of 20,000 a.u. were excluded for 0.4 min. The quadrupole isolation width was set to 2 Th below  $m/z$  of 700 and 3 Th above a  $m/z$  value of 700. The ion mobility (IM) range was configured to 0.6 – 1.51 Vs cm<sup>-2</sup> and calibrated with three Agilent ESI-L TuneMix Ions ( $m/z$ , IM: 622.02, 0.98 Vs cm<sup>-2</sup>; 922.01, 1.19 Vs cm<sup>-2</sup>; 1221.99, 1.38 Vs cm<sup>-2</sup>). The collision energy was decreased as a function of the ion mobility, starting at 1.6 Vs cm<sup>-2</sup> with 59 eV and ending at 0.6 Vs cm<sup>-2</sup> with 20 eV.

## REFERENCES

- Adusumilli, Ravali, and Parag Mallick. 2017. “Data Conversion with ProteoWizard MsConvert.” In *Proteomics*, edited by Lucio Comai, Jonathan E. Katz, and Parag Mallick, 1550:339–68. Methods in Molecular Biology. New York, NY: Springer New York. [https://doi.org/10.1007/978-1-4939-6747-6\\_23](https://doi.org/10.1007/978-1-4939-6747-6_23).
- Bian, Yangyang, Runsheng Zheng, Florian P. Bayer, Cassandra Wong, Yun-Chien Chang, Chen Meng, Daniel P. Zolg, et al. 2020. “Robust, Reproducible and Quantitative Analysis of Thousands of Proteomes by Micro-Flow LC–MS/MS.” *Nature Communications* 11 (1): 157. <https://doi.org/10.1038/s41467-019-13973-x>.
- Chen, Chen, Jie Hou, John J. Tanner, and Jianlin Cheng. 2020. “Bioinformatics Methods for Mass Spectrometry-Based Proteomics Data Analysis.” *International Journal of Molecular Sciences* 21 (8): 2873. <https://doi.org/10.3390/ijms21082873>.
- Collette, Andrew. 2013. *Python and HDF5*. O’Reilly.
- Cox, Jürgen, Marco Y. Hein, Christian A. Luber, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann. 2014. “Accurate Proteome-Wide Label-Free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ.” *Molecular & Cellular Proteomics* 13 (9): 2513–26. <https://doi.org/10.1074/mcp.M113.031591>.
- Cox, Jürgen, and Matthias Mann. 2008. “MaxQuant Enables High Peptide Identification Rates, Individualized p.p.b.-Range Mass Accuracies and Proteome-Wide Protein Quantification.” *Nature Biotechnology* 26 (12): 1367–72. <https://doi.org/10.1038/nbt.1511>.
- Craig, Robertson, and Ronald C. Beavis. 2003. “A Method for Reducing the Time Required to Match Protein Sequences with Tandem Mass Spectra.” *Rapid Communications in Mass Spectrometry* 17 (20): 2310–16. <https://doi.org/10.1002/rcm.1198>.
- Deutsch, Eric W., Attila Csordas, Zhi Sun, Andrew Jarnuczak, Yasset Perez-Riverol, Tobias Ternent, David S. Campbell, et al. 2017. “The ProteomeXchange Consortium in 2017: Supporting the Cultural Change in Proteomics Public Data Deposition.” *Nucleic Acids Research* 45 (D1): D1100–1106. <https://doi.org/10.1093/nar/gkw936>.



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- Folk, Mike, Gerd Heber, Quincey Koziol, Elena Pourmal, and Dana Robinson. 2011. "An Overview of the HDF5 Technology Suite and Its Applications." In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases - AD '11*, 36–47. Uppsala, Sweden: ACM Press. <https://doi.org/10.1145/1966895.1966900>.
- Fondrie, William E., and William S. Noble. 2021. "Mokapot: Fast and Flexible Semisupervised Learning for Peptide Detection." *Journal of Proteome Research*, February, acs.jproteome.0c01010. <https://doi.org/10.1021/acs.jproteome.0c01010>.
- Godoy, Lyris M. F. de, Jesper V. Olsen, Jürgen Cox, Michael L. Nielsen, Nina C. Hubner, Florian Fröhlich, Tobias C. Walther, and Matthias Mann. 2008. "Comprehensive Mass-Spectrometry-Based Proteome Quantification of Haploid versus Diploid Yeast." *Nature* 455 (7217): 1251–54. <https://doi.org/10.1038/nature07341>.
- Goloborodko, Anton A., Lev I. Levitsky, Mark V. Ivanov, and Mikhail V. Gorshkov. 2013. "Pyteomics—a Python Framework for Exploratory Data Analysis and Rapid Software Prototyping in Proteomics." *Journal of The American Society for Mass Spectrometry* 24 (2): 301–4. <https://doi.org/10.1007/s13361-012-0516-6>.
- Gupta, Nitin, and Pavel A. Pevzner. 2009. "False Discovery Rates of Protein Identifications: A Strike against the Two-Peptide Rule." *Journal of Proteome Research* 8 (9): 4173–81. <https://doi.org/10.1021/pr9004794>.
- Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, et al. 2020. "Array Programming with NumPy." *Nature* 585 (7825): 357–62. <https://doi.org/10.1038/s41586-020-2649-2>.
- Käll, Lukas, Jesse D Canterbury, Jason Weston, William Stafford Noble, and Michael J MacCoss. 2007. "Semi-Supervised Learning for Peptide Identification from Shotgun Proteomics Datasets." *Nature Methods* 4 (11): 923–25. <https://doi.org/10.1038/nmeth1113>.
- Keller, Andrew, Alexey I. Nesvizhskii, Eugene Kolker, and Ruedi Aebersold. 2002. "Empirical Statistical Model To Estimate the Accuracy of Peptide Identifications Made by MS/MS and Database Search." *Analytical Chemistry* 74 (20): 5383–92. <https://doi.org/10.1021/ac025747h>.
- Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. "Jupyter Notebooks - a Publishing Format for Reproducible Computational Workflows." In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by Fernando Loizides and Birgit Schmidt, 87–90. IOS Press. <https://eprints.soton.ac.uk/403913/>.
- Knuth, D. E. 1984. "Literate Programming." *The Computer Journal* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.
- Kulak, Nils A, Garwin Pichler, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann. 2014. "Minimal, Encapsulated Proteomic-Sample Processing Applied to Copy-Number Estimation in Eukaryotic Cells." *Nature Methods* 11 (3): 319–24. <https://doi.org/10.1038/nmeth.2834>.
- Lam, Siu Kwan, Antoine Pitrou, and Stanley Seibert. 2015. "Numba: A LLVM-Based Python JIT Compiler." In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*, 1–6. Austin, Texas: ACM Press. <https://doi.org/10.1145/2833157.2833162>.
- Levitsky, Lev I, Joshua A. Klein, Mark V. Ivanov, and Mikhail V. Gorshkov. 2019. "Pyteomics 4.0: Five Years of Development of a Python Proteomics Framework." *Journal of Proteome Research* 18 (2): 709–14. <https://doi.org/10.1021/acs.jproteome.8b00717>.
- MacLean, Brendan, Daniela M. Tomazela, Nicholas Shulman, Matthew Chambers, Gregory L. Finney, Barbara Frewen, Randall Kern, David L. Tabb, Daniel C. Liebler, and Michael J. MacCoss. 2010. "Skyline: An Open Source Document Editor for Creating and Analyzing Targeted Proteomics Experiments." *Bioinformatics* 26 (7): 966–68. <https://doi.org/10.1093/bioinformatics/btq054>.



bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. <https://www.tensorflow.org/>.
- Meier, Florian, Philipp E. Geyer, Sebastian Virreira Winter, Juergen Cox, and Matthias Mann. 2018. "BoxCar Acquisition Method Enables Single-Shot Proteomics at a Depth of 10,000 Proteins in 100 Minutes." *Nature Methods* 15 (6): 440–48. <https://doi.org/10.1038/s41592-018-0003-5>.
- Meyer, Jesse G. 2021. "Deep Learning Neural Network Tools for Proteomics." *Cell Reports Methods* 1 (2): 100003. <https://doi.org/10.1016/j.crmeth.2021.100003>.
- Muntel, Jan, Tejas Gandhi, Lynn Verbeke, Oliver M. Bernhardt, Tobias Treiber, Roland Bruderer, and Lukas Reiter. 2019. "Surpassing 10 000 Identified and Quantified Proteins in a Single Run by Optimizing Current LC-MS Instrumentation and Data Analysis Strategy." *Molecular Omics* 15 (5): 348–60. <https://doi.org/10.1039/C9MO00082H>.
- Nesvizhskii, Alexey I. 2010. "A Survey of Computational Methods and Error Rate Estimation Procedures for Peptide and Protein Identification in Shotgun Proteomics." *Journal of Proteomics* 73 (11): 2092–2123. <https://doi.org/10.1016/j.jprot.2010.08.009>.
- Nesvizhskii, Alexey I., and Ruedi Aebersold. 2005. "Interpretation of Shotgun Proteomic Data." *Molecular & Cellular Proteomics* 4 (10): 1419–40. <https://doi.org/10.1074/mcp.R500012-MCP200>.
- Nesvizhskii, Alexey I, Olga Vitek, and Ruedi Aebersold. 2007. "Analysis and Validation of Proteomic Data Generated by Tandem Mass Spectrometry." *Nature Methods* 4 (10): 787–97. <https://doi.org/10.1038/nmeth1088>.
- Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'textquotesingle Alché-Buc, E. Fox, and R. Garnett, 8024–35. Curran Associates, Inc. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12 (85): 2825–30.
- Rosenberger, George, Isabell Bludau, Uwe Schmitt, Moritz Heusel, Christie L Hunter, Yansheng Liu, Michael J MacCoss, et al. 2017. "Statistical Control of Peptide and Protein Error Rates in Large-Scale Targeted Data-Independent Acquisition Analyses." *Nature Methods* 14 (9): 921–27. <https://doi.org/10.1038/nmeth.4398>.
- Röst, Hannes L, George Rosenberger, Pedro Navarro, Ludovic Gillet, Saša M Miladinović, Olga T Schubert, Witold Wolski, et al. 2014. "OpenSWATH Enables Automated, Targeted Analysis of Data-Independent Acquisition MS Data." *Nature Biotechnology* 32 (3): 219–23. <https://doi.org/10.1038/nbt.2841>.
- Santos, Alberto, Ana R. Colaço, Annelaura B. Nielsen, Lili Niu, Philipp E. Geyer, Fabian Coscia, Nicolai J Wewer Albrechtsen, Filip Mundt, Lars Juhl Jensen, and Matthias Mann. 2020. "Clinical Knowledge Graph Integrates Proteomics Data into Clinical Decision-Making." Preprint. Bioinformatics. <https://doi.org/10.1101/2020.05.09.084897>.
- Savitski, Mikhail M., Mathias Wilhelm, Hannes Hahne, Bernhard Kuster, and Marcus Bantscheff. 2015. "A Scalable Approach for Protein False Discovery Rate Estimation in Large Proteomic Data Sets." *Molecular & Cellular Proteomics* 14 (9): 2394–2404. <https://doi.org/10.1074/mcp.M114.046995>.
- SciPy 1.0 Contributors, Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, et al. 2020. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods* 17 (3): 261–72. <https://doi.org/10.1038/s41592-019-0686-2>.

bioRxiv preprint doi: <https://doi.org/10.1101/2021.07.23.453379>; this version posted July 26, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- Senko, Michael W., Steven C. Beu, and Fred W. McLafferty. 1995. "Determination of Monoisotopic Masses and Ion Populations for Large Biomolecules from Resolved Isotopic Distributions." *Journal of the American Society for Mass Spectrometry* 6 (4): 229–33. [https://doi.org/10.1016/1044-0305\(95\)00017-8](https://doi.org/10.1016/1044-0305(95)00017-8).
- Teleman, Johan, Aakash Chawade, Marianne Sandin, Fredrik Levander, and Johan Malmström. 2016. "Dinosaur: A Refined Open-Source Peptide MS Feature Detector." *Journal of Proteome Research* 15 (7): 2143–51. <https://doi.org/10.1021/acs.jproteome.6b00016>.
- Teleman, Johan, Hannes L. Röst, George Rosenberger, Uwe Schmitt, Lars Malmström, Johan Malmström, and Fredrik Levander. 2015. "DIANA—Algorithmic Improvements for Analysis of Data-Independent Acquisition MS Data." *Bioinformatics* 31 (4): 555–62. <https://doi.org/10.1093/bioinformatics/btu686>.
- The, Matthew, Michael J. MacCoss, William S. Noble, and Lukas Käll. 2016. "Fast and Accurate Protein False Discovery Rates on Large-Scale Proteomics Data Sets with Percolator 3.0." *Journal of The American Society for Mass Spectrometry* 27 (11): 1719–27. <https://doi.org/10.1007/s13361-016-1460-7>.
- Torun, Furkan M., Sebastian Virreira Winter, Sophia Doll, Felix M. Riese, Artem Vorobyev, Johannes B. Mueller-Reif, Philipp E. Geyer, and Maximilian T. Strauss. 2021. "Transparent Exploration of Machine Learning for Biomarker Discovery from Proteomics and Omics Data." Preprint. Biochemistry. <https://doi.org/10.1101/2021.03.05.434053>.
- Tyanova, Stefka, Tikira Temu, Pavel Sinitcyn, Arthur Carlson, Marco Y. Hein, Tamar Geiger, Matthias Mann, and Jürgen Cox. 2016. "The Perseus Computational Platform for Comprehensive Analysis of (Prote)Omics Data." *Nature Methods* 13 (9): 731–40. <https://doi.org/10.1038/nmeth.3901>.
- Välikangas, Tommi, Tomi Suomi, and Laura L. Elo. 2017. "A Comprehensive Evaluation of Popular Proteomics Software Workflows for Label-Free Proteome Quantification and Imputation." *Briefings in Bioinformatics*, May. <https://doi.org/10.1093/bib/bbx054>.
- Vizcaino, Juan A., Eric W. Deutsch, Rui Wang, Attila Csordas, Florian Reisinger, Daniel Ríos, José A. Duanes, et al. 2014. "ProteomeXchange Provides Globally Coordinated Proteomics Data Submission and Dissemination." *Nature Biotechnology* 32 (3): 223–26. <https://doi.org/10.1038/nbt.2839>.
- Wen, Bo, Wen-Feng Zeng, Yuxing Liao, Zhiao Shi, Sara R. Savage, Wen Jiang, and Bing Zhang. 2020. "Deep Learning in Proteomics." *PROTEOMICS* 20 (21–22): 1900335. <https://doi.org/10.1002/pmic.201900335>.
- Wenger, Craig D., and Joshua J. Coon. 2013. "A Proteomics Search Algorithm Specifically Designed for High-Resolution Tandem Mass Spectra." *Journal of Proteome Research* 12 (3): 1377–86. <https://doi.org/10.1021/pr301024c>.
- Wilhelm, Mathias, Marc Kirchner, Judith A.J. Steen, and Hanno Steen. 2012. "Mz5: Space- and Time-Efficient Storage of Mass Spectrometry Data Sets." *Molecular & Cellular Proteomics* 11 (1): O111.011379. <https://doi.org/10.1074/mcp.O111.011379>.
- Zeng, Wen-Feng. 2021. *Jalew188/PyRawDataReader: PyRawDataReader v0.1* (version v0.1). Zenodo. <https://doi.org/10.5281/ZENODO.5053708>.
- Zhang, Fangfei, Weigang Ge, Guan Ruan, Xue Cai, and Tiannan Guo. 2020. "Data-Independent Acquisition Mass Spectrometry-Based Proteomics and Software Tools: A Glimpse in 2020." *PROTEOMICS* 20 (17–18): 1900276. <https://doi.org/10.1002/pmic.201900276>.

### 3.6. Article 6: diaPASEF: parallel accumulation – serial fragmentation combined with data-independent acquisition

Authors: Florian Meier<sup>1,2</sup>, Andreas-David Brunner<sup>1</sup>, Max Frank<sup>3</sup>, Annie Ha<sup>3</sup>, Isabell Bludau<sup>1</sup>, **Eugenia Voytik**<sup>1</sup>, Stephanie Kaspar-Schoenefeld<sup>4</sup>, Markus Lubeck<sup>4</sup>, Oliver Raether<sup>4</sup>, Nicolai Bache<sup>5</sup>, Ruedi Aebersold<sup>6,7</sup>, Ben C. Collins<sup>6,8</sup>, Hannes L. Röst<sup>3</sup> and Matthias Mann<sup>1,9</sup>

<sup>1</sup> Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany.

<sup>2</sup> Functional Proteomics, Jena University Hospital, Jena, Germany.

<sup>3</sup> Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, Canada.

<sup>4</sup> Bruker Daltonik GmbH, Bremen, Germany.

<sup>5</sup> Evosep Biosystems, Odense, Denmark.

<sup>6</sup> Department of Biology, Institute of Molecular Systems Biology, ETH Zurich, Zurich, Switzerland.

<sup>7</sup> Faculty of Science, University of Zurich, Zurich, Switzerland.

<sup>8</sup> School of Biological Sciences, Queen's University of Belfast, Belfast, UK.

<sup>9</sup> NNF Center for Protein Research, University of Copenhagen, Copenhagen, Denmark.

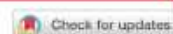
Published in *Nature Methods* (2020).

The concept of data-independent acquisition (DIA) was introduced more than 15 years ago, but recent developments have made it superior to alternatives in a wide range of proteomics applications. The main attractions of DIA are its high data completeness and a wide dynamic range. This acquisition scheme ensures that the selection windows collectively cover the entire  $m/z$  range of interest and that every peptide precursor is isolated and fragmented in every acquisition cycle. However, the overall ion sampling efficiency at the mass-selective quadrupole for conventional DIA methods is as a matter principle limited to 1-3% of all available ions.

In a joint effort of the Aebersold, Röst and Mann groups, in this study we have combined the recently introduced PASEF principle with DIA to overcome this fundamental limitation (13, 125). We employed this novel acquisition method, called diaPASEF, on a trapped ion mobility mass spectrometer (a timsTOF Pro instrument from Bruker), which provides an additional ion mobility dimension of separation. Using the correlation of mass and ion mobility, we acquired up to 100% of the peptide fragment ion current in low-complexity samples. We demonstrate the performance of diaPASEF in typical proteomics experiments, such as single two-hour runs of the HeLa proteome analysis, achieving deep proteome coverage of more than 7,000 proteins. Applying the diaPASEF scan mode to analyze very low sample amounts, we detected exceptional coverage of 4,000 quantified proteins from only 10 ng samples, highlighting the intrinsic high sensitivity of diaPASEF on the TIMS-QTOF setup.

I was involved in data access, data analysis and visualization of timsTOF data in this project.





# diaPASEF: parallel accumulation-serial fragmentation combined with data-independent acquisition

Florian Meier<sup>1,2</sup>, Andreas-David Brunner<sup>1</sup>, Max Frank<sup>3</sup>, Annie Ha<sup>3</sup>, Isabell Bludau<sup>1</sup>, Eugenia Voytik<sup>1</sup>, Stephanie Kaspar-Schoenefeld<sup>4</sup>, Markus Lubeck<sup>4</sup>, Oliver Raether<sup>4</sup>, Nicolai Bache<sup>5</sup>, Ruedi Aebersold<sup>6,7</sup>, Ben C. Collins<sup>8,9</sup>, Hannes L. Röst<sup>3</sup> and Matthias Mann<sup>1,9</sup>

**Data-independent acquisition modes isolate and concurrently fragment populations of different precursors by cycling through segments of a predefined precursor  $m/z$  range. Although these selection windows collectively cover the entire  $m/z$  range, overall, only a few per cent of all incoming ions are isolated for mass analysis. Here, we make use of the correlation of molecular weight and ion mobility in a trapped ion mobility device (timsTOF Pro) to devise a scan mode that samples up to 100% of the peptide precursor ion current in  $m/z$  and mobility windows. We extend an established targeted data extraction workflow by inclusion of the ion mobility dimension for both signal extraction and scoring and thereby increase the specificity for precursor identification. Data acquired from whole proteome digests and mixed organism samples demonstrate deep proteome coverage and a high degree of reproducibility as well as quantitative accuracy, even from 10 ng sample amounts.**

Mass spectrometry-based proteomics, like other omics technologies, aims for an unbiased, comprehensive and quantitative description of the system under investigation<sup>1–3</sup>. Proteomics workflows have become increasingly successful in the characterization of complex proteomes in great depth<sup>4,5</sup>. Application to large sample cohorts requires a high degree of reproducibility and data completeness, which makes data-independent acquisition (DIA) schemes particularly attractive<sup>6,7</sup>. In contrast to data-dependent acquisition (DDA), in which particular precursors are sequentially selected, in DIA, groups of ions are recursively isolated by the quadrupole and concurrently fragmented to generate convoluted fragment ion spectra composed of fragments from many different precursors<sup>8–10</sup>. Although DIA guarantees that each precursor in a predefined mass range is fragmented once per cycle, spectral complexity poses a great challenge to subsequent analysis<sup>11</sup>. Narrower isolation windows result in less complex spectra, but this increases the total number of windows and hence the DIA cycle times needed to cover the entire mass range<sup>9,12,13</sup>. Moreover, as every precursor is isolated only once per cycle, the ion sampling efficiency at the mass-selective quadrupole for DIA methods is limited to 1–3% with typical schemes of 32 or 64 windows.

The addition of ion mobility separation to the chromatographic and mass separation should increase sensitivity and reduce spectral complexity<sup>14–17</sup>. The trapped ion mobility spectrometer (TIMS) is a particularly compact mobility analyzer in which ions are captured in an ion tunnel, between the opposing forces of the gas flow from the source and the counteracting electric field<sup>18–20</sup>. Trapped ions are then sequentially released as a function of their mobility as the electric potential is lowered. In proteomics, ramp times typically range from 50 to 100 ms, in between chromatographic peak

widths (seconds) and the time-of-flight (TOF) spectral acquisition (approximately 100  $\mu$ s per pulse). In a TIMS-quadrupole-TOF configuration, the mobility separation can be synchronized with the quadrupole mass selection in a method termed parallel accumulation-serial fragmentation (PASEF)<sup>21</sup>. Given that multiple precursors are mass selected and fragmented during a single TIMS scan, PASEF achieves a more than tenfold increase in sequencing speed in DDA, without the loss of sensitivity that is otherwise inherent in very fast fragmentation cycles<sup>22,23</sup>. This is because the precursor ion current is compressed into narrow ion mobility peaks and, with two TIMS in series, ions can be accumulated and mobility analyzed in parallel<sup>24</sup>.

Here, we investigate whether the PASEF principle can be extended to DIA, which would combine the advantages of this acquisition method with the inherent efficiency of PASEF. To realize this vision, we modified the mass spectrometer to support ‘diaPASEF’ acquisition cycles. Building on open-source software<sup>25</sup>, we perform targeted extraction of fragment ion traces from the four-dimensional data space for peptide quantification. We explore the performance of the diaPASEF principle in typical proteomics applications such as single-run proteome analysis and label-free quantification, as well as in the characterization of very limited sample amounts.

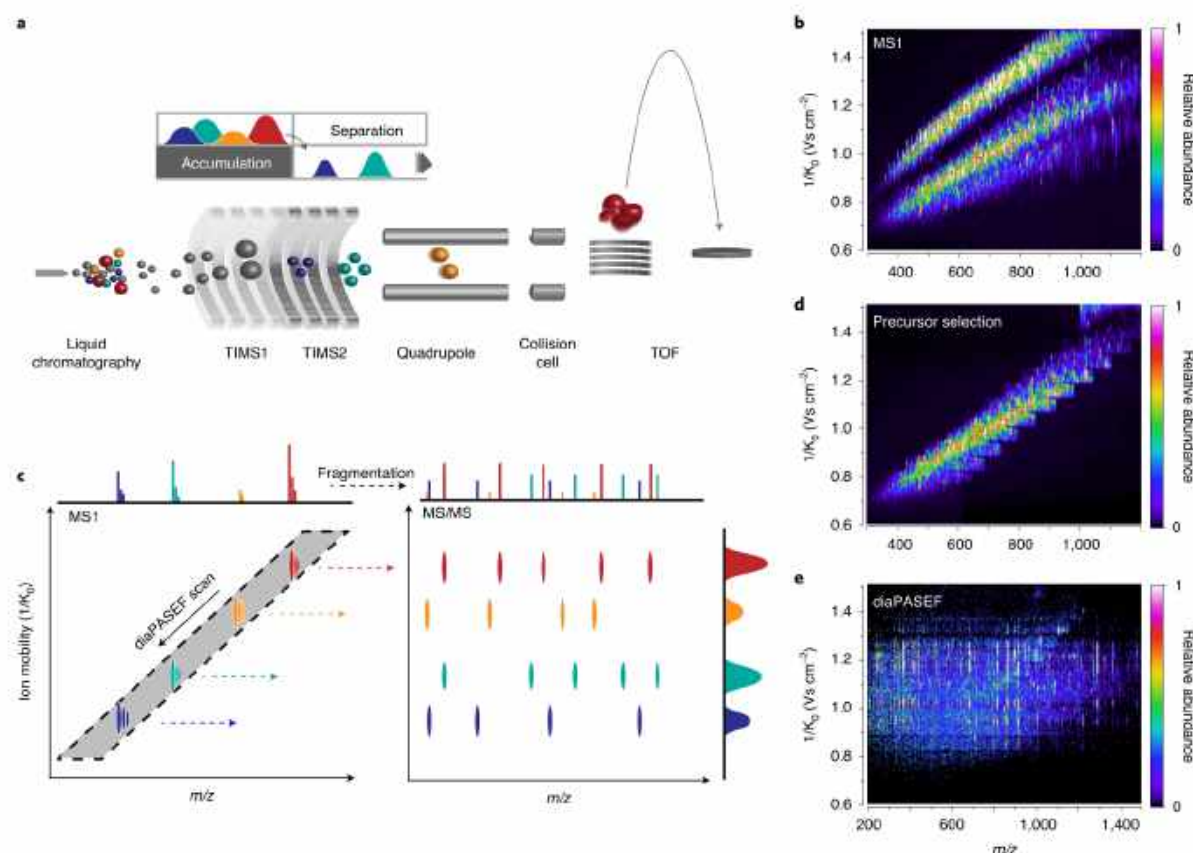
## Results

**The diaPASEF principle.** In the timsTOF Pro instrument (Bruker Daltonik), peptides separated by liquid chromatography are ionized, introduced into the mass spectrometer and immediately trapped in a dual TIMS device (Fig. 1a). Mobility-separated ions reach the orthogonal accelerator, from which rapid TOF pulses result in high-resolution mass spectra (resolution >35,000 over

<sup>1</sup>Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany. <sup>2</sup>Functional Proteomics, Jena University Hospital, Jena, Germany. <sup>3</sup>Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, Canada. <sup>4</sup>Bruker Daltonik GmbH, Bremen, Germany. <sup>5</sup>EvoSep Biosystems, Odense, Denmark. <sup>6</sup>Department of Biology, Institute of Molecular Systems Biology, ETH Zurich, Zurich, Switzerland. <sup>7</sup>Faculty of Science, University of Zurich, Zurich, Switzerland. <sup>8</sup>School of Biological Sciences, Queen's University of Belfast, Belfast, UK. <sup>9</sup>NNF Center for Protein Research, University of Copenhagen, Copenhagen, Denmark. <sup>10</sup>E-mail: ben.collins@qub.ac.uk; hannes.rost@utoronto.ca; mmann@biochem.mpg.de

## ARTICLES

## NATURE METHODS



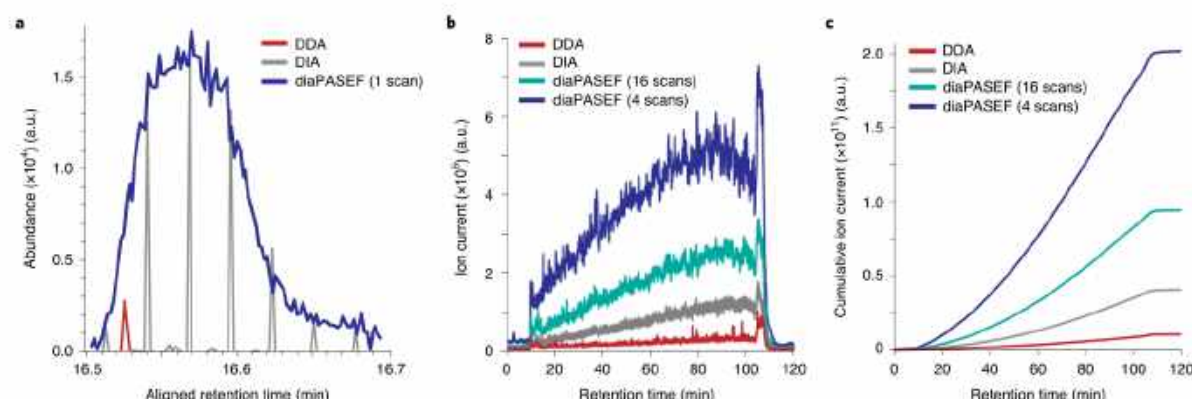
**Fig. 1 | The diaPASEF acquisition method.** **a**, Schematic ion path of the timsTOF Pro mass spectrometer. **b**, Correlation of ion mobility and  $m/z$  in a tryptic digest of HeLa cell lysate. **c**, In diaPASEF, the quadrupole isolation window (gray) is dynamically positioned as a function of ion mobility (arrow). In a single TIMS scan, ions from the selected mass ranges are fragmented to record ion mobility-resolved MS/MS spectra of all precursors. **d**, Implementation of diaPASEF precursor selection with a stepped quadrupole isolation scheme. **e**, Representative example of a single diaPASEF scan with the precursor selection scheme from **d** (Supplementary Fig. 1).

the entire mass range). For peptide ions of a given charge state, ion mobilities and masses are correlated (Fig. 1b). We reasoned that this feature could be used to isolate precursor mass windows for DIA without losing the ions outside the respective windows, in contrast to other DIA acquisition schemes. Given that low-mobility (typically high  $m/z$ ) ions are trapped near the TIMS exit, they are released first, and the mass-selective quadrupole therefore needs to be first positioned at high  $m/z$ . As higher mobility (typically decreasing  $m/z$ ) ions are sequentially released from the TIMS, the quadrupole mass isolation window should slide down to lower  $m/z$  values to fully transmit the ion cloud (Fig. 1c). To approximate this ideal diaPASEF scan, we stepped the isolation window as a function of TIMS ramp time (Methods), covering the vast majority of precursors of the  $2^+$  and  $3^+$  charge states (Fig. 1d and Supplementary Fig. 1). Implementation of this principle required firmware able to synchronize collision energies with the mass selection (Methods). Note that the fragment ions in each DIA window are detected at the exact ion mobility position of the precursor (Fig. 1e). Over the chromatographic elution of a precursor, the intensities of its fragments follow the precursor intensity in time ( $z$  direction). The signal traced out by the set of fragments of an individual precursor is a set of very flat ellipsoids ( $x$  or  $m/z$  dimension), spreading in the ion

mobility direction ( $y$  direction) and elongated in the retention time dimension ( $z$  dimension). For the entire experiment, this leads to a data cuboid in four-dimensional space, containing all fragment ions of all precursors over the entire elution time, with signal intensity as the fourth dimension.

**Quantification of the increase in ion sampling efficiency.** To explore the diaPASEF principle in practice, we measured a tryptic digest of BSA and compared the signals obtained across the DDA, DIA and diaPASEF acquisition methods. As a typical example, the peptide DLGEEHFK eluted over 9 s (Fig. 2a). In DDA, the doubly charged precursor was accumulated at the beginning of the elution peak once for 100 ms before fragmentation. This is approximately 1% of the total elution time and much less than 1% of the entire precursor ion population, as estimated by the relative peak area. In DIA, with a comparably fast cycle time of 1.6 s, the peptide was fragmented seven times over its elution profile. This is sufficient to reconstruct the chromatographic peak shape, but still captured only a small proportion of the total ion signal (less than 5%). By contrast, the diaPASEF scheme (Supplementary Fig. 2) sampled the fragments in each scan for a total of more than 100 times, which resulted in a nearly complete record of the fragments at every time





**Fig. 2 | Efficiency of different data acquisition methods.** **a**, Extracted fragment ion chromatograms of the  $y_1$  ion of the doubly charged DLGEEHFK peptide precursor in a 45 min liquid chromatography–mass spectrometry analysis of BSA digest acquired with typical DDA and DIA methods as well as with a close to 100% duty cycle diaPASEF method shown in Supplementary Fig. 2. a.u., arbitrary units. **b**, Detected ion current from multiply charged precursors in single-run analyses of 200 ng HeLa digest acquired with DDA, DIA and two diaPASEF schemes (Supplementary Figs. 3,4). To extract the ion current after quadrupole isolation, no collision energy was applied and the ion current in the expected peptide space was summed for each TIMS scan. The plot shows the rolling average of 60 TIMS scans. **c**, Same as in **b**, but for cumulative ion current.

point (96% efficiency in terms of acquisition time because of the full scans acquired in between diaPASEF cycles).

We next studied the ion sampling efficiency for a HeLa cell tryptic digest. To address the very high density of fragment ions in the data cuboid, we chose a scheme with four diaPASEF scans, each isolating approximately one-fourth of all precursors with 50  $m/z$  isolation windows, and another scheme with 16 diaPASEF scans and 25  $m/z$  isolation windows (Supplementary Figs. 3,4). To compare acquisition schemes, no collision energy was applied and we extracted the total ion current of isolated precursors in the expected peptide space in the  $m/z$ –ion mobility plane (Methods). In DIA, the sampled fraction of the ion current was approximately three-fold higher than in DDA, whereas the four-scan diaPASEF scheme further increased the accumulated peptide ion current by a factor of five compared with DIA (Fig. 2b,c). We conclude that the diaPASEF principle yields the expected increase in data acquisition efficiency in both simple and complex proteomes.

**Targeted data extraction in four dimensions.** To identify and quantify peptides from this novel data structure, we developed Mobi-DIK (ion mobility DIA analysis kit; Fig. 3a). The workflow is based on the targeted extraction of sets of fragment ions of a specific precursor from the acquired dataset over chromatographic elution time, followed by statistical scoring. Mobi-DIK extends this targeted data analysis principle for DIA<sup>28</sup> (as implemented in the OpenSWATH software suite<sup>29</sup>) to diaPASEF. First, ion mobility-enabled spectral libraries are generated from data-dependent PASEF runs using, for instance, the MaxQuant<sup>37,38</sup> output. The spectral library is processed using OpenMS tools<sup>39,40</sup>, which we here extended to support ion mobility. Calibration between the assay library and experimental data is automatically performed in  $m/z$ , retention time and ion mobility dimensions using a set of high-confidence peptides (Methods). The Mobi-DIK package uses the vendor interface to query diaPASEF raw data, convert them to mzML files, and link the isolation windows to individual TOF scans. The algorithm then uses the targeted extraction paradigm for DIA data to construct four-dimensional data cuboids with a user-defined width in  $m/z$  (ppm), retention time (s) and ion mobility ( $V \cdot s \cdot cm^{-2}$ ). These are projected onto the retention time and ion mobility axes to obtain fragment ion chromatograms and mobilograms for each

precursor-to-fragment transition in the spectral library. Restricting the ion mobility extraction width removes signals from co-eluting peptides in the same precursor mass window that have different ion mobility (Fig. 3b and Supplementary Figs. 5–7). Through investigation of transitions in a single-run analysis of HeLa digest, we found that when the ion mobility extraction window was narrowed to 0.06  $V \cdot s \cdot cm^{-2}$ , this resulted in an average fourfold increase in the signal-to-noise ratios (Supplementary Fig. 8). Note that the acquisition scheme already removes interfering ions with very different ion mobility such as singly charged species, therefore the true gain in signal-to-noise ratio (compared with the respective value of a DIA experiment without ion mobility) is even higher.

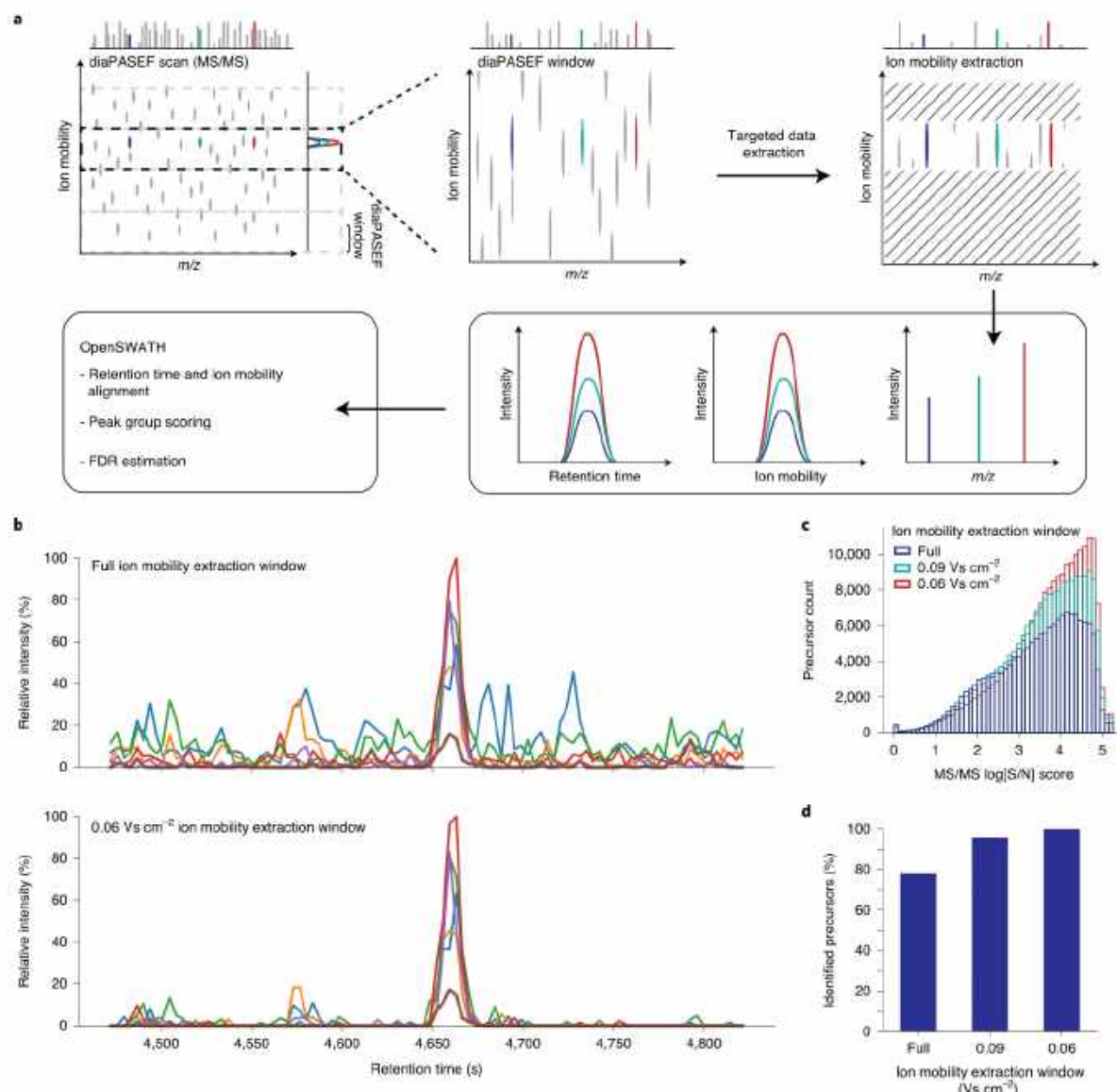
From all projected traces, we next pick peak groups along the chromatographic dimension using established OpenSWATH modules. This step selects putative peak candidates and scores them based on their chromatographic co-elution, goodness of library match and correlation with the precursor profile<sup>25</sup>. For Mobi-DIK, we extended these modules by ion mobility scores. Through use of the high precision of TIMS ion mobility measurements (<1% in replicates of complex samples<sup>12</sup>), a discriminatory score is computed based on the difference between the library and the experimental ion mobility. Additionally, we extract full ion mobilograms for each fragment ion to score the mobility peak shape as well as the peak consistency between all fragment ions. In line with the increased signal-to-noise ratios, the corresponding ‘MS/MS signal-to-noise’ score increased proportionately with narrower ion mobility extraction windows (Fig. 3c). As a result, in a single-run analysis of a full proteome digest (see below), targeted extraction in the ion mobility dimension (combined with ion mobility-aware scoring) increased peptide identifications by 22% compared with a naive analysis (Fig. 3d).

**Single-run proteome analysis.** To investigate diaPASEF in a typical DIA experiment, we first built a project-specific library from 24 high-pH reversed-phase peptide fractions of a HeLa digest with data-dependent PASEF, which consisted of 135,671 target precursors and 9,140 target proteins. For sample amounts on column of at least 200 ng and liquid chromatography–mass spectrometry runs of 120 min, we reasoned that a diaPASEF method with a somewhat lower duty cycle, but higher precursor selectivity, should



## ARTICLES

## NATURE METHODS

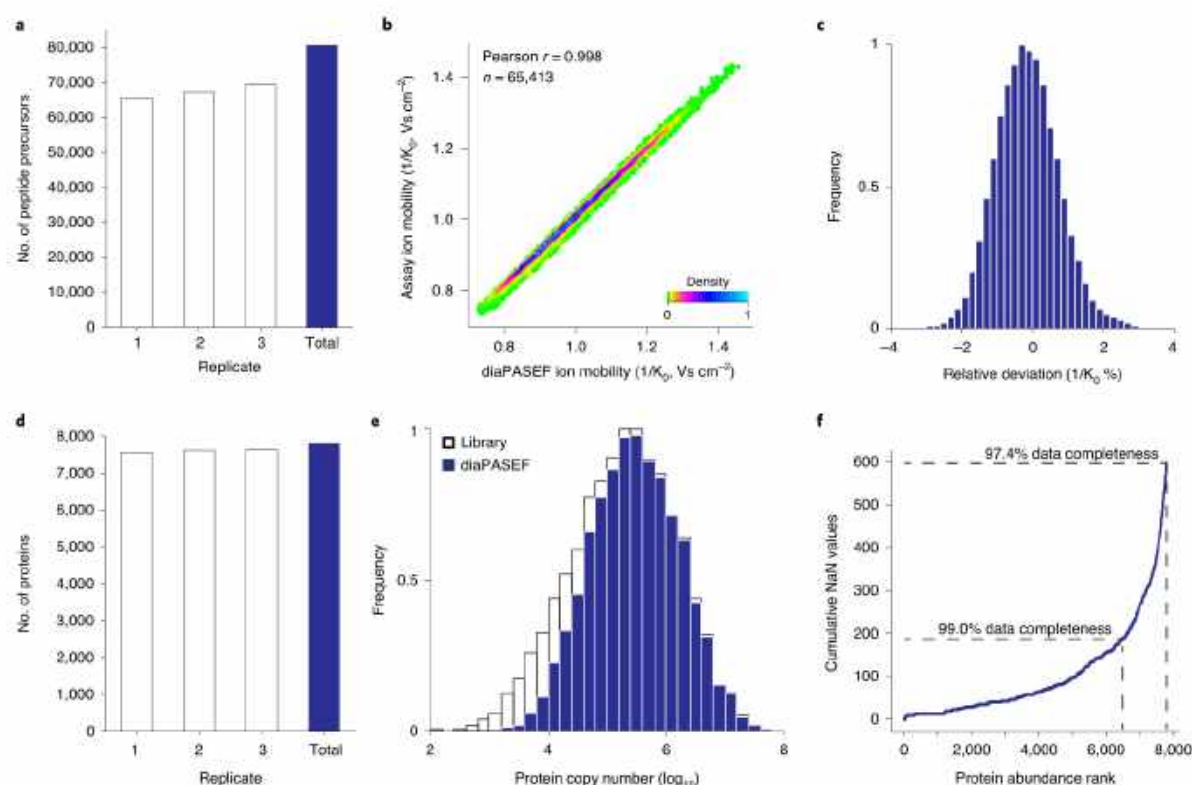


**Fig. 3 | Ion mobility-aware targeted data extraction.** **a**, Steps in the Mobi-DIK workflow to extract fragment ion chromatograms from diaPASEF scans with restricted ion mobility windows and ion mobility-enhanced peak group scoring in OpenSWATH. Colors (red, blue and cyan) indicate fragment ions from a precursor of interest; gray indicates background signals. **b**, Example fragment ion chromatograms of DGLIIGVHSK (color-coded) extracted with (bottom panel) and without (top panel) restriction in the ion mobility dimension from a single-run diaPASEF experiment of HeLa digest. Fragment ions:  $y_{41}$ , orange;  $y_{50}$ , green;  $y_{51}$ , red;  $y_{52}$ , purple;  $y_{53}$ , brown;  $b_{41}$ , blue. **c**, Removal of interfering signals from co-eluting precursors in the same diaPASEF window in triplicate diaPASEF analysis of a HeLa digest. Histograms of the MS/MS signal-to-noise (S/N) scores of identified precursors for different ion mobility extraction windows.  $n = 158,603$  (full), 194,197 (0.09 Vs cm<sup>-2</sup>) and 202,218 (0.06 Vs cm<sup>-2</sup>). **d**, Percentage of detected peptide precursors in triplicate diaPASEF runs from **c** at an FDR of 1% as a function of the ion mobility extraction window.

be beneficial. We devised a method with four windows in each 100 ms diaPASEF scan and 25  $m/z$  precursor isolation windows (Supplementary Fig. 4). Eight of these scans covered the diagonal scan line for doubly charged peptides in the  $m/z$ -ion mobility plane and a second parallel scan line ensured coverage of triply charged species. To reduce potential artifacts from reduced ion transmission at the edges of the diaPASEF windows, we overlapped these scan

lines in the ion mobility dimension (Supplementary Fig. 9). The theoretical coverage of library precursor ions was 99.5% and 92.1% for doubly and triply charged peptides in the analyzed  $m/z$  range 400–1,200, respectively.

In triplicate runs, we detected a total of 80,580 peptide precursors (with 1% precursor and protein false discovery rates (FDRs)), and on average 67,312 peptide precursors per run (Fig. 4a and



**Fig. 4 | Single-run HeLa proteome analysis with diaPASEF.** **a**, Number of peptide precursor ions in triplicate injections of 200 ng HeLa digest with 120 min gradients using the 16-scan diaPASEF scheme shown in Supplementary Fig. 4. **b**, Correlation of precursor ion mobility in a single diaPASEF run with that in the assay library. **c**, Relative deviation of ion mobility values in a single diaPASEF run from the precursor ion mobility in the library. **d**, Number of quantified proteins. **e**, Estimated copy numbers of proteins contained in the assay library and detected with diaPASEF in triplicate single runs. **f**, Cumulative number of missing protein quantification data points (NaN) in the three replicate injections as a function of decreasing protein abundance. Data completeness was calculated as the fraction of valid values in the (number of replicates  $\times$  abundance rank) matrix.

Supplementary Fig. 10). The ion mobility values in the diaPASEF runs were highly correlated with the library values ( $r > 0.99$ , Fig. 4b), and the median absolute deviation of the fragment ion mobility values in diaPASEF from those in the library runs was 0.6% (Fig. 4c). The median summed absolute fragment mass deviation was 6.6 ppm and the median absolute retention time deviation was 17 s. Together, these values define the precision of the position of each precursor and its fragments in the diaPASEF data cuboid.

Overall, 66,998 unique peptide sequences were identified at an FDR of 1%, from which 7,601 proteins per run on average and 7,800 proteins in total were found using only proteotypic peptides as mapped in the low-redundancy Swiss-Prot database and at a global protein FDR of 1% (Fig. 4d and Supplementary Fig. 11). The quantified proteins spanned a dynamic range of approximately four orders of magnitude, as estimated by protein copy numbers derived from the library (Fig. 4e). Of these, 7,348 proteins (94%) were quantified in all three replicates, 307 in two and only 145 proteins in a single replicate, resulting in a virtually complete data matrix (Fig. 4f) with a median coefficient of variation of 7.7%.

**Label-free quantification benchmark.** Next, we set up a two-proteome experiment. We spiked 200 ng HeLa samples with approximately 45 ng and 15 ng of a tryptic yeast digest, respectively, and measured both samples in triplicate single runs as above.

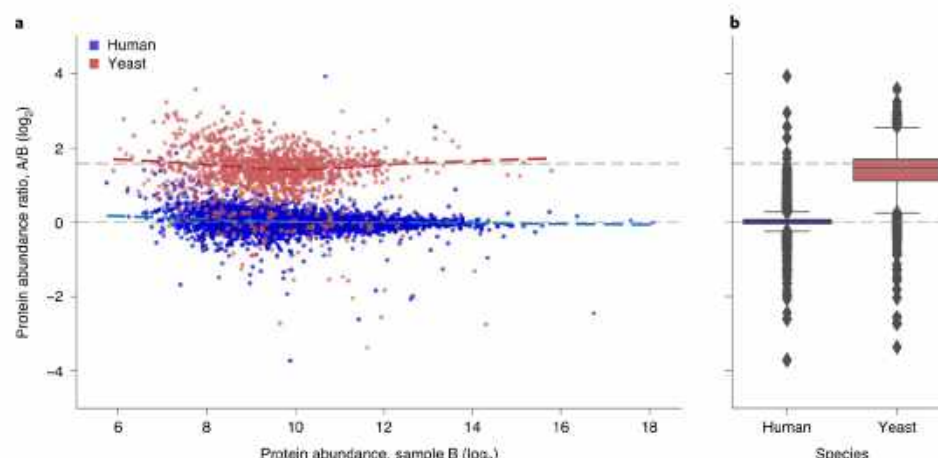
Mobi-DIK analysis using a combined human and yeast library quantified a total of 82,808 human and 7,483 yeast unique peptide sequences from 101,395 human and 7,992 yeast peak groups, for which 7,943 human and 2,250 yeast proteins were inferred. Although the low-abundance yeast spike-in constituted only 7% of the sample, we quantified 7,697 human and 1,394 yeast proteins in at least two replicates in both samples. Their protein abundance ratios split into two distinct populations according to the mixing ratios (median 2.7-fold, Fig. 5). In line with the quantitative precision demonstrated above, the human population clustered precisely around the 1:1 ratio throughout the full abundance range ( $\sigma(\log_2) = 0.22$ ). The low-abundance yeast spike-ins were quantified with a somewhat lower overall precision ( $\sigma(\log_2) = 0.70$ ), although quantitatively similar to human proteins in the same abundance range. We therefore conclude that the label-free diaPASEF workflow precisely and accurately quantifies changes in protein abundance.

**Adaptation of diaPASEF to high-throughput and high-sensitivity proteomics.** The diaPASEF schemes can be optimized to balance selectivity (narrower isolation windows), sensitivity (higher mass spectrometry efficiency, fewer diaPASEF scans) and precursor coverage (Fig. 6a). Fast chromatographic methods typically require shorter mass spectrometry cycle times to achieve a sufficient number of data points for accurate quantification. Hence, we devised



## ARTICLES

## NATURE METHODS



**Fig. 5 | Label-free protein quantification benchmark.** **a**, HeLa digest was spiked with approximately 45 ng (sample A) and 15 ng (sample B) yeast digest, and analyzed in triplicate 120 min diaPASEF single runs each, using the 16-scan diaPASEF scheme (Supplementary Fig. 4), log-transformed ratios are plotted as a function of protein abundance for  $n = 7,697$  human and  $n = 1,394$  yeast proteins. Dashed gray lines indicate the expected ratio. LOESS regression lines are dashed and colored by species. **b**, Boxplots of the data in **a**, showing the median ratio (center line), the 25th and 75th percentiles (lower and upper box limits, respectively), the 1.5x interquartile range (whiskers) and the outliers (diamonds).

an acquisition scheme that focused on a narrower precursor range with a 0.9 s cycle time (Supplementary Fig. 12). To test this scheme, we turned to a liquid chromatography system with fast turnaround times and predefined, standardized gradients for the analysis of 60, 100 and 200 samples per day (EvoSep One)<sup>11</sup>. In triplicate analysis of 200 ng HeLa with the 60 samples per day method (21 min gradient), we quantified on average 4,813 proteins per run and 5,183 in total with a median coefficient of variation of 5.8% (Fig. 6b,c). Remarkably, 4,255 proteins were quantified with a coefficient of variation of <20%. When the throughput was increased to 100 and 200 samples per day, more than 4,000 and 3,000 proteins in triplicate were still quantified, respectively. At 200 samples per day, the median coefficient of variation increased to only 10.3%, which indicates that an even faster diaPASEF method could be viable.

When the number of diaPASEF scans is lowered and the quadrupole isolation width is increased, diaPASEF can be tuned to utilize a higher fraction of the incoming ion beam and still achieve a high precursor selectivity because of the ion mobility separation. To demonstrate this concept, we analyzed only 10 ng of HeLa digest in triplicate 120 min single runs and used a diaPASEF scheme that samples approximately 25% of the ion current of a given precursor (Supplementary Fig. 3). Compared with the standard method, the high duty cycle increased the detected fragment ion signal on average by approximately fourfold and resulted in a more precise quantification of the common peptides, in particular for low-abundance peptides (Fig. 6d). Although the method covers a narrower precursor space, we quantified on average approximately 13,000 peptides with each method and, in effect, the high-sensitivity method extended the detection range of peptides approximately fourfold at the lower end (Fig. 6e). The standard diaPASEF method already quantified on average 3,538 proteins per injection of 10 ng HeLa digest, which highlights the intrinsic high sensitivity of diaPASEF and of the TIMS-QTOF setup. The high-sensitivity method further increased this to 3,835 proteins on average. Cumulatively, we quantified 4,310 proteins in triplicates of 10 ng injections, of which 3,909 were quantified in at least two replicates (Fig. 6f). The increased quantitative precision at the peptide level also translated into higher precision at the protein level, resulting in median coefficients of variation of 9.0% and 11.2% for

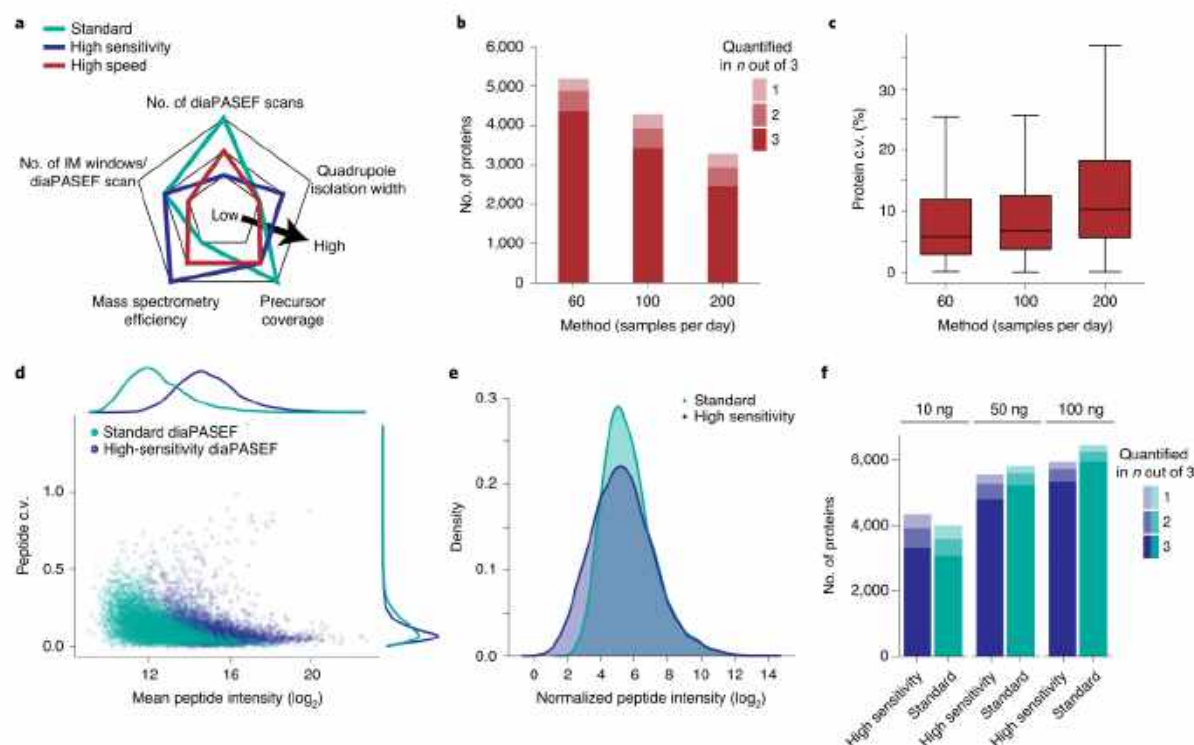
the high-sensitivity and the standard methods, respectively (3,132 and 2,690 proteins quantified with a coefficient of variation of <20%). However, at higher sample amounts, narrower quadrupole windows were more beneficial. With the high-sensitivity and the standard diaPASEF methods we quantified 4,755 and 4,833 proteins, respectively, from 50 ng samples with a coefficient of variation of <20% (median coefficients of variation of 5.1% and 7.3%), and 5,396 and 5,626 proteins, respectively, from 100 ng samples with a coefficient of variation of <20% (median coefficients of variation of 4.4% and 5.7%, respectively).

## Discussion

Here, we have developed and demonstrated a PASEF workflow in a TIMS-TOF mass spectrometer that implements the DIA principle. To make use of the correlation between the ion mobility and the  $m/z$  of peptides, precursors are trapped and then released in synchronization with the quadrupole position in our diaPASEF scheme, which results in almost complete sampling of the precursor ion beam. This is in contrast to DDA methods, which convert only a very small fraction (generally much less than 1%) of the incoming ion beam into fragments, and even to typical DIA workflows, which convert a few per cent of the ion beam at best. For less complex mixtures, we achieve close to 100% of the theoretical maximum, whereas for more complex mixtures, it was beneficial to use the quadrupole to decrease spectral complexity and increase selectivity, and thereby to some extent reduce the fraction of total available ions sampled. Note that results could be further improved by the use of brighter electrospray sources and the minimization of ion losses that may occur along the ion path through the instrument and during mass selection. On the predecessor QTOF instrument (Bruker impact II) we found a >80% ion transmission up to the collision cell and an overall detection probability of approximately 10% for ions transferred into the vacuum<sup>32</sup>, and an ion trapping efficiency of approximately 70% has been reported for the TIMS device<sup>33</sup>.

To extract information using spectral library-based targeted data analysis, we extended the OpenSWATH tool developed for DIA applications to efficiently make use of the ion mobility dimension for library matching, to provide full FDR control and excellent quantification.





**Fig. 6 | High-throughput and high duty cycle diaPASEF analysis.** **a**, Schematic of the parameter space in designing diaPASEF acquisition schemes. IM, ion mobility. **b**, Quantified proteins in *n* out of three replicate injections of 200 ng HeLa digest using different Evosep One liquid chromatography methods (Methods) and a rapid diaPASEF acquisition scheme (Supplementary Fig. 12). **c**, Coefficients of variation of protein abundances measured in at least two replicates. Boxplots show the median (center line), 25th and 75th percentiles (lower and upper box limits, respectively) and the 1.5x interquartile range (whiskers). *n* = 4,884 (60 samples per day), 3,940 (100 samples per day) and 2,939 proteins (200 samples per day). **d**, Mean peptide intensity and coefficient of variation of shared peptide precursors in triplicate injections of 10 ng HeLa digest with two different diaPASEF acquisition schemes (Supplementary Figs. 3,4) and a 120 min gradient. Kernel density estimates of peptide intensities and coefficients of variation are presented in the top traces and right traces, respectively. **e**, Peptide intensity distribution for both experiments in **d**, normalized to the most abundant peptide in each. **f**, Quantified proteins in *n* out of three replicate injections of 10 ng, 50 ng and 100 ng HeLa digest as in **d**.

Even in this first implementation, we achieved deep proteome coverage of more than 7,000 proteins in single, 2 h experiments from 200 ng HeLa peptide sample on column with a high degree of reproducibility. Our two-proteome experiment verifies that the quantitative accuracy of the method is in line with previous strategies even when substantially constrained by the lower loading amount of yeast (15 ng). Even more remarkably, we detected more than 4,000 proteins in triplicate injections of only 10 ng HeLa peptide mass on column. This result points to a perhaps unexpected advantage of diaPASEF, namely that the high ion sampling also fully translates into higher sensitivity. Likewise, the very short cycle time of our new scan mode was found to be advantageous for short gradients, which is an increasingly important attribute because large-scale biological and clinical studies require very large throughput. Given that DIA methods record chromatographic profiles for each fragment ion, they are also increasingly attractive for site-specific analysis of modified peptides<sup>23,24</sup>. With diaPASEF, such strategies could additionally benefit from the separation of positional isomers in the ion mobility dimension<sup>25</sup>. For the future, we imagine that both hardware and software can still be greatly optimized to further increase the amount and quality of the information contained in and extracted from the extremely rich four-dimensional diaPASEF data cuboids. For example, advanced

data acquisition schemes could sample the correlation of precursor mobility and *m/z* more precisely if the isolation window width is varied or the quadrupole is scanned rather than moved in discrete steps. Furthermore, we note that applications of diaPASEF are not restricted to peptides but could equally well be extended to metabolites, lipids or other compound classes<sup>23</sup>.

#### Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at <https://doi.org/10.1038/s41592-020-00998-0>.

Received: 24 February 2020; Accepted: 15 October 2020;  
Published online: 30 November 2020

#### References

- Altelaar, A. F. M., Munoz, J. & Heck, A. J. R. Next-generation proteomics: towards an integrative view of proteome dynamics. *Nat. Rev. Genet.* **14**, 35–48 (2012).
- Larance, M. & Lamond, A. I. Multidimensional proteomics for cell biology. *Nat. Rev. Mol. Cell Biol.* **16**, 269–280 (2015).

## ARTICLES

## NATURE METHODS

3. Aebersold, R. & Mann, M. Mass-spectrometric exploration of proteome structure and function. *Nature* **537**, 347–355 (2016).
4. Bekker-Jensen, D. B. et al. An optimized shotgun strategy for the rapid generation of comprehensive human proteomes. *Cell Syst.* **4**, 587–599.e4 (2017).
5. Wang, D. et al. A deep proteome and transcriptome abundance atlas of 29 healthy human tissues. *Mol. Syst. Biol.* **15**, e8503 (2019).
6. Röst, H. L., Malmström, L. & Aebersold, R. Reproducible quantitative proteotype data matrices for systems biology. *Mol. Biol. Cell* **26**, 3926–3931 (2015).
7. Doerr, A. DIA mass spectrometry. *Nat. Methods* **12**, 35 (2015).
8. Chapman, J. D., Goodlett, D. R. & Musselton, C. D. Multiplexed and data-independent tandem mass spectrometry for global proteome profiling. *Mass Spectrom. Rev.* **33**, 452–470 (2014).
9. Ludwig, C. et al. Data-independent acquisition-based SWATH-MS for quantitative proteomics: a tutorial. *Mol. Syst. Biol.* **14**, e8126 (2018).
10. Gillet, L. C., Leitner, A. & Aebersold, R. Mass spectrometry applied to bottom-up proteomics: entering the high-throughput era for hypothesis testing. *Annu. Rev. Anal. Chem.* **9**, 449–472 (2016).
11. Bilhao, A. et al. Processing strategies and software solutions for data-independent acquisition in mass spectrometry. *Proteomics* **15**, 964–980 (2015).
12. Bruderer, R. et al. Optimization of experimental parameters in data-independent mass spectrometry significantly increases depth and reproducibility of results. *Mol. Cell. Proteom.* **16**, 2296–2309 (2017).
13. Pino, L. K., Just, S. C., MacCoss, M. J. & Searle, B. C. Acquiring and analyzing data independent acquisition proteomics experiments without spectrum libraries. *Mol. Cell. Proteom.* **19**, 1088–1103 (2020).
14. McLean, J. A., Ruotolo, B. T., Gillig, K. J. & Russell, D. H. Ion mobility–mass spectrometry: a new paradigm for proteomics. *Int. J. Mass Spectrom.* **240**, 301–315 (2005).
15. Distler, U. et al. Drift time-specific collision energies enable deep-coverage data-independent acquisition proteomics. *Nat. Methods* **11**, 167–170 (2014).
16. Helm, D. et al. Ion mobility tandem mass spectrometry enhances performance of bottom-up proteomics. *Mol. Cell. Proteom.* **13**, 3709–3715 (2014).
17. Ewing, M. A., Glover, M. S. & Clemmer, D. E. Hybrid ion mobility and mass spectrometry as a separation tool. *J. Chromatogr. A* **1439**, 3–25 (2016).
18. Fernandez-Lima, F. A., Kaplan, D. A. & Park, M. A. Note: Integration of trapped ion mobility spectrometry with mass spectrometry. *Rev. Sci. Instrum.* **82**, 126106 (2011).
19. Fernandez-Lima, F., Kaplan, D. A., Suetering, J. & Park, M. A. Gas-phase separation using a trapped ion mobility spectrometer. *Int. J. Ion Mobil. Spectrom.* **14**, 93–98 (2011).
20. Ridgeway, M. E., Lubeck, M., Jordens, J., Mann, M. & Park, M. A. Trapped ion mobility spectrometry: a short review. *Int. J. Mass Spectrom.* **425**, 22–35 (2018).
21. Meier, F. et al. Parallel accumulation–serial fragmentation (PASEF): multiplying sequencing speed and sensitivity by synchronized scans in a trapped ion mobility device. *J. Proteome Res.* **14**, 5378–5387 (2015).
22. Meier, F. et al. Online parallel accumulation–serial fragmentation (PASEF) with a novel trapped ion mobility mass spectrometer. *Mol. Cell. Proteom.* **17**, 2534–2545 (2018).
23. Vasilopoulou, C. G. et al. Trapped ion mobility spectrometry and PASEF enable in-depth lipidomics from minimal sample amounts. *Nat. Commun.* **11**, 331 (2020).
24. Silveira, J. A., Ridgeway, M. E., Laukien, F. H., Mann, M. & Park, M. A. Parallel accumulation for 100% duty cycle trapped ion mobility–mass spectrometry. *Int. J. Mass Spectrom.* **413**, 168–175 (2017).
25. Röst, H. L. et al. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nat. Biotechnol.* **32**, 219–223 (2014).
26. Gillet, L. C. et al. Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis. *Mol. Cell. Proteom.* **11**, O111.016717 (2012).
27. Cox, J. & Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.* **26**, 1367–1372 (2008).
28. Prianchikov, N. et al. MaxQuant software for ion mobility enhanced shotgun proteomics. *Mol. Cell. Proteom.* **19**, 1058–1069 (2020).
29. Rosenberger, G. et al. Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses. *Nat. Methods* **14**, 921–927 (2017).
30. Röst, H. L. et al. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nat. Methods* **13**, 741–748 (2016).
31. Bache, N. et al. A novel LC system embeds analytes in pre-formed gradients for rapid, ultra-robust proteomics. *Mol. Cell. Proteom.* **17**, 2284–2296 (2018).
32. Beck, S. et al. The impact II, a very high-resolution quadrupole time-of-flight instrument (QTOF) for deep shotgun proteomics. *Mol. Cell. Proteom.* **14**, 2014–2029 (2015).
33. Searle, B. C., Lawrence, R. T., MacCoss, M. J. & Villén, J. Thesaurus: quantifying phosphopeptide positional isomers. *Nat. Methods* **16**, 703–706 (2019).
34. Bekker-Jensen, D. B. et al. Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. *Nat. Commun.* **11**, 787 (2020).
35. Glover, M. S. et al. Examining the influence of phosphorylation on peptide ion structure by ion mobility spectrometry–mass spectrometry. *J. Am. Soc. Mass Spectrom.* **27**, 786–794 (2016).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2020



## Methods

**Sample preparation.** The human cancer cell line (HeLa S3, ATCC) was cultured in Dulbecco's modified Eagle's medium with 10% fetal bovine serum, 20 mM glutamine and 1% penicillin-streptomycin. Cells were collected by centrifugation, washed with phosphate-buffered saline, flash-frozen in liquid nitrogen and stored at  $-80^{\circ}\text{C}$ . Cell lysis, reduction and alkylation were performed in lysis buffer with chloroacetamide (PreOmics) as reported previously<sup>30</sup>. In brief, the cell suspension was heated to  $95^{\circ}\text{C}$  for 10 min and subsequently sonicated to further disrupt cells and shear nucleic acids. Proteins were enzymatically cleaved overnight by adding equal amounts of Lys-C and trypsin in a 1:100 (wt/wt) enzyme:protein ratio. De-salting and purification were performed according to the PreOmics iST protocol on a styrene divinylbenzene reversed-phase sulfonate (SDB-RPS) sorbent. Purified peptides were vacuum-centrifuged to dryness and reconstituted in double-distilled water with 2 vol% acetonitrile (ACN) and 0.1 vol% trifluoroacetic acid (TFA) for single-run LC-MS analysis or fractionation.

To evaluate the quantitative accuracy of diaPASEF, we performed a two-proteome experiment with HeLa and yeast. Furthermore, to evaluate the achievable sample throughput we analyzed HeLa samples using the Evosep One liquid chromatography system. For these experiments, purified and predigested yeast standard was purchased from Promega and resuspended in 0.1 vol% formic acid; whole HeLa cell pellets were purchased from CIL Biotech and lysed using trifluoroethanol<sup>31</sup>. In brief, the cell suspension was kept on ice for 10 min and subsequently incubated for 20 min at  $56^{\circ}\text{C}$ . We used 200 mM dithiothreitol to reduce proteins at  $90^{\circ}\text{C}$  (20 min), and 200 mM iodoacetamide to alkylate cysteine residues during 90 min at room temperature ( $21^{\circ}\text{C}$ ). Proteins were enzymatically cleaved overnight by adding trypsin in a 1:100 (wt/wt) enzyme:protein ratio. The proteome digests were de-salted and purified on a solid phase extraction cartridge (Empore C<sub>18</sub> SPE cartridge, Sigma Aldrich). Samples were washed with 0.1 vol% formic acid and subsequently eluted with 50 vol% ACN in 0.1 vol% formic acid. Purified and dried peptides were reconstituted in 0.1 vol% formic acid for injection. For the two-proteome experiment, the purified peptides from HeLa and yeast were combined as follows: sample A consisted of 200 ng human and 45 ng yeast proteins per LC-MS injection, and sample B of 200 ng human and 15 ng yeast proteins per LC-MS injection. For the Evosep experiments, approximately 200 ng peptides was loaded onto Evtips (EV2001, Evosep) in accordance with the manufacturer's instructions.

**High-pH reversed-phase fractionation.** To generate a comprehensive library of HeLa precursor and fragment ions, peptides were fractionated at pH 10 with a 'spider fractionator' coupled to an EASY-nLC 1000 chromatography system (Thermo Fisher Scientific) as described previously<sup>32</sup>. Approximately 50  $\mu\text{g}$  purified peptides were separated on a 30 cm C<sub>18</sub> column in 96 min and automatically concatenated into 24 fractions by shifting the exit valve every 120 s. The fractions were vacuum-centrifuged to dryness and reconstituted in double-distilled water with 2 vol% ACN and 0.1 vol% TFA for LC-MS analysis. To generate spectral libraries for the Evosep and two-proteome experiments, 100  $\mu\text{g}$  purified peptides from yeast and from HeLa digests were each fractionated at pH 10 on a reversed-phase column (Waters Acquity CSH C18 column, 1.7  $\mu\text{m}$ , 2.1  $\times$  150 mm) using a Dionex Ultimate 3000 system (Thermo Fisher Scientific). For mass spectrometric analysis, the freeze-dried fractions were reconstituted in 0.1% formic acid and placed in the autosampler or loaded onto Evtips.

**Liquid chromatography.** Nanoflow reversed-phase chromatography was performed on an EASY-nLC 1200 system (Thermo Fisher Scientific). Peptides were separated in 120 min at a flow rate of 300 nL min<sup>-1</sup> on a 50 cm  $\times$  75  $\mu\text{m}$  column with a laser-pulled electrospray emitter packed with 1.9  $\mu\text{m}$  ReproSil-Pur C<sub>18</sub>-AQ particles (Dr. Maisch). Mobile phases A and B were water with 0.1 vol% formic acid and 80:20:0.1 vol% ACN:water:formic acid, respectively. The fraction of B was linearly increased from 5% to 30% in 95 min, followed by an increase to 60% in 5 min and a further increase to 95% in 5 min before re-equilibration.

For the two-proteome experiment, we used a nanoElute liquid chromatography system (Bruker Daltonics). Peptides were separated in 120 min at a flow rate of 400 nL min<sup>-1</sup> on a commercially available reversed-phase C<sub>18</sub> column with an integrated CaptiveSpray Emitter (25 cm  $\times$  75  $\mu\text{m}$ , 1.6  $\mu\text{m}$ , IonOpticks). Mobile phases A and B were 0.1 vol% formic acid in water and 0.1 vol% formic acid in ACN, respectively. The fraction of B was linearly increased from 2% to 25% in 90 min, followed by an increase to 35% in 10 min and a further increase to 80% in 10 min before re-equilibration.

For proteome analyses with fast gradients, we used an Evosep One liquid chromatography system<sup>33</sup> and analyzed the samples with the predefined 60, 100 or 200 samples per day methods (Evosep RC.Net 1.3 plugin). For the 60 and 100 samples per day methods, we used an 8 cm  $\times$  150  $\mu\text{m}$  column with 1.5  $\mu\text{m}$  C<sub>18</sub> beads (EV1109, Evosep) and for the 200 samples per day method, we used a 4 cm  $\times$  150  $\mu\text{m}$  column with 1.9  $\mu\text{m}$  C<sub>18</sub> beads (EV1107, Evosep). Mobile phases A and B were 0.1 vol% formic acid in water and 0.1 vol% formic acid in ACN, respectively.

**Mass spectrometry.** Liquid chromatography was coupled online to a hybrid TIMS quadrupole TOF mass spectrometer (Bruker timsTOF Pro) via a CaptiveSpray nano-electrospray ion source. A detailed description of the instrument is available

in ref.<sup>33</sup>. The dual TIMS analyzer was operated at a fixed duty cycle close to 100% using equal accumulation and ramp times of 100 ms each. We performed DDA in PASEF mode with 10 PASEF scans per topN acquisition cycle. Singly charged precursors were excluded by their position in the  $m/z$ -ion mobility plane, and precursors that reached a target value of 20,000 arbitrary units were dynamically excluded for 0.4 min. The quadrupole isolation width was set to  $2m/z$  for  $m/z < 700$  and to  $3m/z$  for  $m/z > 700$ . TIMS elution voltages were calibrated linearly to obtain the reduced ion mobility coefficients ( $1/K_0$ ) using three Agilent ESI-L Tuning Mix ions ( $m/z$  622, 922 and 1,222).

To perform DIA, we extended the instrument control software (Bruker otoControl v6) to define quadrupole isolation windows as a function of the TIMS scan time (diaPASEF). The instrument control electronics were modified to allow seamless and synchronous ramping of all applied voltages. We tested multiple schemes for data-independent precursor windows and placement in the  $m/z$ -ion mobility plane and defined up to eight windows for single 100 ms TIMS scans, as detailed earlier. Acquisition schemes for the diaPASEF methods used herein are shown in Supplementary Figs. 1–4, 12. To limit the number of MS1 scans, we repeated diaPASEF in acquisition schemes; for example, each of the four diaPASEF scans was done twice in the high-sensitivity scheme, and this resulted in one MS1 and eight diaPASEF scans per acquisition cycle. In both scan modes, the collision energy was ramped linearly as a function of the mobility from 59 eV at  $1/K_0 = 1.6 \text{ Vs cm}^{-1}$  to 20 eV at  $1/K_0 = 0.6 \text{ Vs cm}^{-1}$ . To visualize the isolation of precursor ions in Fig. 1d and analyze the ion current from multiply charged precursors (likely peptide precursors) in Fig. 2, we set the collision energy to 5 eV to prevent fragmentation. In the BSA experiment, we distributed the 14 diaPASEF windows to one TIMS scan each and defined  $14 \times 50$  Th precursor isolation windows from  $m/z$  325 to 1,025. In the HeLa DIA experiment, we defined  $32 \times 25$  Th isolation windows from  $m/z$  400 to 1,200. To adapt the MS1 cycle time in diaPASEF, we set the repetitions to 2 in the 16-scan diaPASEF scheme and to 4 in the 4-scan diaPASEF scheme in these experiments.

**Spectral library generation.** To generate spectral libraries for targeted data extraction, we first analyzed high-pH reversed-phase fractions acquired in DDA mode with MaxQuant v1.6.5.0 or 1.6.7.0, which extracts four-dimensional features on the MS1 level (retention time,  $m/z$ , ion mobility and intensity) and links them to peptide spectrum matches. We had acquired the 120 min HeLa library previously for the purpose of predicting ion mobility cross-sections by deep learning<sup>34</sup>. The maximum precursor mass tolerance of the main search was set to 20 ppm and de-isotoping of fragment ions was deactivated. Other than that, we used the default 'TIMS-DDA' parameters. Tandem mass spectrometry spectra were matched against an *in silico* digest of the appropriate Swiss-Prot proteome database (human, 20,402 entries; *Saccharomyces cerevisiae*, 6,721 entries) and a list of common contaminants. The minimum peptide length was set to 7 amino acids, and the peptide mass was limited to 4,600 Da. Carbamidomethylation of cysteine residues was defined as a fixed modification, and methionine oxidation and acetylation of protein N-termini were defined as variable modifications. The FDR was controlled at <1% at both the peptide spectrum match level and the protein level. The Mobi-DIK software package builds on OpenMS tools to compile spectral libraries in the standardized TraML or pep formats from the MaxQuant output tables and retains the full ion mobility information for each precursor-to-fragment ion transition. Only proteotypic peptides with precursor  $m/z > 400$  were included in the library; they were required to have a minimum of six fragment ions with  $m/z > 350$  and to be outside the precursor mass isolation range. We generated separate, project-specific libraries for the 120 min HeLa experiments, the two-proteome experiment and the Evosep experiment (60 samples per day method).

**Targeted data extraction.** To analyze diaPASEF data, we developed an ion mobility DIA analysis kit (Mobi-DIK) that extracts fragment ion traces from the four-dimensional data space, as detailed earlier. Repeated diaPASEF scans were merged. Raw data were automatically re-calibrated using curated reference values in  $m/z$ , retention time and ion mobility dimensions (387 peptides for linear and 3,184 peptides for non-linear alignment). We applied an outlier detection in each dimension before calculating the final fit function to increase robustness. Peak picking and subsequent scoring functionalities in the Mobi-DIK software build on OpenSWATH<sup>35</sup> modules. For diaPASEF, we extended these modules to also consider the additional ion mobility dimension. OpenSWATH (revision: c0b987a) was run with the following parameters: min\_coverage=0.1 (0.3 in Fig. 3), RTNormalization:alignmentMethod=LOWESS, RTNormalization:lowess:span=0.01, Scoring:TransitionGroupPicker:PeakPickerMRM:sgolay\_frame\_length=11, Scoring:stop\_report\_after\_feature=5, rt\_extraction\_window=250, Scoring:Scores:use\_ion\_mobility\_scores, mz\_correction\_function=quadratic\_regression\_delta\_ppm, use\_ms1\_traces, mz\_extraction\_window=25, mz\_extraction\_window\_unit=ppm, mz\_extraction\_window\_ms1=25, mz\_extraction\_window\_ms1\_unit=ppm, irt\_mz\_extraction\_window\_unit=ppm, irt\_mz\_extraction\_window=40, Calibration:ms1\_lm\_calibration, ion\_mobility\_window=0.06, irt\_lm\_extraction\_window=99, RTNormalization:NrRTBins=8, RTNormalization:MinBinsFilled=4. All other parameters were set to default values. PyProphet was used to train an XGBoost



## ARTICLES

## NATURE METHODS

classifier for target-decoy separation by first creating one concatenated and subsampled OpenSwath output for each set of three replicate injections of the same acquisition strategy and sample amount. The classifier was subsequently applied to score all samples, with FDR controlled to <1% at the peak group level per sample, and at both the global peptide and global protein levels. For the two-proteome experiment, the protein FDR was set to <1%, and TRIC alignment<sup>40</sup> was performed using a peak-group level seed  $q$  value threshold of 0.01 and extension  $q$  value threshold of 0.05. In the case of two overlapping diaPASEF windows, the analysis was performed separately for the individual windows, and for FDR estimation the highest scoring peak group was selected. Protein abundances were estimated using an R implementation of the MaxLFQ<sup>41</sup> algorithm for DIA termed *iq* (v1.9) with default parameters<sup>42</sup>. Potential contaminants were excluded from further analysis.

**Bioinformatics.** Output tables from the Mobi-DIK data analysis pipeline were further analyzed and visualized in the R statistical computing environment v4 or in Python v3.6. Ion chromatograms shown in Fig. 2a were extracted from raw data files with the Bruker DataAnalysis software. To estimate the peptide precursor ion current sampled with different acquisition methods in Fig. 2b, we extracted tandem mass spectrometry spectra directly from the raw data files using an SQL interface. Given that the isolated precursors were not fragmented in this experiment, we were able to restrict the analysis to likely multiply charged peptide ions by their position in the ion mobility- $m/z$  space. For this, we empirically estimated a line separating singly from multiply charged species and discarded all signals with  $1/K_0 \geq 0.0009 \cdot m/z + 0.48$ . Protein copy numbers were estimated with the Proteomic Ruler<sup>43</sup> Perseus<sup>44</sup> (v1.6.0.8) plugin from the MaxQuant output table.

**Statistics.** Summary statistics such as coefficients of variation were calculated based on replicate injections of the same sample ( $n=3$  technical replicates) to indicate the technical variation of the mass spectrometry method.

**Reporting summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

### Data availability

The mass spectrometry raw data and spectral libraries generated and analyzed during the current study have been deposited with the ProteomeXchange Consortium via the PRIDE<sup>45</sup> partner repository with the dataset identifier PXD017703. *Homo sapiens* (taxon identifier: 9606) and *S. cerevisiae* (taxon identifier: 559292) proteome databases were downloaded from <https://www.uniprot.org>. Source data are provided with this paper.

### Code availability

Code is available under the three-clause BSD license on <https://github.com/OpenMS/OpenMS> and <https://github.com/Roestlab/dia-pasef>.

### References

36. Kulak, N. A., Pichler, G., Paron, L., Nagaraj, N. & Mann, M. Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nat. Methods* **11**, 319–324 (2014).
37. Wang, H. et al. Development and evaluation of a micro- and nanoscale proteomic sample preparation method. *J. Proteome Res.* **4**, 2397–2403 (2005).
38. Kulak, N. A., Geyer, P. E. & Mann, M. Loss-less nano-fractionator for high sensitivity, high coverage proteomics. *Mol. Cell. Proteom.* **16**, 694–705 (2017).

39. Meier, F. et al. Deep learning the collisional cross sections of the peptide universe from a million training samples. Preprint at *bioRxiv* <https://doi.org/10.1101/2020.05.19.102285> (2020).
40. Röst, H. L. et al. TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics. *Nat. Methods* **13**, 777–783 (2016).
41. Cox, J. et al. Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Mol. Cell. Proteom.* **13**, 2513–2526 (2014).
42. Pham, T. V., Henneman, A. A. & Jimenez, C. R. *iq*: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* **36**, 2611–2613 (2020).
43. Wiśniewski, J. R., Hein, M. Y., Cox, J. & Mann, M. A 'proteomic ruler' for protein copy number and concentration estimation without spike-in standards. *Mol. Cell. Proteom.* **13**, 3497–3506 (2014).
44. Tyanova, S. et al. The Perseus computational platform for comprehensive analysis of (prote)omics data. *Nat. Methods* **13**, 731–740 (2016).
45. Vizcaino, J. A. et al. 2016 update of the PRIDE database and its related tools. *Nucleic Acids Res.* **44**, D447–D456 (2016).

### Acknowledgements

This work was partially supported by the German Research Foundation (DFG-Gottfried Wilhelm Leibniz Prize granted to M.M., grant no. MA 1764/2-1) and by the Max Planck Society for the Advancement of Science (M.M.). This work was partially supported by the Government of Canada through Genome Canada (grant no. 15411) and by the Canadian Institutes for Health Research (H.L.R.). B.C.C. was supported by a Swiss National Science Foundation Ambizione grant (no. PZ00P3\_161435). R.A. was supported by the Swiss National Science Foundation (grant no. 3100A0-688 107679) and the European Research Council (ERC-2014AdG 670821). We thank our colleagues in the Department of Proteomics and Signal Transduction (Max Planck Institute of Biochemistry) and at Bruker Daltonik for discussions and help; in particular J. Müller, A. Strasser, C. Deiml and I. Paron for technical support.

### Author contributions

F.M., R.A., B.C.C., H.L.R. and M.M. conceptualized and designed the study; F.M. and M.M. conceived the acquisition mode; H.L.R. conceived the data analysis software; F.M., A.-D.B., S.K.-S., M.L., O.R., N.B. and B.C.C. performed experiments; A.H. and M.F. contributed to the software development; F.M., A.-D.B., M.F., A.H., I.B., E.V., S.K.-S., B.C.C., H.L.R. and M.M. analyzed the data; F.M., R.A., B.C.C., H.L.R. and M.M. wrote the manuscript.

### Competing interests

S.K.-S., M.L. and O.R. are employees of Bruker Daltonik. N.B. is an employee of and M.M. a shareholder in Evosep Biosystems. All other authors have no competing interests.

### Additional information

**Supplementary information** is available for this paper at <https://doi.org/10.1038/s41592-020-00998-0>.

**Correspondence** and requests for materials should be addressed to B.C.C., H.L.R. or M.M.

**Peer review information** Arunima Singh was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

### 3.7. Article 7: Rapid and in-depth coverage of the (phospho-)proteome with deep libraries and optimal window design for dia-PASEF

Authors: Patricia Skowronek<sup>‡</sup>, Marvin Thielert<sup>‡</sup>, **Eugenia Voytik<sup>‡</sup>**, Maria C. Tanzer<sup>‡</sup>, Fynn M. Hansen<sup>‡</sup>, Sander Willems<sup>‡</sup>, Özge Karayel<sup>‡</sup>, Andreas-David Brunner<sup>‡</sup>, Florian Meier<sup>‡§</sup>, Matthias Mann<sup>‡¶\*</sup>

<sup>‡</sup> Department Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

<sup>§</sup> Functional Proteomics, Jena University Hospital, Jena, Germany

<sup>¶</sup> Protein Research, NNF Center for Protein Research, Copenhagen, Denmark

Accepted (in press) in *Molecular & Cellular Proteomics* (2022).

Our recently published development of data independent acquisition (DIA) on a trapped ion mobility mass spectrometer, called dia-PASEF and introduced in Article 6 is particularly beneficial for acquiring a wide range of proteomics data while maintaining a high sequence coverage and very high sensitivity. However, with the original dia-PASEF method, the placement of the DIA windows in the two-dimensional  $m/z$  and ion mobility space was empirical and – as it turns out – not optimal.

In this study, we address this challenge by developing a new method called py\_diAID. It optimally places variable isolation windows using a Bayesian optimization scheme depending on the precursor density in the  $m/z$  – ion mobility plane. The py\_diAID method is freely available on GitHub as a Python package and a graphical user interface on the major operating systems. In combination with deep project-specific DIA libraries and short gradients, we reproducibly identified and quantified over 6,000 proteins in only 11 minutes LC gradients (100 samples per day) and an astounding 7,700 proteins in 44 minutes gradients. Performing, to our knowledge, the first large-scale study of PTMs on the timsTOF platform, we quantify around 20,000 phosphopeptides in quadruplicate measurements, achieving 93% precursor coverage compared to 34% using the previously published ‘fast’ dia-PASEF method.

For this study, I helped implement the py\_diAID tool and analyze the data.

bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

## **Rapid and in-depth coverage of the (phospho-)proteome with deep libraries and optimal window design for dia-PASEF**

Patricia Skowronek‡, Marvin Thielert‡, Eugenia Voytik‡, Maria C. Tanzer‡, Fynn M. Hansen‡, Sander Willems‡, Özge Karayel‡, Andreas-David Brunner‡, Florian Meier‡§, Matthias Mann‡¶\*

‡ Department Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

§ Functional Proteomics, Jena University Hospital, Jena, Germany

¶ Protein Research, NNF Center for Protein Research, Copenhagen, Denmark

\* Correspondence and material requests may be addressed to M.M.

[mmann@biochem.mpg.de](mailto:mmann@biochem.mpg.de)



bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

#### ABBREVIATIONS

|          |  |
|----------|--|
| ABC      | ammonium bicarbonate   |
| ACN      | acetonitrile   |
| CAA      | 2-chloroacetamide  |
| dda      | data-dependent acquisition   |
| dia      | data-independent acquisition   |
| EGF      | epidermal growth factor  |
| FA       | formic acid  |
| GO       | Gene Ontology  |
| IM       | ion mobility   |
| IPA      | isopropyl alcohol  |
| KEGG     | Kyoto Encyclopedia of Genes and Genomes  |
| MeOH     | methanol   |
| PASEF    | parallel accumulation – serial fragmentation                                       |
| PBS      | phosphate-buffered saline  |
| PTM      | post-translational modification  |
| py_diAID | Python package for Data-Independent Acquisition with an Automated Isolation Design |
| SDC      | sodium deoxycholate  |
| SPD      | samples per day  |
| TBS      | tris-buffered saline   |
| TCEP     | tris(2-carboxy(ethyl)phosphine)  |
| TFA      | trifluoroacetic acid   |
| TIMS     | trapped ion mobility spectrometry  |

## ABSTRACT

Data-independent acquisition (DIA) methods have become increasingly attractive in mass spectrometry (MS)-based proteomics, because they enable high data completeness and a wide dynamic range. Recently, we combined DIA with parallel accumulation – serial fragmentation (dia-PASEF) on a Bruker trapped ion mobility separated (TIMS) quadrupole time-of-flight (TOF) mass spectrometer. This requires alignment of the ion mobility separation with the downstream mass selective quadrupole, leading to a more complex scheme for dia-PASEF window placement compared to DIA. To achieve high data completeness and deep proteome coverage, here we employ variable isolation windows that are placed optimally depending on precursor density in the  $m/z$  and ion mobility plane. This Automatic Isolation Design procedure is implemented in the freely available `py_diAID` package. In combination with in-depth project-specific proteomics libraries and the Evosep LC system, we reproducibly identified over 7,700 proteins in a human cancer cell line in 44 minutes with quadruplicate single-shot injections at high sensitivity. Even at a throughput of 100 samples per day (11 minutes LC gradients), we consistently quantified more than 6,000 proteins in mammalian cell lysates by injecting four replicates. We found that optimal dia-PASEF window placement facilitates in-depth phosphoproteomics with very high sensitivity, quantifying more than 35,000 phosphosites in a human cancer cell line stimulated with an epidermal growth factor (EGF) in triplicate 21 minutes runs. This covers a substantial part of the regulated phosphoproteome with high sensitivity, opening up for extensive systems-biological studies.

## KEY WORDS

TIMS; PASEF; data-independent acquisition; phosphoproteomics; systems biology

## INTRODUCTION

MS-based proteomics has become a powerful tool to study proteomes in a systematic and unbiased manner (1). In recent years, this development has been accelerated by data-independent acquisition (DIA) (2), where predefined isolation windows cycle through the  $m/z$ -range of interest, and regularly subject the covered peptide precursors to fragmentation (3–6). Although the concept of DIA was established more than a decade ago (4, 7), only the most recent DIA implementations and hardware advancements in MS and data analysis are at par or even exceeding data dependent acquisition (DDA) with regards to sensitivity, reproducibility, and dynamic range coverage (2, 6, 8) and surpass targeted approaches in throughput and ease-of-use (9, 10). This holds also true for studying post-translational modifications (11–13).

DIA has recently shown promise in combination with trapped ion mobility spectrometry (TIMS) mass spectrometers, as demonstrated with single-cell analysis (14, 15). The TIMS tunnel is a compact and high-performance implementation of ion mobility separation. It captures the peptides from the incoming ion beam discretizing the continuous LC elution. Within the TIMS tunnel, each ion reaches an equilibrium position based on the opposing forces of a gas flow and an electric field gradient. Decreasing the electric field gradient elutes the peptide ions as a function of their ion mobility (16–19). In the Bruker timsTOF instruments, the TIMS device is placed upstream of mass-selective quadrupole and high-resolution time-of-flight mass analyzer and is itself divided into two parts (20–22). The mobility separation



can be synchronized with the quadrupole isolation, leading to high ion beam utilization, increased sensitivity and decreased spectral complexity due to the additional ion mobility dimension (6, 20, 23). This principle is termed PASEF for parallel accumulation-serial fragmentation (21, 24).

When combined with DIA (dia-PASEF), peptide precursors separate not only in the  $m/z$  but also in the ion mobility dimension, in contrast to standard DIA modes (2, 6). We have observed that dia-PASEF is particularly beneficial for acquiring a wide range of proteomics data while maintaining a high sequence coverage and very high sensitivity (6, 15). Furthermore, ions are detected by inherently fast TOF analysis allowing fast DIA cycle times, which is particularly advantageous for short LC gradients (6). The resulting, complex spectra can be efficiently analyzed by machine learning or deep learning-based algorithms such as DIA-NN (25, 26).

Here, we set out to explore the potential of dia-PASEF to further increase coverage and quantitative accuracy on the fast and sensitive ion mobility-mass spectrometry platform. In dia-PASEF, two-dimensional precursor isolation schemes are defined in the  $m/z$ -ion mobility plane. We used a Bayesian optimization algorithm ensuring optimal placement of the acquisition scheme in both dimensions. Single-runs acquired with these optimal dia-PASEF methods were searched against in-depth project-specific libraries. Furthermore, we combined dia-PASEF with the Evosep One LC system, which features a pre-formed gradient particularly designed for high throughput by eliminating inter-run overhead (6, 27). Together, our optimized dia-PASEF workflow for high throughput proteomics quantified more than 7,000 proteins in only 21 minutes from quadruplicate injections of a tryptic HeLa digest.

Motivated by these proteomic results, we also investigated py\_diAID for phosphorylation analysis. On the Orbitrap MS platform, Olsen and co-workers recently demonstrated an efficient combination of fast chromatography runs with DIA, quantifying more than 13,000 phosphopeptides in very short (15 min) LC/MS runs from HeLa cells using the Spectronaut software (11). In a small scale study, Ishihama and co-workers showed that phosphopeptides analysis benefits from the additional ion mobility dimension in PASEF (28). For large-scale PTM studies, our optimized py\_diAID acquisition schemes cover nearly all theoretical phosphopeptide precursors and quantified expected changes in the well-studied EGF-receptor signaling pathway with minimal time and sample consumption.

## EXPERIMENTAL PROCEDURES

### Experimental Design and Statistical Rationale

All experiments were done using HeLa cell lysate obtained from HeLa S3 cells (ATCC), routinely used for proteomics method development and benchmark experiments (supplemental Fig. S1). Altogether, the data set includes 322 raw data files (uploaded to PRIDE, see below). We used the same HeLa batch for generating libraries and single-run data of both proteome and phosphoproteome measurements. In brief, proteome measurements with different gradient lengths and the technical comparisons of the original and optimal dia-PASEF methods for phosphoproteomics were acquired in quadruplicates. Unless otherwise mentioned, 200 ng HeLa lysate were used for single-run proteome and 100  $\mu$ g for the single-run phosphopeptide enrichment experiments. The libraries were acquired as described below. The experimental design and statistical rationale are described in the respective figure legends. The EGF experiment was performed in biological triplicates to determine significantly different phosphosite levels between the EGF-treated and control samples. Technical quadruplicates



were acquired to evaluate reproducibility and quantitative accuracy by calculating coefficient of variations (CVs) and mean of the replicate injections. Moreover, we alternated the MS run order to avoid potential carryover effects or any similar biases.

### Sample preparation

HeLa S3 cells (ATCC) were cultured in Dulbecco's modified Eagle's medium (Life Technologies Ltd., UK) containing 20 mM glutamine, 10% fetal bovine serum, and 1% penicillin-streptomycin. Sample preparation was essentially performed as previously described in the in-stage tip protocol (29). In brief, the cells were washed with PBS and lysed. Protein reduction and alkylation and digestion with trypsin (Sigma-Aldrich) and LysC (WAKO) (1:100, enzyme/protein, w/w) were performed in one step. Resulting peptides were dried and reconstituted in a solution A\* (0.1% TFA/2% ACN). Peptide concentrations were measured optically at 280 nm (Nanodrop 2000; Thermo Scientific) and 200 ng peptides were loaded onto Evtotips for LC-MS/MS analysis as described previously (15). The Evtotips were washed with 0.1% FA/99.9% ACN, equilibrated with 0.1% FA, loaded with the sample dissolved in 0.1% FA, and washed with 0.1% FA.

For phosphoproteomics, HeLa cells at a plate confluence of 80% were treated for 10 min with 100 ng/mL animal-free recombinant human EGF (PeproTech) or Gibco™ distilled water (Thermo Fisher Scientific) and washed three times with ice-cold TBS before lysis in 2% SDC in 100 mM Tris-HCl (pH 8.5) at 95°C. Protein concentrations were determined using the BCA assay and samples were then reduced and alkylated with 10 mM TCEP and 40 mM CAA, respectively. Altogether, 25 mg protein material of sample was used for the library generation, 8 mg for EGF treated experiments including method benchmarking and 4 mg for untreated experiments. The sample was digested with trypsin (Sigma-Aldrich) and LysC (WAKO) (1:100, enzyme/protein, w/w) overnight and subsequently desalted using Sepax Extraction columns (Generik DBX). Each cartridge was prepared with 100% MeOH and 99% MeOH/1% TFA. After equilibration with 0.2% TFA, the samples were loaded with a protein concentration of 1 mg/mL, washed with 99% IPA/1% TFA, 0.2% TFA/5% ACN, and 0.2% TFA solutions. The peptides were eluted with 5% NH<sub>4</sub>OH/80% ACN. Lyophilized peptides were reconstituted in equilibration solution (1% TFA/80% ACN) and 100 µg peptide material per sample/AssayMAP cartridge, each containing 5 µL Fe(III)-NTA, was enriched for phosphopeptide with the AssayMAP bravo robot (Agilent) (30). Phosphopeptides were dried in a SpeedVac for 20 min at 45°C and loaded onto Evtotips as described above.

### High-pH reversed-phase fractionation for library generation

To generate proteome libraries, 10 µg and 60 µg peptides were separated with high pH reverse-phase chromatography into 24 and 48 fractions, respectively, on a 30 cm C<sub>18</sub> column with an inner diameter of 250 µm at a flow rate of 2 µL/min using the spider sample fractionator (31). The gradient consisted of the binary buffer system (PreOmics GmbH). The buffer B concentration of 3% was increased to 30% in 45 min, 40% in 12 min and 60% in 5 min, and 95% in 10 min. After washing at 95% for 10 min, buffer B concentration was re-equilibrated to 3% in 10 min. The exit valve concatenated the eluted peptides automatically by switching after a defined collection time (80s for 24 and 60s for 48 fractions). The fractions were dried in a SpeedVac and reconstituted in solution A\*. A quarter of each fraction was loaded onto Evtotips for LC-MS/MS analysis. Below we will refer to 'the reference proteome

bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

library' that represents a 24 high pH fractions and dda-PASEF spectral library of a tryptic HeLa digest acquired with a 21 min Evosep gradient.

To generate a phosphoproteome library, peptides obtained from the EGF stimulated cells were separated using an UFLC system (Shimadzu). 6 mg peptide material was fractionated with a binary buffer system: A (2.5 mM ABC) and B (2.5 mM ABC/80% ACN). The peptides were loaded onto a reversed-phase column (ZORBAX 300Extend-C<sub>18</sub>, Agilent) and separated at a 1 mL/min flow rate at 40°C. The buffer B concentration of 2.5% was increased to 38% in 82.5 min, 75% in 2 min, and 100% in 8 min. It stayed at 100% for 2 min and was reduced to 2.5% in 2 min. In total, 95 fractions were collected and fractions with low peptide yield, as determined using Nanodrop, were pooled (supplemental table 1) and dried in a SpeedVac. Next, 76 fractions were enriched for phosphopeptide, which were subsequently loaded onto Evotips.

#### LC-MS/MS analysis

The Evosep One liquid chromatography system coupled with a timsTOF Pro mass spectrometer (Bruker) was used to measure all samples. The 60 and 100 SPD (samples per day) methods required an 8 cm × 150 µm reverse-phase column packed with 1.5 µm C<sub>18</sub>-beads (Pepsep) and the 30 SPD method a 15 cm × 75 µm column with 1.9 µm C<sub>18</sub>-beads (Pepsep) at 40°C. The analytical columns were connected with a fused silica ID emitter (10 µm ID, Bruker Daltonics) inside a nano-electrospray ion source (Captive spray source, Bruker). The mobile phases comprised 0.1% FA as solution A and 0.1% FA/80% ACN as solution B.

The library samples were acquired in dda-PASEF mode with four PASEF/MSMS scans at a throughput of 60 and 100 SPDs and 10 PASEF/MSMS scans at 30 SPD per topN acquisition cycle. Singly charged precursors were filtered out by their position in the *m/z*-ion mobility plane, and only precursor signals over an intensity threshold of 2,500 arbitrary units (a.u.) were picked for fragmentation. While precursors over the target value of 20,000 a.u. were dynamically excluded for 0.4 min, ones below 700 Da were isolated with a 2 Th window and ones above with 3 Th. All spectra were acquired within an *m/z*-range of 100 to 1700 and an ion mobility range from 1.51 to 0.6 Vs cm<sup>-2</sup>.

We described the original dia-PASEF method in Meier et al. (6). The dia-PASEF methods optimized here with py\_diAID cover an *m/z*-range from 300 to 1200 for proteome and from 400 to 1400 for phosphoproteome measurements. Each method includes two ion mobility windows per dia-PASEF scan with variable isolation window widths adjusted to the precursor densities. Eight, 12 and 25 dia-PASEF scans were deployed at a throughput of 100 (cycle time: 0.96 s), 60 (cycle time: 1.38s), and 30 SPDs (cycle time: 2.7s), respectively. We created dia-PASEF methods with equidistant window widths (supplemental Fig. S5) with the software "Compass DataAnalysis" (Bruker Daltonics). These acquisition schemes are plotted on top on a kernel density estimation of precursors from a reference library in supplemental Figure S2-4. The ion mobility range was set to 1.5 Vs cm<sup>-2</sup> and 0.6 Vs cm<sup>-2</sup>. The accumulation and ramp times were specified as 100 ms for all experiments. As a result, each MS1 scan and each MS2/dia-PASEF scan last 100 ms plus additional transfer time, and a dia-PASEF method with 12 dia-PASEF scans has a cycle time of 1.38s. The collision energy was decreased as a function of the ion mobility from 59 eV at 1/K<sub>0</sub> = 1.6 Vs cm<sup>-2</sup> to 20 eV at 1/K<sub>0</sub> = 0.6 Vs cm<sup>-2</sup> and the ion mobility dimension was calibrated with three Agilent ESI Tuning Mix ions (*m/z*, 1/K<sub>0</sub>: 622.02, 0.98 Vs cm<sup>-2</sup>, 922.01, 1.19 Vs cm<sup>-2</sup>, 1221.99, 1.38 Vs cm<sup>-2</sup>). For phosphoproteomics



bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

experiments, the collision energy was decreased from 60 eV at 1.5 Vs cm<sup>-2</sup> to 54 eV at 1.17 Vs cm<sup>-2</sup> to 25 eV at 0.85 Vs cm<sup>-2</sup> and end at 20 eV at 0.6 Vs cm<sup>-2</sup>.

### Raw data analysis

We employed DIA-NN, MSFragger and Spectronaut for transforming raw data into precursor and fragment identifications based on 3D peak position (RT, *m/z* precursor, and ion mobility). In each case, all data was searched against the reviewed human proteome (Uniprot, Nov 2021, 20,360 entries without isoforms) with trypsin/LysC as digestion enzymes. Cysteine carbamidomethylation was set as fixed modification. Methionine oxidation, methionine excision at the N-terminus, and in the case of the phosphoproteome searches, phosphorylation (STY) was selected as variable modifications. A maximum of two missed cleavages and up to three variable modifications were allowed.

The project-specific libraries for DIA-NN analyses were generated with FragPipe (32) (FragPipe 16.2, MSFragger 3.4 (33–35), Philosopher 4.0.0 (36), Python 3.8, EasyPQP 0.1.25 (37)). The default settings were kept except that the precursor mass tolerance was set from -20 to 20 ppm and the fragment mass tolerance to 20 ppm. Additionally, Pyro-Glu or ammonia loss at the peptide N-terminus and water loss on N-terminal glutamic acid were selected as variable modification. The output tables were filtered for an 1% FDR using the Percolator (38, 39) and ProteinProphet (40) option in FragPipe (supplemental table 2).

DIA-NN 1.8 was used to analyze the single-shot experiments against the project-specific libraries generated with FragPipe (32). The default settings were kept except that we changed the charge state to 2 - 4. The precursor's *m/z* range was restricted from 300 to 1200 for proteome and 400 to 1400 for phosphoproteome analysis. The fragment *m/z* range was set from 100 to 1700, and the mass and MS1 accuracy to 15 ppm. 'Match between run' was enabled while 'protein inference' was disabled. We also enabled 'robust LC (high precision)' as the quantification strategy. The proteomics output tables were filtered for a maximum of 1% of q-value at both precursor and global protein levels. For phosphoproteomics, the post-translational modification q-value also had to be a maximum of 1%. The PG.MaxLFQ column integrated in the DIA-NN output tables reports normalized quantity employing the MaxLFQ principle (41) and was used for quantitative analysis on the protein level. For our phosphoproteomics analysis, we used the scoring of post-translational sites implemented in DIA-NN with 'PTM.Site.Confidence' indicating the localization probability (13).

Spectronaut (v16, Biognosys AG, Schlieren, Switzerland) (3) was used for comparative analysis and we used the same search settings as described above if not stated differently. The FDR cutoff was set to 1%. The precursor peptide and q-value cutoffs were 0.2 and 0.01, respectively. The protein q-value experiment and run wide cutoffs were 0.01 and 0.05, respectively. The dataset was analyzed with a sparse q-value and no imputation was performed. For phosphoproteomics experiments, the PTM localization cutoff was set to 0. The results were filtered for the best N fragments per peptide between 3 to 25.

Peptide collapse (v1.4.1), a plug-in tool for Perseus (42), collapsed peptide output tables from DIA-NN or Spectronaut to phosphosite tables using default settings and a localization cutoff of 0.75 (Class I sites) (11). The DIA-NN output table was reformatted by renaming all columns and entries calculating peptide positions to conform to the format required for the plug-in tool. For collapsing, Perseus took only phosphorylation into account. During collapsing phosphopeptide ions to phosphosites, each phosphosite corresponding to the same peptide obtains the same intensity, however imputation may lead to differences in fold changes. If



the same phosphosite was identified on different peptides, which may also have modifications other than phosphorylation or different charge states, the intensities were summed up.

#### Statistical Analysis

Visualization and statistical analyses were performed using the output tables of DIA-NN or Spectronaut with Python (3.8, Jupyter notebook) and the packages pandas (1.4.2) and pyfaidx (0.6.1) for data accession and py\_diAID (0.0.16), AlphaMap (0.1.10), matplotlib (3.4.3), and seaborn (0.11.2) for visualization. The statistical analysis of the EGF experiment was performed in Perseus (1.6.2.2). Log<sub>2</sub>-transformed intensities were filtered for 100 % valid values in at least one condition. The missing values were replaced drawing from a normal distribution (width 0.3 and downshift 1.8). Next, we applied the two-sided Student's t-test ( $S_0=0.1$ , FDR = 0.05) to obtain the significantly changing phosphorylated peptides. A Fisher's exact test was performed for GO term and KEGG pathway enrichment analysis ( $p\text{-value}<0.002$ ).

### RESULTS

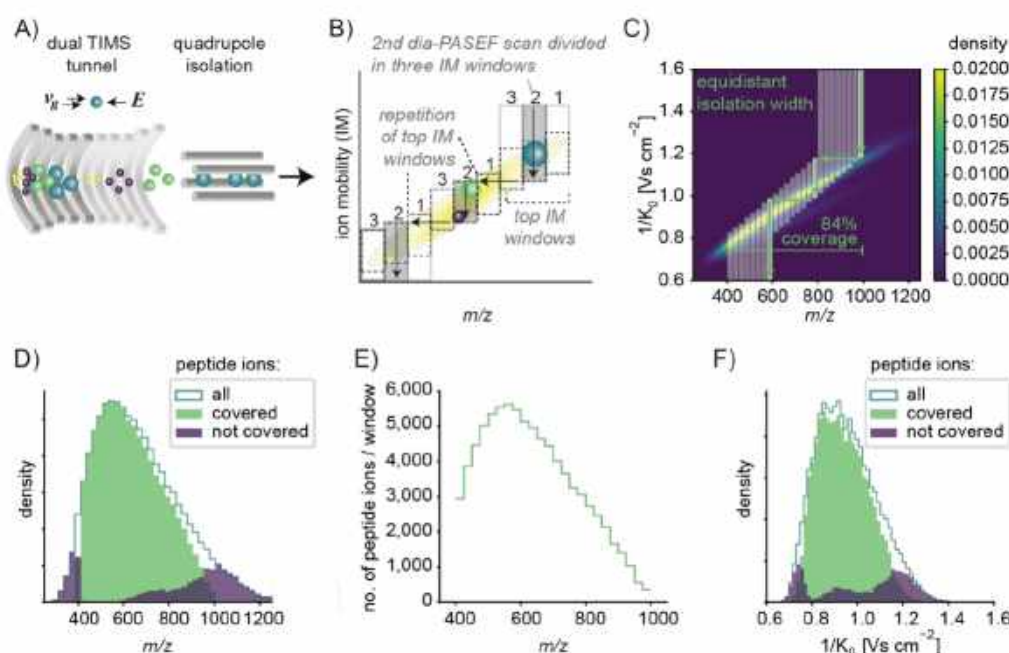
#### Principle and limitations of the original dia-PASEF window design

In the timsTOF mass spectrometer (Bruker Daltonics), a dual TIMS tunnel releases the captured peptide ion species individually as a function of their mobility. In a PASEF MS/MS scan, a quadrupole transmits part of the ion beam where the precursor  $m/z$  values fall into a pre-defined isolation window (Fig. 1A). These precursors are subsequently fragmented by applying a particular collision energy. A downstream TOF analyzer acquires high-resolution mass spectra. In dia-PASEF, changing the quadrupole position is synchronized to the ion mobility elution, increasing the MS efficiency because the isolation window is placed on top of the precursor cloud (6). This movement happens in distinct steps and thereby divides one PASEF scan into multiple ion mobility windows. The quadrupole isolation window is first placed at high  $m/z$  for a certain amount of time, after which it jumps to a position in the lower  $m/z$  range. This transition point corresponds to a particular ion mobility value for each dia-PASEF scan. In each subsequent dia-PASEF scan, the starting  $m/z$  window is offset to lower values (Fig. 1B, C). Together, these isolation windows cover a large proportion of the  $m/z$  and the ion mobility dimensions, constituting a two-dimensional acquisition scheme (Fig. 1B).

Due to software constraints, the original dia-PASEF methods (6) comprise a repeating pattern of the top ion mobility windows per dia-PASEF scan. This leads to a configuration with equidistant quadrupole isolation widths (Fig. 1B). As a result, covering a wide  $m/z$  range comes at the cost of a high cycle time and reduced quantitative accuracy due to lower elution peak coverage. Alternatively, many peptide ions outside the  $m/z$  range would not be included in the acquisition scheme (Fig. 1C, D).

Moreover, when using equidistant isolation windows, the distribution of peptide ions per window is imbalanced, resulting in a high spectral complexity in highly dense regions (Fig. 1E). Lastly, this scheme for acquisition window setting is also suboptimal in the ion mobility dimension (Fig. 1F).

bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



**Figure 1: Principle of dia-PASEF on a timsTOF with equidistant two-dimensional isolation windows.**

- A) Schematic of a TMS tunnel followed by quadrupole isolation
- B) dia-PASEF acquisition scheme depicting three dia-PASEF scans divided into three ion mobility (IM) windows. Vertical arrows indicate the elution of the ions with decreasing electrical field and horizontal arrows indicate the movement of the quadrupole. The pattern of the top ion mobility windows is repeated and the top and bottom ion mobility windows are extended to the upper and lower ion mobility range, respectively.
- C) Original dia-PASEF acquisition scheme (6) plotted on a kernel density distribution of all precursors. One dia-PASEF scan is divided into three ion mobility windows by three distinct movements of quadrupole isolation. This scheme comprises eight dia-PASEF scans with equidistant isolation width covering in total 84% of the peptide ion population.
- D) Histogram of  $m/z$  of all peptides covered by the acquisition method in (C), and peptides not covered by the method but identified in a separately recorded spectral library.
- E) Number of peptide ions per isolation window.
- F) Histogram of ion mobilities of all peptides covered by the acquisition method, and peptides not covered by the method but identified in a separately recorded spectral library.

The subfigures C-F are based on a reference proteome library (see Experimental Procedures).

### Establishing an optimal dia-PASEF window design

We first investigated the optimum balance between the number of dia-PASEF scans and ion mobility windows per dia-PASEF scan to obtain a deep proteome coverage and quantitative accuracy. As described above, the original dia-PASEF method included three ion mobility windows per dia-PASEF scan. Having more ion mobility windows per dia-PASEF scan reduces cycle time but also diminishes precursor coverage due to smaller isolation windows in the ion mobility dimension (supplemental Fig. S5A, B). For instance, splitting the isolation width into two parts halved the complexity per spectrum and thereby increased identifications. However, doubling the number of dia-PASEF scans increases cycle time, which worsens the quantitative accuracy since only half as many data points are collected over one elution peak (supplemental Fig. S5A). We tested the impact of increasing the number of ion mobility windows per dia-PASEF scan and found that two ion mobility windows per dia-PASEF scan are optimal (supplemental Fig. S5C). Six points per elution peak are thought to be necessary for accurate quantitation (43). In the case of 21 minute gradients (60 SPD), we empirically found



a median peak width of 8.27 s with our set-up (supplemental Fig. S5D) only considering precursors with a CV value below 20%. Each individual dia-PASEF scan takes around 100 ms plus one 100 ms MS1 scan per cycle and overhead time. Hence, 12 dia-PASEF scans amount to a cycle time of 1.38 s, ensuring adequate quantitative representation of the LC elution peak (see Experimental Procedures, supplemental Fig. S5E).

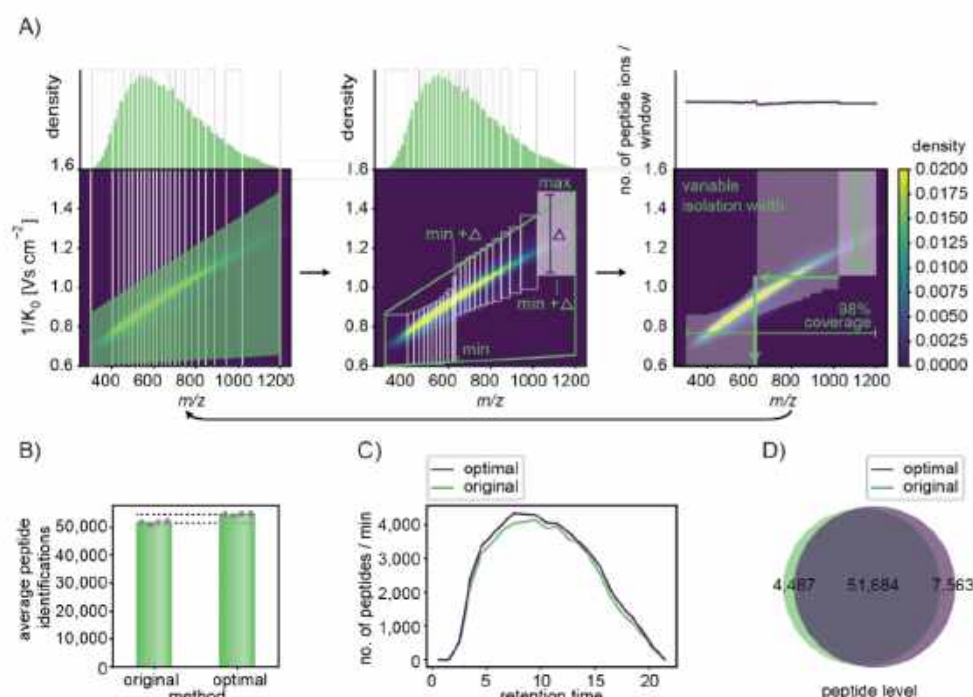
Given the limitations of our previous two-dimensional acquisition scheme, we needed to place and adjust  $m/z$  and ion mobility isolation windows flexibly. Existing tools such as “Define dia-PASEF Region” in Compass DataAnalysis (Bruker) or the “dia-PASEF window Editor” in TimsControl (Bruker) require the manual fitting of the scan area onto the peptide ion population and only generate isolation windows with equidistant widths. Therefore, we developed a Python package for Data Independent Acquisition with an Automated Isolation Design (py\_diAID). It places two-dimensional dia-PASEF acquisition schemes in the  $m/z$ -ion mobility plane based on desired parameters (number of dia-PASEF scans, covered  $m/z$  and ion mobility range, and cycle time) and the empirical acquired reference data, which can be a proteomics library containing precursor ion information. The algorithms in py\_diAID optimally adjust the variable quadrupole isolation widths according to the precursor density, aiming for an equal number of precursors fragmented per isolation window. Our simulations show that variable isolation widths enable short acquisition cycles covering essentially the entire  $m/z$ -ion mobility range (Fig. 2A, right panel).

Our algorithm first bins the precursor ion populations equally along the  $m/z$ -dimension. A trapezoid defines the extent of scan area and the position of the acquisition scheme in the  $m/z$ -ion mobility plane (Fig. 2A, left panel). Based on this, py\_diAID calculates the optimal dimensions of each isolation window (Fig. 2A, middle panel) and extends the top and bottom ion mobility windows to the limits of the measured ion mobility range to maximize the covered peptide ion population (Fig. 2A, right panel and supplemental Fig. S6). The selected mass window of the quadrupole jumps at the determined transition point of each ion mobility window within each dia-PASEF scan. In each subsequent dia-PASEF scan, the starting  $m/z$  window is offset to lower values based on the individual width of the previous window (Fig. 2A). Next, py\_diAID evaluates the generated acquisition scheme based on the covered precursor ions of an experimentally acquired library or subset thereof, for example one filtered by a charge state or by a population of modified peptides. This is a multivariate non-linear optimization problem, and we used the `gp_minimize` module provided by the Scikit-Optimize (skopt) library in Python to perform this task that is highly used in machine and deep learning for the hyperparameter optimization (see Experimental Procedures). Its inputs are the trapezoid corners and it iteratively decides which parameters should be tested next based on the above evaluation. This process is repeated for many iterations (about 200 in practice, supplemental Fig. S7) until it converges to the best window placement. py\_diAID is available as a Python module, a command-line interface, and a graphical user interface on all major operating systems under an Apache 2.0 license (supplemental Fig. S8). The source code is freely available on GitHub (<https://github.com/MannLabs/pydiAID>).

We first benchmarked the optimal dia-PASEF methods designed with py\_diAID against the original dia-PASEF method. The optimal dia-PASEF method calculated by py\_diAID covered 99% of all doubly and 94% of all triply charged precursors in the ‘reference library’, that was generated with FragPipe. This compares to 88% and 71% coverage with the original dia-PASEF. The original dia-PASEF method had already been extensively and manually optimized for the short gradient lengths and the tryptic HeLa digest employed here. This explains why the number of experimentally identified proteins is very similar between both methods



(supplemental Table 1). However, even in this case, py\_diAID's optimal acquisition scheme increased the number of identified peptides by 6% in single-run injections (Fig. 2B) and across the entire retention time (Fig. 2C), while the number of peptide identifications in replicate injections deviates only by 1%. Inspection of the data shows that the additional peptides originate both from the previously not covered regions and from the most dense elution times. More than 80% of all identified peptides were commonly identified by both methods (Fig. 2D). In other applications, such as phosphoproteomics, the gains by py\_diAID were much larger (see below Fig. 5).



**Figure 2: py\_diAID algorithm and evaluation.**

A) py\_diAID design of the optimal acquisition scheme and window placement for a 21 min gradient (60 SPD, Evosep) with variable widths to balance the distribution of peptide ions, providing nearly complete peptide ion coverage.

The left panel illustrates the first steps of the py\_diAID algorithm: defining the  $m/z$ -range of interest, binning the peptide ions in the  $m/z$ -dimension and definition of the scan area in the IM dimension.

middle panel: Calculation of the isolation window dimensions and coordinates based on the scan area.

right panel: Extension of the isolation windows to the limits of the ion mobility ranges. The arrow at the bottom indicates that the py\_diAID algorithm evaluates the new acquisition scheme, defines the following test set of scan area parameters by Bayesian optimization, and resumes with the steps in the left panel. This is repeated for a user-defined number of iterations (more details in supplemental Fig. S6).

A is plotted on top of a kernel density distribution based on the reference proteome library.

B) Average peptide identifications by the original and optimal dia-PASEF methods.

C) Number of peptides identified per minute over the entire retention time.

D) Venn diagram showing the shared and unique peptides identified by both methods.

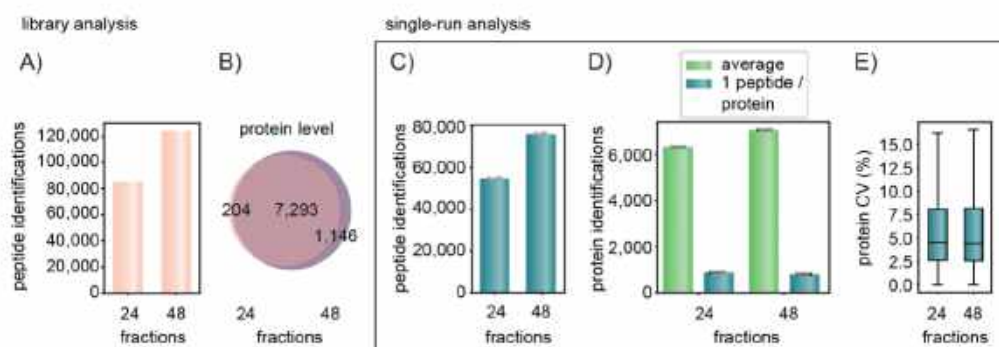
Data in B to D are from quadruplicate injections of 200 ng tryptic HeLa digest with a 21 min gradient and analyzed with the reference proteome library.

### Deep proteome coverage in short LC-gradients

We next investigated if coupling our optimized dia-PASEF methods with project-specific, in-depth libraries yields higher peptide identification and improves quantification accuracy. To generate such an in-depth library, we separated 15  $\mu$ g of the HeLa sample that we also used for single dia-PASEF acquisitions into 48 concatenated fractions using high pH reverse phase chromatography of the Spider fractionator (see Experimental Procedures) (31). These fractions were measured in dda-PASEF mode and again analyzed with FragPipe and its SpecLib workflow. We compared our 'reference library' generated with limited sample amount (2.5  $\mu$ g proteolytic digest) and 24 fractions to the new one with ample sample amount (15  $\mu$ g) and twice as many fractions. As expected, the latter was substantially larger, containing 45% more peptides (counting all modifications) and 13% more proteins. Altogether, this deep library constructed from 21 min runs comprised 124,155 peptides and 8,439 different protein groups (Fig. 3A, B).

Next, we compared single dia-PASEF runs with reference vs deep library using DIA-NN and found a corresponding increase in the proteome depth (39% more peptides and 12% more proteins) (Fig. 3C, D). Using the deep library identified  $76,214 \pm 1,021$  peptides and the reference library  $51,711 \pm 641$  peptides (Fig. 3C). With the deep library, an astounding  $7,056 \pm 8$  proteins were identified with our optimized acquisition scheme in each of four replicate runs on average. Specifically, with the reference library, DIA-NN reported 14% significant protein identifications on the basis of one peptide and this percentage decreased slightly to 11% with the deeper library (Fig. 3D).

Quantitative reproducibility between the quadruplicates was virtually identical when using the reference or deep library (4.5% vs 4.4% on protein level and 12.1% vs 13.45% on peptide level) (Fig. 3E). Taken together, we found that single run identification benefited from a project-specific, in-depth library while maintaining the accuracy of quantification. We therefore used the 48 fractions library for all 21 min runs to generate equivalent libraries for evaluating a range of gradient lengths as described next (referred to as 'project-specific deep libraries').



**Figure 3: Workflow optimization for the 21 min gradient with project-specific deep libraries**

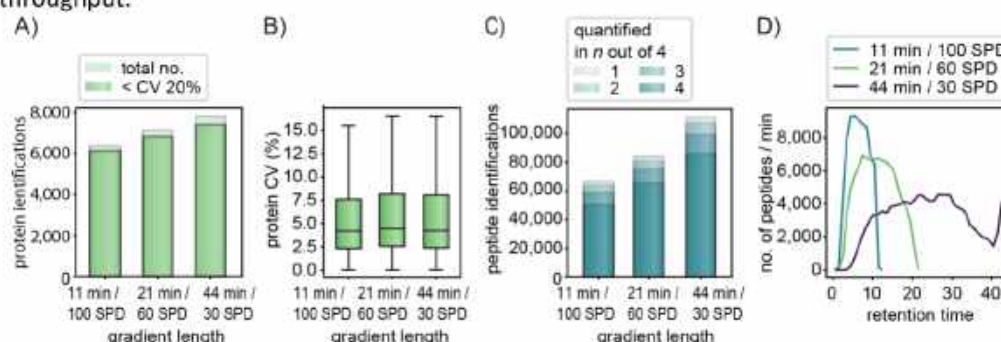
- A) Peptides identified of the reference vs. the project-specific deep library for 21 min runs.  
 B) Shared proteins and depth on the protein level in the two libraries.  
 C) Average peptide identification of four single-run injections. This data and the one in (D) and (E) were generated from quadruplicate injections of 200 ng tryptic HeLa digest acquired with a 21 min gradient and searched with the reference (24 fractions) or project-specific library (48 fractions).  
 D) Average protein identifications and identifications with only one peptide in the single runs.  
 E) Coefficients of variation at the protein level based on the MaxLFQ algorithm of DIA-NN. Boxplots show the median (center line), 25<sup>th</sup>, and 75<sup>th</sup> percentiles (lower and upper box limits, respectively), and the 1.5 $\times$  interquartile range (whiskers). n = 6,384 (24 fractions) and 7,121 (48 fractions) shown in panel C.



We next investigated the effect of even shorter gradients as well as somewhat longer gradients on proteome depths and quantitative accuracy. As before, each library was acquired with dda-PASEF and 15  $\mu$ g HeLa lysate separated into 48 fractions. Extending the gradient to 44 min (30 SPD method on the Evosep One system) identified an average of  $7,756 \pm 6$  proteins based on  $100,900 \pm 634$  peptides (including all modifications). This represents an identification increase of 10% on protein level in comparison to the 21 minutes gradient. The median CV between the quadruplicates was 4% at the protein level for these technical replicates, and 7,393 protein groups had CVs below 20% (Fig. 4A, C).

We expected that the fast scan rate of the timsTOF, together with our optimized method might still accurately measure a large part of the proteome even in very short gradients (6, 32). Indeed, the 100 SPD method (11 min gradient) still identified  $6,285 \pm 18$  proteins ( $59,811 \pm 368$  peptides). Quantitative accuracy reported by DIA-NN did not suffer and remained at a median CV of 4%. Taking only the proteins with CVs equal or below 20%, the 100 SPD method still resulted in 6,121 proteins, covering 83% of proteins that could be accurately quantified with the 44 min gradient while substantially reducing the analysis time (Fig. 4A). Rank order reproducibility was also high for these technical replicates for all gradient lengths (supplemental Fig. S9-10,  $r=0.999$  for proteins and  $r=0.992$  for peptides). As expected, the number of peptides identified per minute decreased when increasing the gradient length while the 11-min gradient reached the highest numbers (9,330 peptides per minute translating to 155 peptide identifications per second at the apex, Fig. 4D).

In conclusion, our data show that our improved workflow constitutes a powerful technological platform capable of accurately quantifying a large part of the proteome at high throughput.



**Figure 4: Comparison of different gradient lengths/ throughput based on single-run analysis.**

A) All single-run identifications and those with a CV < 20% for the 11 min, 21 min and 44 min gradients.  
 B) Coefficients of variation at the protein level based on the MaxLFQ algorithm of DIA-NN. Boxplots show the median (center line), 25<sup>th</sup>, and 75<sup>th</sup> percentiles (lower and upper box limits, respectively), and the 1.5 $\times$  interquartile range (whiskers).  $n = 6,341$  (11min / 100 SPD) and  $7,121$  (21min / 60 SPD), and  $7,802$  (44min / 30 SPD) shown in panel A.  
 C) Analysis of peptide quantification in n out of four technical replicates shows that the large majority is quantified consistently.  
 D) The number of peptides per second over the retention time for the three gradient lengths.  
 The data was acquired in quadruplicate injections of 200 ng HeLa digest and analyzed with 48 fraction, dda-PASEF libraries each recorded with the corresponding gradient length. 11-min library: 8,553 proteins and 122,105 peptides, 21-min library: 8,439 proteins and 124,155 peptides, 44-min library: 9,461 proteins, 175,839 peptides.



#### Comparison of proteome results between DIA-NN and Spectronaut

The above analyses were all performed with the DIA-NN package. To determine if our results depend on the software used, we employed Spectronaut (Biognosys) (3), another widely used software package (11, 44). This revealed that both packages identified comparable numbers of proteins. For instance, in the 60 samples per day method, Spectronaut reported 7,285 significant protein groups, whereas DIA-NN reported 7,056 (supplemental Fig. S11 A). In the version tested (Spectronaut 16), this also held for even shorter gradients (6,250 vs. 6,285). Having established that the overall protein numbers are similar, we next investigated the overlap between the found proteins. As DIA-NN has a different protein grouping algorithm from Spectronaut, we performed this analysis on the level of genes and peptide precursors. Employing similar grouping schemes at the gene level showed a high level of concordance, with 548 genes unique to Spectronaut and 208 unique to DIA-NN out of a total of 7,668 identified genes for both (supplemental Fig. S11 B). For the total of 128,002 identified peptide precursors, the discrepancy was somewhat larger, with 28% unique identifications for Spectronaut and 5% for DIA-NN (supplemental Fig. S11 C). Overall, based on these proteome results, we conclude that the gains achieved by py\_diAID are independent of the DIA analysis software used.

#### Rapid phosphoproteomics with optimal isolation window design

Phosphorylation, one of the most prevalent and most studied post-translational modification, refers to the addition of a phosphoryl group – usually on serine, threonine or tyrosine amino acid residues. This introduces a mass and ion mobility shift on the modified peptides, indicating that analysis of phosphopeptides can benefit from the additional ion mobility dimension in PASEF (45, 46). To date, dia-PASEF has not been explored in a large-scale study of the phosphoproteome or any other post-translationally modified sub-proteome.

It is well known that the ion mobility dimension separates peptides in clouds primarily reflecting their charge status. In the timsTOF case, Figure 5A depicts dense clouds containing doubly, triply, and quadruply charged peptide ions (47). In the case of phospho-enriched samples, projecting the distribution of phosphorylated peptides into the  $m/z$  and IM space revealed a substantial shift of ion cloud to higher  $m/z$  values and higher IM values, due to the 80 Da increase in their mass, higher charge states and conformational changes upon phosphorylation (Fig. 5B). These observations suggest that dia-PASEF methods need to be tailored for phosphoproteomics. To this end, we first generated an in-depth phospho-library from EGF stimulated HeLa cells that were separated into 76 fractions and then enriched for phosphorylated peptides. These enriched fractions were measured with the 60 SPD method, dda-PASEF in little more than one day. We analyzed the results both by FragPipe combined with DIA-NN and by Spectronaut 16 (see Experimental Procedures). This generated an in-depth library of 187,730 modified or unmodified peptides, 123,133 phosphopeptides and 107,154 phosphosites for DIA-NN. Spectronaut 16 obtained very similar results (194,309 modified or unmodified peptides, 132,270 phosphopeptides and 114,158 phosphosites). The overlap between phosphopeptides was 50% (supplemental Fig. S13A).

When we simulated the coverage of the original dia-PASEF method for the 21 min gradient (6), we found that it only reached a coverage of 34% of phosphopeptide ions in our deep phospho-library, in contrast to the 81% achieved for unmodified peptides (Fig. 5C). Therefore, we used our phospho-library as input for py\_diAID to obtain a dia-PASEF method

tailored for phosphoproteomics. This resulted in a theoretical coverage of 93% of all doubly charged and 92% of all triply charged phosphopeptide ions (Fig. 5D).

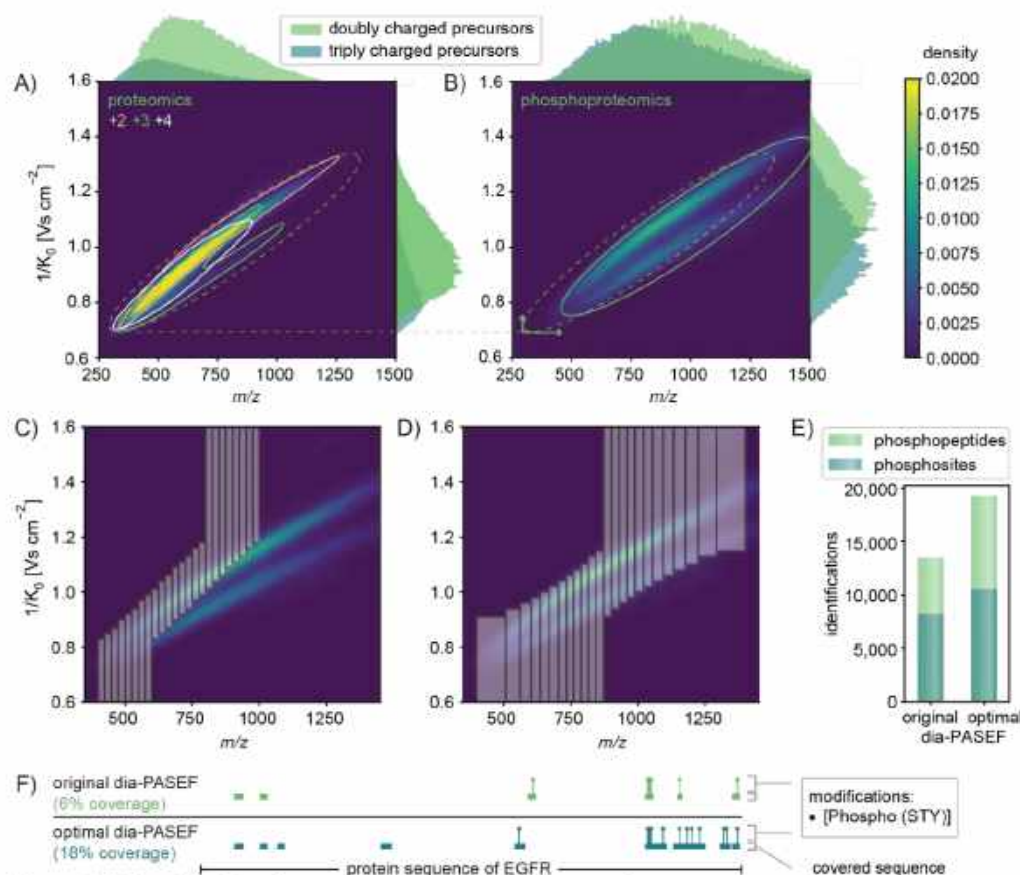
We next utilized this optimal dia-PASEF phospho-method to measure the samples containing phosphorylated peptides enriched from 100 µg digest of EGF stimulated HeLa cells. We first analyzed the resulting files with DIA-NN against our deep phospho-library. In agreement with our simulations, the original dia-PASEF method identified 8,199 phosphosites and 13,485 phosphopeptides whereas the optimal method detected 28% more phosphosites (10,510) and 43% more phosphorylated peptides (19,258) (Fig. 5E). To illustrate this further, we mapped the experimentally acquired phosphopeptides to the EGF receptor (EGFR) sequence essential for transmitting the EGF signal using AlphaMap (48). This revealed that the optimal dia-PASEF phospho-method doubled the number of detected phosphosites to a total of 14 (Fig. 5F).

The intensities of the phosphopeptides detected in our deep FragPipe phospho-library in dda-PASEF mode and 76 fractions span almost seven orders of magnitude (supplemental Fig. S12A). When searching single dia-PASEF phospho-runs against our phospho-library using DIA-NN, we found that single short gradients covered 21% of the phosphopeptide sequences, ranging from 12% in the most abundant quintile to 0.3% in the least abundant one (Suppl. Fig. S12A). Apart from the statistical analysis, the AlphaViz package (49), based on AlphaTims (50), allows visualization of any phosphopeptides of interest. This is shown for the phosphopeptide ELVEPLT[Phospho (STY)]PSGEAPNQALLR on EGFR, where the distinct precursor and fragment peaks are clearly visible in the retention time dimension and even more important in the retention time – ion mobility plane, supporting the DIA-NN assignment (supplemental Fig. S12B, C).

Next, we analyzed the same single-run phospho dataset with Spectronaut. To our surprise – especially given the comparable results at the proteome level – Spectronaut drastically increased the number of identified phosphosites to 28,980 (supplemental Fig. S13B). This was even more pronounced for identified phospho-peptides (72,216, supplemental Fig. S13C). Accordingly, the common overlap of phosphosites was only 26% (supplemental Fig. S13B).

We do not know the origin of this large discrepancy, but we encourage the providers of these software packages to resolve this, especially as the code is not available for inspection. In the context of our study, we decided to continue with the more extensive Spectronaut results, as they appeared to still correctly represent the regulation in the EGFR signaling experiment described below.





**Figure 5: Method optimization specifically for phosphoproteomics.**

- A) Peptide distribution of a proteomics digest displayed as kernel density estimation dependent on the charge and histograms of the abundance of differently charged precursors based on our deep proteomics library.
- B) Peptide distribution of a phosphoproteomics digest displayed as kernel density estimation and histograms of the abundance of differently charged precursors based on our phosphopeptide library.
- C) Original dia-PASEF method plotted on top of the phosphopeptide library.
- D) Optimal dia-PASEF method tailored to the phospho-library.
- E) Identified phosphosites and phosphopeptides based on quadruplicates of 100  $\mu$ g EGF-stimulated and enriched HeLa digest, separated within 21 min and searched with DIA-NN against the phospho-library.
- F) AlphaMap visualization (48): Protein sequence coverage of the epidermal growth factor receptor (EGFR) depending on the acquisition method.

### In-depth phosphoproteomics analysis of the EGF-signaling pathway

To benchmark our optimal dia-PASEF workflow, we chose the well-studied epidermal growth factor (EGF) signaling pathway in HeLa cells. The binding of EGF to the EGF receptor (EGFR) results in the activation of downstream kinases, which phosphorylate a repertoire of numerous substrates, regulating diverse cellular processes (51). We aimed to quantitatively and accurately measure the differential phosphorylation of proteins involved in this signaling pathway using our rapid and sensitive method. To this end, EGF-treated and control samples were collected in three biological replicates, digested into peptides and enriched for phosphorylated peptides (see Experimental Procedures). Subsequently, we measured the enriched phosphopeptides with dia-PASEF in 21 minutes and searched the deep



phosphopeptide library that we already employed for the method optimization described above with Spectronaut 16.

With our workflow, we quantified 46,136 phosphorylation sites on 4,300 proteins. Of these, 35,537 sites were identified with a high site localization probability (75%, Class I sites (52)) and 20,001 were quantified in all replicates of at least one experimental condition (Fig. 6A). The dia-PASEF workflow allowed high reproducible quantification demonstrated by a median Pearson coefficient above 0.92 for replicates within conditions (Fig. 6B). Remarkably, a full 26% (5,200, 5% FDR) and 10.5% (2,117, 1% FDR) of phosphorylation sites were significantly modulated upon EGF treatment (Fig. 6C).

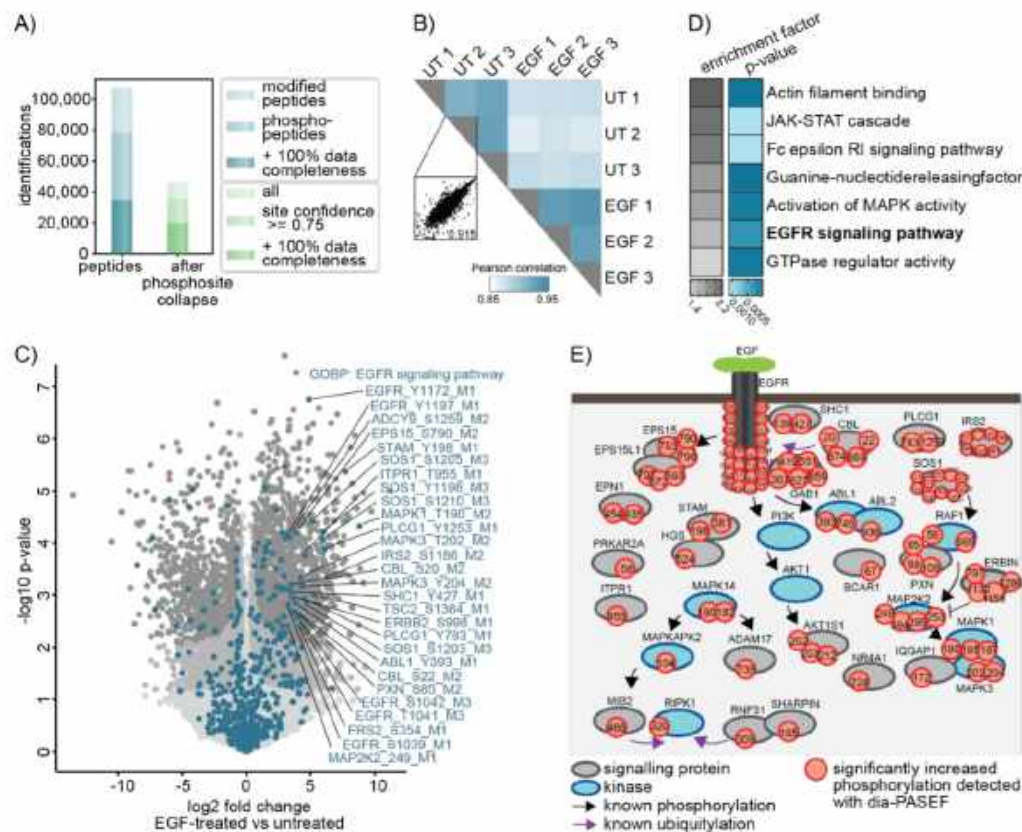
As expected, Gene Ontology (GO) enrichment analysis revealed strong overrepresentation of proteins involved in the EGFR signaling pathway (GOBP) and related pathways among the significantly EGF-upregulated phosphoproteins (Fig. 6D). Most are known to be critical for intact EGF signaling. For example, we detected phosphorylation of T693, Y1110, Y1172, Y1197 on the receptor EGFR itself, Y427 on the adaptor protein Src Homology 2 Domain-Containing-Transforming Protein C1 (SHC1), Y659 on Growth Factor Receptor Bound Protein 2-Associated Protein 1 (GAB1) and on the downstream kinases Mitogen-Activated Protein Kinase 2 (MAP2K2) (T394) and Mitogen-Activated Protein Kinase 1 and 3 (MAPK1/3) (T185/Y187, T202/Y204) (53) (Fig. 6C, E). These phosphosites are typically used to examine EGF signaling with classical methods such as immunoblotting or with targeted mass spectrometry (9, 10, 54). These approaches, however, only allow relatively low throughput analyses, that require dedicated assay development procedures or the generation of phosphospecific antibodies. In contrast, by combining the automated phosphoenrichment on the BRAVO platform with the robust Evosep and timsTOF setup, our approach achieves 60 SPD. This allows us to track and accurately quantify the induction of more than 60 phosphorylation events on proteins critical for EGF signaling (part of GOBP: EGFR signaling pathway) within a single 21-min run (supplemental Fig. S14). Importantly, besides the phosphorylations of the classical EGF signaling members, many other signaling events that, for example, result from signaling crosstalk downstream of the EGF receptor can also be detected, including S897 of the Ephrin Type-A Receptor 2 (EPHA2), S339 of the C-X-C Motif Chemokine Receptor 4 (CXCR4) and T701 of Erb-B2 Receptor Tyrosine Kinase 2 (ERBB2) (supplemental Fig. S14).

To identify functionally important phosphorylation events not directly linked to EGF signaling, we matched the functionality prediction score developed by Beltrao and co-workers to the upregulated phosphorylation events (55). We identified 659 phosphosites with a high functional score of >0.5 to be significantly upregulated, which are not part of the GOBP term 'EGFR signaling pathway' (FDR<0.05) (supplemental data 1). These include EGF-induced phosphorylation of E3 ligases like Mindbomb Homolog 2 (MIB2) (S309) and members of the linear ubiquitin chain assembly complex Ring Finger Protein 31 (RNF31) (S466) and Sharpin (S165), which are most frequently studied in the context of TNF signaling (supplemental Fig. S14) (56–59). Similarly, phosphorylation of Receptor Interacting Serine/Threonine Protein Kinase 1 (RIPK1) on S320, which prevents TNF-induced cell death, was also increased upon EGF signaling (supplemental Fig. S14) (60, 61). This phosphorylation is mediated by MAP kinase-Activated Protein Kinase 2 (MAPKAPK2), which is activated upon EGF stimulation demonstrated by its increased phosphorylation at T334. These are just some examples of functional candidates whose role in EGF signaling has still to be determined.

Together, this EGF study demonstrates the quantitative capabilities of the dia-PASEF-based phosphoproteomics workflow. We conclude that efficient analysis of ions separated in the IM

bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

and  $m/z$  space enables the investigation of signaling pathways with high sensitivity in a high-throughput manner.



**Figure 6: The dia-PASEF workflow allows the robust detection of characteristic EGF signaling events.**

- A) Numbers of all identified phosphopeptides and phosphosites before and after filtering for localization probability and data completeness.
- B) Phosphoproteome Pearson correlation matrix. Scatter plot shows the correlation of replicates within a condition.
- C) Volcano plot of phosphosites regulated upon 15 min of EGF-treatment in HeLa cells vs untreated cells. (two-sided Student's t-test,  $FDR < 0.1$  = grey,  $FDR < 0.5$  = dark grey). Protein's part of the GOBP term 'EGFR signaling pathway' are highlighted in turquoise.
- D) Fisher's exact test of proteins with significantly increased phosphosites upon EGF treatment (p-value  $< 0.002$ ). Enrichment annotations are GOBP, GOMF and KEGG.
- E) Scheme of significantly upregulated phosphosites that were detected in this study and are part of the GOBP term 'EGFR signaling pathway' and/or changed significantly upon EGF stimulation ( $FDR < 0.05$ ).



## DISCUSSION

The optimal placement of dia-PASEF windows in the two-dimensional  $m/z$  and ion mobility space is not trivial. We here developed py\_diAID which is available on GitHub at MannLabs and is installable as a Python module with a command line interface or as a GUI on Windows, Mac and Linux. It adjusts the isolation window width to the precursor density, and optimally positions the isolation design in the  $m/z$ -IM space. This leads to near-complete theoretical precursor coverage for proteomics. Compared to the original dia-PASEF method (6), the gains for phosphorylated precursors are especially striking (34% vs. 93%).

MS-based proteomics is a rapidly developing technology. For perspective, to cover ten thousand proteins we had to measure the samples for twelve days with four-hour gradients ten years ago (62). Here, we coupled a robust, high throughput LC system to the TIMS-qTOF instrument employing the rapid sampling speed of a TOF analyzer. It offers short gradients and also low overhead time, enhancing the overall throughput capabilities (27). With this, we generated in-depth project-specific libraries of 9,461 proteins in only 13% of the previous measurement time. Furthermore, once the libraries are ready, subsequent proteome characterization using py\_diAID generated methods happens in only 44 minutes to a depth of 7,700 proteins (less than 1% of the measurement time necessary ten years ago). Our workflow is also twice as fast as currently employed high throughput screening strategies for cancer proteomics, while achieving greater proteome depth on cell lysate (63–65).

So far, there have been only a few reports of the timsTOF principle on phosphoproteomics (28). Here, we show that this instrument is capable of in-depth phosphoproteomics with very high sensitivity. Specifically, we identified thirty-five thousand phosphosites in only 21 minutes in triplicates from 100  $\mu$ g EGF-stimulated HeLa cell digests. Our workflow opens up the possibility to measure multiple pathways in a short time. We demonstrated that quantitatively analyzing the regulated phosphoproteome covers the well-studied EGF signaling pathway together with auxiliary pathways. Interestingly, our workflow is even faster than selected reaction monitoring employed as a targeted screening method for assessing the activation of signaling pathways (9). However, our method is generic to any pathway and applicable in principle to the entire phosphoproteome.

In the current implementation, the dia-PASEF windows are adjusted based on empirical data before the acquisition. These adjustments could also be implemented in real-time based on the precursor density achieving an acquisition design optimized to the individual time points of an entire gradient. Furthermore, we employed in-depth libraries. While they can be generated quickly, current developments of *in silico* generated DIA libraries or direct DIA methods may soon obviate the need for this step. Likewise, we expect that py\_diAID will perform similarly for other PTMs.

## ACKNOWLEDGEMENT

We thank Nagarjuna Nagaraj for introducing us to the acquisition software timsControl, which is a basis for py\_diAID, and our colleagues in the Department of Proteomics and Signal Transduction at the Max Planck Institute of Biochemistry. We are particularly grateful for the help from Ankit Sinha, Igor Paron, Maria Wahle, Corazon Ericka Mae Itang, Isabell Bludau, Constantin Ammar and Medini Steger.



bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

#### FUNDING AND ADDITIONAL INFORMATION

This study was supported by the Max-Planck Society for Advancement of Science, the Deutsche Forschungsgemeinschaft (DFG) project 'Chemical proteomics inside us' (grant 412136960) and by the Bavarian State Ministry of Health and Care through the research project DigiMed Bayern ([www.digimed-bayern.de](http://www.digimed-bayern.de)).

#### DATA AVAILABILITY

All dia-PASEF parameter files required for the acquisition, mass spectrometry raw files corresponding to the spectral libraries and single-run experiments, and output information from DIA-NN and MS-Fragger have been deposited with the ProteomeXchange Consortium via the PRIDE partner (66) repository with the dataset identifier PXD034128. Supplemental data 2 is a roadmap linking the raw files. Homo sapiens (taxon identifier: 9606) proteome databases were downloaded from <https://www.uniprot.org>. py\_diAID is a fully open-source package, and the code is freely available under the Apache 2.0 license at <https://github.com/MannLabs/pydiAID>.

#### AUTHOR CONTRIBUTIONS

P.S., M.T., M.C.T., F.M.H., Ö.K., A.-D.B., F.M and M.M. conceptualized and designed the study; P.S. and F.M. conceived the tool py\_diAID; P.S., M.T., M.C.T. and F.M.H. performed experiments; P.S., E.V. and S.W. developed the tool py\_diAID; P.S., M.T., M.C.T., E.V., F.M.H., Ö.K., A.-D.B., F.M and M.M. analyzed the data; P.S., M.T., S.W., Ö.K. and M.M. wrote the manuscript with input from all authors.

#### CONFLICT OF INTEREST

M. M. is an indirect investor in Evosep Biosystems. All other authors declare that they have no conflicts of interest with the contents of this article.

#### SUPPLEMENTAL DATA

This article contains supplemental data.

#### REFERENCES

1. Aebersold, R., and Mann, M. (2016) Mass-spectrometric exploration of proteome structure and function. *Nature* 537, 347–355
2. Ludwig, C., Gillet, L., Rosenberger, G., Amon, S., Collins, B. C., and Aebersold, R. (2018) Data-independent acquisition-based SWATH - MS for quantitative proteomics: a tutorial. *Mol. Syst. Biol.* 14, 1–23
3. Bruderer, R., Bernhardt, O. M., Gandhi, T., Miladinović, S. M., Cheng, L. Y., Messner, S., Ehrenberger, T., Zanotelli, V., Butscheid, Y., Escher, C., Vitek, O., Rinner, O., and Reiter, L. (2015) Extending the limits of quantitative proteome profiling with data-independent acquisition and application to acetaminophen-treated three-dimensional liver microtissues. *Mol. Cell. Proteomics* 14, 1400–1410
4. Gillet, L. C., Navarro, P., Tate, S., Röst, H., Selevsek, N., Reiter, L., Bonner, R., and

bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- Aebersold, R. (2012) Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: A new concept for consistent and accurate proteome analysis. *Mol. Cell. Proteomics* 11, 1–17
5. Chapman, J. D., Goodlett, D. R., and Masselon, C. D. (2014) Multiplexed and data-independent tandem mass spectrometry for global proteome profiling. *Mass Spectrom. Rev.* 33, 452–470
  6. Meier, F., Brunner, A. D., Frank, M., Ha, A., Bludau, I., Voytik, E., Kaspar-Schoenefeld, S., Lubeck, M., Raether, O., Bache, N., Aebersold, R., Collins, B. C., Röst, H. L., and Mann, M. (2020) diaPASEF: parallel accumulation–serial fragmentation combined with data-independent acquisition. *Nat. Methods* 17, 1229–1236
  7. Venable, J. D., Dong, M. Q., Wohlschlegel, J., Dillin, A., and Yates, J. R. (2004) Automated approach for quantitative analysis of complex peptide mixtures from tandem mass spectra. *Nat. Methods* 1, 39–45
  8. Bekker-Jensen, D. B., Martínez-Val, A., Steigerwald, S., Rüther, P., Fort, K. L., Arrey, T. N., Harder, A., Makarov, A., and Olsen, J. V. (2020) A compact quadrupole-orbitrap mass spectrometer with FAIMS interface improves proteome coverage in short LC gradients. *Mol. Cell. Proteomics* 19, 716–729
  9. Picotti, P., and Aebersold, R. (2012) Selected reaction monitoring-based proteomics: Workflows, potential, pitfalls and future directions. *Nat. Methods* 9, 555–566
  10. Keshishian, H., McDonald, E. R., Mundt, F., Melanson, R., Krug, K., Porter, D. A., Wallace, L., Forestier, D., Rabasha, B., Marlow, S. E., Jane-Valbuena, J., Todres, E., Specht, H., Robinson, M. L., Jean Beltran, P. M., Babur, O., Olive, M. E., Golji, J., Kuhn, E., Burgess, M., MacMullan, M. A., Rejtar, T., Wang, K., Mani, D., Satpathy, S., Gillette, M. A., Sellers, W. R., and Carr, S. A. (2021) A highly multiplexed quantitative phosphosite assay for biology and preclinical studies. *Mol. Syst. Biol.* 17, e10156
  11. Bekker-Jensen, D. B., Bernhardt, O. M., Hogrebe, A., Martinez-Val, A., Verbeke, L., Gandhi, T., Kelstrup, C. D., Reiter, L., and Olsen, J. V. (2020) Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. *Nat. Commun.* 11, 1–12
  12. Hansen, F. M., Tanzer, M. C., Brüning, F., Bludau, I., Stafford, C., Schulman, B. A., Robles, M. S., Karayel, O., and Mann, M. (2021) Data-independent acquisition method for ubiquitinome analysis reveals regulation of circadian biology. *Nat. Commun.* 12, 2020.07.24.219055
  13. Steger, M., Demichev, V., Backman, M., Ohmayer, U., Ihmor, P., Müller, S., Ralser, M., and Daub, H. (2021) Time-resolved in vivo ubiquitinome profiling by DIA-MS reveals USP7 targets on a proteome-wide scale. *Nat. Commun.* 12, 5399
  14. Mund, A., Coscia, F., Hollandi, R., Kovács, F., Kriston, A., Brunner, A.-D., Bzorek, M., Naimy, S., Gjerdrum, L. M. R., Dyring-Andersen, B., Bulkescher, J., Lukas, C., Gnann, C., Lundberg, E., Horvath, P., and Mann, M. (2021) AI-driven Deep Visual Proteomics defines cell identity and heterogeneity. *bioRxiv*, 2021.01.25.427969
  15. Brunner, A., Thielert, M., Vasilopoulou, C., Ammar, C., Coscia, F., Mund, A., Hoerning, O. B., Bache, N., Apalategui, A., Lubeck, M., Richter, S., Fischer, D. S., Raether, O., Park, M. A., Meier, F., Theis, F. J., and Mann, M. (2022) Ultra-high sensitivity mass spectrometry quantifies single-cell proteome changes upon perturbation. *Mol. Syst. Biol.* 18, e10798
  16. Ridgeway, M. E., Lubeck, M., Jordens, J., Mann, M., and Park, M. A. (2018) Trapped ion mobility spectrometry: A short review. *Int. J. Mass Spectrom.* 425, 22–35



bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

17. Fernandez-Lima, F., Kaplan, D. A., Suetering, J., and Park, M. A. (2011) Gas-phase separation using a trapped ion mobility spectrometer. *Int. J. Ion Mobil. Spectrom.* 14, 93–98
18. Fernandez-Lima, F. A., Kaplan, D. A., and Park, M. A. (2011) Note: Integration of trapped ion mobility spectrometry with mass spectrometry. *Rev. Sci. Instrum.* 82,
19. McLean, J. A., Ruotolo, B. T., Gillig, K. J., and Russell, D. H. (2005) Ion mobility-mass spectrometry: A new paradigm for proteomics. *Int. J. Mass Spectrom.* 240, 301–315
20. Meier, F., Brunner, A. D., Koch, S., Koch, H., Lubeck, M., Krause, M., Goedecke, N., Decker, J., Kosinski, T., Park, M. A., Bache, N., Hoerning, O., Cox, J., Räther, O., and Mann, M. (2018) Online parallel accumulation–serial fragmentation (PASEF) with a novel trapped ion mobility mass spectrometer. *Mol. Cell. Proteomics* 17, 2534–2545
21. Meier, F., Beck, S., Grassl, N., Lubeck, M., Park, M. A., Raether, O., and Mann, M. (2015) Parallel accumulation–serial fragmentation (PASEF): Multiplying sequencing speed and sensitivity by synchronized scans in a trapped ion mobility device. *J. Proteome Res.* 14, 5378–5387
22. Beck, S., Michalski, A., Raether, O., Lubeck, M., Kaspar, S., Goedecke, N., Baessmann, C., Hornburg, D., Meier, F., Paron, I., Kulak, N. A., Cox, J., and Mann, M. (2015) The impact II, a very high-resolution quadrupole time-of-flight instrument (QTOF) for deep shotgun proteomics. *Mol. Cell. Proteomics* 14, 2014–2029
23. Meier, F., Park, M. A., and Mann, M. (2021) Trapped ion mobility spectrometry and parallel accumulation–serial fragmentation in proteomics. *Mol. Cell. Proteomics* 20, 100138
24. Silveira, J. A., Ridgeway, M. E., Laukien, F. H., Mann, M., and Park, M. A. (2017) Parallel accumulation for 100% duty cycle trapped ion mobility-mass spectrometry. *Int. J. Mass Spectrom.* 413, 168–175
25. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S., and Ralser, M. (2020) DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. *Nat. Methods* 17, 41–44
26. Demichev, V., Yu, F., Teo, G. C., Szyrwiel, L., Rosenberger, G. A., Decker, J., Kaspar-Schoenefeld, S., Lilley, K. S., Mülleder, M., Nesvizhskii, A. I., and Ralser, M. (2021) High sensitivity dia-PASEF proteomics with DIA-NN and FragPipe. *bioRxiv*, 2021.03.08.434385
27. Bache, N., Geyer, P. E., Bekker-Jensen, D. B., Hoerning, O., Falkenby, L., Treit, P. V., Doll, S., Paron, I., Müller, J. B., Meier, F., Olsen, J. V., Vorm, O., and Mann, M. (2018) A novel LC system embeds analytes in pre-formed gradients for rapid, ultra-robust proteomics. *Mol. Cell. Proteomics* 17, 2284–2296
28. Ogata, K., Chang, C. H., and Ishihama, Y. (2021) Effect of phosphorylation on the collision cross sections of peptide ions in ion mobility spectrometry. *Mass Spectrom.* 10, 1–8
29. Kulak, N. A., Pichler, G., Paron, I., Nagaraj, N., and Mann, M. (2014) Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nat. Methods* 11, 319–324
30. Stukalov, A., Girault, V., Grass, V., Bergant, V., Karayel, O., Urban, C., Haas, D. A., Huang, Y., Oubraham, L., Wang, A., Hamad, S. M., Piras, A., Tanzer, M., Hansen, F. M., Engleitner, T., Reinecke, M., Lavacca, T. M., Ehmann, R., Wölfel, R., Jores, J., Küster, B., Protzer, U., Rad, R., Ziebuhr, J., Thiel, V., Scaturro, P., Mann, M., and Pichlmair, A. (2020) Multi-level proteomics reveals host-perturbation strategies of SARS-CoV-2 and



bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- SARS-CoV. *bioRxiv*, 2020.06.17.156455
31. Kulak, N. A., Geyer, P. E., and Mann, M. (2017) Loss-less nano-fractionator for high sensitivity, high coverage proteomics. *Mol. Cell. Proteomics* 16, 694–705
  32. Demichev, V., Yu, F., Teo, G. C., Szyrwiell, L., Rosenberger, G. A., Decker, J., Kaspar-Schoenefeld, S., Lilley, K. S., Mülleder, M., Nesvizhskii, A. I., and Ralser, M. (2021) High sensitivity dia-PASEF proteomics with DIA-NN and FragPipe. *bioRxiv*, 2021.03.08.434385
  33. Yu, F., Teo, G. C., Kong, A. T., Haynes, S. E., Avtonomov, D. M., Geiszler, D. J., and Nesvizhskii, A. I. (2020) Identification of modified peptides using localization-aware open search. *Nat. Commun.* 11, 4065
  34. Yu, F., Haynes, S. E., Teo, G. C., Avtonomov, D. M., Polasky, D. A., and Nesvizhskii, A. I. (2020) Fast Quantitative Analysis of timsTOF PASEF Data with MSFragger and IonQuant. *Mol. Cell. Proteomics* 19, 1575–1585
  35. Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D., and Nesvizhskii, A. I. (2017) MSFragger: Ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nat. Methods* 14, 513–520
  36. da Veiga Leprevost, F., Haynes, S. E., Avtonomov, D. M., Chang, H. Y., Shanmugam, A. K., Mellacheruvu, D., Kong, A. T., and Nesvizhskii, A. I. (2020) Philosopher: a versatile toolkit for shotgun proteomics data analysis. *Nat. Methods* 17, 869–870
  37. Rosenberger, G. EasyPQP: Simple library generation for OpenSWATH.
  38. Käll, L., Storey, J. D., MacCoss, M. J., and Noble, W. S. (2008) Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J. Proteome Res.* 7, 29–34
  39. Käll, L., Canterbury, J. D., Weston, J., Noble, W. S., and MacCoss, M. J. (2007) Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods* 4, 923–925
  40. Nesvizhskii, A. I., Keller, A., Kolker, E., and Aebersold, R. (2003) A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem.* 75, 4646–4658
  41. Cox, J., Hein, M. Y., Lubner, C. A., Paron, I., Nagaraj, N., and Mann, M. (2014) Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Mol. Cell. Proteomics* 13, 2513–2526
  42. Tyanova, S., Temu, T., Sinitcyn, P., Carlson, A., Hein, M. Y., Geiger, T., Mann, M., and Cox, J. (2016) The Perseus computational platform for comprehensive analysis of (prote)omics data. *Nat. Methods* 13, 731–740
  43. Bruderer, R., Bernhardt, O. M., Gandhi, T., Xuan, Y., Sondermann, J., Schmidt, M., Gomez-Varela, D., and Reiter, L. (2017) Optimization of experimental parameters in data-independent mass spectrometry significantly increases depth and reproducibility of results. *Mol. Cell. Proteomics* 16, 2296–2309
  44. Zhang, F., Ge, W., Ruan, G., Cai, X., and Guo, T. (2020) Data-Independent Acquisition Mass Spectrometry-Based Proteomics and Software Tools: A Glimpse in 2020. *Proteomics* 20, 1900276
  45. Olsen, J. V., and Mann, M. (2013) Status of large-scale analysis of posttranslational modifications by mass spectrometry. *Mol. Cell. Proteomics* 12, 3444–3452
  46. Doll, S., and Burlingame, A. L. (2015) Mass spectrometry-based detection and assignment of protein posttranslational modifications. *ACS Chem. Biol.* 10, 63–71
  47. Meier, F., Köhler, N. D., Brunner, A. D., Wanka, J. M. H., Voytik, E., Strauss, M. T., Theis, F. J., and Mann, M. (2021) Deep learning the collisional cross sections of the

bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- peptide universe from a million experimental values. *Nat. Commun.* 12, 1–24
48. Voytik, E., Bludau, I., Willems, S., Hansen, F. M., Brunner, A.-D., Strauss, M. T., and Mann, M. (2022) AlphaMap: an open-source Python package for the visual annotation of proteomics data with sequence-specific knowledge. *Bioinformatics* 38, 849–852
49. Voytik, Eugenia, Willems, S. AlphaViz.
50. Willems, S., Voytik, E., Skowronek, P., Strauss, M. T., and Mann, M. (2021) AlphaTims: Indexing trapped ion mobility spectrometry-TOF data for fast and easy accession and visualization. *Mol. Cell. Proteomics* 20, 100149
51. Wee, P., and Wang, Z. (2017) Epidermal growth factor receptor cell proliferation signaling pathways. *Cancers (Basel)*. 9, 52
52. Olsen, J. V., Blagoev, B., Gnäd, F., Macek, B., Kumar, C., Mortensen, P., and Mann, M. (2006) Global, In Vivo, and Site-Specific Phosphorylation Dynamics in Signaling Networks. *Cell* 127, 635–648
53. Martinez-Val, A., Bekker-Jensen, D. B., Steigerwald, S., Koenig, C., Østergaard, O., Mehta, A., Tran, T., Sikorski, K., Torres-Vega, E., Kwasniewicz, E., Brynjólfssdóttir, S. H., Frankel, L. B., Kjøbsted, R., Krogh, N., Lundby, A., Bekker-Jensen, S., Lund-Johansen, F., and Olsen, J. V. (2021) Spatial-proteomics reveals phospho-signaling dynamics at subcellular resolution. *Nat. Commun.* 12, 7113
54. Mahmood, T., and Yang, P. C. (2012) Western blot: Technique, theory, and trouble shooting. *N. Am. J. Med. Sci.* 4, 429–434
55. Ochoa, D., Jarnuczak, A. F., Viéitez, C., Gehre, M., Soucheray, M., Mateus, A., Kleefeldt, A. A., Hill, A., Garcia-Alonso, L., Stein, F., Krogan, N. J., Savitski, M. M., Swaney, D. L., Vizcaino, J. A., Noh, K. M., and Beltrao, P. (2020) The functional landscape of the human phosphoproteome. *Nat. Biotechnol.* 38, 365–373
56. Feltham, R., Jamal, K., Tenev, T., Liccardi, G., Jaco, I., Domingues, C. M., Morris, O., John, S. W., Annibaldi, A., Widya, M., Kearney, C. J., Clancy, D., Elliott, P. R., Glatte, T., Qiao, Q., Thompson, A. J., Nesvizhskii, A., Schmidt, A., Komander, D., Wu, H., Martin, S., and Meier, P. (2018) Mind Bomb Regulates Cell Death during TNF Signaling by Suppressing RIPK1's Cytotoxic Potential. *Cell Rep.* 23, 470–484
57. Tanzer, M. C., Bludau, I., Stafford, C. A., Hornung, V., and Mann, M. (2021) Phosphoproteome profiling uncovers a key role for CDKs in TNF signaling. *Nat. Commun.* 12, 6053
58. Goffeau, A., Barrell, G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M., Louis, E. J., Mewes, H. W., Murakami, Y., Philippsen, P., Tettelin, H., and Oliver, S. G. (1996) Life with 6000 genes. *Science (80- )*. 274, 546–567
59. Thys, A., Trillet, K., Rosińska, S., Gayraud, A., Douanne, T., Danger, Y., Renaud, C. C. N., Antigny, L., Lavigne, R., Pineau, C., Com, E., Vérité, F., Gavard, J., and Bidère, N. (2021) Serine 165 phosphorylation of SHARPIN regulates the activation of NF-κB. *iScience* 24, 101939
60. Jaco, I., Annibaldi, A., Lalaoui, N., Wilson, R., Tenev, T., Laurien, L., Kim, C., Jamal, K., Wicky John, S., Liccardi, G., Chau, D., Murphy, J. M., Brumatti, G., Feltham, R., Pasparakis, M., Silke, J., and Meier, P. (2017) MK2 Phosphorylates RIPK1 to Prevent TNF-Induced Cell Death. *Mol. Cell* 66, 698–710.e5
61. Mohideen, F., Paulo, J. A., Ordureau, A., Gygi, S. P., and Harper, J. W. (2017) Quantitative phospho-proteomic analysis of TNFα/NFκB signaling reveals a role for



bioRxiv preprint doi: <https://doi.org/10.1101/2022.05.31.494163>; this version posted May 31, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

- RIPK1 phosphorylation in suppressing necrotic cell death. *Mol. Cell. Proteomics* 16, 1200–1216
62. Nagaraj, N., Wisniewski, J. R., Geiger, T., Cox, J., Kircher, M., Kelso, J., Pääbo, S., and Mann, M. (2011) Deep proteome and transcriptome mapping of a human cancer cell line. *Mol. Syst. Biol.* 7, 548
  63. Poulos, R. C., Hains, P. G., Shah, R., Lucas, N., Xavier, D., Manda, S. S., Anees, A., Koh, J. M. S., Mahboob, S., Wittman, M., Williams, S. G., Sykes, E. K., Hecker, M., Dausmann, M., Wouters, M. A., Ashman, K., Yang, J., Wild, P. J., deFazio, A., Balleine, R. L., Tully, B., Aebersold, R., Speed, T. P., Liu, Y., Reddel, R. R., Robinson, P. J., and Zhong, Q. (2020) Strategies to enable large-scale proteomics for reproducible research. *Nat. Commun.* 11, 3793
  64. Muazzam, A., Chiasserini, D., Kelsall, J., Geifman, N., Whetton, A. D., and Townsend, P. A. (2021) A prostate cancer proteomics database for swath-ms based protein quantification. *Cancers (Basel)*. 13, 5580
  65. Tully, B., Balleine, R. L., Hains, P. G., Zhong, Q., Reddel, R. R., and Robinson, P. J. (2019) Addressing the Challenges of High-Throughput Cancer Tissue Proteomics for Clinical Application: ProCan. *Proteomics* 19, 1900109
  66. Perez-Riverol, Y., Bai, J., Bandla, C., García-Seisdedos, D., Hewapathirana, S., Kamatchinathan, S., Kundu, D. J., Prakash, A., Frericks-Zipper, A., Eisenacher, M., Walzer, M., Wang, S., Brazma, A., and Vizcaino, J. A. (2022) The PRIDE database resources in 2022: a hub for mass spectrometry-based proteomics evidences. *Nucleic Acids Res.* 50, D543–D552



### 3.8. Article 8: Deep learning the collisional cross sections of the peptide universe from a million experimental values

Authors: Florian Meier<sup>1,5</sup>, Niklas D. Köhler<sup>2</sup>, Andreas-David Brunner<sup>1</sup>, Jean-Marc H. Wanka<sup>2</sup>, **Eugenia Voytik**<sup>1</sup>, Maximilian T. Strauss<sup>1</sup>, Fabian J. Theis<sup>2,3</sup> & Matthias Mann<sup>1,4</sup>

<sup>1</sup> Department Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany.

<sup>2</sup> Institute of Computational Biology, Helmholtz Zentrum München — German Research Center for Environmental Health, Neuherberg, Germany.

<sup>3</sup> Department of Mathematics, TU München, Munich, Germany.

<sup>4</sup> NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark.

<sup>5</sup> Functional Proteomics, Jena University Hospital, Jena, Germany.

Published in *Nature Communications* (2021).

Despite its relatively short history, the new timsTOF instrument in combination with the PASEF method have already become a widely used technology in proteomics laboratories. This has made it possible to measure ion mobility values and to derive collisional cross section (CCS) values representing ion size and shape on a very large scale. However, despite numerous attempts to understand the nature and predict the utility of the peptide CCS values using machine learning approaches, the relations between linear amino acid sequence and CCS had proven too complex to be generalized or predicted using simple rules.

To tackle this challenge, we measured more than two million CCS values of about 500,000 peptides with unique sequences from whole-proteome digests of five biological species with high precision and an accuracy of a few percent. This revealed on a global scale the sequence-specific factors that determine CCS values, such as hydrophobicity, the proportion of prolines and the location of histidines. The size of the acquired dataset made it possible for the first time to develop a deep learning model capable of learning complex interactions between amino acids that influence peptide shape and finally to predict CCS values directly for any peptide sequence. We found that CCS values are intrinsic properties of the molecule, similar to molecular weights and that they can be predicted by our model with a median deviation of 1.4%, not far from the experimental variation.

My contribution to the project was mainly to the data analysis in this project.



## ARTICLE



OPEN

# Deep learning the collisional cross sections of the peptide universe from a million experimental values

Florian Meier<sup>1,5,6</sup>, Niklas D. Köhler<sup>2,6</sup>, Andreas-David Brunner<sup>1,6</sup>, Jean-Marc H. Wanka<sup>2</sup>, Eugenia Voytik<sup>1</sup>, Maximilian T. Strauss<sup>1</sup>, Fabian J. Theis<sup>2,3</sup> & Matthias Mann<sup>1,4</sup>

The size and shape of peptide ions in the gas phase are an under-explored dimension for mass spectrometry-based proteomics. To investigate the nature and utility of the peptide collisional cross section (CCS) space, we measure more than a million data points from whole-proteome digests of five organisms with trapped ion mobility spectrometry (TIMS) and parallel accumulation-serial fragmentation (PASEF). The scale and precision (CV < 1%) of our data is sufficient to train a deep recurrent neural network that accurately predicts CCS values solely based on the peptide sequence. Cross section predictions for the synthetic ProteomeTools peptides validate the model within a 1.4% median relative error ( $R > 0.99$ ). Hydrophobicity, proportion of prolines and position of histidines are main determinants of the cross sections in addition to sequence-specific interactions. CCS values can now be predicted for any peptide and organism, forming a basis for advanced proteomics workflows that make full use of the additional information.

<sup>1</sup>Department Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany. <sup>2</sup>Institute of Computational Biology, Helmholtz Zentrum München—German Research Center for Environmental Health, Neuherberg, Germany. <sup>3</sup>Department of Mathematics, TU München, Munich, Germany. <sup>4</sup>NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark. <sup>5</sup>Present address: Functional Proteomics, Jena University Hospital, Jena, Germany. <sup>6</sup>These authors contributed equally: Florian Meier, Niklas D. Köhler, Andreas-David Brunner. ✉email: [fabian.theis@helmholtz-muenchen.de](mailto:fabian.theis@helmholtz-muenchen.de); [mmann@biochem.mpg.de](mailto:mmann@biochem.mpg.de)

## ARTICLE

The combination of ion mobility spectrometry (IMS) and mass spectrometry (MS) extends conventional liquid chromatography–mass spectrometry (LC–MS) by an extra dimension of separation, increasing peak capacity, selectivity, and depth of analysis<sup>1–5</sup>. Recent advances have greatly improved the sensitivity of commercially available IMS devices and the technology is now set for a broader application in MS-based proteomics<sup>6–10</sup>.

IMS separates ions in the gas phase (typically in the mbar pressure range) based on their size and shape within milliseconds. This time scale allows recording full ion mobility spectra between typical chromatographic peaks (seconds) and the acquisition pulses of time-of-flight (TOF) instruments (~100  $\mu$ s). We have recently integrated trapped ion mobility spectrometry (TIMS)<sup>11,12</sup>, a relatively new and particularly compact ion mobility device, with a high-resolution quadrupole TOF mass analyzer<sup>10,13,14</sup>. In MS/MS mode, this opens up the possibility to step the precursor selection window as a function of ion mobility, allowing the fragmentation of multiple precursors during a single TIMS scan<sup>13</sup>. We termed this novel scan mode parallel accumulation-serial fragmentation (PASEF) and demonstrated that it increases MS/MS rates more than ten-fold without any loss in sensitivity as is otherwise inherent to faster scanning rates<sup>10,15</sup>.

An intriguing feature of the combination of TIMS and PASEF is that it should allow the acquisition of ion mobility values on a very large scale. Such data have previously been measured on a case by case basis by classical drift tube IMS, in which a weak electric field drags ions through an inert buffer gas<sup>16–18</sup>. Larger ions collide more frequently with gas molecules and hence traverse the drift tube with a lower speed as compared with their smaller counterparts. In TIMS the physical process is the same, except that the setup is reversed with the electric field holding ions stationary against an incoming gas flow, prior to their controlled release from the device by lowering the electric field<sup>19,20</sup>. In both cases, the measured ion mobility (reported as the reduced ion mobility coefficient  $K_0$ ) can be used to derive a collisional cross section (CCS), which is the rotational average of an ion's gas-phase conformation<sup>21,22</sup>. The CCS intrinsically depends on the ion structure, which is also illustrated by the fact that different classes of biomolecules (e.g., metabolites, carbohydrates, peptides) show different trends in their ion mobilities as a function of molecular mass<sup>23</sup>. Interestingly, conformations also vary within a compound class – to the extent that isobaric peptide sequences can be distinguishable by their different CCS<sup>24,25</sup>.

The link between the amino acids of a peptide and its measured cross section has the potential to increase the confidence in its identifications through reference or predicted CCS values. This has motivated researchers to develop various (machine learning) models based on amino acid-specific parameterization and physicochemical properties<sup>16,26–29</sup>. However, as comprehensive experimental data are not available, predicting the full complexity of the peptide conformational space remains elusive. Furthermore, it is not clear which properties should be considered to best parameterize such models and make them generalizable. We reasoned that a combination of very large and consistent datasets acquired by PASEF with state of the art deep learning methods would address both challenges. Due to their inherent flexibility and their ability to scale to large datasets, deep learning methods have proven very successful in genomics<sup>30,31</sup> and more recently in proteomics for the prediction of retention times and fragmentation spectra<sup>32–35</sup>.

We here set out to explore the nature and utility of the peptide CCS space in proteomics by first measuring a very large dataset of CCSs by TIMS-TOF PASEF across five different biological species. Building on this dataset, we develop and train a bi-directional recurrent neural network with long short-term

memory (LSTM) units to predict CCS values for any peptide sequence in the tryptic peptide universe. Interpreting our network based on recent approaches from explainable AI allows us to investigate the nature of the underlying relationship between linear peptide sequence and peptide cross section.

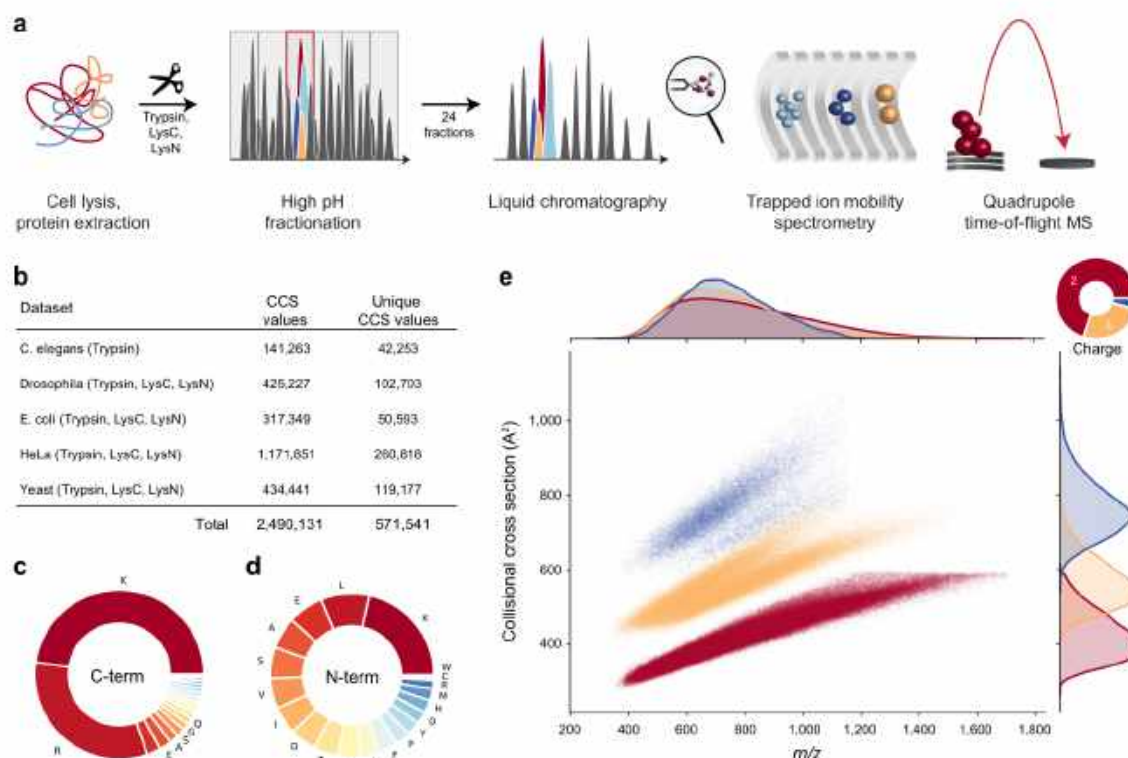
## Results

**Construction of a very large-scale peptide CCS dataset.** To fully capture the conformational diversity of peptides in the gas phase, we generated peptides from whole-cell proteomes of *Caenorhabditis elegans*, *Drosophila melanogaster*, *Escherichia coli*, *HeLa*, and *budding yeast* using up to three different enzymes with complementary cleavage specificity (trypsin, LysC, and LysN). To increase the depth of our analysis, we split peptide mixtures into 24 fractions per organism and analyzed each of them separately with PASEF on a TIMS-quadrupole TOF MS (Methods; Fig. 1a). As this is the same setup we used before, we combined our new experimental data with our previously reported dataset from a tryptic HeLa digest<sup>10</sup>.

In total, we compiled 360 LC-MS/MS runs and processed them in the MaxQuant software<sup>36,37</sup>. This resulted in about 2.5 million peptide spectrum matches and 426,845 unique peptide sequences at globally controlled false discovery (FDR) rates of less than 1% at the peptide and protein levels for each organism and enzyme. MaxQuant links each peptide spectrum match to a four-dimensional (4D) isotope cluster (or ‘feature’) in mass, retention time, ion mobility, and intensity dimension. For each of these, the ion mobility value is determined as the intensity-weighted average of the corresponding mobilogram trace and can be converted into an ion-neutral CCS value using the Mason-Schamp equation<sup>21</sup>. Some peptides occur in more than one conformation and have multiple peaks in an LC-TIMS-MS experiment, but for simplicity we here chose to keep only the most abundant feature per charge state (Supplementary Fig. 1).

Overall, our dataset comprises over two million CCS values, which we collapsed to about 570,000 unique combinations of peptide sequence, charge state and, if applicable, side chain modifications such as oxidation of methionine (Fig. 1b). Peptide sequence lengths ranged from 7 up to 55 amino acids with a median length of 14. The trypsin and LysC datasets contributed 79% of the peptide sequences (C-terminal R or K), whereas LysN peptide (N-terminal K) accounted for the remaining 21%. Within the two classes of peptides, the proportion of the terminal amino acids conformed to their expected frequencies from the database (Fig. 1c, d). Due to our selection of enzymes, peptides should have at least one basic amino acid. Consequently, singly charged ions were a small minority (2%), which we excluded from further analysis. We detected 69% of the peptides in the doubly charged, and 25% in the triply charged and 4% in the quadruply charged state. Plotting the mass-to-charge ( $m/z$ ) vs. CCS distribution of all peptides separates them by their charge state over the  $m/z$  range 400–1700  $\text{\AA}^2$  and 300–1000  $\text{\AA}^2$  in cross section (Fig. 1e). Within each charge state,  $m/z$  and CCS were correlated in accordance with previous observations in smaller datasets<sup>10,18,23,38–40</sup>. Overall, 95% of all tryptic peptides were distributed within  $\pm 8\%$  around power-law trend lines for each charge state (Supplementary Fig. 2). Interestingly, the deviation increases with charge state and mass—to the extent that there are two distinct sub-populations for charge state 3—perhaps due to the increased amino acid variability and structural flexibility in longer sequences. Our data show that peptides occupy about one-quarter of the 2D  $m/z$ -mobility space, whereas a fully orthogonal 2D separation would occupy the full space. Assuming an average ion mobility resolution of 60, this translates into an at least ten-fold increased analytical peak capacity as compared with only MS (Supplementary Fig. 3).





**Fig. 1 Large-scale peptide collisional cross section (CCS) measurement with TIMS and PASEF.** **a** Workflow from extraction of whole-cell proteomes through digestion, fractionation, and chromatographic separation of each fraction. The TIMS-quadrupole TOF mass spectrometer was operated in PASEF mode. **b** Overview of the CCS dataset in this study by organism. **c** Frequency of peptide C-terminal amino acids. **d** Frequency of peptide N-terminal amino acids. **e** Distribution of 559,979 unique data points, including modified sequence and charge state, in the CCS vs.  $m/z$  space color-coded by charge state. Density distributions for  $m/z$  and CCS are projected on the top and right axes, respectively. Source data are provided as a Source Data file.

**Evaluating the accuracy, precision, and utility of TIMS CCS measurements.** Peak capacity indicates how many peptides can be analytically resolved from each other. However, for their identification it is sufficient to determine their apex positions with adequate precision. In MS-based proteomics, accurate measurement of the peptide mass greatly reduces the number of candidates in database searches<sup>36</sup>, and the retention time can likewise be employed as a filter, as is typically done in the analysis of data-independent acquisition (DIA) experiments<sup>41</sup>. We reasoned that ion mobility values should be precise and reproducible as they are based on gas-phase interactions and defined electric fields, in contrast to chromatographic retention times, which depend on surface interactions that vary according to sample matrices and over time. We therefore investigated the precision, accuracy and added benefit of ion mobility measurements in our dataset.

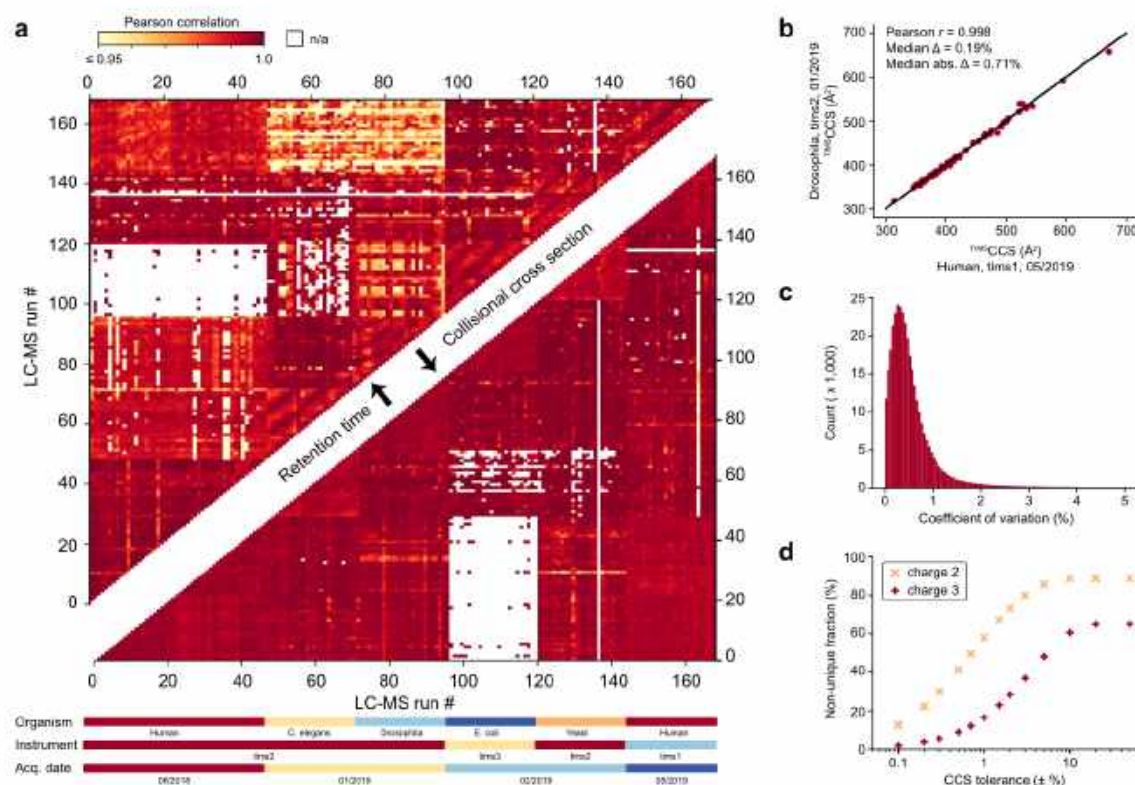
First, we calculated correlation coefficients for retention times and CCS values from pair-wise overlapping tryptic peptides in the 168 LC-MS/MS runs that had the highest number of shared peptides across organisms. Depending on evolutionary distance, this number ranged from none to hundreds and these formed the basis of our calculations. We obtained two triangular half-matrices of color-coded Pearson correlation coefficients—one for the retention time correlations and one for CCS (upper and lower part of Fig. 2a, respectively). Correlation values were generally above 0.9 for both retention time and cross section, although experiments were done over several months on three different instruments. However, correlations of CCS values were systematically higher than those for retention times, for example, the

median correlation for the HeLa runs between June 2018 and May 2019 is  $r = 0.990$  for retention times and  $r = 0.995$  for cross sections (based on 1264 peptides per pairwise comparison on average). Further, the upper triangle of the heatmap shows patches of similar color, unlike the mirrored positions in the lower triangle (Fig. 2a). This indicates chromatographic batch-effects resulting in non-linear shifts or changes in the peptide elution order. In contrast, the absence of similar patterns in the CCS comparisons supports our starting hypothesis that the ion mobility is largely independent of experimental circumstances.

Closer inspection of the variation in CCS values revealed mostly linear shifts, which do not affect the correlation coefficient. These shifts were only in the range from absolute 0 to 40 Å<sup>2</sup> (median 9.4 Å<sup>2</sup>) even for very distant measurements, and they are mainly due to variations of the gas flow in the TIMS tunnel. Importantly, a linear alignment based on a few peptide CCS values almost completely corrects for these shifts (Methods, Fig. 2b). With such an alignment, CCS values can be compared across disparate datasets, which we did for all analyses shown here. Across the 347,885 peptide CCS values measured at least in duplicate, the median coefficient of variation (CV) was 0.4%, which highlights the excellent reproducibility of TIMS CCS measurements also over longer periods of time and across instruments (Fig. 2c). This may even be improvable as suggested by our previously reported CVs of 0.1% for replicate injections of a whole-proteome digest on a single instrument<sup>10</sup>. Reassuringly, we found an excellent correlation of TIMS-CCS<sub>N2</sub> values and drift



## ARTICLE



**Fig. 2 Precision, accuracy, and utility of experimental peptide CCS values.** **a** Color-coded pairwise Pearson correlation values of peptide retention time (upper triangular matrix) and CCS values (lower triangular matrix) between the 168 LC-MS/MS runs of fractionated tryptic digests. Experimental metadata are indicated below the x-axis. White (n/a) indicates less than 5 data points for pairwise comparison. **b** CCS values of shared tryptic peptides independently measured in two typical LC-MS runs of fractions from *Drosophila* and HeLa ( $n = 68$ ). **c** CVs of repeatedly measured peptide CCS values in the full dataset ( $n = 347,885$  peptides). **d** Specificity of combined peptide  $m/z$  and CCS information for doubly and triply charged peptides with C-terminal arginine or lysine ( $n = 324,246$  and  $112,015$ ) with a fixed  $m/z$  tolerance of  $\pm 1.5$  ppm and as a function of CCS tolerance. For details, see main text and Methods.

tube ion mobility experiments<sup>42–44</sup> (Pearson  $r = 0.997$ ) with an average absolute deviation  $< 2\%$  (Supplementary Fig. 4).

To investigate the utility of the additional CCS information for peptide identification, we returned to Fig. 1e and defined tolerance windows in  $m/z$  and CCS dimensions for each peptide with C-terminal arginine or lysine as expected in tryptic digests (identified by MS/MS at an FDR  $< 1\%$ ). We then determined the fraction of windows in this map that were exclusively occupied by a single peptide, meaning a unique match between experimental measurement and our large peptide dataset (Fig. 2d). We set the mass tolerance at the median mass accuracy ( $\pm 1.5$  ppm) and varied the CCS tolerance separately for doubly and triply charged peptides, because they occupy different cross section areas (Methods). Without the CCS information, at  $\pm 50\%$  tolerance, about 90% of the doubly charged and 67% of the triply charged peptides had at least one other peptide within 1.5 ppm distance ('non-unique'). The fraction of unique peptides increased once the CCS window was restricted to less than  $\pm 10\%$ , in accordance with the roughly 20% spread of CCS values in Fig. 1e. Within three standard deviations ( $\pm 1.5\%$ ) of the measured CCS values, about two-thirds of the doubly charged and 75% of the triply charged species were unique and these fractions increased progressively for narrower CCS windows. We thus conclude that ion mobility can substantially reduce the number of potential peptides that need to be considered, benefiting peptide

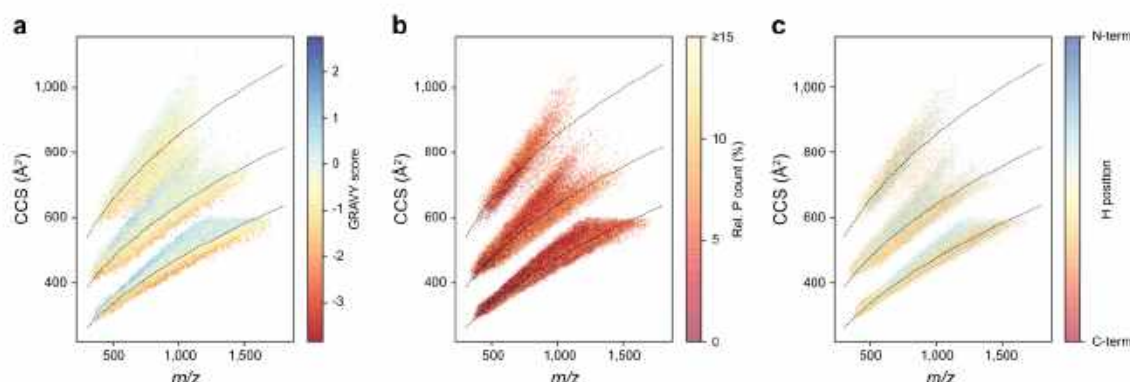
identification or MS1 level feature matching. At current CCS value accuracy, this is about a factor of two to three. As Fig. 2d also shows, an increase in accuracy down to 0.1% could make the large majority of peptides unique (56% for  $2^+$  and 90% and  $3^+$  in a  $\pm 0.5\%$  CCS window).

#### Dependence of CCS values on linear sequence determinants.

Having investigated the accuracy and utility of peptide CCS values, we asked whether a dataset of this scale could also shed a light on potential substructures in the  $m/z$  vs. ion mobility space and the relationships between linear peptide sequences and their corresponding gas-phase structures. In the  $m/z$  vs. CCS space of Fig. 1e, more compact conformations appear below and more extended conformations appear above the overall trend lines for CCS values as a function of  $m/z$ .

We first explored whether amino acids with preferences for secondary protein structures<sup>45</sup>, would also effect peptide ion structures in the gas phase and form clusters in this global view (Supplementary Fig. 5). This is a long-standing interest in ion mobility research and detailed studies of model peptides revealed that in particular helical structures can be stable in the gas phase<sup>46–48</sup>. Mapping the amino acids in each peptide sequence that favor helices in proteins, we found a tendency toward higher CCS with an increasing fraction of A, L, M, H, Q, and E. This suggests that such peptides, indeed, have a propensity to adopt





**Fig. 3 A global view on peptide cross sections.** **a** Mass-to-charge vs. collisional cross section distribution of all peptides in this study colored by the GRAVY hydrophobicity index ( $n = 559,979$ ). **b** Subset of peptides with C-terminal arginine or lysine colored by the fraction of prolines in the linear sequence ( $n = 452,592$ ). **c** Histidine-containing peptides of **(b)** colored by the relative position of histidine ( $n = 171,429$ ). Trend lines (dashed) are fitted to the overall peptide distribution to visualize the correlation of ion mass and mobility in each charge state.

extended helical rather than more compact globular structures. In contrast, peptides with a high fraction of amino acids favoring turn structures (G, S, D, N, and P) tended to more compact conformations. Note, however, that these are subtle, population-wide effects. An interesting result was that peptides with <10% of the mostly non-polar amino acids V, I, F, T, and Y (favoring sheet structures in proteins) formed a narrow band of compact gas-phase conformations.

Such tendencies have been ascribed to intra-molecular interactions such as coulombic repulsion, charge solvation and hydrogen bonding<sup>47–51</sup>. We reasoned that the hydrophobicity of peptides could thus be a good indicator of these interactions in a global view. Indeed, the GRAVY score<sup>52</sup>, a commonly used index of hydrophobicity, highlighted distinct areas of the  $m/z$  vs. ion mobility space and within the CCS value distributions of each charge state, the peptides below the trend line had lower GRAVY scores than those above (Fig. 3a). The two major subgroups of the triply charged peptides also followed this trend in that hydrophobic peptides had a higher propensity to be in the upper population and vice versa. Interestingly, and perhaps counter-intuitively, this correlation was less apparent when comparing the relative bulkiness of amino acid residues even though these properties are related (Supplementary Fig. 6). These results are, however, in line with early work in ion mobility, indicating that non-polar amino acids contribute over-proportionately to the peptide CCS value<sup>26,53</sup> and stabilize helices in the absence of solvent<sup>47</sup>. When rotationally averaged, this results in larger, effective cross sections.

To resolve structural trends at the level of individual amino acids, we visualized their relative distribution in the same 2D space. Proline is unique due to its cyclic structure, which results in an inability to donate hydrogen bonds and to disruption of secondary structures in proteins. We found that peptides with more prolines had somewhat smaller CCS values on a global scale (Fig. 3b). In line with the above reasoning, this could be explained by a disruption of extended conformations and preference for globular ones.

A peptide's CCS value is not only determined by its amino acid composition, but also by its amino acid sequence. As a large-scale example of this, we generated complementary peptide sequences with lysine either at the N-terminus (LysN digestion) or at the C-terminus (LysC digestion). As described before<sup>39</sup>, the two peptide populations are most distinct in triply charged species (Supplementary Fig. 7). Comparing 43,463 complementary sequences of

doubly charged peptides, we found changing CCS values in the range of –5% up to +10% with a slight median shift of about 1% toward higher CCS values for peptides with C-terminal lysine. The 14,388 triply charged species split in two sub-populations, with one maximum at about +1% similar to the doubly charged species and a second maximum at a shift of about +8%. This indicates that for the latter, switching the position of lysine from the C- to the N-terminus destabilizes the extended conformation. Assuming that the LysC peptides have a more extended conformation due to charge repulsion of the terminal charges, this again conforms to the above considerations.

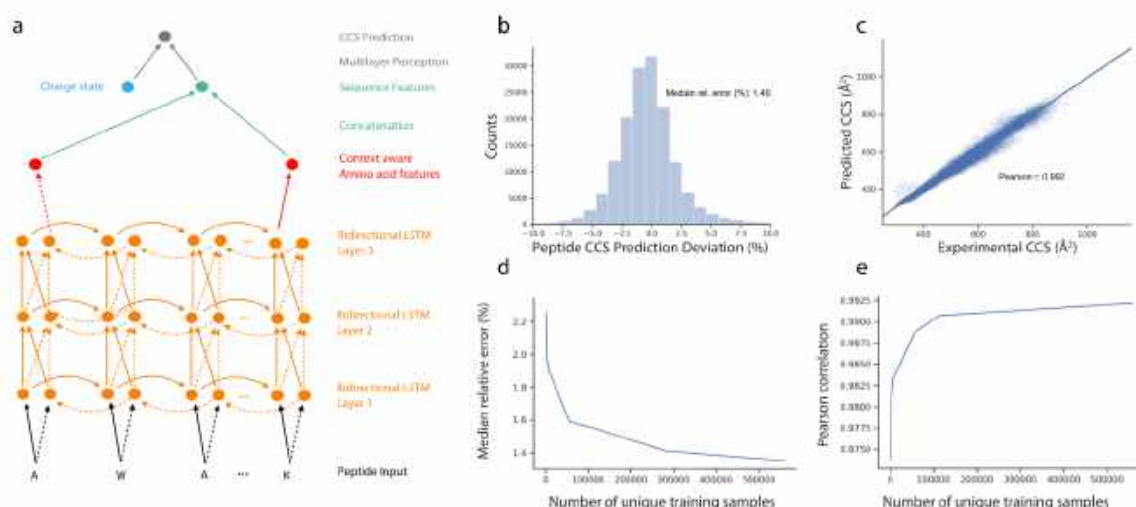
We next investigated such effects in histidine-containing tryptic peptides, by color-coding them by their relative histidine position in the linear sequences (Fig. 3c). Peptides with histidines close to the N-terminus are more likely to adopt an extended conformation and peptides with histidines closer to the C-terminal lysine or arginine are more compact in the gas phase. This again emphasizes that the internal charge distribution and the ability to solvate charges intra-molecularly have a strong influence on peptide CCS values.

Although our analysis revealed interesting general trends and suggested common principles, it is challenging to combine them into robust models that rationalize the trends and determine the CCS value of a given peptide from its linear sequence. More importantly, peptide CCS values do not lend themselves to global *ab initio* calculations as this is beyond the capabilities of computational chemistry. To that end, we next turned to deep learning.

**Deep learning accurately predicts peptide CCS values.** To construct an accurate CCS predictor that can incorporate these large-scale peptide measurements, we decided to employ a flexible deep learning model. We set out to define a network architecture that is capable of learning a non-linear mapping function connecting the linear amino acid peptide sequence with associated charge states to the experimentally measured CCS value with the following properties: (i) Exploit the sequential structure of the data where each peptide is encoded as a string of amino acid sequences; (ii) Account for the influence of an amino acid in the context of the entire peptide sequence; and (iii) Process peptide sequences of arbitrary length. An architecture fulfilling those properties is a bi-directional LSTM network on top of the raw sequence followed by a two-layer multilayer perceptron (MLP)



## ARTICLE



**Fig. 4 Deep learning peptide CCS values.** **a** Architecture of the neural network. Bi-directional long short-term model (LSTM): (i) amino acid sequence input, (ii) vectorization of amino acid information for processing, (iii) bi-directional LSTM layers, (iv) reduction to fixed length peptide feature vector by concatenating the last output neurons of both directional LSTMs, and (v) CCS prediction. **b** Relative deviation of predicted CCS values from an independent experimental validation dataset of synthetic peptides from the ProteomeTools project. **c** Correlation of predicted versus experimental CCS values ( $n = 155,004$ ). **d** Dependence of the median relative error on training dataset size. **e** Same for Pearson correlation coefficient. Source Data are provided as a Source Data file.

(Fig. 4a, Methods). Similar models have already proven successful in proteomics<sup>32,34,35</sup>. The bi-directional LSTM layers enable the model to interpret each amino acid in the context of neighboring amino acids, while the following concatenation layer reduces the resulting  $N$  (sequence length) vectors into a single set of 256 features, together encoding the properties of the entirety of the peptide sequence. Together with the charge state, this vector constitutes the input to the MLP module for the final CCS value regression. The entire architecture is implemented with differentiable modules and is end-to-end trainable. We trained our model with the set of 559,979 unique CCS values from our experimental data of the five organisms.

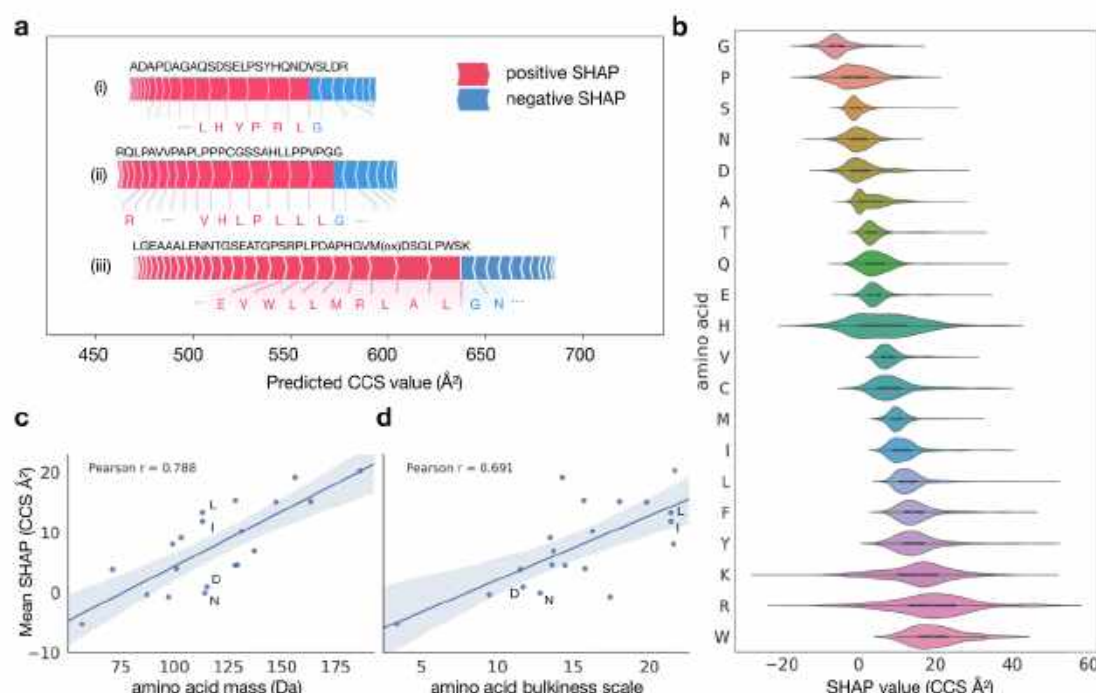
Machine learning models, in particular deep learning models, can easily be over-fitted, resulting in poor generalization performance on new datasets. While holding out samples within the dataset helps, for a more rigorous safeguard, we acquired an independent additional dataset from the synthetic ProteomeTools peptides<sup>34</sup>. This yielded 155,004 unique peptide sequences as an external test set, which was never seen by the model during training. In this test set, our model reached a high accuracy with a 1.4% absolute median deviation and a Pearson correlation coefficient of 0.992 (Fig. 4b, c). For the subset of doubly charged peptides the median absolute deviation was 1.2%, and for triply and quadruply charged species it was 1.8% and 2.0%, respectively (Supplementary Fig. 8). Presumably as a result of an increasing number of accessible conformations, we found that the median absolute deviation increased from 1.2% for CCS values  $<400 \text{ Å}^2$ , to 1.5% for CCS values between 400 and  $800 \text{ Å}^2$  ( $n = 129,710$ ) and 2.2% for 2580 peptides with CCS values  $>800 \text{ Å}^2$  (Supplementary Fig. 9). Of all predicted CCS values, 90% were within  $\pm 4.0\%$  deviation from the experimental data. In comparison, the experimental median absolute deviation between tryptic peptides from ProteomeTools and endogenous peptides was 0.6% ( $r = 0.995$ ,  $n = 54,914$ ).

In our ProteomeTools data we also found a subset of 7% of the peptide sequences, for which MaxQuant identified at least one secondary feature with a CCS difference  $>2\%$  relative to the most

abundant feature. As we trained our model with CCS values of the latter, it is expected to predict the CCS value of the main conformation in such cases. However, for peptides with a more compact secondary conformation, we observed a bias toward lower CCS values and vice versa (Supplementary Fig. 10). Future prediction models may therefore benefit from considering multiple conformations, in particular for longer peptides and higher charge states.

To independently validate the accuracy of our predictions in a real-world example, we replaced experimental CCS values in a spectral library for DIA, built from the 24 HeLa fractions, with our predictions. We then used the experimental and the predicted libraries individually to re-analyze a triplicate diaPASEF experiment of a whole-proteome HeLa sample<sup>35</sup>. Targeted data analysis in the Spectronaut<sup>36</sup> software makes use of library values to score peptide signals and to restrict the data extraction window in the ion mobility dimension, thereby removing interfering signals from precursors with similar mass and retention time, but different ion mobility. The software automatically performs an alignment of the diaPASEF experiment to the library and optimized the median ion mobility extraction window to 0.07 and  $0.09 \text{ Vs cm}^{-2}$  for the experimental and predicted library, respectively. The median absolute deviation of peptide ion mobility values were 0.74% and 0.93%. Overall, the experimental and predicted libraries performed very similarly, resulting in 7766 (experimental) and 7685 (predicted) identified protein groups on average (Supplementary Fig. 11).

Given that datasets in hundreds of thousands may still not be seen as large in deep learning, we next investigated the dependency between model accuracy in the test set and training dataset size (Fig. 4d, e). We observed a monotonous improvement in relative prediction accuracy as well as in the Pearson correlation with growing training dataset size. The model error decreased from 1.91% median relative error at 5600 samples to 1.42% for a set of 279,990 training samples, reflecting a substantial decrease in relative error of more than 20%. In contrast, moving from 279,990 samples to the full set of



**Fig. 5 Explainable artificial intelligence reveals context-dependent amino acid contributions.** **a** Example peptide sequences with SHAP value attributions of the most influential amino acids in the linear sequence. **b** Amino acid-specific SHAP value distributions over the test dataset. Data are presented as violin plots showing kernel density estimates and boxplots with the following elements: median (center), 25<sup>th</sup> and 75<sup>th</sup> percentiles (lower and upper box limits), the 1.5× interquartile range (whiskers);  $n = 100,000$  sequences. **c** Correlation between amino acid mass and mean SHAP value. **d** Correlation between amino acid bulkiness<sup>59</sup> and mean SHAP value.

559,979 samples resulted in a relative improvement of only 1.4% to a median relative error or 1.4%. These diminishing returns in accuracy of prediction indicated that the number of CCS values was sufficient—at least for currently achievable data quality.

**Resolving amino acids contributions.** Deep learning models are often deemed black boxes, as they are powerful predictors but learned relationships are typically hard to interpret. To make our model interpretable in relation to our experimental findings and to extract further molecular insights we calculated Shapley Additive Explanation (SHAP)<sup>57,58</sup> values for each amino acid in each sequence. In this case, SHAP values indicate the influence of a specific amino acid on the peptide CCS value by comparing it to reference values determined by randomly sampling sequences. This allowed us to interpret the CCS prediction for a peptide sequence by determining the individual, contextual attribution of each amino acid (Methods).

Figure 5a illustrates our analysis of sequence-specific amino acid SHAP values for three representative peptide sequences. In the regular tryptic peptide sequence (i), arginine and leucine with long side-chains shifted the prediction value to larger CCS as compared with a random sequence, while the smaller glycine contributed less than average. In the atypical peptide sequence (ii), the attribution of leucine was similar, however, the attribution of arginine was largely reduced in the N-terminal position. The context-dependent attribution of each amino acid was also evident from the long peptide sequence (iii), indicating a relatively large contribution of the small amino acid alanine to the prediction value. Interestingly, in this particular sequence, glutamic acid had a positive attribution, whereas asparagine

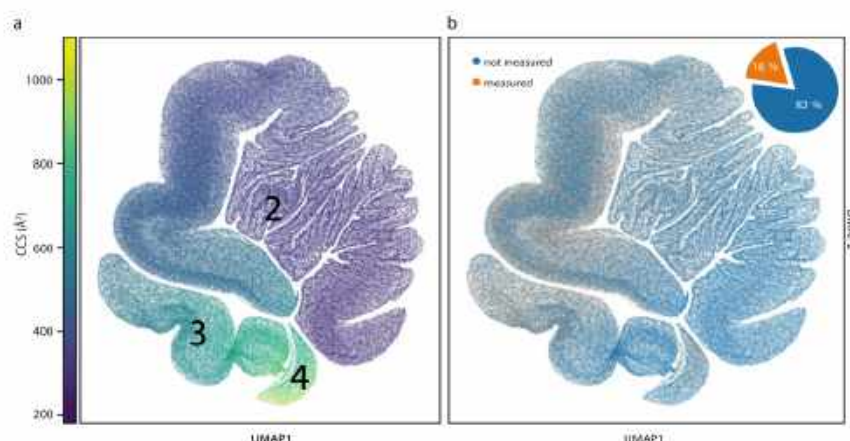
somewhat reduced the prediction value, despite the fact that both are similar in size and mass.

Plotting the aggregated SHAP value distribution over the entire test dataset for each individual amino acid, showed the expected relative order in terms of their average contribution (Fig. 5b): light and small amino acids such as glycine and proline had smaller SHAP values, whereas large and bulky amino acids such as tryptophan, arginine and lysine had larger attributions on average. In line with this observation, the average SHAP values correlated well with the amino acid mass and bulkiness<sup>59</sup>, as indicated by Pearson correlation coefficients of 0.79 and 0.69, respectively (Fig. 5c, d). Deviations from these correlations, for example, for asparagine, aspartic acid, leucine, and isoleucine, which all have similar mass, could be explained by differences in their bulkiness and hydrophobicity, in line with our experimental results above. Collectively, these results highlight that our deep learning model learned plausible features, extracting related physical quantities on the level of individual amino acids automatically from the training data, even though we solely used the linear peptide sequence as an input.

Beyond the average values, the contribution of individual amino acids to a CCS prediction had vastly different values depending on their position in a sequence (Fig. 5b). Whereas the contributions of glycine, serine, glutamic acid, and methionine were quite constant, those of lysine, arginine, and histidine nearly varied over the entire range of observed SHAP values. In particular for histidine, this agrees with our empirical observation that the position in the linear sequence had a distinct effect on the cross section (Fig. 3c). We thus conclude that our model resolves substantial structural effects for some of the amino acids within each sequence to provide a very accurate CCS estimate for the entire peptide.



## ARTICLE



**Fig. 6 The human peptide CCS universe.** **a** Two-dimensional UMAP representation of 616,948 unique tryptic peptide sequences colored by their predicted CCS value. **b** Same UMAP plot. Peptide sequences with experimental values in this study are highlighted in orange (18%).

**Human whole-proteome level CCS prediction.** The human proteome gives rise to 616,948 unique tryptic peptide sequences (considering a minimum length of 7 amino acids and no missed cleavages), of which we measured about 18% in the course of this study. To investigate the entire peptide universe and to create a reference database of all tryptic peptides in the human organism, we next used our trained deep learning model to predict CCS values for the remaining 82%. Given the importance of charge in ion mobility and the fact that it does not follow from the linear sequence in a trivial manner, we first trained a second deep learning model on our experimental training data to also predict the charge state (Methods). We then fed each human peptide sequence together with its predicted charge state into the trained CCS model, resulting in a virtually complete compendium of human peptide CCS values (Supplementary Data 1).

To provide a bird's-eye view of the structure of these data, we visualized the data manifold learned in the last layer of the neural network, in which each sequence is described by a vector of 256 neural network features. These features represent all information relevant to the prediction and were used to regress the final CCS values. However, the data manifold is too high dimensional to be directly accessible to human interpretation, hence we used a non-linear dimension reduction algorithm (Uniform Manifold Approximation and Projection, UMAP<sup>60</sup>) to visualize the data in a 2D space. In this view, each point represents a single peptide sequence and each local structure represents classes of peptides with similar features. Distances in this space can be interpreted as similarities between sequences in terms of the features extracted by the network, meaning that sequences with similar gas-phase properties are close to each other. Figure 6a reveals that the neural network organized the data in three connected manifolds, in which the sequences are ordered in terms of their associated CCS value, starting with low CCS values (<300 Å<sup>2</sup>) in the first cluster and increasing to high values (>900 Å<sup>2</sup>) in the third cluster. Similar to the representation in *m/z* vs. CCS space, we found that the main clusters were directly associated with the charge state and, within each charge state, there were apparent local structures.

Importantly, our experimental CCS values are distributed across the entire predicted peptide universe (orange and blue points in Fig. 6b), with very high densities in the CCS regions 400–800 Å<sup>2</sup>, and lower densities in the region below 300 Å<sup>2</sup>. This reassures that the depth of our experimental dataset was sufficient to sample the full feature space, and therefore suggests that our

model can be applied to predict CCS values of any tryptic peptide sequence with similar high accuracy.

### Discussion

Technological advances have rekindled the interest in IMS, which is now about to become mainstream in proteomic laboratories. Differential ion mobility spectrometers act as filters, only allowing selected ions to enter the mass spectrometer. In contrast, TIMS allows to measure ion mobility values and to derive CCS values that reflect an ion's size and shape. To investigate the benefit of this additional information in proteomics and making use of the speed and sensitivity of PASEF, we measured over two million CCS values of about 500,000 unique peptide sequences from five biological species. This covers a substantial proportion of the peptide space and is by far the most comprehensive dataset of CCS values to date.

This scale allowed us to first assess the analytical benefits of CCS values, which turn out to correspond to a roughly ten-fold increase in separation power. We further established that at an accuracy of 1%, the number of possible precursors of a peptide in a proteomics experiment decreases about two- to three-fold. Such an accuracy can be achieved with a simple linear re-calibration across distant measurements and different instruments. With this re-calibration, CCS values essentially become intrinsic properties of a molecule—meaning they do not depend on external circumstances—similar to their molecular weights, and unlike their retention times. In this regard, we note ongoing research on minimizing ion heating effects in TIMS measurements, as this may also influence the observed cross section or result in fragmentation before MS/MS, depending on instrument settings and space-charge effects<sup>61–64</sup>. However, results presented here and in other studies<sup>15,22,65,66</sup> indicate that TIMSCCS values are generally in excellent agreement with the current gold-standard drift tube ion mobility.

The scale and uniformity of our dataset makes it a valuable resource to investigate fundamentals of peptide gas-phase structures in detail. Beyond the well-known correlation of CCS values with peptide mass, they also correlated with physicochemical amino acid properties such as hydrophobicity, while the contribution of certain amino acids varied based on their position in the sequence. While this scale allowed us to compare a multitude of different peptide sequences, a limitation of our analysis is that we considered only one CCS per peptide and charge state for



simplicity. However, ions from a single peptide may occur in multiple gas-phase conformations that can be resolved by IMS<sup>50</sup>. Even more information could thus be derived by resolving the ion-mobility fine structure, for example, of higher charge states<sup>51</sup> or proline-containing peptides<sup>67</sup>. As peptide CCS values in the gas phase are fully determined by their linear amino acid sequences, we reasoned that they should also be predictable with high accuracy. Indeed, after training our state-of-the-art deep learning model on our extensive dataset, it achieved a median accuracy of about 1% for independently measured synthetic peptides, close to the experimental uncertainty. Our model generalized very well to the extent that it accurately predicted CCS values even for unseen peptides, such as those from the 'missing genes' subset in ProteomeTools<sup>54</sup>. Adding even more data values would have diminishing returns, however, prediction accuracy could be further improved with even more consistent measurements and higher ion mobility resolution or by considering multiple conformations. To obtain a sufficient number of CCS values for deep learning, we trained and validated our model with complex samples of proteolytic digests and pooled synthetic peptides. In the future, this work could be complemented with manual investigation of isolated peptides, for example, to study mobility peak shapes and multiple conformations in more detail and independent of MS feature detection algorithms or other factors.

We also interrogated our deep learning model with regard to the determinants of its predictions with Shapley Additive Explanation (SHAP). Amino acids greatly differ in the extent to which their CCS contribution depends on their sequence context—ranging from almost none to a rather wide positive or negative contribution compared to an average amino acid. This highlights how our model, indeed, learned underlying principles. These could readily be extended to other peptide classes, such as modified<sup>68</sup> or cross-linked<sup>69</sup> peptides, using transfer learning<sup>70</sup>, with little additional experimental effort.

Our study complements recent efforts in predicting properties of peptides on the basis of their sequences alone, especially those using deep learning for retention times and MS/MS spectra intensities<sup>32,34,35</sup>. Taken together, almost any peptide property relevant to proteomics workflows can now be predicted accurately, even in an ion mobility setup. Conceptually, this allows the community to nearly fully reconstruct the expected experimental values of a MS-based proteomics experiment, given a list of identified and quantified peptides. In more narrow terms, there is great potential to render time- and cost-intensive experimental libraries largely dispensable as exemplified here for diaPASEF. The CCS model presented here further extends the capabilities of such strategies to make full use of the ion mobility dimension. Similarly, predicted CCS values open up the possibility to reuse comprehensive community libraries such as the Pan Human library<sup>71</sup> for ion mobility-enhanced DIA or targeted workflows. We further envision that the combination of predicted CCS, retention time, and MS/MS spectra may improve scoring in database searches and narrow down the list of candidates. This is especially important in challenging applications such as peptidomics or proteomics of microbiomes<sup>34</sup> that have a very large search space. To foster its application and further developments, we make the source code available for training and predictions, in addition to the ready-to-use predictions of the human peptide universe included here.

## Methods

**Sample preparation.** The human HeLa cell line (S3, ATCC), *C. elegans* (N2 wild-type), *D. melanogaster* (CantonS), *E. coli* (XL1 Blue), and *Saccharomyces cerevisiae* (BY4741) were cultivated following standard protocols. All animal experiments

were performed in compliance with the institutional regulations of the Max Planck Institute of Biochemistry and the government agencies of Upper Bavaria. Whole organisms were first grinded in liquid nitrogen and cell pellets were directly suspended in lysis buffer with chloroacetamide (PreOmics, Germany) to simultaneously lyse cells, reduce protein disulfide bonds, and alkylate cysteine side chains<sup>72</sup>. The samples were boiled at 95 °C for 10 min and subsequently sonicated at maximum power (Bioruptor, Diagenode, Belgium). Proteolytic digestion was performed overnight at 37 °C by adding either (i) equal amounts of LysC and trypsin, (ii) LysC, or (iii) LysN in a 1:100 enzyme:protein (wt/wt) ratio. The resulting peptides were de-salted and purified via solid-phase extraction on styrenedivinylbenzene reversed-phase sulfonate (SDB-RPS) sorbent according to our 'in-StageTip' protocol (PreOmics). The dried eluates were reconstituted in water with 2% acetonitrile (ACN) and 0.1% trifluoroacetic acid (TFA) for further analysis. The synthetic ProteomeTools<sup>54</sup> peptides were reconstituted in the same buffer. To make the data comparable and reusable, we spiked iRT standards (Biognosys) into all samples.

**High-pH reversed-phase fractionation.** Peptide fractionation was performed at pH 10 on an EASY-nLC 1000 (Thermo Fisher Scientific, Germany) using a 30 cm × 250 µm C<sub>18</sub> reversed-phase column (PreOmics). Approximately 50 µg of peptides were separated at a flow rate of 2 µL min<sup>-1</sup> with a binary gradient starting from 3% B, which was linearly increased to 30% B within 45 min, to 60% B within 17 min, and to 95% B within 5 min before re-equilibration. Fractions were collected into 24 wells by switching the rotor valve of an automated concatenation system<sup>73</sup> (Spider fractionator, PreOmics) in 90 s intervals. Peptide fractions were vacuum-centrifuged to dryness and reconstituted in water with 2% ACN and 0.1% TFA.

**Liquid chromatography and mass spectrometry.** LC-MS was performed on an EASY-nLC 1200 (Thermo Fisher Scientific) system coupled online to a hybrid TIMS-quadrupole TOF mass spectrometer<sup>10</sup> (Bruker Daltonik timstOF Pro, Germany) via a nano-electrospray ion source (Bruker Daltonik Captive Spray). Approximately 200 ng of peptides were separated on an in-house 45 cm × 75 µm reversed-phase column at a flow rate of 300 nL min<sup>-1</sup> in an oven compartment heated to 60 °C. The column was packed in-house with 1.9 µm C<sub>18</sub> beads (Dr. Maisch Reprosil-Pur AQ, Germany) up to the laser-pulled electrospray emitter tip. Mobile phases A and B were water and 80%/20% ACN/water (v/v), respectively, and both buffered with 0.1% formic acid (v/v). To analyze fractionated peptides from whole-proteome digests, we used a gradient starting with a linear increase from 5% B to 30% B over 95 min, followed by further linear increases to 60% B and finally to 95% B in 5 min each, which was held constant for 5 min before returning to 5% in 5 min and re-equilibration for 5 min. The pooled synthetic peptides were analyzed with a gradient starting from 5% B to 30% B in 35 min, followed by linear increases to 60% B and 95% in 2.5 min each before washing and re-equilibration for a total of 5 min.

The mass spectrometer was operated in data-dependent PASEF<sup>13</sup> mode with 1 survey TIMS-MS and 10 PASEF MS/MS scans per acquisition cycle. We analyzed an ion mobility range from  $1/K_0 = 1.51$  to  $0.6 \text{ Vs cm}^{-2}$  using equal ion accumulation and ramp time in the dual TIMS analyzer of 100 ms each. Suitable precursor ions for MS/MS analysis were isolated in a window of 2 Th for  $m/z < 700$  and 3 Th for  $m/z > 700$  by rapidly switching the quadrupole position in sync with the elution of precursors from the TIMS device. The collision energy was lowered stepwise as a function of increasing ion mobility, starting from 52 eV for 0–19% of the TIMS ramp time, 47 eV for 19–38%, 42 eV for 38–57%, 37 eV for 57–76%, and 32 eV until the end. We made use of the  $m/z$  and ion mobility information to exclude singly charged precursor ions with a polygon filter mask and further used 'dynamic exclusion' to avoid re-sequencing of precursors that reached a 'target value' of 20,000 a.u. The ion mobility dimension was calibrated linearly using three ions from the Agilent ESI LC/MS tuning mix ( $m/z$ ,  $1/K_0$ : 622.0289, 0.9848  $\text{Vs cm}^{-2}$ ; 922.0097, 1.1895  $\text{Vs cm}^{-2}$ ; and 1221.9906, 1.3820  $\text{Vs cm}^{-2}$ ). All experimental parameters with relevance to the measurement of CCS values are summarized in Supplementary Table 1.

**Data processing.** MS raw files were analyzed with MaxQuant<sup>36,37</sup> version 1.6.5.0, which extracts 4D isotope patterns ('features') and associated MS/MS spectra. The built-in search engine Andromeda<sup>74</sup> was used to match observed fragment ions to theoretical peptide fragment ion masses derived from in silico digests of a reference proteome and a list of 245 potential contaminants using the appropriate digestion rules for each proteolytic enzyme (trypsin, LysC or LysN). We allowed a maximum of two missing values and required a minimum sequence length of 7 amino acids while limiting the maximum peptide mass to 4600 Da. Carbamidomethylation of cysteine was defined as a fixed modification, and oxidation of methionine and acetylation of protein N-termini were included in the search as variable modifications. Reference proteomes for each organism including isoforms were accessed from UniProt (*Homo sapiens*: 91,618 entries, 2019/05; *E. coli*: 4403 entries, 2019/01; *C. elegans*: 28,403 entries, 2019/01; *S. cerevisiae*: 6049 entries, 2019/01; *D. melanogaster*: 23,304 entries, 2019/01). The synthetic peptide library (ProteomeTools<sup>54</sup>) was searched against the entire human reference proteome. The maximum mass tolerances were set to 20 and 40 ppm for precursor and fragment ions,



## ARTICLE

respectively. False discovery rates were controlled at 1% on both the peptide spectrum match and protein level with a target-decoy approach. The analyses were performed separately for each organism and each set of synthetic peptides ('proteotypic set', 'SRM atlas', and 'missing gene set'). To demonstrate the utility of CCS prediction, we re-analyzed three dataPASEF experiments from Meier et al.<sup>55</sup> with Spectronaut 14.7.201007.47784 (Biognosys AG), replacing experimental ion mobility values in the spectral library with our predictions. Singly charged peptide precursors were excluded from this analysis as the neural network was exclusively trained with multiply charged peptides.

**Bioinformatic analysis.** Bioinformatic analysis of the MaxQuant output files and data visualization was performed with Python version 3.6 employing the following packages: NumPy, pandas, SciPy<sup>75</sup>, Biopython<sup>76</sup>, Matplotlib, and Seaborn. Decoy database hits were excluded from the analysis as well as peptide features assigned with zero intensity values. Peptides can adopt multiple conformations, both in the liquid and in the gas phase. For simplification, we here selected only the most abundant feature for each modified peptide sequence and charge state per LC-TIMS-MS run. To account for experimental drifts in the measurements of TIMS-CCS values over time, we performed a hierarchical clustering (similar to<sup>77</sup>) and aligned all experiments by calculating pair-wise linear offsets ( $y = x + b$ ) going from the closest to the most distant nodes. Multiple measurements of the same modified peptide and charge state in different LC-MS experiments were merged to one unique CCS value by calculating the mean. To perform nearest neighbor analysis in the  $m/z$  vs. CCS space, we represented the data in a Kd-tree structure using the Chebyshev distance metric to define a rectangular area with a given mass and CCS tolerance surrounding a node of interest.

**Deep learning model for CCS prediction.** The deep learning model takes a raw (modified) peptide sequence as input. First, each amino acid gets one-hot encoded into a 26-dimensional vector representation for processing. This one-hot encoding also is applied to the elements 'ox' and 'ac', resulting in a total feature vector with dimension  $L \times 26$  with  $L$  being the length of a given peptide. This vector is connected to a two-layer bi-directional recurrent network with LSTM<sup>77</sup> units with 500 hidden nodes each, which extract context-based features for each individual amino acid. This feature embedding gets further reduced to a global 256-dimensional peptide feature vector by concatenating the last output neurons of both the LSTM networks aggregating from left or right over the sequence. This peptide feature vector is further concatenated with additional charge state of the sequence and then is fed to a logistic regression layer which regresses the expected CCS value for the sequence. The most significant hyperparameters, namely: number of hidden neurons, number of layers were chosen by running a small search in a first preliminary step on a validation set consisting of 10% of the training data. The combination of recurrent layers with the concatenation step allows the model architecture to process peptide sequences with arbitrary lengths. The final model is end-to-end optimized by an ADAM Optimizer on 559,979 unique CCS values (modified peptide sequence and charge state) and validated on 155,004 holdout peptides from the synthetic ProteomeTools library. The full framework is implemented using Python with TensorFlow<sup>78</sup> as the autograd library, enabling the neural network optimization. On an i7-4930K CPU machine equipped with an NVIDIA GeForce 1080 our model was trained within 8 h and the prediction of single peptide CCS values takes approximately 1 ms.

**Deep learning model for peptide charge state prediction.** To predict the most probable (most abundant) charge state from the linear peptide sequence, we built a charge prediction neural network which has the identical structure as our CCS prediction model. It takes the raw peptide sequence as input following the same one-hot encoding procedure and predicts a single associated charge value. We trained the charge prediction model on the same 559,979 unique training values and validated it on the holdout set of 155,004 peptides from ProteomeTools. The charge prediction model reaches a final accuracy of 93.5% for predicting the three observed charge states 2, 3, and 4.

**Analysis of amino acid feature attribution of the learnt network.** For a given sequence and its CCS prediction, every amino acid is associated with a SHAP value<sup>57,58</sup>. This SHAP value quantifies how the presence of the amino acid influences the final prediction. By the summation-to-delta property, the SHAP values are constrained in a way such that the sum of all SHAP values in a sequence results in the final CCS prediction. SHAP values are a unification of multiple existing approaches<sup>79–83</sup> for explaining predictions by feature attribution. For interpreting the predictions of our model we use the DeepExplainer from the official SHAP implementation (<https://github.com/slundberg/shap>). The DeepExplainer approximates SHAP values and is based on DeepLift<sup>84</sup>. Here the importance of individual features is approximated by comparing the model output for an input that contains the specific feature value to the model output where the feature is set to a reference value. A crucial step for this approach is to define the reference values. In our case, the inputs are sequences of one-hot-encoded amino acids and we use 128 randomly chosen background sequences from the dataset in order to define meaningful reference values for all neurons. In order to capture non-linearities, the DeepLift approach approximates feature attributions for every neuron in the model. It starts

at the output layer and propagates the values to the input by backpropagation, which is called applying the chain rule for multipliers in the original publication<sup>81</sup>. Applying this approach to the input sequences in our CCS model we are able to capture the SHAP value for an individual amino acid in a peptide sequence.

**Visualization of learnt network representation of the human proteome.** To visualize the 256-dimensional neural network feature space, we apply the UMAP<sup>60</sup> algorithm, which is a dimension reduction technique for general non-linear dimension reduction and it assumes uniform distribution of the data on a Riemannian manifold. Under certain conditions this manifold can be modeled with a fuzzy topological structure. The 2D embedding, which is used for visualization is found by searching for a low-dimensional projection of the data that has the closest possible equivalent fuzzy topological structure. Therefore, pairwise similarities between peptide sequences in the high-dimensional NN space approximately resemble positions in the low-dimensional embedding visualization.

**Reporting summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

### Data availability

The MS raw files and associated MaxQuant output files generated and analyzed throughout this study have been deposited at the ProteomeXchange Consortium via the PRIDE partner repository<sup>85</sup> with the dataset identifier PXD019086. The previously acquired HeLa data is available through the dataset identifier PXD010012. The dataPASEF raw files are available through the dataset identifier PXD017703. *H. sapiens* (taxon identifier: 9606), *S. cerevisiae* (taxon identifier: 559292), *D. melanogaster* (taxon identifier: 7227), *E. coli* (taxon identifier: 83333) and *C. elegans* (taxon identifier: 6239) proteome databases were downloaded from UniProt (<https://www.uniprot.org>). Source data are provided with this paper.

### Code availability

The source code of our deep learning model and data analysis scripts are available on GitHub (<https://github.com/theislabs/DeepCollisionalCrossSection> and <https://github.com/manlabs/DeepCollisionalCrossSection>).

Received: 18 May 2020; Accepted: 22 January 2021;

Published online: 19 February 2021

### References

- McLean, J. A., Ruotolo, B. T., Gillig, K. J. & Russell, D. H. Ion mobility-mass spectrometry: a new paradigm for proteomics. *Int. J. Mass Spectrom.* **240**, 301–315 (2005).
- Baker, E. S. et al. An LC-IMS-MS platform providing increased dynamic range for high-throughput proteomic studies. *J. Proteome Res.* **9**, 997–1006 (2010).
- Kanu, A. B., Dwivedi, P., Tam, M., Matz, L. & Hill, H. H. Ion mobility-mass spectrometry. *J. Mass Spectrom.* **43**, 1–22 (2008).
- Disler, U. et al. Drift time-specific collision energies enable deep-coverage data-independent acquisition proteomics. *Nat. Methods* **11**, 167–170 (2014).
- Helm, D. et al. Ion mobility tandem mass spectrometry enhances performance of bottom-up proteomics. *Mol. Cell. Proteom.* **13**, 3709–3715 (2014).
- Pfammatter, S. et al. A novel differential ion mobility device expands the depth of proteome coverage and the sensitivity of multiplex proteomic measurements. *Mol. Cell. Proteom.* **17**, 2051–2067 (2018).
- Hebert, A. S. et al. Comprehensive single-shot proteomics with FAIMS on a hybrid orbitrap mass spectrometer. *Anal. Chem.* **90**, 9529–9537 (2018).
- Bekker-Jensen, D. B. et al. A compact quadrupole-orbitrap mass spectrometer with FAIMS interface improves proteome coverage in short LC gradients. *Mol. Cell. Proteom.* **19**, 716–729 (2020).
- Yu, Q. et al. Benchmarking the orbitrap tribrid eclipse for next generation multiplexed proteomics. *Anal. Chem. Anal. Chem.* **92**, 6478–6485 (2020).
- Meier, F. et al. Online parallel accumulation–serial fragmentation (PASEF) with a novel trapped ion mobility mass spectrometer. *Mol. Cell. Proteom.* **17**, 2534–2545 (2018).
- Fernandez-Lima, F., Kaplan, D. A., Suetering, J. & Park, M. A. Gas-phase separation using a trapped ion mobility spectrometer. *Int. J. Ion. Mobil. Spectrom.* **14**, 93–98 (2011).
- Fernandez-Lima, F. A., Kaplan, D. A. & Park, M. A. Note: integration of trapped ion mobility spectrometry with mass spectrometry. *Rev. Sci. Instrum.* **82**, 126106 (2011).
- Meier, F. et al. Parallel accumulation–serial fragmentation (PASEF): multiplying sequencing speed and sensitivity by synchronized scans in a trapped ion mobility device. *J. Proteome Res.* **14**, 5378–5387 (2015).



14. Ridgeway, M. E., Lubeck, M., Jordens, J., Mann, M. & Park, M. A. Trapped ion mobility spectrometry: a short review. *Int. J. Mass Spectrom.* **425**, 22–35 (2018).
15. Vasilopoulou, C. G. et al. Trapped ion mobility spectrometry and PASEF enable in-depth lipidomics from minimal sample amounts. *Nat. Commun.* **11**, 331 (2020).
16. Valentine, S. J., Counterterman, A. E. & Clemmer, D. E. A database of 660 peptide ion cross sections: use of intrinsic size parameters for bona fide predictions of cross sections. *J. Am. Soc. Mass Spectrom.* **10**, 1188–1211 (1999).
17. Tao, L., McLean, J. R., McLean, J. A. & Russell, D. H. A collision cross-section database of singly-charged peptide ions. *J. Am. Soc. Mass Spectrom.* **18**, 1232–1238 (2007).
18. May, J. C., Morris, C. B. & McLean, J. A. Ion mobility collision cross section compendium. *Anal. Chem.* **89**, 1032–1044 (2017).
19. Michelmann, K., Silveira, J. A., Ridgeway, M. E. & Park, M. A. Fundamentals of trapped ion mobility spectrometry. *J. Am. Soc. Mass Spectrom.* **26**, 14–24 (2014).
20. Silveira, J. A., Michelmann, K., Ridgeway, M. E. & Park, M. A. Fundamentals of trapped ion mobility spectrometry part II: fluid dynamics. *J. Am. Soc. Mass Spectrom.* **27**, 585–595 (2016).
21. Mason, E. A. & McDaniel, E. W. *Transport Properties of Ions in Gases* (John Wiley & Sons, Inc., 1988).
22. Gabelica, V. et al. Recommendations for reporting ion mobility mass spectrometry measurements. *Mass Spectrom. Rev.* **38**, 291–320 (2019).
23. May, J. C. et al. Conformational ordering of biomolecules in the gas phase: nitrogen collision cross sections measured on a prototype high resolution drift tube ion mobility-mass spectrometer. *Anal. Chem.* **86**, 2107–2116 (2014).
24. Wu, C., Sierra, W. F., Klasmeyer, J. & Hill, H. H. Separation of isomeric peptides using electrospray ionization/high-resolution ion mobility spectrometry. *Anal. Chem.* **72**, 391–395 (2000).
25. Srebalus Barnes, C. A., Hilderbrand, A. E., Valentine, S. J. & Clemmer, D. E. Resolving isomeric peptide mixtures: a combined HPLC/ion mobility-TOFMS analysis of a 4000-component combinatorial library. *Anal. Chem.* **74**, 26–36 (2002).
26. Shvartsburg, A. A., Siu, K. W. M. & Clemmer, D. E. Prediction of peptide ion mobilities via a priori calculations from intrinsic size parameters of amino acid residues. *J. Am. Soc. Mass Spectrom.* **12**, 885–888 (2001).
27. Wang, B., Valentine, S., Plasencia, M., Raghuraman, S. & Zhang, X. Artificial neural networks for the prediction of peptide drift time in ion mobility mass spectrometry. *BMC Bioinformatics* **11**, 182 (2010).
28. Shah, A. R. et al. Machine learning based prediction for peptide drift times in ion mobility spectrometry. *Bioinformatics* **26**, 1601–1607 (2010).
29. Wang, B. et al. Prediction of peptide drift time in ion mobility mass spectrometry from sequence-based features. *BMC Bioinformatics* **14**, S9 (2013).
30. Zou, J. et al. A primer on deep learning in genomics. *Nat. Genet.* **51**, 12–18 (2019).
31. Eraslan, G., Avsec, Z., Gagneur, J. & Theis, F. J. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* **20**, 389–403 (2019).
32. Zhou, X. X. et al. PDeep: predicting MS/MS spectra of peptides with deep learning. *Anal. Chem.* **89**, 12690–12697 (2017).
33. Ma, C. et al. Improved peptide retention time prediction in liquid chromatography through deep learning. *Anal. Chem.* **90**, 10881–10888 (2018).
34. Gessulat, S. et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nat. Methods* **16**, 509–518 (2019).
35. Tiwary, S. et al. High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis. *Nat. Methods* **16**, 519–525 (2019).
36. Cox, J. & Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.* **26**, 1367–1372 (2008).
37. Priamichnikov, N. et al. MaxQuant software for ion mobility enhanced shotgun proteomics. *Mol. Cell. Proteomics* **19**, 1058–1069 (2020).
38. Valentine, S. J., Counterterman, A. E., Hoaglund, C. S., Reilly, J. P. & Clemmer, D. E. Gas-phase separations of protease digests. *J. Am. Soc. Mass Spectrom.* **9**, 1213–1216 (1998).
39. Lietz, C. B., Yu, Q. & Li, L. Large-scale collision cross-section profiling on a traveling wave ion mobility mass spectrometer. *J. Am. Soc. Mass Spectrom.* **25**, 2009–2019 (2014).
40. Taraszka, J. A., Counterterman, A. E. & Clemmer, D. E. Gas-phase separations of complex tryptic peptide mixtures. *Fresenius. J. Anal. Chem.* **369**, 234–245 (2001).
41. Ludwig, C. et al. Data-independent acquisition-based SWATH-MS for quantitative proteomics: a tutorial. *Mol. Syst. Biol.* **14**, e8126 (2018).
42. Bush, M. F., Campuzano, I. D. G. & Robinson, C. V. Ion mobility mass spectrometry of peptide ions: effects of drift gas and calibration strategies. *Anal. Chem.* **84**, 7124–7130 (2012).
43. Stow, S. M. et al. An interlaboratory evaluation of drift tube ion mobility-mass spectrometry collision cross section measurements. *Anal. Chem.* **89**, 9048–9055 (2017).
44. Picache, J. A. et al. Collision cross section compendium to annotate and predict multi-omic compound identities. *Chem. Sci.* **10**, 983–993 (2019).
45. Levitt, M. Conformational preferences of amino acids in globular proteins. *Biochemistry* **17**, 4277–4285 (1978).
46. Jarrold, M. F. Peptides and proteins in the vapor phase. *Annu. Rev. Phys. Chem.* **51**, 179–207 (2000).
47. Jarrold, M. F. Helices and sheets in vacuo. *Phys. Chem. Chem. Phys.* **9**, 1659 (2007).
48. Wyttenbach, T., Pierson, N. A., Clemmer, D. E. & Bowers, M. T. Ion mobility analysis of molecular dynamics. *Annu. Rev. Phys. Chem.* **65**, 175–196 (2014).
49. McLean, J. R. et al. Factors that influence helical preferences for singly charged gas-phase peptide ions: the effects of multiple potential charge-carrying sites. *J. Phys. Chem. B* **114**, 809–816 (2010).
50. Pierson, N. A., Chen, L., Valentine, S. J., Russell, D. H. & Clemmer, D. E. Number of solution states of bradykinin from ion mobility and mass spectrometry measurements. *J. Am. Chem. Soc.* **133**, 13810–13813 (2011).
51. Xiao, C., Pérez, L. M. & Russell, D. H. Effects of charge states, charge sites and side chain interactions on conformational preferences of a series of model peptide ions. *Analyst* **140**, 6933–6944 (2015).
52. Kyte, J. & Doolittle, R. F. A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.* **157**, 105–132 (1982).
53. Valentine, S. J., Counterterman, A. E., Hoaglund-Hyzer, C. S. & Clemmer, D. E. Intrinsic amino acid size parameters from a series of 113 lysine-terminated tryptic digest peptide ions. *J. Phys. Chem. B* **103**, 1203–1207 (1999).
54. Zolg, D. P. et al. Building ProteomeTools based on a complete synthetic human proteome. *Nat. Methods* **14**, 259–262 (2017).
55. Meier, F. et al. diPASEF: parallel accumulation–serial fragmentation combined with data-independent acquisition. *Nat. Methods* **17**, 1229–1236 (2020).
56. Bruderer, R. et al. Extending the limits of quantitative proteome profiling with data-independent acquisition and application to acetaminophen-treated three-dimensional liver microtissues. *Mol. Cell. Proteom.* **14**, 1400–1410 (2015).
57. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. et al.) 4765–4774 (Curran Associates, Inc., 2017).
58. Lundberg, S. M. et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nat. Biomed. Eng.* **2**, 749–760 (2018).
59. Zimmerman, J. M., Eliezer, N. & Simha, R. The characterization of amino acid sequences in proteins by statistical methods. *J. Theor. Biol.* **21**, 170–201 (1968).
60. McInnes, L., Healy, J. & Melville, J. UMAP: uniform manifold approximation and projection for dimension reduction. Preprint at <https://arxiv.org/abs/1802.03426> (2018).
61. Morsa, D. et al. Effective temperature and structural rearrangement in trapped ion mobility spectrometry. *Anal. Chem.* **92**, 4573–4582 (2020).
62. Bleiholder, C., Liu, F. C. & Chai, M. Comment on effective temperature and structural rearrangement in trapped ion mobility spectrometry: TIMS enables native mass spectrometry applications. *Anal. Chem.* **92**, 16329–16333 (2020).
63. Naylor, C. N., Ridgeway, M. E., Park, M. A. & Clowers, B. H. Evaluation of trapped ion mobility spectrometry source conditions using benzylammonium thermometer ions. *J. Am. Soc. Mass Spectrom.* **31**, 1593–1602 (2020).
64. Yu, F. et al. Fast quantitative analysis of timsTOF PASEF data with MSFragger and IonQuant. *Mol. Cell. Proteom.* **19**, 1575–1585 (2020).
65. Silveira, J. A., Ridgeway, M. E. & Park, M. A. High resolution trapped ion mobility spectrometry of peptides. *Anal. Chem.* **86**, 5624–5627 (2014).
66. Hernandez, D. R. et al. Ion dynamics in a trapped ion mobility spectrometer. *Analyst* **139**, 1913–1921 (2014).
67. Counterterman, A. E. & Clemmer, D. E. Cis–trans signatures of proline-containing tryptic peptides in the gas phase. *Anal. Chem.* **74**, 1946–1951 (2002).
68. Glover, M. S. et al. Examining the influence of phosphorylation on peptide ion structure by ion mobility spectrometry-mass spectrometry. *J. Am. Soc. Mass Spectrom.* **27**, 786–794 (2016).
69. Steigenberger, B. et al. Benefits of collisional cross section assisted precursor selection (caps-PASEF) for cross-linking mass spectrometry. *Mol. Cell. Proteom.* **19**, 1677–1687 (2020).
70. Weiss, K., Khoshgofaar, T. M. & Wang, D. A survey of transfer learning. *J. Big Data* **3**, 9 (2016).
71. Rosenberger, G. et al. A repository of assays to quantify 10,000 human proteins by SWATH-MS. *Sci. Data* **1**, 140031 (2014).
72. Kulak, N. A., Pichler, G., Paron, L., Nagaraj, N. & Mann, M. Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nat. Methods* **11**, 319–324 (2014).



## ARTICLE

73. Kulak, N. A., Geyer, P.E. & Mann, M. Loss-less nano-fractionator for high sensitivity, high coverage proteomics. *Mol. Cell. Proteomics* **16**, 694–705 (2017).
74. Cox, J. et al. Andromeda: a peptide search engine integrated into the MaxQuant environment. *J. Proteome Res.* **10**, 1794–1805 (2011).
75. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
76. Cock, P. J. A. et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).
77. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
78. Abadi, M. et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. *OSDI'16: Proc. 12th USENIX Conf. Operating Systems Design and Implementation* 265–283 (USENIX, 2016).
79. Štrumbelj, E. & Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **41**, 647–665 (2014).
80. Datta, A., Sen, S. & Zick, Y. Algorithmic transparency via quantitative input influence: theory and experiments with learning systems. *IEEE Symp. Security and Privacy (SP)* 598–617 (IEEE, 2016). <https://doi.org/10.1109/SP.2016.42>.
81. Bach, S. et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**, e0130140 (2015).
82. Lipovetsky, S. & Conklin, M. Analysis of regression in game theory approach. *Appl. Stoch. Model. Bus. Ind.* **17**, 319–330 (2001).
83. Ribeiro, M. T., Singh, S. & Guestrin, C. ‘Why Should I Trust You?’: Explaining the predictions of any classifier. *Proc. 2016 Conf. North American Chapter of the Association for Computational Linguistics: Demonstrations* 97–101 (ACL, 2016).
84. Shrikumar, A., Greenside, P. & Kundaje, A. Learning important features through propagating activation differences. In *ICML'17: Proc. 34th Int. Conf. Machine Learning* (eds. Precup, D. & Whye Teh, Y.) Vol. 70, 3145–3153 (ACM, 2017).
85. Perez-Riverol, Y. et al. The PRIDE database and related tools and resources in 2019: improving support for quantification data. *Nucleic Acids Res.* **47**, D442–D450 (2019).

## Acknowledgements

We thank our colleagues in the department of Proteomics and Signal Transduction for help and discussions, and in particular I. Paron, B. Splettsöber, J. Müller, and A. Strasser for technical assistance. We acknowledge the ProteomeTools project led by B. Küster for providing a clone of the synthetic peptide library. This work was partially supported by the German Research Foundation (DFG-Gottfried Wilhelm Leibniz Prize granted to M.M., grant# MA 1764/2-1) and by the Max-Planck Society for the Advancement of Science. F.J.T. acknowledges support by the BMBF (grant #L031L0214A, grant# 01IS18036B and grant# 01IS18053A) and by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI (grant number: ZT-I-PF-5-01) and sparse2big (grant number ZT-I-007).

## Author contributions

F.M., A.B., and M.M. designed the proteomics experiments. F.M. and A.B. performed the experiments. F.M., A.B., and M.M. analyzed the data and interpreted the results. E.V. and M.T.S. contributed to the data analysis. N.D.K., with contributions from F.J.T., designed and developed the deep learning model as well as the prediction interpretation and visualization pipeline. J.M.W. performed neural network training runs and supported N.D.K. in integrating the feature attribution functionality. F.M., N.D.K., F.J.T., and M.M. wrote the manuscript. F.J.T. and M.M. supervised the project.

## Funding

Open Access funding enabled and organized by Projekt DEAL.

## Competing interests

F.J.T. reports receiving consulting fees from Roche Diagnostics GmbH and Cellarity Inc., and ownership interest in Cellarity, Inc. and Dermagnostix. The other authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41467-021-21352-8>.

**Correspondence** and requests for materials should be addressed to F.J.T. or M.M.

**Peer review information** *Nature Communications* thanks Aivett Bilbao, Zheng-jang Zhu and the other, anonymous, reviewer for their contribution to the peer review of this work. Peer reviewer reports are available.

**Reprints and permission information** is available at <http://www.nature.com/reprints>.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021

### **3.9. Article 9: AlphaPeptDeep: A modular deep learning framework to predict peptide properties for proteomics**

Authors: Wen-Feng Zeng<sup>1</sup>, Xie-Xuan Zhou<sup>1</sup>, Sander Willems<sup>1</sup>, Constantin Ammar<sup>1</sup>, Maria Wahle<sup>1</sup>, Isabell Bludau<sup>1</sup>, Eugenia Voytik<sup>1</sup>, Maximillian T. Strauss<sup>2</sup>, Matthias Mann<sup>1,2,\*</sup>

<sup>1</sup> Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

<sup>2</sup> Proteomics Program, NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark

Pre-print published online: *bioRxiv* (2022), doi: 10.1101/2022.07.14.499992v1.

Under revision in *Nature Communications*.

Recent advances in deep learning have proven very useful in proteomics, especially for predicting various experimental peptide properties. The initial development of deep learning models to predict each of these properties individually has subsequently led to a plethora of high-performance deep learning models. However, most of these efforts have focused only on the prediction of individual properties and furthermore many of them have not fully complied with open-source standards. This makes them difficult to adopt or extend and prevents from being updated to improve results in line with improvements in deep learning technologies.

Therefore, our group has recently introduced a highly modular deep learning system called AlphaPeptDeep. It provides models with comparable or superior performance for all peptide properties at the same time. Using a transfer learning approach, our integrated deep learning models eliminate the need to provide large data sets to refine models for specific experimental conditions. Moreover, the ‘model shop’ in AlphaPeptDeep enables non-specialists to build and train a model from scratch using few lines of codes. This is exemplified by building a deep learning model that predicts the propensity of any peptide sequence to be presented as a human leukocyte antigen (HLA) peptide by the immune system. Given this model, the search space for identifying HLA peptides can be drastically reduced, greatly increasing identification rates in this very challenging sample type.

As a part of this project, I contributed to the visualization functionality of AlphaPeptDeep and extensive testing of the tool through its successful integration into AlphaViz, described in Article 2.

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

## AlphaPeptDeep: A modular deep learning framework to predict peptide properties for proteomics

Wen-Feng Zeng<sup>1</sup>, Xie-Xuan Zhou<sup>1</sup>, Sander Willems<sup>1</sup>, Constantin Ammar<sup>1</sup>, Maria Wahle<sup>1</sup>,  
Isabell Bludau<sup>1</sup>, Eugenia Voytik<sup>1</sup>, Maximilian T. Strauss<sup>2</sup>, Matthias Mann<sup>1,2,\*</sup>

1. Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry,  
Martinsried, Germany

2. Proteomics Program, NNF Center for Protein Research, Faculty of Health Sciences,  
University of Copenhagen, Copenhagen, Denmark

\* E-mail: [mmann@biochem.mpg.de](mailto:mmann@biochem.mpg.de)

### Abstract

Machine learning and in particular deep learning (DL) are increasingly important in mass spectrometry (MS)-based proteomics. Recent DL models can predict the retention time, ion mobility and fragment intensities of a peptide just from the amino acid sequence with good accuracy. However, DL is a very rapidly developing field with new neural network architectures frequently appearing, which are challenging to incorporate for proteomics researchers. Here we introduce AlphaPeptDeep, a modular Python framework built on the PyTorch DL library that learns and predicts the properties of peptides (<https://github.com/MannLabs/alphapeptdeep>). It features a model shop that enables non-specialists to create models in just a few lines of code. AlphaPeptDeep represents post-translational modifications in a generic manner, even if only the chemical composition is known. Extensive use of transfer learning obviates the need for large data sets to refine models for particular experimental conditions. The AlphaPeptDeep models for predicting retention time, collisional cross sections and fragment intensities are at least on par with existing tools. Additional sequence-based properties can also be predicted by AlphaPeptDeep, as demonstrated with a novel HLA peptide prediction model to improve HLA peptide identification for data-independent acquisition.

### Introduction

The aim of MS-based proteomics is to obtain an unbiased view of the identity and quantity of all the proteins in a given system<sup>1,2</sup>. This challenging analytical task requires advanced liquid chromatography – mass spectrometry (LC/MS) systems as well as equally sophisticated bioinformatic analysis pipelines<sup>3</sup>. Over the last decade, machine learning (ML) and in particular deep neural network (NN)-based deep learning (DL) approaches have become very powerful and are increasingly beneficial in MS-based proteomics<sup>4,5</sup>.

Identification in proteomics entails the matching of fragmentation spectra (MS2) and other



properties to a set of peptides. Bioinformatics can now predict peptide properties for any given amino acid sequences so that they can be compared to actually measured data. This can markedly increase the statistical confidence in peptide identifications.

To do this, a suitable ML/DL model needs to be chosen which is then trained on the experimental data. There are a number of peptide properties that can be predicted from the sequence and for each of them different models may be most appropriate. For the peptide retention times in LC, relatively straightforward approaches such as iRT-calculator, RTPredict, and ELUDE have shown good results<sup>6-8</sup>. However, large volumes of training data are readily available in public repositories today and DL models currently tend to perform best<sup>9</sup>. This is also the case for predicting the fragment intensities in the MS2 spectra, where DL models such as our previous model pDeep<sup>10,11</sup>, DeepMass:Prism<sup>12</sup>, Prosit<sup>13</sup> and many subsequent ones now represent the state-of-the-art. They mostly use long-short term memory (LSTM<sup>14</sup>) or gated recurrent unit (GRU<sup>15</sup>) models. Recently, transformers have been adopted in proteomics and show better performance<sup>16,17</sup>. This illustrates the rapid pace of advance in DL and the need for updating proteomics analysis pipelines with them. However, the focus of existing efforts has not been on extensibility or modularity, making it difficult or in some cases impossible to change or extend their NN architectures.

Here we set out to address this limitation by creating a comprehensive and easy to use framework, termed AlphaPeptDeep. As part of the AlphaPept ecosystem<sup>18</sup>, we keep its principles of open source, community orientation as well as robustness and high performance. Apart from Python and its scientific stack, we decided to use PyTorch,<sup>19</sup> one of the most popular DL libraries.

AlphaPeptDeep contains pre-trained models for predicting MS2 intensities, retention time (RT), and collisional cross sections (CCS) of arbitrary peptide sequences or entire proteomes. It also handles peptides containing post-translational modifications (PTMs), including unknown ones with user-specified chemical compositions. AlphaPeptDeep makes extensive use of transfer learning, drastically reducing the amount of training data required.

In this paper, we describe the design and use of AlphaPeptDeep and we benchmark its performance for predicting MS2 intensities, RT, and CCS on peptides with or without PTMs. On challenging samples like HLA peptides, AlphaPeptDeep dramatically boosts performance of peptide identification for data-dependent acquisition. We also describe how AlphaPeptDeep can easily be applied to build and train models for different peptide properties such as an HLA prediction model, which narrows the database search space for data-independent acquisition, and hence improves the identification of HLA peptides with the AlphaPeptDeep-predicted spectral library.

## Results

#### AlphaPeptDeep overview and model training

For any given set of peptide properties that depend on their sequences, the goal of the AlphaPeptDeep framework is to enable easy building and training of deep learning (DL) models, that achieve high performance given sufficient training data (Fig. 1a). Although modern DL libraries are more straightforward to use than before, designing a neural network (NN) or developing a deployable DL model for proteomics studies is not as simple as it could be, even for biologists with programming experience. This is because of the required domain knowledge and the complexity of the different steps involved in building a DL model. The framework of AlphaPeptDeep is designed to address these issues (Fig 1b).

The first challenge is the embedding, which maps amino acid sequences and their associated PTMs into a numeric tensor space that the NN needs as an input. For each amino acid, a 'one-hot encoder' is customarily used to convert it into a 27-length fingerprint vector consisting of 0s and 1s (Online Methods). In contrast, PTM embedding is not as simple. Although recent studies also used one-hot encoding to embed phosphorylation for MS2 prediction via three additional amino acids<sup>16</sup>, this is not extendable to arbitrary PTMs. In pDeep2 (ref<sup>11</sup>), the numbers of C, H, N, O, S, P atoms for a site-specific modification are prepended to the embedding vector which is flexible and can be applied to many different PTMs. AlphaPeptDeep inherits this feature from pDeep2 but adds the ability to embed all the other chemical elements. To make the input space manageable, we use a linear NN that reduces the size of the input vector for each PTM (Online Methods, Extended Fig. 1). This allows efficient embedding for most modification types, except for very complex ones such as glycans. The PTM embedding can be called directly from AlphaPeptDeep building blocks.

To build a new model, AlphaPeptDeep provides modular application programming interfaces (APIs) to use different NN architectures. Common ones like LSTM, convolutional NN (CNN) as well as many others are readily available from the underlying PyTorch library. Recently transformers – attention-based architectures to handle long sequences – have achieved breakthrough results in language processing but were then also found to be applicable to many other areas like image analysis<sup>20</sup>, gene expression<sup>21</sup> and protein folding<sup>22</sup>. AlphaPeptDeep includes a state-of-the-art HuggingFace transformer library<sup>23</sup>. Our framework also easily allows combining different NN architectures for different prediction tasks.

The training and transfer learning steps are mostly generic tasks, even for different NNs. Therefore, we designed a universal training interface allowing users to train the models using just a single line of Python code – `model.train()`. In our training interface, we also provide a "warmup" training strategy to schedule the learning rate for different training epochs (Online Methods). This has proven very useful in different tasks to reduce the bias at the early training stage<sup>24</sup>. Almost all DL tasks can be done on graphical processing units (GPUs) and training a model from scratch on a standard GPU usually take not more than hours in AlphaPeptDeep and is performed only once. Transfer learning from a



pre-trained model is feasible within minutes, even without GPU.

After training, all learned NN parameters should be saved for persistent applications. This can be readily done using DL library functionalities, and is also implemented in AlphaPeptDeep – *model.save()*. In the latter case, AlphaPeptDeep will save the source code of the NN architectures in addition to the training hyperparameters. Thus, the NN code and the whole training snapshot can be recovered even if the source code was accidentally changed in the AlphaPeptDeep or developers' codebase. This is especially useful for dynamic computational graph-based DL libraries such as PyTorch and TensorFlow in 'eager mode' because they allow dynamically changing the NN architectures.

The most essential functionality of the AlphaPeptDeep framework is the prediction of a property of a given peptide of interest. When using only the CPU, one can choose multiprocessing (predicting with multiple CPU cores), making the prediction speed acceptable on regular personal computers (PCs) and laptops (nearly 2h for the entire reviewed human proteome). Prediction on GPU is still an order of magnitude faster. As PyTorch caches the GPU RAM in the first prediction batch, subsequent batches for the same model will be even faster. However, GPU random access memory (RAM) should be released after the prediction stage, thus making the RAM available for other DL models. These steps are automatically done in AlphaPeptDeep within the *model.predict()* functionality.

AlphaPeptDeep provides several templates in the "model shop" module to develop new DL models from scratch for classification or regression with very little code. All these high-level functionalities in AlphaPeptDeep give the user a quick on-ramp and they minimize the effort needed to build, train and use the models. As an illustrative example, we built a classifier to predict if a peptide elutes in the first or second half of the LC gradient using only several lines of code. Training took only ~16 minutes on nearly 350K peptide-spectrum matches (PSMs) on a standard HeLa dataset<sup>25</sup> and the model achieved 95% accuracy in the testing set (Extended Fig. 2).



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

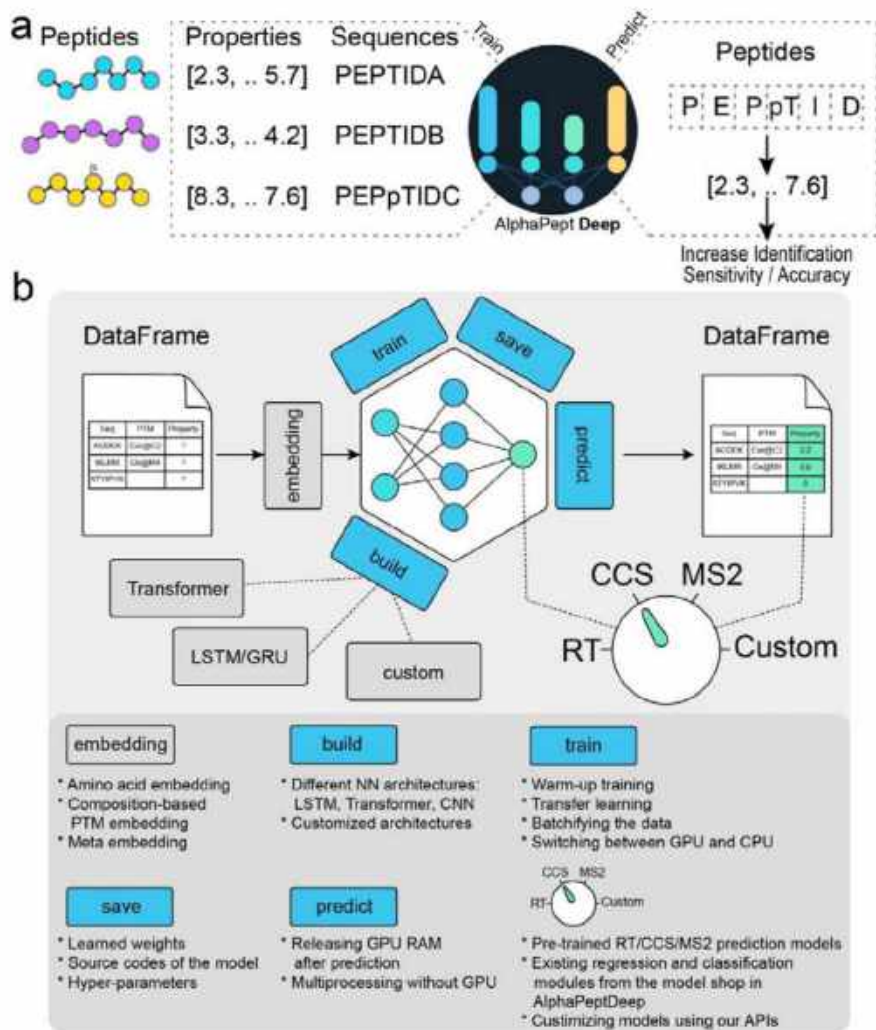


Figure 1. Overview of the AlphaPeptDeep framework. (a) Measured peptide properties are encoded with the respective amino acid sequences and used to train a network in AlphaPeptDeep (left). Once a model is trained, it can be used on arbitrary sets of peptide sequences to predict the property of interest. This then improve the sensitivity and accuracy of peptide identification. (b) The AlphaPeptDeep framework reads and embeds the peptide sequences of interest. Its components include the build functionality in which the model can build. It is then trained, saved and used to predict the property of interest. The dial represents the different standard properties that can be predicted (RT, retention time; CCS, collision section; MS2, intensities of fragment spectra). Custom refers to any other peptide property of interest. lower part lists aspects of the functionalities in more detail.

The MS2, RT, and CCS prediction models in AlphaPeptDeep are all publicly available in

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

our Python modules (Fig. 2). The MS2 prediction model was inherited from pDeep2 but reimplemented on transformers which have been shown very useful in MS2 prediction<sup>16,17</sup>. The pre-trained MS2 model in AlphaPeptDeep is much smaller than other models without sacrificing accuracy (4M parameters vs 64M in the Prosit-Transformer<sup>17</sup>), making the prediction extremely fast (Extended Fig. 3). We also applied the same principle of light-weight models to our RT and CCS models (less than 1M parameters for each, Online Methods), which we built on previous LSTM models<sup>25-27</sup>.

We trained and tested the MS2 models with tens of millions of spectra from a variety of instruments, collision energies and peptides, and trained the RT and CCS models with about half a million RT and CCS values of peptides (Suppl. Data 1). The results of this initial training were then stored as pre-trained models for further use or as a basis for refinement with transfer learning.

Using these pre-trained models and specifically designed data structures (Online Methods), the prediction of a spectral library with MS2 intensities, RT, and ion mobilities (converted from CCS, Online Methods) for the human proteome took only 10 min on a regular GPU and 100 minutes on the CPU with multiprocessing (Extended Fig. 3). As this prediction only needs to be done at most once per project, we conclude that the prediction of libraries by DL is not a limitation in data analysis workflows.

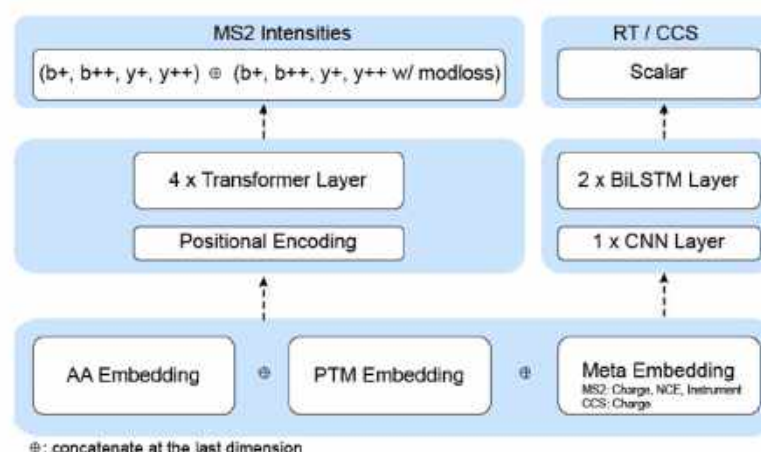


Figure 2. The built-in and pre-trained MS2, RT, and CCS prediction models. The MS2 model is built on four transformer layers, and the RT/CCS models consist of a convolutional neural network (CNN) layer followed by two bidirectional long short-term memory (BiLSTM) layers. The pre-trained MS2 model currently supports predicting the intensities of backbone b/y ions as well as their modification-associated neutral losses if any (e.g. -98 Da loss of phosphorylation on Ser/Thr). However, the user can easily configure the MS2 model to train and predict water and ammonium losses from backbone fragments as well.

### Prediction performance of the AlphaPeptDeep model for MS2 spectra

With the AlphaPeptDeep framework for prediction of MS2 intensities, RT and CCS in hand, we first benchmarked the MS2 model against datasets of tryptic peptides (phase 1 in Fig. 3a). The training and testing data were collected from various instruments and collisional energies (Suppl. Data 1), including ProteomeTools<sup>28</sup>, which were derived from synthetic peptides with known ground truth. We split the data sets in two and trained on a LSTM model similar to pDeep or on the new transformer model. As expected, transformer performed better than the LSTM model on the test datasets (Extended Fig. 4). Overall, on ProteomeTools, 97% of all significantly matching PSMs had Pearson correlation coefficients (PCC) of the predicted vs. the measured fragment intensities of at least 90% (Fig. 3a), which we term 'PCC90' in this manuscript. Note that the experimental replicates also exhibit some variation, making the best possible prediction accuracy somewhat less than 100%. On the ProteomeTools replicates measured with the Lumos mass spectrometer, 99% had PCCs above 90% (Suppl. Data 1), meaning that our predicted intensities mirrored the measured ones almost within experimental uncertainties (99% experimental vs. 97% predicted). Next, we tested the model on the same ProteomeTools sample but measured on a trapped ion mobility Time of Flight mass spectrometer (timsTOF) in dda-PASEF mode<sup>25,29</sup>, and achieved a PCC90 of 87.9% (Suppl. Data 1), showing that the prediction from the pre-trained model is already very good for timsTOF even without adaption.

As expected, our pretrained model performed equally well across different organisms. Interestingly, it did almost as well on chymotrypsin or GluC-digested peptides although it had not been trained on them (Fig. 3a).

HLA class 1 peptides are short pieces of cellular proteins (about 9 amino acids) that are presented to the immune system at the cell surface, which is of great interest to biomedicine<sup>30</sup>. Because of their low abundance and non-tryptic nature, they are very challenging to identify by standard computational workflow, a task in which DL can help<sup>31</sup>. In a second training phase, we added a synthetic HLA dataset, which was also from ProteomeTools<sup>32</sup>, into the training set and trained the model for additional 20 epochs ('fine tuning the model'). We first checked if the new model negatively impacted performance on the tryptic data sets, but this turned out not to be the case (phase 2 in Fig. 3a). On the HLA peptides, however, performance substantially increased the PCC90 from 79% to 92%.

Finally, we extended our model to predict phosphorylated and ubiquitylated peptides, which have spectra somewhat distinct from unmodified peptides. In this case, in addition to backbone fragmentation intensities, AlphaPeptDeep also needs to learn the intensities of fragments with or without modifications. For phosphopeptide prediction, performance of the pre-trained model was much worse, with PCC90 values of only around 30%. However, after training on PTM datasets at phase 3, the performance dramatically increased, almost to the level of tryptic peptides (Fig. 3a). The ubiquitylation prediction (rightmost in Fig. 3a) was already reasonable with the pre-trained model but increased further after phase 3 training (PCC90 from 75% to 93%).



### Prediction performance of the AlphaPeptDeep models for RT and CCS

RT and CCS models are quite similar to each other as their inputs are the peptide sequences and PTMs, and outputs are scalar values. For both we used LSTM architectures. In the CCS prediction model, precursor charge states are considered in the model as well. Taking advantage of the PTM embedding in AlphaPeptDeep, the RT and CCS models naturally consider PTM information, and hence can predict peptide properties given arbitrary PTMs. We trained the RT model on datasets with regular peptides (unmodified and Met-oxidated) from our HeLa measurements<sup>25</sup>.

We first tested the trained RT model on regular peptides from the pan human library<sup>33</sup>. As shown in Fig. 3b, the pre-trained model gave very good predictions in most of the RT range, but failed to accurately predict the last few minutes (iRT (ref<sup>7</sup>) values higher than 100) possibly due to the different flushing settings of the LC in training and testing data. These differences could be addressed by fine-tuning the model with experiment-specific samples. Few-shot fine-tuning with only 500 training samples improved the accuracies of the RT prediction from an  $R^2$  of 0.927 to 0.986.

We also tested the RT model on a phosphopeptide dataset,<sup>34</sup> although the model had not been trained on such data. After fine-tuning on 500 peptides, the  $R^2$  increased from 0.958 to 0.984 (Fig. 3b). As RT behavior of peptides varies with the LC conditions in different experiments, we highly recommend fine-tuning whenever possible. It turns out that few-shot fine-tuning worked well to fit short LC conditions as well (Extended Fig. 5). Finally, as expected, the more training peptides we used, the better the fine-tuning, and with many peptides our model reached  $R^2$  values up to 0.99 (Fig. 3b).

While the CCS model was trained on regular human peptides from our prior HeLa dataset<sup>25</sup> (Suppl. Data. 1), we tested the trained model on E. coli and yeast peptides from the same instrument in the same publication. For these regular peptides the CCS model achieved an  $R^2 > 0.98$  of the predicted and detected CCS values. Next, we searched the HeLa and drosophila data by the open-search mode in Open-pFind<sup>35</sup>, to obtain experimental CCS values for modified peptides (Online Methods). For these peptides in the testing set, the  $R^2$  was 0.965 and 0.953, respectively, a prediction accuracy quite close to the one for regular peptides, even for unexpected modifications. The predicted CCS values can be converted to ion mobilities of Bruker timsTOF using the Mason Schamp equation.<sup>36</sup>

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

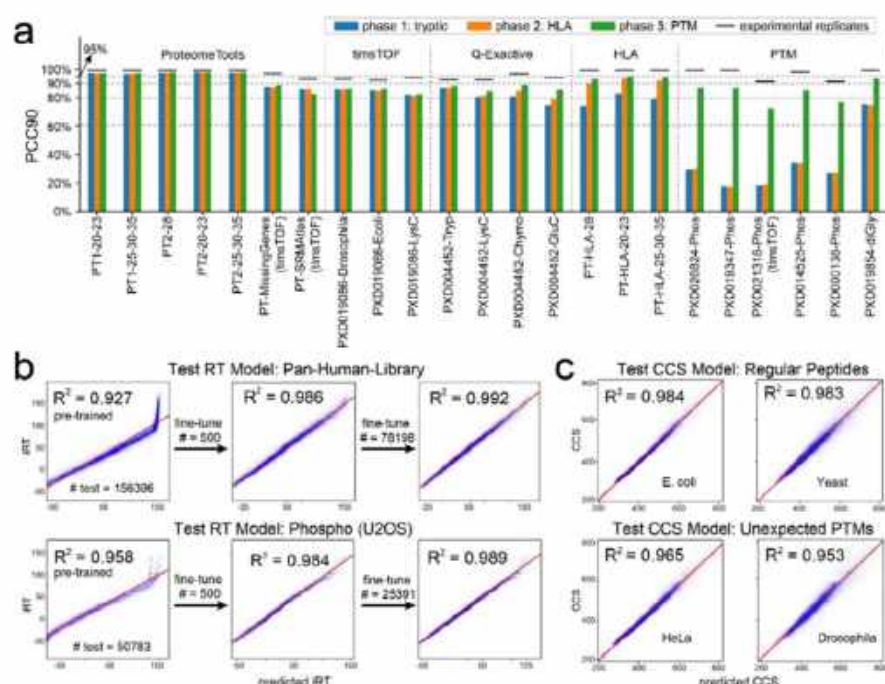


Figure 3. The performance of MS2/RT/CCS models. (a) The MS2 prediction accuracies of the three training phases on different testing datasets. The performance is evaluated by "PCC90" (percentage of PCC values larger than 0.9). The prefix 'PT' of each data set refers to ProteomeTools. PT1 and PT2 refer to ProteomeTools part I and II, respectively. The black bars are the PCC90 values of experimental spectra. (b) For RT prediction, few-shot learning can correct the RT bias between different LC conditions. (c) Our CCS model works well for both regular (top panels) and unexpectedly modified (bottom panels) peptides.

### Prediction performance for 21 PTMs with transfer learning

To further demonstrate the powerful and flexible support for PTMs in AlphaPeptDeep, we tested the pre-trained tryptic MS2 (phase 1 in Fig. 3a) and RT models using the 21 PTMs, which were synthesized based on 200 template peptide sequences<sup>37</sup>.

Interestingly, there is a group of modifications for which the prediction of MS2 spectra is as good as the values of unmodified peptides (Fig. 4a). These include Hydroxypro@P, Methyl@R, and Dimethyl@R for which the PCC90 was greater than 80%. This is presumably because these modifications do not change the overall fragmentation pattern much. In contrast, most of the other PTMs cannot be well predicted by the pre-trained models, for example, the PCC90 values were less than 10% for Malonyl@K and Citrullin@R. Remarkably, transfer learning for each PTM type using as few as ten peptides with different charge states and collisional energies greatly improved the

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

prediction accuracies on the testing data. The largest improvements of PCC90 were as high as 60% (Citrullin@R and Malonyl@K, Fig. 4a). Overall, compared with the pre-trained model, the ones tuned by ten peptides improved the PCC90 from a median of 48% to 87% (Fig. 4b). We speculate that this is because the fragmentation properties of amino acids at different collisional energies have been well learned by the pre-trained model after which transfer learning only needs to learn the properties of modified ones. Including 50 PTM bearing peptides improved this number to 93% whereas using 80% of all the identified peptides ( $n \leq 200$ ) with these PTMs only improved prediction by another 2%. This demonstrates that our models can be adapted to novel situations with very little additional data, due to the power of transfer learning.

AlphaPeptDeep has been included in AlphaViz<sup>38</sup>, a tool suite for RAW MS data visualization (<https://github.com/MannLabs/alphaviz>), which among other features allows users to visualize a mirrored plot between experimental and predicted spectra. As an example, the MS2 prediction of the peptide "AGPNASIIISLKSDK-Biotin@K11" before and after transfer learning is displayed in Fig. 4c. The y12++ ion was first wrongly predicted by the pre-trained model, but this was corrected after transfer learning with only 50 other biotinylated peptides. AlphaPeptDeep also allows users to visualize the 'attention' weights— a key feature of transformer models — showing what data attributes were important for the prediction. To depict the attention changes between pre-trained and transfer learning transformer models, we used the BertViz package (<https://github.com/jessevig/bertviz>) (Extended Fig. 6).

Next, we tested the performance of our pre-trained RT model using the datasets of 21 PTMs. Although the model was never trained on any of these PTMs, the accuracy of RT prediction on these peptides exceeds that of DeepLC<sup>39</sup>, an RT prediction model designed for unseen PTMs ( $R^2$  of 0.95 of AlphaPeptDeep vs. 0.89 of DeepLC, Fig. 4d and 4e). In this case, transfer learning only slightly improves the results, presumably because some of these synthetic modified peptides elute in very broad peaks, which makes them hard to predict.



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

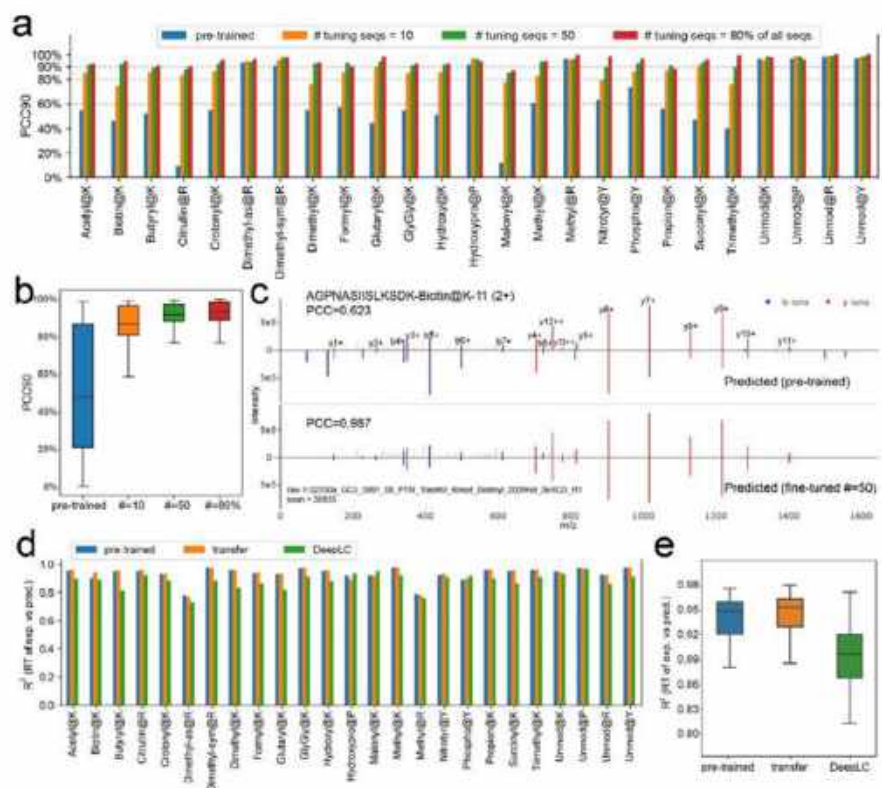


Figure 4. Model performance with transfer learning on 21 PTMs from ProteomeTools. (a) The accuracy of MS2 prediction with different numbers of peptides for transfer learning for each PTM. Each PTM is tested separately, "80% seqs" refers to using 80% of the identified modified sequences for transfer learning. (b) Overall accuracy without unmodified peptides from (a). (c) Transfer learning dramatically improves the MS2 prediction of the example peptide "AGPNASIIISLKSDK-Biotin@K11" (tuned by 50 other peptides). (d) Comparisons of RT prediction for each PTM on pre-trained and transfer learning (by 50% of all the identified peptides) models, as well as DeepLC models. (e) Overall R<sup>2</sup> distribution without unmodified peptides from (d).

**Boosting Data-Dependent Acquisition (DDA) identification of HLA peptides**

As explained above, HLA peptides are among the most challenging samples for MS-based proteomics. Given the excellent model performance of the transformers in AlphaPeptDeep, we hypothesized that prediction of their MS2 spectra could substantially improve their identification.

The non-tryptic nature of these peptides results in an extremely large number of peptides that need to be searched, leading to a decreased statistical sensitivity at a given false discovery rate (FDR) level (usually 1%). The key idea of using MS2, RT and CCS

prediction to support HLA peptide identification is that, for correct peptides of the searched spectra, the predicted properties should be very close to the detected ones, while the predicted properties of the irrelevant peptides tend to be randomly distributed. Therefore, the similarities or differences between the predicted and detected properties can be used as machine learning features to distinguish correct from false identifications using semi-supervised learning. Such an approach has long been implemented in Percolator and later in other tools to re-score PSMs<sup>40</sup>, which increases the sensitivity at the same FDR level<sup>31,32</sup>. However, due to the lack of support for arbitrary PTMs with DL models it has not been for open-search of HLA peptides. Modern protein open-search engines like pFind<sup>35</sup> can perform very fast unspecific peptide search without limiting the peptide mass window using the sequence tag technique<sup>41</sup>, enabling the identification of unexpected PTMs.

AlphaPeptDeep fully supports the Percolator algorithm for regular as well as open-search of HLA peptides (Online Methods). To accelerate the rescoring, we calculate the fragment intensity similarities between predicted and detected spectra on a GPU, making the rescoring process extremely fast (~1 hour to rescore 16,812,335 PSMs from 424 MS runs using a PC with a GeForce RTX 3080 GPU, where ~60% of the time was used for loading the RAW files). This means that the rescoring by AlphaPeptDeep is not a bottleneck for HLA peptide search.

To investigate how much AlphaPeptDeep can boost the HLA peptide search, we applied it on two datasets, MSV000084172 from samples in which particular mono-allelic HLA-I types were enriched<sup>42</sup>, here referred to as the 'mono-allelic dataset' and our published dataset from tumor samples (PXD004894<sup>43</sup>) referred to as the 'tumor dataset'. These two datasets had been analyzed with a regular search engine (MaxQuant<sup>44</sup>) by the Kuster group<sup>32</sup> (Fig. 5a) and we here used pFind in the open-search mode (Fig. 5b).

First, we wanted to compare the AlphaPeptDeep results with MaxQuant as well as Prosit, a recently published DL based tool that has also been applied to HLA peptides<sup>32</sup>. Since Prosit only supports fixed iodoacetamide modification on alkylated peptides (IAA in Fig. 5a), we only used the results of the same IAA RAW files in rescoring. On the mono-allelic and the tumor datasets, AlphaPeptDeep covered 93% and 96% of the MaxQuant results while more than doubling the overall numbers at the same FDR of 1% (Fig. 5a). Compared to Prosit, AlphaPeptDeep captured 91% of their peptides and still improved the overall number on the mono-allelic dataset by 7%.

Next, we searched both datasets with the open-search mode of pFind (Fig. 6b), and rescored the results in AlphaPeptDeep. Here, both alkylated and non-alkylated peptides were analyzed. Interestingly, the open-search itself already identified similar numbers of peptides as the DL-boosted regular search, but AlphaPeptDeep further improved the total number of identified peptides by 29% and 42%, while retaining 99% and 98% of the pFind hits at the same FDR for the mono-allelic and tumor datasets, respectively (Fig. 5b). This demonstrates the benefits of AlphaPeptDeep's support of open-search for HLA

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

peptide analysis.

AlphaPeptDeep with open-search identified PTMs such as phosphorylation, which are known to exist on HLA peptides but are very difficult to identify by regular unspecific search. For the mono-allelic dataset we identified a total of 490 phosphopeptides. To gauge the biological reasonability of these peptides, we searched for sequence motifs of both the phosphorylated and non-phosphorylated peptides. This revealed the expected HLA peptide motifs, dominated by the anchor residues for their cognate major histocompatibility complex proteins. Only the phospho-HLA peptides additionally had linear phospho-motifs, like the prominent SP motif common to proline directed kinases (Fig. 5c and Extended Fig. 7). We also identified 359 phospho-HLA peptides from the tumor dataset, with similar phospho-motifs (Extended Fig. 7). We further used AlphaPeptDeep to inspect retention time and MS2 spectrum similarities (Extended Fig. 8). Note that the MS2 and RT models were only fine-tuned by at most 100 phospho-PSMs from eight RAW files (Online Methods), so most of the phosphopeptides from other RAW files were not used in fine-tuning. Our method was also able to identify other PTMs associated with HLA peptides, such as cysteinylolation<sup>45</sup> (Extended Fig. 9). Overall, most of the HLA peptides additionally identified by this method had modifications related to sample preparation, such as deamidation, N-terminal pyro-Glu, and N-terminal carbamidomethylation (Extended Fig. 9).



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

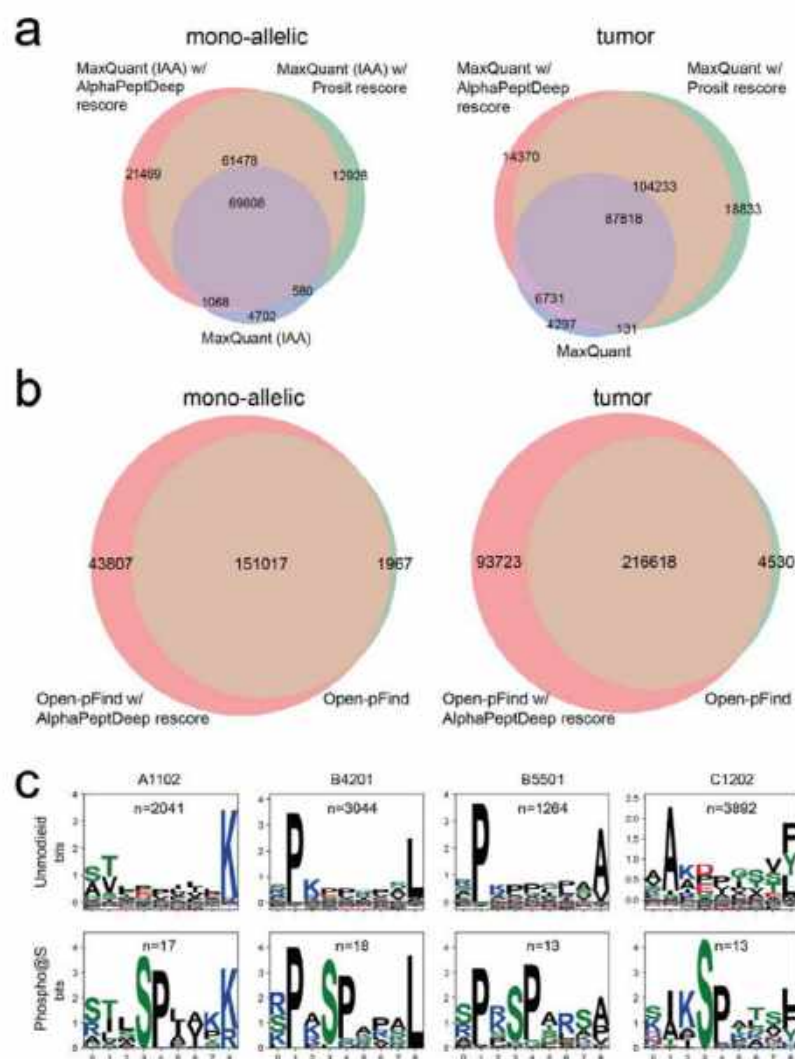


Figure 5. AlphaPeptDeep drastically improves upon regular (a) and open-search (b) results for DDA identification of HLA peptides. IAA refers iodoacetamide alkylated peptides. (c) Logo plots of unmodified and phosphorylated peptides with nine amino acids identified by open-search for four different HLA types. Logo plots were generated by LogoMaker.<sup>46</sup>

### Building an HLA prediction model for HLA DIA Search

In recent years, DIA has become a method of choice to generate large-scale proteome datasets. DIA data analysis traditionally requires DDA experiments to generate a library to which the data is then matched<sup>47</sup>. These libraries contain RT, ion mobility (if applicable)

and the most intense and specific fragments for each peptide. The generation of experimental libraries is laborious and sample consuming. With the development of DL in proteomics, libraries with predicted RT, CCS/ion mobilities and fragment intensities from whole proteome sequences are becoming more and more popular, although there is still a debate about whether measured or predicted libraries are preferable. This is because the large search space introduced by purely in silico libraries can make FDR control difficult.

DIA for HLA peptide analysis is also getting more attention<sup>48,49</sup>. So far, these efforts have been restricted to experimental DIA libraries because analysis with a predicted HLA library is far more challenging than with an experimental one. This is mainly because HLA peptides are not tryptic, meaning they do not follow specific cleavage rules and do not necessarily have a favorable fragmentation pattern. The number of theoretical peptides with amino acid lengths between 8 and 14 from a reviewed human proteome is more than 70M, which is nearly two orders of magnitude more than that of tryptic peptides in the same length range (~900K). Due to this enormous search space, a predicted library is difficult or even impossible to search by state-of-the-art DIA search tools such as DIA-NN<sup>50</sup> and Spectronaut<sup>51</sup>.

Fortunately, HLA peptides follow certain sequence motifs guided by the HLA-types that are present. We reasoned that these motifs could be learned by DL for more efficient peptide identification. To test this hypothesis, we built an HLA prediction model using the model shop functionalities in our AlphaPeptDeep framework (Online Methods). This model - a binary LSTM classifier predicts if a given sequence is likely to be an HLA peptide presented to the immune system and extracts these peptides from the human proteome sequence. There are two main goals of the model: (1) keep as many actually presented HLA peptides as possible (i.e., high sensitivity); and (2) reduce the number of predicted peptides to a reasonable number (i.e., high specificity). Note that sensitivity is more important here as we hope that all measured HLA peptides are still in the predicted set.

Based on these goals, we developed a pipeline which enables predicted library search for DIA data (Fig. 6a). In brief, we first trained a pan-HLA prediction model with peptides from all known HLA types ('pre-trained model' in Fig. 6a). Normally, only a few HLA types are actually present in the samples from any given individual. Therefore, we used transfer learning to create a person-specific model with sample-specific peptides ('tuned model' in Fig. 6a). This model should then be able to predict whether an HLA peptide is potentially present in the sample or not, thus further reducing the number of peptides to be searched and increasing prediction accuracy. For this strategy, we need to identify a number of sample specific HLA peptides. This can be done directly from the already acquired DIA data by a 'direct-DIA search'<sup>52</sup> obviating the need for a separate DDA experiment. This involves grouping eluting fragment detected peaks belonging to the same peptide signal into a pseudo-spectrum for DIA data, and then searching the pseudo-spectrum with conventional DDA search algorithms.

To test this pipeline, we used the HLA-I dataset of the RA957 cell line in PXD022950<sup>48</sup>. We started with our pan-HLA prediction model from 94 known HLA types (Fig. 5). It reduced the number of sequences from 70M to 7M with 82% sensitivity. However, 7M peptides are still too many to search and the model would have lost 18% of true HLA peptides. Furthermore, the pre-trained model is not able to identify unknown HLA types as it is only trained on already known ones.

To enable transfer learning, we searched RA957 data with DIA-Umpire<sup>52</sup>. It identified 12,998 unique sequences with length from 8 to 14. We used transfer learning on 80% of this data to train the sample specific HLA model while keeping 20% for testing. This dramatically increased the specificity to 96% with 92% sensitivity (note that this is judged on the identifications by direct-DIA; thus our sensitivity may be even higher). The number of HLA peptides predicted by this model is 3M, which is comparable to the tryptic human proteome library.

Having predicted our sample-specific HLA peptides, including their MS2 fragment spectra and RTs, we used this as input for a DIA-NN search of the DIA data. Our workflow identified 36,947 unique sequences. PEAKS-Online<sup>53</sup> is a very recently published tool which combines searching a public library, direct-DIA, and *de novo* sequencing. It identified 30,733 unique sequences within the same length range. Our workflow almost tripled the number of unique sequences of DIA-Umpire and obtained 20% more than PEAKS-Online. As a reference, MaxQuant identified 14,563 sequences in the 8 to 14 aa range on DDA of the same sample in the original publication<sup>48</sup>.

To judge the reliability of the identified HLA peptides, we used MixMHCpred<sup>54</sup> to deconvolute these identified peptides at the 5% rank level based on the HLA type list in the original publication of the datasets<sup>48</sup> (Fig. 6b). The overall peptide distribution identified by our pipeline for different HLA types was very similar to that of the original DDA data, indicating that our additionally identified HLA peptides were reliable at the same level.

Finally, we directly tested if our pipeline is also able to identify peptides with unknown HLA types. To simulate this situation, we removed all peptides of the dominant HLA-A\*68:01 and used the rest to train a new pan-HLA model. This means that all HLA-A\*68:01 peptides in the RA957 sample were now unknown. Then we used only 100 HLA-A\*68:01 and all non-HLA-A\*68:01 peptides identified by direct-DIA and deconvoluted by MixMHCpred for transfer learning. The resulting library then identified 29,331 peptides including 7,868 from HLA-A\*68:01 (Transfer learning with 1000 HLA-A\*68:01 peptides retrieved almost all of them) (Fig. 6b). This demonstrates that few-shot transfer learning is able to rescue many of the peptides of an unknown HLA type even if the peptide number is low after direct-DIA identification.



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

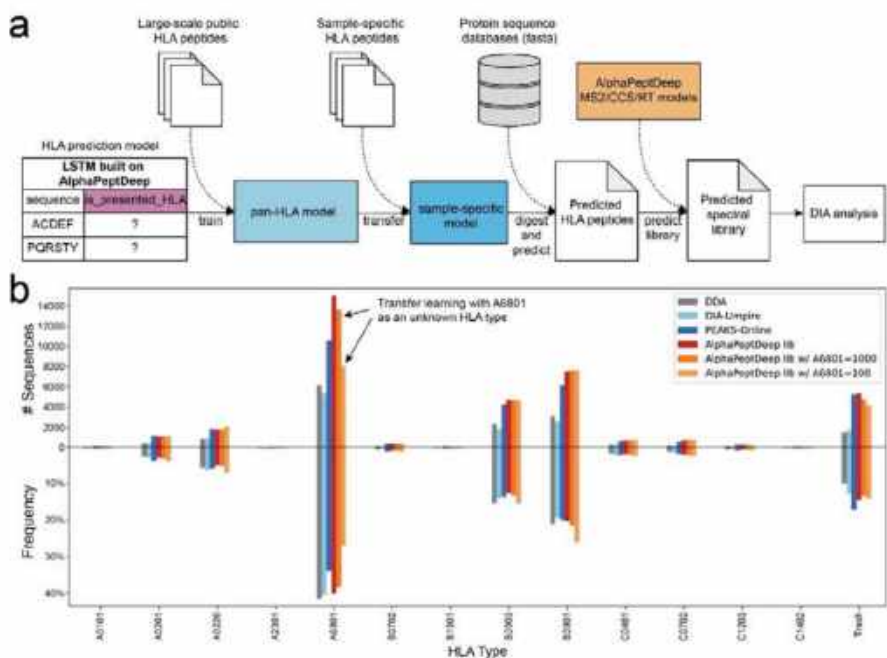


Figure 6. HLA prediction model built on AlphaPeptDeep functionalities. (a) The pipeline with the HLA prediction model to extract potential HLA peptides from the proteome databases. The HLA model is a binary classifier that predicts if a given sequence is a potentially presented HLA sequence. (b) Our HLA prediction model boosts the number of identified HLA-I peptides compared to other tools. Cell line HLA data from RA957 with sequence lengths from 8 to 14 were used. The top bar plots show the number of identified unique sequences of HLA types for each search method. The bottom bar plots the relative frequency of these HLA types. 'Trash' means the peptides cannot be assigned to any HLA types by MixMHCpred at 5% rank level. 'AlphaPeptDeep lib' (red) refers to the library predicted by the sample-specific HLA model and our MS2 and RT models. The bars represent DDA data analyzed by MaxQuant, and the DIA data analyzed by DIA-Umpire, or PEAKS-Online including de novo sequencing. AlphaPeptDeep with the sample-specific HLA library clearly outperforms these. The results of omitting the dominant HLA-A\*68:01 (A6801) HLA type in the pan-model and using transfer learning with including 1000 or 100 of these peptides identified by direct-DIA from the data are shown in the last two bars of the A6801 type (see arrows in the panel).

### Conclusion

We developed a deep learning framework called AlphaPeptDeep that unifies high-level functionalities to train, transfer learn and use the models for peptide property prediction. Based on these functionalities, we built MS2, RT and CCS models, which enabled the prediction for a large variety of different PTM types. These models can boost DDA identification of for example, HLA peptides, not only in regular search but also in open-search. We also provided a module called 'model shop' which contains generic models so that users can develop new ones from scratch with just a few lines of code. Based on the model shop, we also built an HLA prediction model to predict whether a

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

peptide sequence is a presented HLA peptide. With the HLA model and the MS2, RT and CCS models in AlphaPeptDeep, we predicted the HLA spectral libraries directly from the whole human proteome, and searched them using HLA DIA data. This is the first time that the predicted libraries at the proteome-level have been used to search DIA data for HLA peptides. Our predicted libraries out-performed other methods including recently published pipelines specifically designed for HLA DIA analysis.

Although AlphaPeptDeep is both powerful and easy to use, we note that traditional machine learning issues, such as overfitting in the framework, still need to be kept in mind. For instance, users still need to split the data, train and test the models on different sets. Trying different hyperparameters such as the number of training epochs is still necessary as well. Different mini-batch sizes and learning rates may also impact on the model training. However, the model shop at least provides baseline models for any property prediction problem.

We hope AlphaPeptDeep will minimize the challenges for researchers that are not AI experts to build their own models either from scratch or on top of our pre-trained models. As we pointed out in our recent review<sup>4</sup>, peptide property prediction can be involved in almost all steps to improve the computational proteomics workflow. Apart from specific properties of interest in MS-based proteomics, it can in principle be used to solve any problem where a peptide property is a function of the amino acid sequence, as we demonstrated by successfully predicting potential HLA peptides to narrow the database search. Therefore, with sufficient and reliable training data, we believe AlphaPeptDeep will be a valuable DL resource for proteomics.

## Online Methods

### Infrastructure development

To develop AlphaPeptDeep, we built an infrastructure package named AlphaBase (<https://github.com/MannLabs/alphabase>) which contains many necessary functionalities for proteins, peptides, PTMs, and spectral libraries. In AlphaBase, we use the pandas DataFrame as the base data structure, which allows transparent data processing in a tabular format and is compatible with many other Python packages. AlphaPeptDeep uses the AlphaBase DataFrames as the input to build models and predicts properties of peptides. Amino acid and PTM embedding is performed directly from 'sequence' (amino acid sequence), 'mods' (modification names), and 'mod\_sites' (modification sites) columns in the peptide DataFrame.

### Amino acid embedding

Each amino acid of a sequence is converted to a unique integer, for example, 1 for 'A', 2 for 'B', ..., and 26 for 'Z'. Zero is used as a padding value for N- and C-terminals, and other "padding" positions. As a result, there are 27 unique integers to represent an amino acid sequence. A 'one-hot encoder' is used to map each integer into a 27-D vector with

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

zeros and ones. These vectors are mapped to an N-dimensional embedded vector using a linear layer (Extended Fig. 1). For this, we additionally make use of the `'torch.Embedding'` method, which is more efficient and flexible and can support more letters such as all the 128 ASCII codes.

#### **PTM embedding**

For each PTM, we use a 6-D embedding vector to represent the C, H, N, O, S, and P atoms. All other atoms of a PTM are embedded into a 2-D vector with a fully connected (FC) layer. The 6-D and 2-D vectors are concatenated into an 8-D vector to represent the PTM (Extended Fig. 1).

#### **MS2 model**

The MS2 model consists of an embedding layer, positional encoder layer, and four transformer layers followed by two FC layers. The embedding layer embeds not only amino acid sequences and modifications but also metadata (if necessary) including charge states, normalized collisional energies, and instrument type. All these embedded tensors are concatenated for the following layer.

We added an additional transformer layer to predict the 'modloss', which refers to neutral loss intensities of PTMs, for example, the -98 Da of the phospho-group. This modloss layer can be turned off by setting 'mask\_modloss' as 'True'. The output layer dimension is  $(n - 1) \times 8$  for each peptide, where  $n$  is the length of the peptide sequence, and 8 refers to eight fragment types, i.e. b+, b++, y+, y++, b\_modloss+, b\_modloss++, y\_modloss+, and y\_modloss++. With 'mask\_modloss=True', the modloss layer is disabled and the predicted modloss intensities are always zero. The hidden layer size of transformers is 256. The total number of the model parameters is 3,988,974.

All matched b/y fragment intensities in the training and testing datasets were normalized by dividing by the highest matched intensity for each spectrum. The MS2 models were trained based on these normalized intensities. For prediction, negative values will be clipped to zero, hence the predicted values will be between zero and one.

In training phase 1, we only used tryptic peptides in the training datasets. The training parameters were: epoch=100, warmup epoch=20, learning rate (lr)=1e-5, dropout=0.1. In training phase 2, HLA peptides were added to the training set and the parameters were: epoch=20, warmup epoch=5, lr=1e-5, dropout=0.1, mini-batch size=256. In phase 3, phosphorylation and ubiquitylation datasets were added for training, and only phosphorylation sites with >0.75 localized probabilities were considered. The training parameters were: epoch=20, warmup epoch=5, lr=1e-5, dropout=0.1, mini-batch size=256. For transfer learning of the 21 PTMs, the parameters were: epoch=10, warmup epoch=5, lr=1e-5, dropout=0.1, mini-batch size depends on the peptide length. L1 loss was used for all training phases. We used the "cosine schedule with warmup" method implemented in HuggingFace for warmup training of these models including all the following models.



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

For Thermo Orbitrap instruments, the fragment intensities of each identified PSM are directly extracted from the raw data. For this, we imported the centroided MS2 spectra with Thermo's RawFileReader API that is integrated in AlphaPept, hence the extracted intensities are reproducible across different search engines. For dda-PASEF data, the b/y ion intensities are extracted directly from the msms.txt file of MaxQuant results. Note that different search engines may have different centroiding algorithms for dda-PASEF, resulting in quite different fragment intensities, so fine-tuning is highly recommended for dda-PASEF data analyzed by different software.

A fragment DataFrame is designed to store the predicted intensities. Its columns are fragment ion types (e.g., 'b\_z1' for b+ and 'y\_z2' for y++ ions), and the rows refer to the different fragmented positions of peptides from which the fragments originate. The start and end pointers of the rows ('frag\_start\_idx' and 'frag\_end\_idx') belonging to peptides are stored in the peptide DataFrame to connect between peptides and their fragments. The fragment DataFrame is pre-allocated only once for all peptides before prediction. While predicting, the predicted values of a peptide are assigned to the region of the peptide located by 'frag\_start\_idx' and 'frag\_end\_idx'. The fragment DataFrame allows fast creation and storage of the predicted intensities. The tabular format further increases human readability and enables straightforward access by programming.

#### RT model

The RT model consists of an embedding layer for sequences and modifications, and a CNN layer followed by two LSTM layers with a hidden layer size of 128. The outputs of the last LSTM layer are summed over the peptide length dimension and processed by two FC layers with output sizes of 64 and 1. The total number of the model parameters is 708,224.

All RT values of PSMs in the training datasets were normalized by dividing by the time length of each LC gradient, resulting in normalized RT values ranging from 0 to 1. As a result, the predicted RTs are also normalized. The training parameters were: epoch=300, warmup epoch=30, lr=1e-4, dropout=0.1, mini-batch size=256. The fine-tuning parameters are: epoch=30, warmup epoch=10, lr=1e-4, dropout=0.1, mini-batch size=256. L1 loss was used for training.

To compare predicted RT values with experimental ones, each value is multiplied with the time length of each LC gradient. For testing on peptides with iRT values, we used 11 peptides with known iRT values<sup>7</sup> to build a linear model between their iRT and predicted RT values. Then all the predicted RTs in the testing sets are converted to iRT values using the linear model.

#### CCS model

The CCS model consists of an embedding layer for sequence, modifications and charge states, and a CNN layer followed by two LSTM layers with a hidden layer size of 128. The

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

outputs of the last LSTM layer are summed over the peptide length dimension and processed by two FC layers with output sizes 64 and 1. The total number of the model parameters is 713,452.

The training parameters are: epoch=300, warmup epoch=30, lr=1e-4, dropout=0.1, mini-batch size=256. L1 loss was used for training. The predicted CCS values are converted to mobilities of Bruker timsTOF using the Mason Schamp equation.<sup>36</sup>

#### Rescoring

Rescoring includes three steps:

1. Model fine-tuning. 5,000 PSMs are randomly sampled from at most eight RAW files at 1% FDR to fine-tune the MS2, RT and CCS (if applicable) models to obtain project-specific models. The top-10 frequent modifications are also selected for fine-tuning from the eight RAW files. At most 100 PSMs are sampled for each modification. Therefore, the fine-tuning covers not only unmodified peptides, but also modified peptides.
2. Deep learning feature extraction. The tuned MS2, RT and CCS models are used to predict MS2, RT and CCS values for all the reported PSMs including decoys. All PSMs are matched against the MS2 spectra in the RAW files to obtain detected fragment intensities. Then the predicted and detected values are used to calculate 61 score features, which include correlations of fragments, RT differences, mobility differences, and so on (Suppl. Data. 2).
3. Percolator for rescoring. We use the cross-validation schema<sup>55</sup> to perform the semi-supervised Percolator algorithm to reduce the chance of overfitting. All the peptides are divided into K folds (K=2 in the analyses of this work) and rescored by 5 iterations in Percolator. In each iteration, a Logistic regression model from scikit-learn<sup>56</sup> is trained with the 61 features on the K-1 folds, and the model is used to re-score on the remainder. All the K folds will be re-scored after repeating this for K times on each of the folds.

Multiprocessing is used in step 2 for faster rescoring. Because GPU RAM is often limited, it can become a bottleneck meaning that only one process is allowed to access the GPU space at a time for prediction. We developed a producer-consumer schema to schedule the tasks with different processes (Extended Fig. 10). The PSMs are matched against MS2 spectra in parallel with multiprocessing grouped by RAW files. Then, they are sent back to the main process for prediction in GPU. At last, the 61 Percolator features are extracted in parallel again. All correlation values between matched and predicted MS2 intensities are also calculated in GPU for acceleration. As this is not memory intensive, the GPU RAM can be shared and used in parallel from different processes. For multiprocessing without GPU, all predictions are done with separate processes and results merged into the main process to run Percolator.

#### HLA prediction model

The HLA prediction model consists of an embedding layer for sequences, a CNN layer



followed by two LSTM layers with a hidden layer size of 256. The outputs of the last LSTM layer are summed over the sequence length dimension and processed by two linear layers with output sizes of 64 and 1. The sigmoid activation function is applied for last linear layer to obtain probabilities. The total number of the model parameters is 1,669,697.

For training and transfer learning, identified HLA peptides with sequence lengths from 8 to 14 are regarded as positive samples. Negative samples were randomly picked from the reviewed human protein sequences. The sequence number and length distribution were the same for the positive and negative samples. These samples were then split 80% for training and 20% for testing. The parameters for training the pre-trained model were: epoch=100, warmup epoch=20, lr=1e-4, dropout=0.1. For transfer learning, the DIA data were searched by DIA-Umpire and MSFragger<sup>57</sup> in HLA mode at 1% FDR with reviewed human protein sequence. The parameters for transfer learning were: epoch=50, warmup epoch=20, lr=1e-5, dropout=0.1, mini-batch size=256. Binary cross-entropy loss was used for training.

To predict HLA peptides from fasta files, we first concatenate protein sequences into a long string separated by the "\$" symbol. Next, we use the longest common prefix (LCP) algorithm<sup>58</sup> to accelerate the unspecific digestion for the concatenated sequence. Only the start and end indices of the peptides in concatenated sequence are saved, thus minimizing the usage of RAM. These indices are used to generate peptide sequences on the fly for prediction. The LCP functionalities have been implemented in AlphaBase. All sequences with a predicted probability larger than 0.7 were regarded as potential HLA peptides.

#### Open-search for Orbitrap and dda-PASEF data

We performed an open search on the Thermo RAW data with Open-pFind. For HLA DDA data, the reviewed human protein sequences from UniProt (<https://www.uniprot.org/>) were searched with the following parameters: open-search mode=True, enzyme=Z at C-terminal (i.e., unspecific enzyme), specificity=unspecific. The search tolerance was set to  $\pm 10$  ppm for MS1 and  $\pm 20$  ppm for MS2. All modifications marked as 'isotopic label' in UniMod ([www.unimod.org](http://www.unimod.org)) were removed from the searched modification list. The FDR was set as 1% at the peptide level.

To enable Open-pFind search for dda-PASEF data, the spectra were loaded by AlphaPept APIs<sup>18</sup> and exported as pFind compatible MGF files using our in-house Python script. The reviewed drosophila and human sequences were used to search the respective tryptic DDA data with parameters: open-search mode=True, enzyme=KR at C-terminal, enzyme specificity=specific. The search tolerance was set to  $\pm 30$  ppm for both MS1 and MS2.

#### Spectral libraries

Functionalities for spectral libraries are implemented in AlphaBase. When providing DataFrames with sequence, modification and charge columns, the fragment m/z values and intensities are calculated and stored in fragment DataFrames. AlphaBase also



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

integrates functionalities to load and save DataFrames in a single Hierarchical Data Format (HDF) file for fast access. For subsequent use with DIA-NN or Spectronaut, all the DataFrames are then converted into a tab-separated values file (\*.tsv) which is compatible with these tools.

For HLA DIA analysis, we used reviewed human protein sequences to predict HLA peptides. We considered charge states from one to three for each peptide. All RT, CCS, and MS2 were predicted using the model from training phase 3. The 12 most abundant b/y ions with 1+ and 2+ charge states were written to the \*.tsv file. Fragment m/z range was set to be from 200 to 1800, precursor m/z range was from 300 to 1800.

In DIA-NN, the mass tolerance for MS1 and MS2 were set to 20 and 10 ppm respectively, with a scan window of 8. All other parameters were the default values of DIA-NN. The results identified from the first pass were used for post-search analysis.

#### Data availability

The reviewed protein sequence databases of human, E. coli, fission yeast, and drosophila were downloaded from uniprot (<https://www.uniprot.org/>). The training and testing data were from PRIDE with IDs: PXD010595, PXD004732, PXD021013, PXD009449, PXD000138, PXD019854, PXD019086, PXD004452, PXD014525, PXD017476, PXD019347, PXD021318, PXD026805, PXD026824, PXD029545, PXD000269, and PXD001250.

The mono-allelic HLA DDA dataset was downloaded from MassIVE with ID MSV000084172. The tumor HLA dataset was downloaded from PRIDE with ID PXD004894.

HLA DIA data and the MaxQuant results of DDA data from the RA957 cell line were downloaded from PRIDE with ID PXD022950. HLA DIA results of PEAKS-Online were downloaded from the PEAKS-Online publication.<sup>53</sup> Only results from RAW files '20200317\_QE\_HFX2\_LC3\_DIA\_RA957\_R01.raw' and '20200317\_QE\_HFX2\_LC3\_DIA\_RA957\_R02.raw' from RA957 were used to compare different methods.

Result files and Python notebooks to reproduce the analysis results in this study (total of 7 GByte) can be found in <https://doi.org/10.6084/m9.figshare.20260761>.

#### Code availability

The source code of AlphaBase and AlphaPeptDeep are fully opened on GitHub: <https://github.com/MannLabs/alphabase> and <https://github.com/MannLabs/alphapeptdeep>. They are also available through PyPI with "pip install alphabase" and "pip install peptdeep". The versions used in this study of AlphaBase and AlphaPeptDeep are 0.1.2 and 0.1.2 respectively. All the pre-trained MS2,

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

RT, and CCS models can be found in

[https://github.com/MannLabs/alphapeptdeep/releases/download/pre-trained-models/pretrained\\_models.zip](https://github.com/MannLabs/alphapeptdeep/releases/download/pre-trained-models/pretrained_models.zip). These models will be automatically downloaded when using the AlphaPeptDeep package for the first time.

The versions of other software are displayed in the Reporting Summary.

### Acknowledgements

We thank Marvin Thielert for the testing of the spectral libraries. We thank Mario Oroshi and Igor Paron for help with retrieval of MS RAW data. This study was supported by The Max-Planck Society for the Advancement of Science and by the Bavarian State Ministry of Health and Care through the research project DigiMed Bayern ([www.digimed-bayern.de](http://www.digimed-bayern.de)). I.B. acknowledges funding support from her Postdoc.Mobility fellowship granted by the Swiss National Science Foundation [P400PB\_191046].

### Author contributions

W.-F.Z. developed AlphaBase, AlphaPeptDeep and models, and analyzed the data. X.-X.Z. developed the models, contributed to AlphaPeptDeep code, and analyzed the data. S.W. designed the template code structure on GitHub and developed HDF functionalities in AlphaBase. C.A. reviewed the code and contributed to the model shop functionalities. M.W. came up with the idea of HLA prediction. I.B. contributed to AlphaBase. E.V. developed functionalities in AlphaViz for mirrored MS2 plots and testing the integration with AlphaPeptDeep. M.T.S. reviewed almost all the source code of AlphaBase and AlphaPeptDeep, and provided a lot of suggestions. M.M. supervised this project. W.-F.Z., M.T.S. and M.M. wrote the manuscript. All the authors revised the manuscript.

### Ethics declarations

Competing interests

The authors declare no competing interests.

### References

1. Aebersold, R. & Mann, M. Mass-spectrometric exploration of proteome structure and function. *Nature* vol. 537 Preprint at <https://doi.org/10.1038/nature19949> (2016).
2. Meissner, F., Geddes-McAlister, J., Mann, M. & Bantscheff, M. The emerging role of mass spectrometry-based proteomics in drug discovery. *Nature Reviews Drug Discovery* (2022) doi:10.1038/s41573-022-00409-3.
3. Li, S. & Tang, H. Computational methods in mass spectrometry-based proteomics. in *Advances in Experimental Medicine and Biology* vol. 939 (2016).
4. Mann, M., Kumar, C., Zeng, W. F. & Strauss, M. T. Artificial intelligence for

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

- proteomics and biomarker discovery. *Cell Systems* vol. 12 Preprint at <https://doi.org/10.1016/j.cels.2021.06.006> (2021).
5. Wen, B. *et al.* Deep Learning in Proteomics. *Proteomics* vol. 20 Preprint at <https://doi.org/10.1002/pmic.201900335> (2020).
6. Moruz, L., Tomazela, D. & Käll, L. Training, selection, and robust calibration of retention time models for targeted proteomics. *Journal of Proteome Research* **9**, (2010).
7. Escher, C. *et al.* Using iRT, a normalized retention time for more targeted measurement of peptides. *Proteomics* **12**, (2012).
8. Pfeifer, N., Leinenbach, A., Huber, C. G. & Kohlbacher, O. Statistical learning of peptide retention behavior in chromatographic separations: A new kernel-based approach for computational proteomics. *BMC Bioinformatics* **8**, (2007).
9. Ma, C. *et al.* Improved Peptide Retention Time Prediction in Liquid Chromatography through Deep Learning. *Analytical Chemistry* **90**, (2018).
10. Zhou, X. X. *et al.* PDeep: Predicting MS/MS Spectra of Peptides with Deep Learning. *Analytical Chemistry* **89**, (2017).
11. Zeng, W. F. *et al.* MS/MS Spectrum prediction for modified peptides using pDeep2 Trained by Transfer Learning. *Analytical Chemistry* **91**, (2019).
12. Tiwary, S. *et al.* High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis. *Nature Methods* **16**, (2019).
13. Gessulat, S. *et al.* Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods* **16**, (2019).
14. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **9**, (1997).
15. Cho, K., van Merriënboer, B., Bahdanau, D. & Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. (2014).
16. Lou, R. *et al.* DeepPhospho accelerates DIA phosphoproteome profiling through in silico library generation. *Nature Communications* **12**, (2021).
17. Ekvall, M., Truong, P., Gabriel, W., Wilhelm, M. & Käll, L. Prosit Transformer: A transformer for Prediction of MS2 Spectrum Intensities. *Journal of Proteome Research* (2021) doi:10.1021/acs.jproteome.1c00870.
18. Strauss, M. T. *et al.* AlphaPept, a modern and open framework for MS-based proteomics. *bioRxiv* (2021).
19. Paszke, A. *et al.* PyTorch: An imperative style, high-performance deep learning library. in *Advances in Neural Information Processing Systems* vol. 32 (2019).
20. Dosovitskiy, A. *et al.* An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (2020).
21. Avsec, Ž. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods* **18**, (2021).
22. Tunyasuvunakool, K. *et al.* Highly accurate protein structure prediction for the human proteome. *Nature* **596**, (2021).
23. Wolf, T. *et al.* HuggingFace's Transformers: State-of-the-art Natural Language Processing. (2019).
24. Goyal, P. *et al.* Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour.



bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

- (2017).
25. Meier, F. *et al.* Deep learning the collisional cross sections of the peptide universe from a million experimental values. *Nature Communications* **12**, (2021).
  26. Wen, B., Li, K., Zhang, Y. & Zhang, B. Cancer neoantigen prioritization through sensitive and reliable proteogenomics analysis. *Nature Communications* **11**, (2020).
  27. Müller, J. B. *et al.* The proteome landscape of the kingdoms of life. *Nature* **582**, (2020).
  28. Zolg, D. P. *et al.* Building ProteomeTools based on a complete synthetic human proteome. *Nature Methods* **14**, (2017).
  29. Meier, F., Park, M. A. & Mann, M. Trapped ion mobility spectrometry and parallel accumulation–serial fragmentation in proteomics. *Molecular and Cellular Proteomics* vol. 20 Preprint at <https://doi.org/10.1016/j.mcpro.2021.100138> (2021).
  30. Chong, C., Coukos, G. & Bassani-Sternberg, M. Identification of tumor antigens with immunopeptidomics. *Nature Biotechnology* vol. 40 Preprint at <https://doi.org/10.1038/s41587-021-01038-8> (2022).
  31. Li, K., Jain, A., Malovannaya, A., Wen, B. & Zhang, B. DeepRescore: Leveraging Deep Learning to Improve Peptide Identification in Immunopeptidomics. *Proteomics* **20**, (2020).
  32. Wilhelm, M. *et al.* Deep learning boosts sensitivity of mass spectrometry-based immunopeptidomics. *Nature Communications* **12**, (2021).
  33. Rosenberger, G. *et al.* A repository of assays to quantify 10,000 human proteins by SWATH-MS. *Scientific Data* **1**, (2014).
  34. Wang, S. *et al.* NAGuideR: Performing and prioritizing missing value imputations for consistent bottom-up proteomic analyses. *Nucleic Acids Research* **48**, (2020).
  35. Chi, H. *et al.* Comprehensive identification of peptides in tandem mass spectra using an efficient open search engine. *Nature Biotechnology* **36**, (2018).
  36. Mason, E. A. & McDaniel, E. W. *Transport Properties of Ions in Gases*. *Transport Properties of Ions in Gases* (1988). doi:10.1002/3527602852.
  37. Paul Zolg, D. *et al.* Proteometools: Systematic characterization of 21 post-translational protein modifications by liquid chromatography tandem mass spectrometry (lc-ms/ms) using synthetic peptides. *Molecular and Cellular Proteomics* **17**, (2018).
  38. Voytik, E. *et al.* AlphaViz: Visualization and validation of critical proteomics data directly at the raw data level. *bioRxiv* 2022.07.12.499676 (2022) doi:10.1101/2022.07.12.499676.
  39. Bouwmeester, R., Gabriels, R., Hulstaert, N., Martens, L. & Degroove, S. DeepLC can predict retention times for peptides that carry as-yet unseen modifications. *Nature Methods* **18**, (2021).
  40. Käll, L., Canterbury, J. D., Weston, J., Noble, W. S. & MacCoss, M. J. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods* **4**, (2007).
  41. Mann, M. & Wilm, M. Error-Tolerant Identification of Peptides in Sequence Databases by Peptide Sequence Tags. *Analytical Chemistry* **66**, (1994).

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

42. Sarkizova, S. *et al.* A large peptidome dataset improves HLA class I epitope prediction across most of the human population. *Nature Biotechnology* **38**, (2020).
43. Bassani-Sternberg, M. *et al.* Direct identification of clinically relevant neoepitopes presented on native human melanoma tissue by mass spectrometry. *Nature Communications* **7**, (2016).
44. Cox, J. & Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology* **26**, (2008).
45. Sturm, T. *et al.* Mild Acid Elution and MHC Immunoaffinity Chromatography Reveal Similar Albeit Not Identical Profiles of the HLA Class I Immunoepitidome. *Journal of Proteome Research* **20**, (2021).
46. Tareen, A. & Kinney, J. B. Logomaker: Beautiful sequence logos in Python. *Bioinformatics* **36**, (2020).
47. Ludwig, C. *et al.* Data-independent acquisition-based SWATH - MS for quantitative proteomics: a tutorial . *Molecular Systems Biology* **14**, (2018).
48. Pak, H. S. *et al.* Sensitive immunoepitidomics by leveraging available large-scale multi-HLA spectral libraries, data-independent acquisition, and MS/MS prediction. *Molecular and Cellular Proteomics* **20**, (2021).
49. Ritz, D., Kinzi, J., Neri, D. & Fugmann, T. Data-Independent Acquisition of HLA Class I Peptidomes on the Q Exactive Mass Spectrometer Platform. *Proteomics* **17**, (2017).
50. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S. & Ralser, M. DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. *Nature Methods* **17**, (2020).
51. Martinez-Val, A., Bekker-Jensen, D. B., Høgrebe, A. & Olsen, J. V. Data Processing and Analysis for DIA-Based Phosphoproteomics Using Spectronaut. in *Methods in Molecular Biology* vol. 2361 (2021).
52. Tsou, C. C. *et al.* DIA-Umpire: Comprehensive computational framework for data-independent acquisition proteomics. *Nature Methods* **12**, (2015).
53. Xin, L. *et al.* A streamlined platform for analyzing tera-scale DDA and DIA mass spectrometry data enables highly sensitive immunoepitidomics. *Nature Communications* **13**, 3108 (2022).
54. Gfeller, D. *et al.* The Length Distribution and Multiple Specificity of Naturally Presented HLA-I Ligands. *The Journal of Immunology* **201**, (2018).
55. Granholm, V., Noble, W. S. & Käll, L. A cross-validation scheme for machine learning algorithms in shotgun proteomics. *BMC Bioinformatics* **13 Suppl 16**, (2012).
56. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, (2011).
57. Kong, A. T., Lèpèrevost, F. v., Avtonomov, D. M., Mellacheruvu, D. & Nesvizhskii, A. I. MSFragger: Ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nature Methods* **14**, (2017).
58. Zhou, C. *et al.* Speeding up tandem mass spectrometry-based database searching

bioRxiv preprint doi: <https://doi.org/10.1101/2022.07.14.499992>; this version posted July 16, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

by longest common prefix. *BMC Bioinformatics* **11**, (2010).



## 4. Discussion

Although the timsTOF instrument has only very recently been introduced, it has already gained much attention and has shown very promising results in proteomics, including its wide dynamic range, high sequence coverage and very high sensitivity. The release of such a novel instrument and other subsequent instrumental advances creates a strong need for the development of software allowing to efficiently handle the data. Although perhaps a natural duty for the manufacturer, experience shows that commercial suppliers seldomly and in any case slowly meet this need.

Therefore, once I started working with the timsTOF data, it became clear that the first and absolutely essential step was to replace the existing, but extremely inconvenient and time-consuming data access. This step was to influence, if successful, all subsequent projects. Having become aware of the common problems in scientific software development, where code, even if released to the public, often does not meet basic software engineering standards, we decided to build an open-source proteomics framework, called AlphaPept, allowing users to explore the complex steps of proteomics data processing in a simple way in Jupyter notebooks or to focus on their own new algorithmic ideas, replacing only parts of existing code (**Article 5**). Once built, this framework allowed MS data to be analyzed several orders of magnitude faster, while also providing an environment for developing all other software projects in our group to robust software engineering standards, such as high-quality code, extensive documentation, automated testing, and continuous integration. All these 'good software engineering practices' help to develop stable, robust, easy-to-use and extend software tools. However, due to the specific nature of scientific projects, e.g. an imprecise or ill-defined research question at the beginning of the development process, it is quite difficult to adopt these practices to this domain. At the moment there are no set guidelines for development specifically in science and generally there are only attempts by various individual groups to internally establish some good practice through trial and error. Given this background, I thought that it would be interesting to see what methods could be adopted to standardize the scientific development process to make it more efficient and generally accepted. The development of the AlphaPept ecosystem provided a great opportunity to explore these abstract principles in practice.

The AlphaPept ecosystem became a solid basis for the timsTOF data accession tool implemented in the AlphaTims project (**Article 4**). AlphaTims provided access to complex timsTOF data across any available dimension in just milliseconds, indexing sparse four-dimensional timsTOF data of billions of detector events in a matter of

seconds, thereby removing a serious bottleneck. Now it also takes only milliseconds to interactively visualize raw data such as TICs, XICs, mobilograms and mass spectra. Due to the easy usability and extensibility of the tool it has also been quite well received by the community, and I myself have been able to use it as a basis for further projects. This example illustrates how important community efforts can be in a complex scientific field such as proteomics. Clearly community involvement can benefit researchers by saving time and funding resources by simply using the tools available as a basis for further projects, especially if they already meet the aforementioned standards for software development.

With AlphaTims on hand to quickly access and visualize unprocessed timsTOF data, I decided to validate the identifications reported by popular proteomics workflows by exploring their underlying raw information. This project, named AlphaViz, arose as a result of the need in our group to be able to critically evaluate individual biologically or clinically important proteins and their (modified) peptides (**Article 2**). This had been unavailable for the timsTOF platform, or indeed for most other proteomics workflows. The AlphaViz package helps to quickly visually (in)validate the identification of peptides regardless of the score given to them by automated workflows. AlphaViz is also the first software tool to use the latest advances in the deep learning MS-property prediction to allow the visual comparison of the expected vs. the measured peptide results. We have successfully established that peptide signals present in the raw data can be unambiguously ascertained, although they were not reported by the search engine. This project has provided a tool that combines the previous developments with the deep learning prediction to greatly facilitate data visualization. It is also turned out to be a key advance to communicate the results of complex experiments not only quantitatively, but also visually.

The importance of visualization as well as its ability to help understand data integrated from different resources data is further demonstrated by the AlphaMap project (**Article 3**). AlphaMap enables visual exploration of proteomics data at the peptide level, while additionally integrating prior knowledge gathered by the community, such as UniProt annotations and proteolytic cleavage sites. AlphaMap has already been extended by adding an additional layer of information, such as the three-dimensional structures of proteins predicted by AlphaFold (123).

With these tools implemented, I have contributed to a range of specific project in the group. For instance, a novel diaPASEF scanning mode on the timsTOF instrument and its further optimization allows to acquire up to 100% of the peptide fragment ion current and achieve deep proteome coverage even in short gradients, such as over 6,000

proteins in 11 minutes gradient (**Article 6, 7**). These improvements enhance the overall throughput capabilities, enabling it to be successfully used for single-cell analysis, and achieve very high sensitivity for phosphoproteomics data. In the future, I look forward to seeing this extended towards optimizing the acquisition method in real time, as well as its application to other PTMs.

As mentioned, the field of proteomics is greatly benefiting from the latest developments in the machine learning and, in particular, deep learning. Several projects in which I have been involved complement recent efforts to predict peptides properties, in our case CCS values based on measured ion mobilities, on the basis of their sequences alone (**Article 8, 9**). This allowed us to better understand the nature of CCS values but also to apply a combination of predicted peptide properties, such as retention time, ion mobility and the fragment intensities in MS<sup>2</sup> spectra, to predict the spectral libraries needed for DIA. This also narrows down the list of possible candidates and improves scoring in such a challenging application as peptidomics.

A biological focus of my thesis is on post-translational modifications, in particular phosphorylation, and its functional exploration. With the described tools and methods at hand, I applied the timsTOF principle to an in-depth study of phosphoproteomics (**Article 2, 3, 7**). For the first time, we were able to quantitatively and accurately identify 35,000 phosphosites in just 21 minutes LC gradients, covering a substantial fraction of the regulated phosphoproteome with high sensitivity. Further detailed analysis of the well-studied EGF signaling pathway in HeLa cells revealed differential phosphorylation of proteins involved in this signaling pathway. Visualization and validation of the presence of (un)reported key signal nodes helped to enhance confidence in these results, which would be interesting for biological interpretation or follow-up experiments.

Based on my several years of working in a scientific and bioinformatics environment, I have come to the conclusion that there are several important aspects that should be further improved in scientific software development. In recent years we can observe a positive trend where more and more groups are working towards the ideals of open science. In a complex scientific field such as proteomics, this would give a great boost both to software development as well as to the advancement of the MS-based proteomics in general. An important aspect that remains is how to instill software development practices in researchers, given the nature of scientific problems and their lack of specific training. I believe that changes in the culture of research and software development are necessary and can be brought about by better integration of agile practices, which are naturally useful for exploratory, iterative and collaborative development. This should help improve the quality of code, expand project



documentation and ensure better testing, all of which should lead to better quality and reuse of tools in the field. In my view, journals should encourage or make it compulsory for the development of scientific tools to follow these principles prior to submission, just as data quality are now routinely be checked in the case of experimental data.

## 5. References

1. Wasinger, V. C., Cordwell, S. J., Poljak, A., Yan, J. X., Gooley, A. A., Wilkins, M. R., Duncan, M. W., Harris, R., Williams, K. L., and Humphery-Smith, I. (1995) Progress with gene-product mapping of the Mollicutes: *Mycoplasma genitalium*. *ELECTROPHORESIS* 16, 1090–1094
2. Griffiths, J. (2008) A brief history of mass spectrometry. *Analytical Chemistry* 80, 5678–5683
3. Thomson, J. J. (1913) Bakerian Lecture:—Rays of positive electricity. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 89, 1–20
4. Fenn, J. B., Mann, M., Meng, C. K., Wong, S. F., and Whitehouse, C. M. (1989) Electrospray ionization for mass spectrometry of large biomolecules. *Science* 246, 64–71
5. Karas, M., and Hillenkamp, F. (1988) Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. *Anal Chem* 60, 2299–2301
6. Tanaka, K., Waki, H., Ido, Y., Akita, S., Yoshida, Y., Yoshida, T., and Matsuo, T. (1988) Protein and polymer analyses up to  $m/z$  100 000 by laser ionization time-of-flight mass spectrometry. *Rapid Communications in Mass Spectrometry* 2, 151–153
7. Müller, J. B., Geyer, P. E., Colaço, A. R., Treit, P. v., Strauss, M. T., Oroshi, M., Doll, S., Virreira Winter, S., Bader, J. M., Köhler, N., Theis, F., Santos, A., and Mann, M. (2020) The proteome landscape of the kingdoms of life. *Nature* 2020 582:7813 582, 592–596
8. Wiśniewski, J. R., Zougman, A., Nagaraj, N., and Mann, M. (2009) Universal sample preparation method for proteome analysis. *Nature Methods* 2009 6:5 6, 359–362
9. Kulak, N. A., Pichler, G., Paron, I., Nagaraj, N., and Mann, M. (2014) Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nature Methods* 2014 11:3 11, 319–324
10. Bache, N., Geyer, P. E., Bekker-Jensen, D. B., Hoerning, O., Falkenby, L., Treit, P. v., Doll, S., Paron, I., Müller, J. B., Meier, F., Olsen, J. v., Vorm, O., and Mann, M. (2018) A novel LC system embeds analytes in pre-formed gradients for rapid, ultra-robust proteomics. *Molecular and Cellular Proteomics* 17, 2284–2296
11. Meier, F., Beck, S., Grassl, N., Lubeck, M., Park, M. A., Raether, O., and Mann, M. (2015) Parallel accumulation-serial fragmentation (PASEF): Multiplying sequencing speed and sensitivity by synchronized scans in a trapped ion mobility device. *Journal of Proteome Research* 14, 5378–5387
12. Meier, F., Geyer, P. E., Winter, S. V., Cox, J., and Mann, M. (2018) BoxCar acquisition method enables single-shot proteomics at a depth of 10,000 proteins in 100 minutes. *Nature Methods* 15, 440–448
13. Meier, F., Brunner, A. D., Koch, S., Koch, H., Lubeck, M., Krause, M., Goedecke, N., Decker, J., Kosinski, T., Park, M. A., Bache, N., Hoerning, O., Cox, J., Räther, O., and Mann, M. (2018) Online parallel accumulation–serial fragmentation (PASEF) with a novel trapped ion mobility mass spectrometer. *Molecular and Cellular Proteomics* 17, 2534–2545
14. Meier, F., Brunner, A.-D., Frank, M., Ha, A., Bludau, I., Voytik, E., Kaspar-Schoenefeld, S., Lubeck, M., Raether, O., Bache, N., Aebersold, R., Collins, B. C., Röst, H. L., and Mann, M. (2020) diaPASEF: parallel accumulation-serial fragmentation combined with data-independent acquisition. *Nat Methods* 17, 1229–1236

15. Messner, C. B., Demichev, V., Bloomfield, N., Yu, J. S. L., White, M., Kreidl, M., Egger, A. S., Freiwald, A., Ivosev, G., Wasim, F., Zelezniak, A., Jürgens, L., Suttorp, N., Sander, L. E., Kurth, F., Lilley, K. S., Mülleder, M., Tate, S., and Ralser, M. (2021) Ultra-fast proteomics with Scanning SWATH. *Nature Biotechnology* 2021 39:7 39, 846–854
16. Beck, S., Michalski, A., Raether, O., Lubeck, M., Kaspar, S., Goedecke, N., Baessmann, C., Hornburg, D., Meier, F., Paron, I., Kulak, N. A., Cox, J., and Mann, M. (2015) The impact II, a very high-resolution quadrupole time-of-flight instrument (QTOF) for deep shotgun proteomics. *Molecular and Cellular Proteomics* 14, 2014–2029
17. Hebert, A. S., Prasad, S., Belford, M. W., Bailey, D. J., McAlister, G. C., Abbatiello, S. E., Huguet, R., Wouters, E. R., Dunyach, J. J., Brademan, D. R., Westphall, M. S., and Coon, J. J. (2018) Comprehensive Single-Shot Proteomics with FAIMS on a Hybrid Orbitrap Mass Spectrometer. *Anal Chem* 90, 9529–9537
18. MacLean, B., Tomazela, D. M., Shulman, N., Chambers, M., Finney, G. L., Frewen, B., Kern, R., Tabb, D. L., Liebler, D. C., and MacCoss, M. J. (2010) Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics* 26, 966–968
19. Bruderer, R., Bernhardt, O. M., Gandhi, T., Miladinović, S. M., Cheng, L.-Y., Messner, S., Ehrenberger, T., Zanotelli, V., Butscheid, Y., Escher, C., Vitek, O., Rinner, O., and Reiter, L. (2015) Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues. *Molecular & Cellular Proteomics : MCP* 14, 1400
20. Kong, A. T., Leprevost, F. v., Avtonomov, D. M., Mellacheruvu, D., and Nesvizhskii, A. I. (2017) MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nat Methods* 14, 513–520
21. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S., and Ralser, M. (2020) DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. *Nature Methods* 17, 41–44
22. Cox, J., and Mann, M. (2008) MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology* 2008 26:12 26, 1367–1372
23. Tran, J. C., Zamdborg, L., Ahlf, D. R., Lee, J. E., Catherman, A. D., Durbin, K. R., Tipton, J. D., Vellaichamy, A., Kellie, J. F., Li, M., Wu, C., Sweet, S. M. M., Early, B. P., Siuti, N., Leduc, R. D., Compton, P. D., Thomas, P. M., and Kelleher, N. L. (2011) Mapping intact protein isoforms in discovery mode using top-down proteomics. *Nature* 2011 480:7376 480, 254–258
24. Schmidt, A., Claassen, M., and Aebersold, R. (2009) Directed mass spectrometry: towards hypothesis-driven proteomics. *Current Opinion in Chemical Biology* 13, 510–517
25. Gillet, L. C., Navarro, P., Tate, S., Röst, H., Selevsek, N., Reiter, L., Bonner, R., and Aebersold, R. (2012) Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis. *Mol Cell Proteomics* 11,
26. Nesvizhskii, A. I., and Aebersold, R. (2005) Interpretation of Shotgun Proteomic Data. *Molecular & Cellular Proteomics* 4, 1419–1440
27. Hein, M. Y., Sharma, K., Cox, J., and Mann, M. (2013) Proteomic Analysis of Cellular Systems. *Handbook of Systems Biology*, 3–25
28. Goldberg, S. (2008) Mechanical/physical methods of cell disruption and tissue homogenization. *Methods Mol Biol* 424, 3–22



29. Han, J. C., and Han, G. Y. (1994) A Procedure for Quantitative Determination of Tris(2-Carboxyethyl)phosphine, an Odorless Reducing Agent More Stable and Effective Than Dithiothreitol. *Analytical Biochemistry* 220, 5–10
30. Smejkal, G. B., Li, C., Robinson, M. H., Lazarev, A. v., Lawrence, N. P., and Chernokalskaya, E. (2006) Simultaneous reduction and alkylation of protein disulfides in a centrifugal ultrafiltration device prior to two-dimensional gel electrophoresis. *Journal of Proteome Research* 5, 983–987
31. Glatter, T., Ludwig, C., Ahrné, E., Aebersold, R., Heck, A. J. R., and Schmidt, A. (2012) Large-scale quantitative assessment of different in-solution protein digestion protocols reveals superior cleavage efficiency of tandem Lys-C/trypsin proteolysis over trypsin digestion. *Journal of Proteome Research* 11, 5145–5156
32. Tsiatsiani, L., Heck, A. J. R., Heck, A. J. R., and Bijvoet, P. (2015) Proteomics beyond trypsin. *The FEBS Journal* 282, 2612–2626
33. Rappsilber, J., Mann, M., and Ishihama, Y. (2007) Protocol for micro-purification, enrichment, pre-fractionation and storage of peptides for proteomics using StageTips. *Nature Protocols* 2:8 2, 1896–1906
34. Wiśniewski, J. R., Zougman, A., and Mann, M. (2009) Combination of FASP and StageTip-based fractionation allows in-depth analysis of the hippocampal membrane proteome. *Journal of Proteome Research* 8, 5674–5678
35. Kulak, N. A., Pichler, G., Paron, I., Nagaraj, N., and Mann, M. (2014) Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nature Methods* 2014 11:3 11, 319–324
36. Geyer, P. E., Kulak, N. A., Pichler, G., Holdt, L. M., Teupser, D., and Mann, M. (2016) Plasma Proteome Profiling to Assess Human Health and Disease. *Cell Systems* 2, 185–195
37. Doll, S., and Burlingame, A. L. (2015) Mass spectrometry-based detection and assignment of protein posttranslational modifications. *ACS Chem Biol* 10, 63–71
38. Kulak, N. A., Geyer, P. E., and Mann, M. (2017) Loss-less Nano-fractionator for High Sensitivity, High Coverage Proteomics. *Molecular & Cellular Proteomics : MCP* 16, 694
39. Michalski, A., Cox, J., and Mann, M. (2011) More than 100,000 detectable peptide species elute in single shotgun proteomics runs but the majority is inaccessible to data-dependent LC-MS/MS. *Journal of Proteome Research* 10, 1785–1793
40. Zhang, Y., Fonslow, B. R., Shan, B., Baek, M. C., and Yates, J. R. (2013) Protein analysis by shotgun/bottom-up proteomics. *Chemical Reviews* 113, 2343–2394
41. Wilm, M., and Mann, M. (1996) Analytical properties of the nanoelectrospray ion source. *Analytical Chemistry* 68, 1–8
42. Yinon, J. (1990) Tandem Mass Spectrometry (MS/MS) and Collision Induced Dissociation (CID) — an Introduction. *Chemistry and Physics of Energetic Materials*, 685–693
43. Olsen, J. v., Macek, B., Lange, O., Makarov, A., Horning, S., and Mann, M. (2007) Higher-energy C-trap dissociation for peptide modification analysis. *Nat Methods* 4, 709–712
44. Syka, J. E. P., Coon, J. J., Schroeder, M. J., Shabanowitz, J., and Hunt, D. F. (2004) Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry. *Proc Natl Acad Sci U S A* 101, 9528–9533
45. Steen, H., and Mann, M. (2004) The abc's (and xyz's) of peptide sequencing. *Nature Reviews Molecular Cell Biology* 2004 5:9 5, 699–711

46. Wiesner, J., Premisler, T., and Sickmann, A. (2008) Application of electron transfer dissociation (ETD) for the analysis of posttranslational modifications. *Proteomics* 8, 4466–4483
47. Senko, M. W., Beu, S. C., and McLafferty, F. W. (1995) Determination of monoisotopic masses and ion populations for large biomolecules from resolved isotopic distributions. *J Am Soc Mass Spectrom* 6, 229–233
48. Cox, J., and Mann, M. (2009) Computational Principles of Determining and Improving Mass Precision and Accuracy for Proteome Measurements in an Orbitrap. *J Am Soc Mass Spectrom* 20, 1477–1485
49. Cox, J., Michalski, A., and Mann, M. (2011) Software lock mass by two-dimensional minimization of peptide mass errors. *J Am Soc Mass Spectrom* 22, 1373–1380
50. Johnson, R. S., and Taylor, J. A. (2002) Searching sequence databases via de novo peptide sequencing by tandem mass spectrometry. *Mol Biotechnol* 22, 301–315
51. Muth, T., and Renard, B. Y. (2018) Evaluating de novo sequencing in proteomics: already an accurate alternative to database-driven peptide identification? *Briefings in Bioinformatics* 19, 954–970
52. Tran, N. H., Zhang, X., Xin, L., Shan, B., and Li, M. (2017) De novo peptide sequencing by deep learning. *Proc Natl Acad Sci U S A* 114, 8247–8252
53. Tran, N. H., Qiao, R., Xin, L., Chen, X., Liu, C., Zhang, X., Shan, B., Ghodsi, A., and Li, M. (2018) Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry. *Nature Methods* 2018 16:1 16, 63–66
54. Nesvizhskii, A. I. (2010) A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics* 73, 2092–2123
55. Cox, J., and Mann, M. (2008) MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat Biotechnol* 26, 1367–1372
56. Prieto, G., and Vázquez, J. (2020) Calculation of false discovery rate for peptide and protein identification. *Methods in Molecular Biology* 2051, 145–159
57. Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., and Kuster, B. (2007) Quantitative mass spectrometry in proteomics: a critical review. *Analytical and Bioanalytical Chemistry* 2007 389:4 389, 1017–1031
58. Pappireddi, N., Martin, L., and Wühr, M. (2019) A Review on Quantitative Multiplexed Proteomics. *Chembiochem* 20, 1210
59. Thomson, J., London, E. R.-T., Edinburgh, undefined, Dublin, and, and 1896, undefined (2009) XL. On the passage of electricity through gases exposed to Röntgen rays. *Taylor & Francis* 42, 392–407
60. Eiceman, G. A., and Karpas, Z. (2005) Ion Mobility Spectrometry. *Ion Mobility Spectrometry*,
61. Barnes, W. S., Martin, D. W., and McDaniel, E. W. (1961) Mass Spectrographic Identification of the Ion Observed in Hydrogen Mobility Experiments. *Physical Review Letters* 6, 110
62. McAfee, K. B., and Edelson, D. (1963) Identification and Mobility of Ions in a Townsend Discharge by Time-resolved Mass Spectrometry. *Proceedings of the Physical Society* 81, 382
63. Bloomfield, C., Society, J. H.-D. of the F., and 1964, undefined New technique for the study of ion-atom interchange. *pubs.rsc.org*,

64. McLean, J. A., Ruotolo, B. T., Gillig, K. J., and Russell, D. H. (2005) Ion mobility–mass spectrometry: a new paradigm for proteomics. *International Journal of Mass Spectrometry* 240, 301–315
65. Ruotolo, B. T., Benesch, J. L. P., Sandercock, A. M., Hyung, S. J., and Robinson, C. v. (2008) Ion mobility–mass spectrometry analysis of large protein complexes. *Nature Protocols* 2008 3:7 3, 1139–1152
66. Helm, D., Vissers, J. P. C., Hughes, C. J., Hahne, H., Ruprecht, B., Pachl, F., Grzyb, A., Richardson, K., Wildgoose, J., Maier, S. K., Marx, H., Wilhelm, M., Becher, I., Lemeer, S., Bantscheff, M., Langridge, J. I., and Kuster, B. (2014) Ion mobility tandem mass spectrometry enhances performance of bottom-up proteomics. *Molecular and Cellular Proteomics* 13, 3709–3715
67. May, J. C., and McLean, J. A. (2015) Ion mobility-mass spectrometry: Time-dispersive instrumentation. *Analytical Chemistry* 87, 1422–1436
68. Fernandez-Lima, F. A., Kaplan, D. A., and Park, M. A. (2011) Note: Integration of trapped ion mobility spectrometry with mass spectrometry. *Review of Scientific Instruments* 82, 126106
69. Fernandez-Lima, F., Kaplan, D. A., Suetering, J., and Park, M. A. (2011) Gas-phase separation using a trapped ion mobility spectrometer. *International Journal for Ion Mobility Spectrometry* 14, 93–98
70. Silveira, J. A., Ridgeway, M. E., Laukien, F. H., Mann, M., and Park, M. A. (2017) Parallel accumulation for 100% duty cycle trapped ion mobility-mass spectrometry. *International Journal of Mass Spectrometry* 413, 168–175
71. Beck, S., Michalski, A., Raether, O., Lubeck, M., Kaspar, S., Goedecke, N., Baessmann, C., Hornburg, D., Meier, F., Paron, I., Kulak, N. A., Cox, J., and Mann, M. (2015) The Impact II, a Very High-Resolution Quadrupole Time-of-Flight Instrument (QTOF) for Deep Shotgun Proteomics \*. *Molecular & Cellular Proteomics* 14, 2014–2029
72. Brunner, A.-D., Thielert, M., Vasilopoulou, C., Ammar, C., Coscia, F., Mund, A., Hoerning, O. B., Bache, N., Apalategui, A., Lubeck, M., Richter, S., Fischer, D. S., Raether, O., Park, M. A., Meier, F., Theis, F. J., and Mann, M. (2022) Ultra-high sensitivity mass spectrometry quantifies single-cell proteome changes upon perturbation. *Molecular Systems Biology* 18, e10798
73. Kicman, A. T., Parkin, M. C., and Iles, R. K. (2007) An introduction to mass spectrometry based proteomics-Detection and characterization of gonadotropins and related molecules. *Molecular and Cellular Endocrinology* 260–262, 212–227
74. Sinha, A., and Mann, M. (2020) A beginner's guide to mass spectrometry–based proteomics. *The Biochemist* 42, 64–69
75. Cho, N. H., Cheveralls, K. C., Brunner, A. D., Kim, K., Michaelis, A. C., Raghavan, P., Kobayashi, H., Savy, L., Li, J. Y., Canaj, H., Kim, J. Y. S., Stewart, E. M., Gnann, C., McCarthy, F., Cabrera, J. P., Brunetti, R. M., Chhun, B. B., Dingle, G., Hein, M. Y., Huang, B., Mehta, S. B., Weissman, J. S., Gómez-Sjöberg, R., Itzhak, D. N., Royer, L. A., Mann, M., and Leonetti, M. D. (2022) OpenCell: Endogenous tagging for the cartography of human cellular organization. *Science* (1979) 375,
76. Vasilopoulou, C. G., Sulek, K., Brunner, A. D., Meitei, N. S., Schweiger-Hufnagel, U., Meyer, S. W., Barsch, A., Mann, M., and Meier, F. (2020) Trapped ion mobility spectrometry and PASEF enable in-depth lipidomics from minimal sample amounts. *Nature Communications* 2020 11:1 11, 1–11
77. Nothias, L. F., Petras, D., Schmid, R., Dührkop, K., Rainer, J., Sarvepalli, A., Protisyuk, I., Ernst, M., Tsugawa, H., Fleischauer, M., Aicheler, F., Aksenov, A. A., Alka, O., Allard, P.



- M., Barsch, A., Cachet, X., Caraballo-Rodriguez, A. M., da Silva, R. R., Dang, T., Garg, N., Gauglitz, J. M., Gurevich, A., Isaac, G., Jarmusch, A. K., Kameník, Z., Kang, K. bin, Kessler, N., Koester, I., Korf, A., le Gouellec, A., Ludwig, M., Martin H, C., McCall, L. I., McSayles, J., Meyer, S. W., Mohimani, H., Morsy, M., Moyne, O., Neumann, S., Neuweiger, H., Nguyen, N. H., Nothias-Esposito, M., Paolini, J., Phelan, V. v., Pluskal, T., Quinn, R. A., Rogers, S., Shrestha, B., Tripathi, A., van der Hooft, J. J. J., Vargas, F., Weldon, K. C., Witting, M., Yang, H., Zhang, Z., Zubeil, F., Kohlbacher, O., Böcker, S., Alexandrov, T., Bandeira, N., Wang, M., and Dorrestein, P. C. (2020) Feature-based molecular networking in the GNPS analysis environment. *Nature Methods* 2020 17:9 17, 905–908
78. Meier, F., Park, M. A., and Mann, M. (2021) Trapped ion mobility spectrometry and parallel accumulation–serial fragmentation in proteomics. *Molecular and Cellular Proteomics* 20, 100138
79. Michalski, A., Cox, J., and Mann, M. (2011) More than 100,000 detectable peptide species elute in single shotgun proteomics runs but the majority is inaccessible to data-dependent LC-MS/MS. *Journal of Proteome Research* 10, 1785–1793
80. Gillet, L. C., Leitner, A., and Aebersold, R. (2016) Mass Spectrometry Applied to Bottom-Up Proteomics: Entering the High-Throughput Era for Hypothesis Testing. <http://dx.doi.org/10.1146/annurev-anchem-071015-041535> 9, 449–472
81. Bourmaud, A., Gallien, S., and Domon, B. (2016) Parallel reaction monitoring using quadrupole-Orbitrap mass spectrometer: Principle and applications. *PROTEOMICS* 16, 2146–2159
82. Wichmann, C., Meier, F., Winter, S. V., Brunner, A. D., Cox, J., and Mann, M. (2019) MaxQuant.Live Enables Global Targeting of More Than 25,000 Peptides. *Molecular & Cellular Proteomics : MCP* 18, 982
83. Lesur, A., Schmit, P. O., Bernardin, F., Letellier, E., Brehmer, S., Decker, J., and Dittmar, G. (2021) Highly multiplexed targeted proteomics acquisition on a TIMS-QTOF. *Analytical Chemistry* 93, 1383–1392
84. Ribet, D., and Cossart, P. (2010) Post-translational modifications in host cells during bacterial infection. *FEBS Letters* 584, 2748–2758
85. Olsen, J. v., Blagoev, B., Gnad, F., Macek, B., Kumar, C., Mortensen, P., and Mann, M. (2006) Global, In Vivo, and Site-Specific Phosphorylation Dynamics in Signaling Networks. *Cell* 127, 635–648
86. Ozlu, N., Akten, B., Timm, W., Haseley, N., Steen, H., and Steen, J. A. J. (2010) Phosphoproteomics. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 2, 255–276
87. Hornbeck, P. v., Kornhauser, J. M., Tkachev, S., Zhang, B., Skrzypek, E., Murray, B., Latham, V., and Sullivan, M. (2012) PhosphoSitePlus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. *Nucleic Acids Research* 40, D261–D270
88. Ubersax, J. A., and Ferrell, J. E. (2007) Mechanisms of specificity in protein phosphorylation. *Nature Reviews Molecular Cell Biology* 2007 8:7 8, 530–541
89. Savage, S. R., and Zhang, B. (2020) Using phosphoproteomics data to understand cellular signaling: A comprehensive guide to bioinformatics resources. *Clinical Proteomics* 17, 1–18
90. Ardito, F., Giuliani, M., Perrone, D., Troiano, G., and Muzio, L. lo (2017) The crucial role of protein phosphorylation in cell signaling and its use as targeted therapy (Review). *International Journal of Molecular Medicine* 40, 271–280

91. Wee, P., and Wang, Z. (2017) Epidermal Growth Factor Receptor Cell Proliferation Signaling Pathways. *Cancers (Basel)* 9,
92. Chakraborty, S., Li, L., Puliappadamba, V. T., Guo, G., Hatanpaa, K. J., Mickey, B., Souza, R. F., Vo, P., Herz, J., Chen, M. R., Boothman, D. A., Pandita, T. K., Wang, D. H., Sen, G. C., and Habib, A. A. (2014) Constitutive and ligand-induced EGFR signalling triggers distinct and mutually exclusive downstream signalling networks. *Nature Communications* 2014 5:1 5, 1–15
93. Ochoa, D., Jarnuczak, A. F., Viéitez, C., Gehre, M., Soucheray, M., Mateus, A., Kleefeldt, A. A., Hill, A., Garcia-Alonso, L., Stein, F., Krogan, N. J., Savitski, M. M., Swaney, D. L., Vizcaíno, J. A., Noh, K. M., and Beltrao, P. (2019) The functional landscape of the human phosphoproteome. *Nature Biotechnology* 2019 38:3 38, 365–373
94. Solari, F. A., Dell'Aica, M., Sickmann, A., and Zahedi, R. P. (2015) Why phosphoproteomics is still a challenge. *Molecular BioSystems* 11, 1487–1493
95. Grimsrud, P. A., Swaney, D. L., Wenger, C. D., Beauchene, N. A., and Coon, J. J. (2010) Phosphoproteomics for the masses. *ACS Chem Biol* 5, 105
96. Potel, C. M., Lemeer, S., and Heck, A. J. R. (2019) Phosphopeptide Fragmentation and Site Localization by Mass Spectrometry: An Update. *Analytical Chemistry* 91, 126
97. Chalkley, R. J., and Clauser, K. R. (2012) Modification Site Localization Scoring: Strategies and Performance. *Molecular & Cellular Proteomics : MCP* 11, 3
98. Bekker-Jensen, D. B., Bernhardt, O. M., Högberg, A., Martínez-Val, A., Verbeke, L., Gandhi, T., Kelstrup, C. D., Reiter, L., and Olsen, J. v. (2020) Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. *Nature Communications* 2020 11:1 11, 1–12
99. Oliinyk, D., and Meier, F. (2022) Ion mobility-resolved phosphoproteomics with dia-PASEF and short gradients. *bioRxiv*, 2022.06.02.494482
100. Neumann, G. C., Antillanca Espina, H., Ctor, V., and Daza, P. (2017) Development of Scientific Software and Practices for Software Development: A Systematic Literature Review. *Journal of Software* 12, 114–124
101. Arvanitou, E. M., Ampatzoglou, A., Chatzigeorgiou, A., and Carver, J. C. (2021) Software engineering practices for scientific software development: A systematic mapping study. *Journal of Systems and Software* 172, 110848
102. Crüwell, S., van Doorn, J., Etz, A., Makel, M. C., Moshontz, H., Niebaum, J. C., Orben, A., Parsons, S., and Schulte-Mecklenbeck, M. (2019) Seven Easy Steps to Open Science. <https://doi.org/10.1027/2151-2604/a000387> 227, 237–248
103. Baker, M., and Penny, D. (2016) 1,500 scientists lift the lid on reproducibility. *Nature* 533, 452–454
104. Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016) Jupyter Notebooks – a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*, 87–90
105. Knuth, D. E. (1984) Literate Programming. *The Computer Journal* 27, 97–111
106. Bittremieux, W., Adams, C., Laukens, K., Dorrestein, P. C., and Bandeira, N. (2021) Open Science Resources for the Mass Spectrometry-Based Analysis of SARS-CoV-2. *Journal of Proteome Research* 20, 1464–1475

107. Santos, A., Colaço, A. R., Nielsen, A. B., Niu, L., Strauss, M., Geyer, P. E., Coscia, F., Albrechtsen, N. J. W., Mundt, F., Jensen, L. J., and Mann, M. (2022) A knowledge graph to interpret clinical proteomics data. *Nature Biotechnology* 2022 40:5 40, 692–702
108. Bisong, E. (2019) Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 59–64
109. Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., Fufezan, C., Ternent, T., Eglen, S. J., Katz, D. S., Pollard, T. J., Konovalov, A., Flight, R. M., Blin, K., and Vizcaíno, J. A. (2016) Ten Simple Rules for Taking Advantage of Git and GitHub. *PLOS Computational Biology* 12, e1004947
110. Gou, Y., Graff, F., Kilian, O., Kafkas, S., Katuri, J., Kim, J. H., Marinos, N., McEntyre, J., Morrison, A., Pi, X., Rossiter, P., Talo, F., Vartak, V., Coleman, L. A., Hawkins, C., Kinsey, A., Mansoor, S., Morris, V., Rowbotham, R., Chaplin, D., MacIntyre, R., Patel, Y., Ananiadou, S., Black, W. J., McNaught, J., Rak, R., and Rowley, A. (2015) Europe PMC: a full-text literature database for the life sciences and platform for innovation. *Nucleic Acids Res* 43, D1042–D1048
111. Oveland, E., Muth, T., Rapp, E., Martens, L., Berven, F. S., and Barsnes, H. (2015) Viewing the proteome: How to visualize proteomics data? *PROTEOMICS* 15, 1341–1355
112. Tyanova, S., Temu, T., Carlson, A., Sinitcyn, P., Mann, M., and Cox, J. (2015) Visualization of LC-MS/MS proteomics data in MaxQuant. *Proteomics* 15, 1453–1456
113. Petras, D., Phelan, V. v., Acharya, D., Allen, A. E., Aron, A. T., Bandeira, N., Bowen, B. P., Belle-Oudry, D., Boecker, S., Cummings, D. A., Deutsch, J. M., Fahy, E., Garg, N., Gregor, R., Handelsman, J., Navarro-Hoyos, M., Jarmusch, A. K., Jarmusch, S. A., Louie, K., Maloney, K. N., Marty, M. T., Meijler, M. M., Mizrahi, I., Neve, R. L., Northen, T. R., Molina-Santiago, C., Panitchpakdi, M., Pullman, B., Puri, A. W., Schmid, R., Subramaniam, S., Thukral, M., Vasquez-Castro, F., Dorrestein, P. C., and Wang, M. (2021) GNPS Dashboard: collaborative exploration of mass spectrometry data in the web browser. *Nature Methods* 2021 19:2 19, 134–136
114. Perez-Riverol, Y., Xu, Q.-W., Wang, R., Uszkoreit, J., Griss, J., Sanchez, A., Reisinger, F., Csordas, A., Ternent, T., Del-Toro, N., Dianes, J. A., Eisenacher, M., Hermjakob, H., and Vizcaíno, J. A. (2016) PRIDE Inspector Toolsuite: Moving Toward a Universal Visualization Tool for Proteomics Data Standard Formats and Quality Assessment of ProteomeXchange Datasets. *Mol Cell Proteomics* 15, 305–317
115. Perez-Riverol, Y., Wang, R., Hermjakob, H., Müller, M., Vesada, V., and Vizcaíno, J. A. (2014) Open source libraries and frameworks for mass spectrometry based proteomics: A developer's perspective. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* 1844, 63–76
116. Gatto, L., Breckels, L. M., Naake, T., and Gibb, S. (2015) Visualization of proteomics data using R and bioconductor. *Proteomics* 15, 1375–1389
117. Martin-Jaular, L., Nevo, N., Schessner, J. P., Tkach, M., Jouve, M., Dingli, F., Loew, D., Witwer, K. W., Ostrowski, M., Borner, G. H. H., and Théry, C. (2021) Unbiased proteomic profiling of host cell extracellular vesicle composition and dynamics upon HIV-1 infection. *EMBO J* 40,
118. Hansen, F. M., Tanzer, M. C., Brüning, F., Bludau, I., Stafford, C., Schulman, B. A., Robles, M. S., Karayel, O., and Mann, M. (2021) Data-independent acquisition method for ubiquitinome analysis reveals regulation of circadian biology. *Nature Communications* 2021 12:1 12, 1–13
119. Bittremieux, W., Valkenburg, D., Martens, L., and Laukens, K. (2017) Computational quality control tools for mass spectrometry proteomics. *PROTEOMICS* 17, 1600159



120. Noga, M., Sucharski, F., Suder, P., and Silberring, J. (2007) A practical guide to nano-LC troubleshooting. *J Sep Sci* 30, 2179–2189
121. Bateman, A., Martin, M. J., O'Donovan, C., Magrane, M., Alpi, E., Antunes, R., Bely, B., Bingley, M., Bonilla, C., Britto, R., Bursteinas, B., Bye-Ajee, H., Cowley, A., da Silva, A., de Giorgi, M., Dogan, T., Fazzini, F., Castro, L. G., Figueira, L., Garmiri, P., Georghiou, G., Gonzalez, D., Hatton-Ellis, E., Li, W., Liu, W., Lopez, R., Luo, J., Lussi, Y., MacDougall, A., Nightingale, A., Palka, B., Pichler, K., Poggioli, D., Pundir, S., Pureza, L., Qi, G., Rosanoff, S., Saidi, R., Sawford, T., Shypitsyna, A., Speretta, E., Turner, E., Tyagi, N., Volynkin, V., Wardell, T., Warner, K., Watkins, X., Zaru, R., Zellner, H., Xenarios, I., Bougueleret, L., Bridge, A., Poux, S., Redaschi, N., Aimo, L., ArgoudPuy, G., Auchincloss, A., Axelsen, K., Bansal, P., Baratin, D., Blatter, M. C., Boeckmann, B., Bolleman, J., Boutet, E., Breuza, L., Casal-Casas, C., de Castro, E., Coudert, E., Cuche, B., Doche, M., Dornevil, D., Duvaud, S., Estreicher, A., Famiglietti, L., Feuermann, M., Gasteiger, E., Gehant, S., Gerritsen, V., Gos, A., Gruaz-Gumowski, N., Hinz, U., Hulo, C., Jungo, F., Keller, G., Lara, V., Lemercier, P., Lieberherr, D., Lombardot, T., Martin, X., Masson, P., Morgat, A., Neto, T., Noupikell, N., Paesano, S., Pedruzzi, I., Pilbout, S., Pozzato, M., Pruess, M., Rivoire, C., Roechert, B., Schneider, M., Sigrist, C., Sonesson, K., Staehli, S., Stutz, A., Sundaram, S., Tognolli, M., Verbregue, L., Veuthey, A. L., Wu, C. H., Arighi, C. N., Arminski, L., Chen, C., Chen, Y., Garavelli, J. S., Huang, H., Laiho, K., McGarvey, P., Natale, D. A., Ross, K., Vinayaka, C. R., Wang, Q., Wang, Y., Yeh, L. S., and Zhang, J. (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Research* 45, D158–D169
122. Hornbeck, P. v., Kornhauser, J. M., Tkachev, S., Zhang, B., Skrzypek, E., Murray, B., Latham, V., and Sullivan, M. (2012) PhosphoSitePlus: A comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. *Nucleic Acids Research* 40,
123. Bludau, I., Willems, S., Zeng, W. F., Strauss, M. T., Hansen, F. M., Tanzer, M. C., Karayel, O., Schulman, B. A., and Mann, M. (2022) The structural context of posttranslational modifications at a proteome-wide scale. *PLOS Biology* 20, e3001636
124. Wilding-Mcbride Id, D., Dagley Id, L. F., Spall, S. K., Id, G. I., and Webb Id, A. I. (2022) Simplifying MS1 and MS2 spectra to achieve lower mass error, more dynamic range, and higher peptide identification confidence on the Bruker timsTOF Pro. *PLOS ONE* 17, e0271025
125. Meier, F., Beck, S., Grassl, N., Lubeck, M., Park, M. A., Raether, O., and Mann, M. (2015) Parallel accumulation-serial fragmentation (PASEF): Multiplying sequencing speed and sensitivity by synchronized scans in a trapped ion mobility device. *Journal of Proteome Research* 14, 5378–5387

## Appendix

1) *Article 10*: Plasma Proteome Profiling to detect and avoid sample-related biases in biomarker studies



# Plasma Proteome Profiling to detect and avoid sample-related biases in biomarker studies

Philipp E Geyer<sup>1,2</sup> , Eugenia Voytik<sup>1</sup>, Peter V Treit<sup>1</sup>, Sophia Doll<sup>1,2</sup>, Alisa Kleinhempel<sup>3</sup>, Lili Niu<sup>2</sup>, Johannes B Müller<sup>1</sup>, Marie-Luise Buchholtz<sup>3</sup>, Jakob M Bader<sup>1</sup>, Daniel Teupser<sup>3</sup>, Lesca M Holdt<sup>3</sup> & Matthias Mann<sup>1,2,\*</sup>

## Abstract

Plasma and serum are rich sources of information regarding an individual's health state, and protein tests inform medical decision making. Despite major investments, few new biomarkers have reached the clinic. Mass spectrometry (MS)-based proteomics now allows highly specific and quantitative readout of the plasma proteome. Here, we employ Plasma Proteome Profiling to define quality marker panels to assess plasma samples and the likelihood that suggested biomarkers are instead artifacts related to sample handling and processing. We acquire deep reference proteomes of erythrocytes, platelets, plasma, and whole blood of 20 individuals (> 6,000 proteins), and compare serum and plasma proteomes. Based on spike-in experiments, we determine sample quality-associated proteins, many of which have been reported as biomarker candidates as revealed by a comprehensive literature survey. We provide sample preparation guidelines and an online resource ([www.plasmaproteomeprofiling.org](http://www.plasmaproteomeprofiling.org)) to assess overall sample-related bias in clinical studies and to prevent costly miss-assignment of biomarker candidates.

**Keywords** biomarker discovery; mass spectrometry; plasma proteomics; sample quality; study design

**Subject Categories** Biomarkers; Proteomics

**DOI** 10.15252/emmm.201910427 | Received 4 February 2019 | Revised 26 August 2019 | Accepted 3 September 2019 | Published online 30 September 2019  
**EMBO Mol Med (2019) 11: e10427**

## Introduction

Protein levels determined in blood-based laboratory tests can be useful proxies of diseases. These biomarkers assess normal physiological status, pathogenic processes, or a response to an exposure or intervention (FDA-NIH-Biomarker-Working-Group, 2016). Proteins and enzymes constitute the largest proportion of laboratory tests, reflecting the importance of the plasma proteome in clinical diagnostics (Geyer *et al.*, 2017). Typical protein biomarkers such as the

enzymes aspartate aminotransferase (ASAT) and alanine aminotransferase (ALAT) for the diagnosis of liver diseases or cardiac troponins indicating myocardial necrosis are used routinely in clinical decision making. Enzymatic activity or antibody-based laboratory tests are performed in high-throughput and at relatively low costs, as the standard of health care. However, specific biomarkers are only available for a very limited number of conditions and most have been introduced decades ago (Anderson *et al.*, 2013). There is thus a critical need to make the biomarker discovery process more efficient.

Protein-binder assays quantifying many plasma proteins in parallel have become available (Gold *et al.*, 2010; Assarsson *et al.*, 2014), resulting in large-scale biomarker mining efforts (Ganz *et al.*, 2016; Herder *et al.*, 2018; Sun *et al.*, 2018). Orthogonal to those technologies, mass spectrometry (MS)-based proteomics has become increasingly powerful in all domains of protein research (Aebersold & Mann, 2003, 2016; Munoz & Heck, 2014). MS measures the mass and fragmentation spectra of tryptic peptides derived from the sample with very high accuracy. Because these peptide and fragment masses are unique, MS-based proteomics is inherently specific, which can be an advantage over enzyme tests and immunoassays (Wild, 2013). Within its limit of detection, MS-based proteomics can analyze all proteins in a system and is unbiased and hypothesis-free in this sense.

The proteomic community has developed guidelines for the development, specificity, and potential clinical application of biomarkers. These discuss quality standards and emphasize the importance of selecting cohorts that are appropriate in size, thus ensuring the statistical significance of potential findings (Mischak *et al.*, 2010; Surinova *et al.*, 2011; Skates *et al.*, 2013; Hoofnagle *et al.*, 2016; Geyer *et al.*, 2017). That being said, there are no systematic procedures in place to assess the proteome-wide effects of pre-analytical handling of blood-based samples. Considering that plasma samples are often collected during daily clinical routine and variably processed, sample collection and processing clearly have the potential to negatively influence clinical studies, making it difficult to uncover true biomarkers, while potentially contributing incorrect ones. Especially in case-control studies, any difference in the

<sup>1</sup> Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

<sup>2</sup> NNF Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark

<sup>3</sup> Institute of Laboratory Medicine, University Hospital, LMU Munich, Munich, Germany

\*Corresponding author. Tel: +49 89 8578 2557; E-mail: [mmanin@biochem.mpg.de](mailto:mmanin@biochem.mpg.de)



collection and processing of samples may result in systematic bias. So far, relatively little attention has been paid to this crucial aspect on a proteome-wide scale and these studies mainly investigate pre-analytical effects (Rai *et al*, 2005; Timms *et al*, 2007; Schroh *et al*, 2008; Qundos *et al*, 2013; Hassis *et al*, 2015).

Recently, we developed "Plasma Proteome Profiling", an automated MS-based pipeline for high-throughput screening of plasma samples (Geyer *et al*, 2016a). In this article, we apply this technology to systematically assess the quality of individual samples and clinical studies with the aim to identify generally applicable quality marker panels. Blood collection and subsequent errors in preparation are likely sources of plasma contamination. To address this issue, we construct proteomic catalogs of contaminating cell types as well as proteomic changes that may be induced during processing. This results in three panels of contaminating proteins, recommendations for assessing the quality of plasma samples and for consistent sample processing. We develop an online tool for biomarker studies and test the applicability of the panels on a recent investigation on the effects of weight loss on the plasma proteome (Geyer *et al*, 2016b). A comprehensive literature review of plasma proteome studies highlights that about half of them potentially suffer from limitations related to sample processing.

## Results

### Erythrocyte and platelet proteins in the plasma proteome

During the development of our Plasma Proteome Profiling pipeline and its optimization for high-throughput screening of human cohorts (Geyer *et al*, 2016a), we repeatedly observed proteins that tended to emerge as groups of statistically significant outliers but appeared to be independent of the particular study. We hypothesized that they reflected sample quality issues. Manual and bioinformatic inspection revealed three classes of origin: erythrocytes, platelets, and the blood coagulation system. Consequently, we designed experiments to systematically characterize these main quality issues of the plasma proteome.

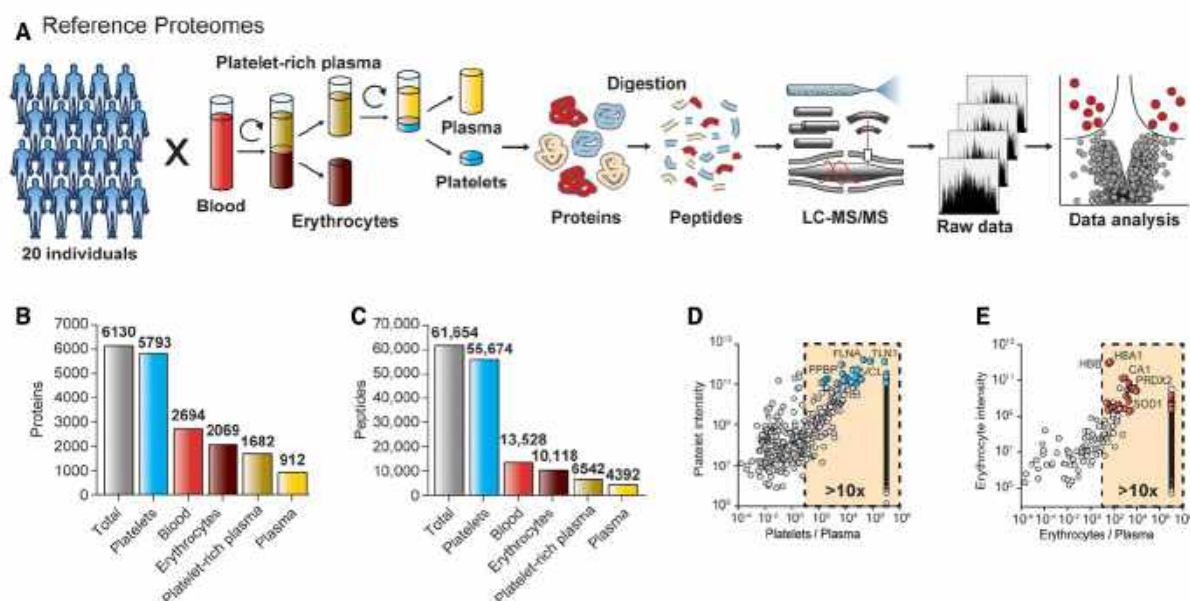
First, we acquired reference proteomes of erythrocytes and platelets, which are by far the most abundant cellular components ( $5 \times 10^6$  and  $3 \times 10^5$  cells per  $\mu\text{l}$ ). We harvested these cellular components from 10 healthy females and 10 males to obtain representative erythrocytes, platelets, and pure (platelet-free) plasma and further collected platelet-rich plasma and whole blood (Fig 1A; see Materials and Methods). Cell counting confirmed the purity of the samples (Table EV1). All five blood fractions were separately prepared for each individual by our automated proteomic sample preparation pipeline, followed by liquid chromatography coupled to high-resolution mass spectrometry (LC-MS/MS). To create reference proteomes, we generated a very deep library from pooled samples by analyzing extensively pre-fractionated peptides (Kulak *et al*, 2017; see Materials and Methods). A total of 6,130 different proteins were identified from 61,654 sequence-unique peptides (Fig 1B and C). The platelet proteome was the most extensive (5,793 proteins), whereas we detected 2,069 proteins in erythrocytes, 1,682 in platelet-rich plasma, and 912 in platelet-free plasma. The comparison of platelet-rich plasma to platelet-free plasma (84% additional

proteins) demonstrates the extent of proteins that can be introduced by platelets.

Next, we investigated purified samples for all 20 study participants individually. The average numbers of identified proteins and peptides were very consistent in all individuals (Appendix Fig S1). To construct panels of easily detectable and robust quality markers, we calculated the average protein intensities and the coefficient of variation (CV) across the study participants. As a prerequisite, we required that the proteins should be substantially more abundant in erythrocytes as well as platelets rather than in plasma. According to these criteria, we selected the 30 most abundant proteins with CVs below 30% and at least a 10-fold higher expression level in the contaminating cell type than in plasma (Fig 1D and E). NIF3-like protein 1 (NIF3L1), a low-abundance erythrocyte-specific protein, was excluded, because it was inconsistently identified as was the platelet-bound coagulation factor F13A1, whose function makes it an unsuitable platelet marker. The remaining proteins represent our cellular quality marker panels (Table EV2). They overlap by just two proteins (actin/ACTB and glyceraldehyde-3-phosphate dehydrogenase/GAPDH), and their quantities were not correlated with each other (Appendix Fig S2). Thus, they are specific and independent indicators for the origin of plasma quality.

Comparing median expression values of proteins shared between the blood components revealed that plasma proteins do correlate with whole blood (Pearson's correlation coefficient  $R = 0.43$ ), as expected. In contrast, there was no correlation between the platelet, erythrocyte, and plasma proteomes (Appendix Fig S2). This indicates that the levels of cellular proteins in plasma are not a constant fraction of those in the cellular proteomes. The platelet panel was enriched in platelet-rich plasma compared to normal (platelet-free) plasma. Both panels are de-enriched in pure plasma compared to whole blood, however, this effected the erythrocyte panel even more strongly, because centrifugation removes erythrocytes more efficiently than platelets. A histogram of both panels over the abundance range visualizes their distribution in the different blood compartments (Appendix Fig S2). Erythrocytes are 10-fold more abundant and fourfold larger than platelets, and indeed, the corresponding panel proteins have a 42-fold difference in whole blood. In plasma, however, their ratio was nearly one to one, again pinpointing a more efficient removal of erythrocytes than of platelets in standard sample preparation. The fact that several proteins of both panels were still detectable in pure plasma indicates a baseline level of contaminants due to imperfect de-enrichment or the life cycle of these cells. The four most abundant erythrocyte proteins, HBA1, HBB, CA1, and HBD, were present in pure plasma of almost all individuals, whereas lower abundant proteins were only sporadically identified. In contrast, platelet proteins were quantified over a larger abundance range and some of them were found in every individual.

In addition to the sum of panel protein abundances, we calculated their correlation to the standard reference panel defined by the 20 participants to several hundred plasma samples of a previous study (Geyer *et al*, 2016b). A distinct contamination of erythrocyte proteins seems to be a part of the plasma proteome as the erythrocyte panel has in general a relatively high correlation between the reference cohort erythrocyte levels and the plasma samples in the above-mentioned study. In contrast, in many plasma samples there



**Figure 1. Identification of blood cell markers.**

- A** Study outline and proteomic workflow. Erythrocytes, thrombocytes, platelet-rich, and platelet-free plasma were generated from 10 healthy female and male individuals by differential centrifugation and successive purification steps. To generate reference proteomes for each of the blood compartments, the respective protein samples of the 20 study participants were digested to peptides.
- B, C** Proteins (**B**) and peptides (**C**) identified for platelets, erythrocytes, platelet-rich, and platelet-free plasma.
- D, E** Selection of the most suitable quality marker proteins for (**D**) platelet contamination (blue dots) and (**E**) erythrocyte contamination (red dots) based on their abundance, the platelet/erythrocyte-to-plasma ratio, and the coefficient of variation. Proteins that were only detected in platelets or erythrocytes, but not in plasma are aligned on the right side of the graph.

was no correlation detectable between the reference cohort platelet levels and the plasma samples in the study. In practice, a correlation  $> 0.5$  indicated that the proteins are present as a result of contamination (Appendix Fig S3A–C). Note that an apparent contaminant protein could still be applied as a biomarker—however, in this case its abundance value should be different from the pattern in the reference quality panel.

#### Serial dilution experiments validate the erythrocyte and platelet quality marker panels

To determine whether the two protein panels correctly quantify contamination in plasma, we generated four pools of erythrocytes and platelets from five study participants at a time. These pools were diluted in nine steps into platelet-free plasma for a total range of  $10^7$ , followed by cell counting and proteomic analysis (Fig 2A). This resulted in an expected decrease in the cellular proteome ratio to plasma (Fig 2B and C). All but two of the panel proteins were consistently quantified over the dilution range. As the protein within each panel has the same origin, we defined a single variable for each cell type by summing their intensities and dividing by the summed intensities of all quantified plasma proteins. This yielded two remarkably robust “contamination indices” that turned out to be linear with respect to the cell numbers determined by cell cytometry (Table EV3;  $R = 0.98$  and  $0.99$ , Fig 2D and E). Spiked-in

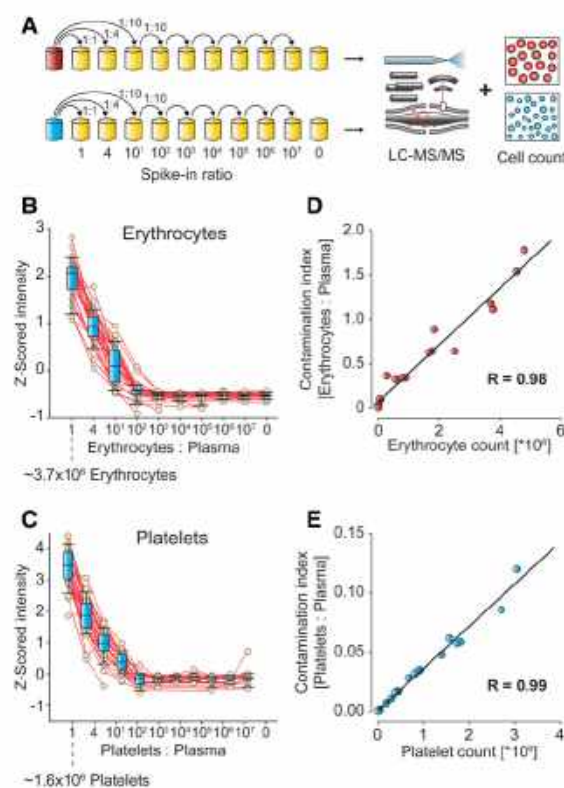
contaminations of 1:100 could readily be detected, which corresponds to a concentration of 70,000 erythrocytes or 30,000 platelets per  $\mu$ l plasma.

#### Quality marker panel for blood coagulation

In addition to contamination due to cellular constituents, partial and variable coagulation could contribute to systematic bias in biomarker studies. Indeed, we had found coagulation-related proteins to be connected to sample handling from finger pricks while developing our plasma proteomics pipeline (Geyer *et al.*, 2016a). In clinical practice, an anticoagulant is pre-added to commercially available containers so that it is combined with blood upon withdrawal. Prompt inversion mixes the anticoagulant with the blood, yielding pure plasma after centrifugation (Fig 3A). Any delay in adding or mixing could cause partial coagulation—in the extreme case of missing anticoagulant and waiting for 30 min, one would obtain serum instead of plasma.

To generate a panel for assessing blood coagulation, we systematically compared 72 plasma vs. 72 serum samples (four individuals, 18 aliquots). From a total of 2,099 quantified proteins, 299 were significantly altered (Fig 3B). The most significantly de-enriched proteins after clotting were typical constituents of the coagulation cascade such as fibrinogen chains alpha (FGA), beta (FCB), and gamma (FCG) ( $P < 10^{-150}$ ,  $> 40$ -fold), whereas the platelet-associated





**Figure 2.** Spike-in of erythrocyte and platelet fractions into pure plasma.

- A Dilution and analysis scheme.  
 B, C Protein intensities were Z-scored across the dilution series (B) for the 29 quality markers of the erythrocyte panel and (C) for the 29 markers of the platelet panel as a function of their spike-in proportion to plasma. Whiskers indicate 10–90 percentiles, and horizontal lines denote the mean.  
 D Correlation of erythrocyte count to the “contamination index” for the erythrocyte marker panel.  
 E Correlation of platelet count to contamination index for the platelet marker panel.

coagulation factor F13A1 and antithrombin-III (SERPINC1) decreased by more than half. Interestingly, the strongest elevated proteins in serum were highly abundant platelet proteins: platelet basic protein (PPBP), platelet glycoprotein Ib alpha chain (GP1BA), thrombospondin 1 (THBS1), and platelet glycoprotein V (GP5) ( $P < 10^{-10}$ ; twofold to fivefold increase). In total, 208 proteins increased and 91 decreased due to coagulation. The former set of proteins, which have higher levels in serum than in plasma, were also quantitatively enriched with high-abundant platelet proteins ( $P < 10^{-5}$ ; median rank 699 of 3,150 proteins), indicating coagulation-induced activation of platelets.

To define a robust panel of quality markers for the extent of coagulation, we first selected the 30 most significantly altered proteins between serum and plasma. Although not among the top 30, we added the platelet factor 4 variant 1 (PF4v1;  $P < 10^{-11}$ , 2.2-fold up in serum), because it was an excellent indicator of

coagulation in our studies and has already been reported in the context of pre-analytical variation (Timms *et al.*, 2007).

In contrast to the erythrocyte and platelet panels, proteins of the coagulation panel increase or decrease due to blood clotting and the fold changes vary strongly between them. Because fold changes are greatest for the decreasing proteins, we calculated the coagulation marker ratio only from them (sum of all plasma proteins divided by sum of plasma-elevated coagulation proteins). This ratio was very robust when comparing serum and plasma, clearly separating them with median ratios of 9 and 120 for these distinct sample types (Fig 3C). Of the coagulation marker panel, only F13A1, PPBP, and THBS1 were in common with the platelet panel and none with the erythrocyte panels (Fig 3D). The low overlap observed for the three quality marker panels should make them highly specific tools to elucidate the presence and origin of sample-related bias.

#### Application of the quality marker panels to a biomarker study

The above-defined marker panels can assess sample-related issues at three levels: the quality of each sample in a clinical cohort, potential systematic bias in the entire study, and the likelihood that individual biomarker candidates belong to the contaminant proteomes.

We recently investigated changes in the plasma proteome upon weight loss (Geyer *et al.*, 2016a,b). Briefly, caloric restriction in 52 individuals for 2 months was followed by weight maintenance for 1 year. Plasma Proteome Profiling of seven longitudinal samples revealed significant changes in the profile of apolipoproteins, a decrease in inflammatory proteins and markers correlating with insulin sensitivity. Given that protein abundance changes of  $< 20\%$  were often highly significant, we expected that overall sample quality was high, making this study suitable for testing the practical applicability of the quality marker panels.

First, we assessed the quality of each sample separately by calculating the three contamination indices and plotting their distribution in the total of 318 measurements. For each index, we initially defined potentially contaminated samples as those with a value more than two standard deviations above the mean (red lines in Fig 4A). This flagged 12 samples, six with platelet contamination, one with increased erythrocyte levels, and five with signs of partial coagulation. Resolving the three quality marker panels to the levels of individual proteins resulted in almost perfectly parallel trajectories (Appendix Fig S4A–C). Accordingly, the correlations to the reference quality marker panels were substantial ( $R > 0.77$ ). Overall, the variation of the contamination indices was highest for the platelets also visible by a contamination index difference (max/min ratio) of a factor 182 between the least and the most contaminated sample, followed by erythrocytes (max/min 23), and lowest for coagulation (max/min 5). The platelet proteins talin-1 (TLN1), myosin-9 (MYH9), and alpha-actinin-1 (ACTN1) had the largest variations, all with maximal changes  $> 5,000$ -fold. Catalase (CAT), carbonic anhydrase 1 and 2 (CA1, CA2) from the erythrocyte index varied maximally by more than 500-fold. The three fibrinogens in the coagulation panel changed by up to 20-fold, indicating that only partial coagulation events took place (Fig 4A).

Note that evaluating individual sample quality based on the standard deviation of all samples, as done here, has the benefit of being independent of the specific proteomic method used to measure protein amounts. However, this requires that most samples have



low levels of contamination, so that outliers of the statistical distribution are clearly apparent. If this is not the case, we propose using general, study-independent cutoff values to differentiate between samples of high and poor quality in such studies.

To assess potential systematic bias for groups of samples such as cases and controls or different time points, we applied a *t*-test based volcano plot. Most of the significantly upregulated proteins at time point 4 were members of the platelet panel (Fig 4B). With this information in hand, we contacted our collaboration partners, who tracked down the platelet contamination to a switch of the blood-taking equipment due to low supplies.

In practice, such sample issues will occasionally happen in a clinical study, and our quality marker panels would allow elimination of the affected samples. However, if contaminating proteins can reliably be distinguished from relevant biomarker candidates, the data could still be used. In our example, six of the eight significant outliers were from the platelet panel, and the other two proteins—GP1BA and NRPI—could still be of interest. To investigate this further, we inspected the global correlation map of all proteins, time points, and participants (Albrechtsen *et al*, 2018). In this hierarchical clustering analysis, proteins that are co-regulated have a high correlation to each other and appear in groups, visualized as red patches (Fig 4C). Here, the platelet cluster was the second largest one with 38 proteins ( $R = 0.69$ ). All quantified platelet panel proteins were in this cluster, as was GP1BA, flagging them as likely contaminants (Fig 4C and inset). Interestingly, NRPI, a receptor involved in angiogenesis, did not group with the platelet proteins, suggesting a potential biological role. This is supported by the fact that NRPI was significantly regulated over all time points compared to the baseline, in contrast to the platelet cluster proteins.

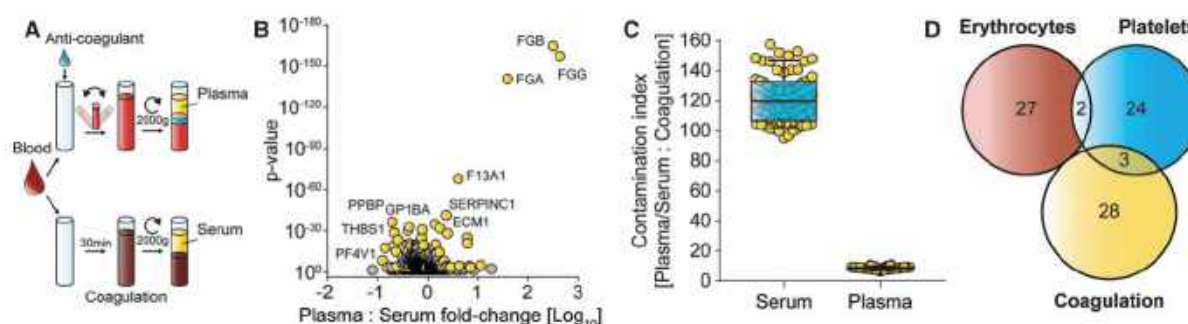
The other two quality marker panels are also readily apparent in the global correlation map. Ten members of the erythrocyte panel cluster tightly as do the three fibrinogen chains (Appendix Fig S5). However, in this study the fibrinogens group with proteins involved in low-grade inflammation, reduction of which was one of the main findings of our study (Appendix Fig S5). In contrast, the coagulation

marker PF4v1, which is also a highly abundant protein in platelets, clustered in the platelet group in this analysis, indicating that it varied as a result of sample preparation.

To make the above-described analysis readily available, we created an online platform at [www.plasmaproteomeprofiling.org](http://www.plasmaproteomeprofiling.org). It provides a toolbox for the interactive assessment of the quality of plasma proteomic data. Lists of protein abundances from MaxQuant search result tables or the template (Table EV4) can be uploaded by a simple drag and drop system. The system automatically generates the three contamination index values as shown in Fig 4A. If the user indicates cases and controls, the data set will be analyzed for systematic bias as visualized in a volcano plot (Fig 4B). The global correlation map is also displayed with the clusters of the quality marker panels (Fig 4C). The website is designed in the Dash data visualization framework, which allows further interactive analysis of the data (see Materials and Methods). Potential biomarker candidates in the volcano plot can be selected and displayed in the global correlation map to check whether the protein falls into or near one of the quality marker clusters.

#### Revisiting results of published biomarker studies

Having examined one study in detail, we set out to survey the extent to which quality marker proteins are reported as biomarker candidates in the literature. To this end, we performed a comprehensive PubMed search requiring the terms 'proteomics', 'proteome', 'plasma OR serum', 'biomarker' and 'mass spectrometry' spanning the time frame from 2002 to April 2018. We excluded review papers, purely technological publications without biomarker candidates, animal studies, and publications without proteins as qualitative or quantitative variables. From the resulting 210 publications, we manually extracted the lists of the biomarker candidates that were reported as "significantly altered proteins" by the authors. Gene and protein names were mapped to the corresponding protein identifiers in our reference panels and analyzed for their frequencies.



**Figure 3. Quality marker panel for blood coagulation.**

- A** Preparation of plasma and serum samples. EDTA was used as anticoagulation agent, and incubation and centrifugation values are indicated.
- B** Volcano plot comparing 72 plasma vs. 72 serum proteomes. Proteins highlighted in yellow were chosen according to their *P*-value as markers for coagulation. Only the plasma-enriched proteins (compared to serum) were used in the calculation of the coagulation contamination index.
- C** Ratio of the summed intensities of all plasma or serum proteins to the sum of the plasma-enriched panel proteins is plotted for all samples. Whiskers indicate the 10–90 percentile, and horizontal lines denote the mean.
- D** Overlap of the three quality marker panels.

Remarkably, 113 studies (54%) reported at least one potential quality marker as a biomarker candidate or as a statistically significant association (Fig 4D). As the total quality marker panel consists of 84 proteins and the median number of candidates per clinical study was seven, a certain overlap is not entirely unexpected. However, the candidates in question almost always were near the top of most abundant proteins of the quality marker panels, making it highly likely that they are indeed contaminants. Furthermore, while an individual protein could still be a genuine biomarker candidate, the fact that 22 studies (11%) reported two of them, and a further 23 studies (11%) three or more, again makes quality issues the likely explanation.

The majority of these studies reported proteins as potential biomarkers or as significant outliers of the coagulation panel, followed by the erythrocyte and platelet panels (Fig 4E). The most frequent one was clusterin (CLU; 27 times), followed by the fibrinogens (alpha, beta, and gamma; 22, 10, and 15 times), prothrombin (F2; 17 times), kininogen (KNG1; 15 times), antithrombin-III (SERPINC1; 13 times), and platelet basic protein (PPBP; 10 times). It is worth noting that proteins related to erythrocyte leakage may falsely be taken to indicate activation of oxidative pathways. For example, the hemoglobin subunits (e.g. HBA1, HBB, and HBD, listed 1, 6, and 1 time), carbonic anhydrases (CA1 and CA2, 6 and 6 times), fructose-bisphosphate aldolase (ALDOA, 5 times), peroxiredoxin 2 (PRDX2, 3 times), and superoxide dismutase (SOD1; 2 times) are annotated with keywords linked to oxidation. To illustrate this, a recent publication connected plasma proteome alterations in type 1 diabetes to oxidative stress. This may be a spurious link because the reported proteins were mostly members of the erythrocyte quality marker panel (Liu *et al*, 2018). Although platelet panel proteins are not prominent in the biomarker literature yet, we expect that they—along with lower abundant erythrocyte-specific proteins—will play an increasing role as technological progress enables higher plasma proteome coverage. We caution that platelet proteins already found in the biomarker literature such as PPBP, THBS1, and PF4 are often linked to coagulation events.

#### Recommendations for future proteomic studies

Based on our experience with the above-defined three quality marker panels (Table EV2) and analysis of thousands of plasma proteomes, we devised a general guideline for minimizing and detecting biases related to sample taking and processing (Table 1).

To further document the influence of common variables in the blood-taking process, we invited 10 healthy individuals and collected blood in 10 different blood sampling tubes. In this experiment, we systematically varied the type of plasma/serum, the blood specimen tubes (with or without gel), and the deposition of blood into the sampling tube (vacuum vs. pull system).

The most prominent differences were again between serum and plasma (Fig 3B; Appendix Fig S6). Apart from this, we found that contaminations with high-abundant erythrocyte-specific proteins appeared in several comparisons. Serum and EDTA plasma both had significantly higher levels than lithium heparin and citrate plasma (Appendix Fig S6A–F). Moreover, vacuum sampling can have an influence on erythrocyte-specific protein levels for some tubes. For instance, we found significantly increased levels of HBA1 and HBB in lithium heparin plasma tubes after vacuum sampling compared to a pull system, but not in the same comparison when using serum tubes (Appendix Fig S7A–D). Furthermore, erythrocyte-specific

proteins were significantly increased in lithium heparin pull tubes (more than twofold), which contain a gel plug compared to pull tubes without a gel plug (Appendix Fig S8A–D). In contrast, there were no differences between serum tubes with and without gel. These findings illustrate how even seemingly minor changes in blood-taking equipment can result in statistically significant differences of protein levels, which could confound biomarker studies. They also highlight the value of unbiased, system-wide investigation of the blood proteome and our quality marker panels.

We also found that the procedure of sampling the plasma from the tubes has a prominent effect on platelet contamination (Appendix Figs S9 and S10). Thus, we recommend not to collect the lowest layer of the plasma above the platelet bed after centrifugation. Furthermore, any delay from centrifugation to plasma harvest has the potential to induce platelet protein contamination. These factors mainly influence the platelet rather than the erythrocyte contamination index, indicating that proteins from the platelet proteome are the most likely cause of erroneous assignment of biomarker candidates.

## Discussion

Blood plasma remains the predominant biological matrix to assess health and disease in clinical settings. Around the world, every day hundreds of thousands of samples are analyzed to determine the levels of individual proteins. Likewise, blood plasma is directly or indirectly assessed in most clinical trials. Protein levels in plasma can readily be affected by cellular contamination or handling-related issues, and in clinical practice, this is partially addressed by simple tests such as those for hemoglobin contamination. However, these tests are not systematic or quantitative and they can only be used to exclude clearly contaminated samples.

Because of its high specificity and unbiased nature, MS-based proteomics is ideally suited to characterize the quality of blood plasma and it requires < 1 µl of material. So far, research on sample quality involving MS has mainly been restricted to the stability of internal standards in targeted assays and has rarely addressed overall sample quality (Schrohl *et al*, 2008; Hassis *et al*, 2015; Hoofnagle *et al*, 2016). Employing our Plasma Proteome Profiling pipeline to various clinical studies suggested that platelets, erythrocytes, and coagulation are by far the most important causes of plasma quality issues. We acquired very deep reference proteomes for these cell types and blood compartments, which we provide to the community to evaluate the possible origin of proteins emerging from biomarker studies. We defined three panels of about 30 proteins each that can serve as contamination indices (Table EV2). Using the example of a longitudinal Plasma Proteome Profiling study of weight loss and our online resource, we illustrated how the contamination indices can flag individual suspect samples and systematic biases. Furthermore, correlation analysis reveals whether potential biomarkers emerging from a given study are likely to be associated with quality-related proteome changes instead. Conversely, this procedure can “rescue” genuine biomarker candidates that are part of the quality marker proteomes. As an example, fibrinogens, a member of the coagulation quality marker panel, can also change during an inflammatory condition and might be correlated with classical inflammation markers such as CRP. In certain diseases, the entire set of proteins of a quality marker panel can be altered. For example, increased platelet



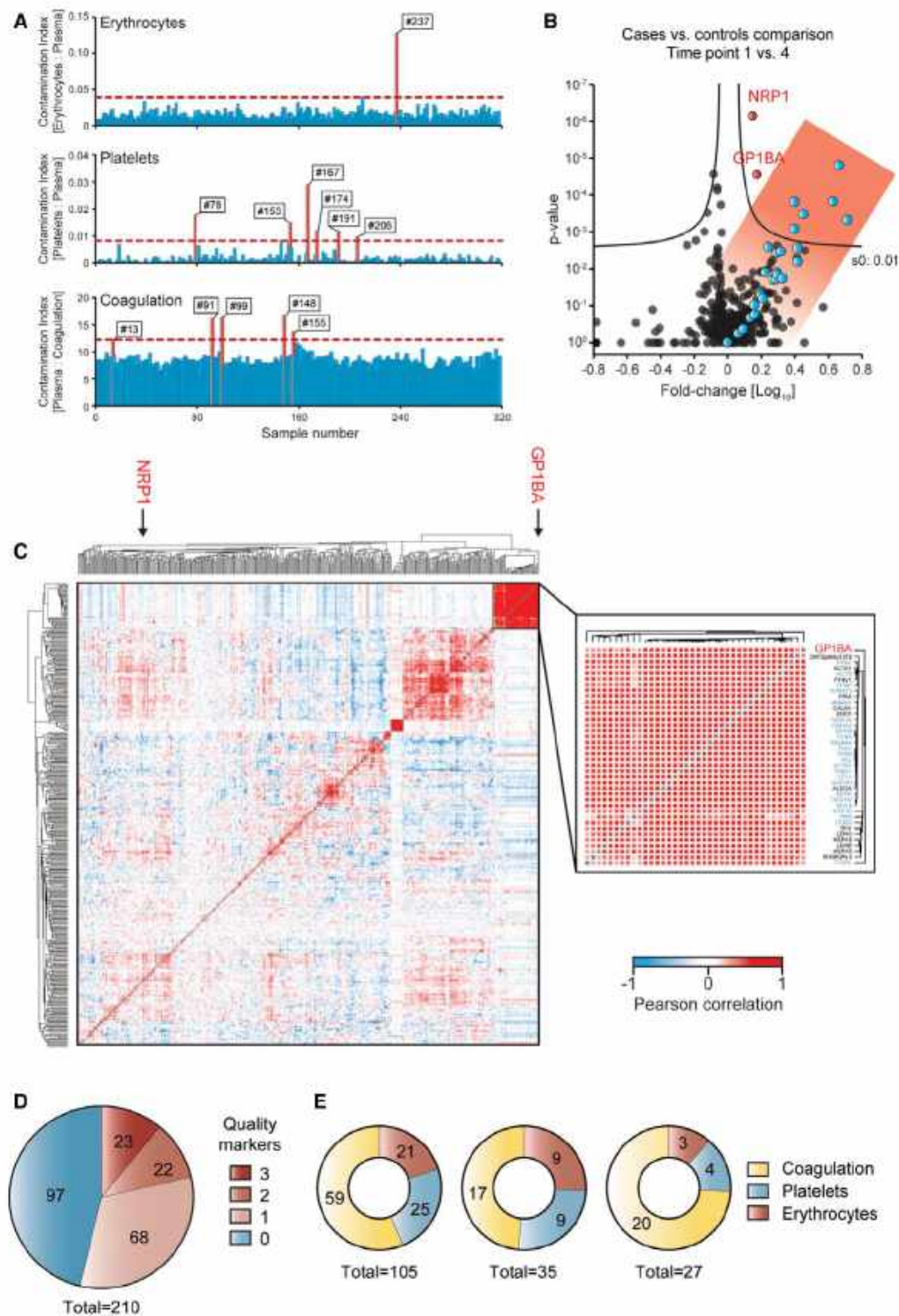


Figure 4.



**Figure 4. Quality marker panels in a weight loss study and literature study.**

- A Assessment of individual sample quality with respect to the three contamination indices using the online tool at [www.plasmaproteomeprofiling.org](http://www.plasmaproteomeprofiling.org). Samples with indices that are more than two standard deviations from the mean (horizontal red lines) are flagged as potentially contaminated (red bars and sample numbers).
- B Volcano plot of the proteome comparison of time point 1 vs. 4. Proteins of the platelet panel are highlighted in blue and two additional significantly regulated proteins in red.
- C Global correlation map on the left with an inset of the platelet cluster on the right. The two significant outliers of the volcano plot in (B) are marked in red. Platelet panel proteins are highlighted in blue in the inset. Red patches in the global correlation map indicate positive and blue patches negative correlations.
- D Literature analysis of 210 publications using MS-based plasma proteomics to identify new biomarkers. The number of quality markers reported as biomarker candidates in these studies is indicated.
- E Distribution of the reported quality markers according to the three types of likely contaminations. The distribution is shown across studies that report one, two, or three proteins of the same quality marker panel.

levels—thrombocytopenia—can have a variety of causes ranging from chronic inflammation to myeloproliferative diseases. Likewise, increased concentration of erythrocyte-specific proteins can be caused by hemolytic diseases such as in autoimmunity. While these cases are not the usual reasons why a quality marker panel is altered, they need to be considered when judging the analytical validity of a plasma measurement.

The clinical potential of the plasma proteome has long been realized and is also emphasized by the fact that more than 50

FDA-approved biomarkers can be quantified even in relatively shallow proteomic measurements of plasma (Geyer *et al*, 2016a). If there are as many new biomarkers among the less abundant proteins, there should be a diagnostic treasure trove still to be discovered (Geyer *et al*, 2017). Millions of plasma samples are stored in biobanks worldwide, representing an immense untapped resource that could be analyzed by MS-based proteomics or large-scale affinity-based methods. Despite initial enthusiasm and community efforts such as the Human Proteome Organization's plasma proteomic initiative (Omenn *et al*, 2005; Schwenk *et al*, 2017), few if any new protein biomarkers have entered the clinic in recent decades. This is probably at least partially due to technological limitations to characterize the vast dynamic range of the plasma proteome, which in turn has led to underpowered study designs (Geyer *et al*, 2017). While many of these challenges are already being addressed, we suspect that problems with sample quality represent another important reason for the paucity of new biomarkers and, even more seriously, for incorrect biomarkers being used. Examining our own data as well as the scientific literature, we here show that sample quality issues indeed have an impact on reported results. Nearly half of the reviewed studies reported at least one potential biomarker that is in our quality marker panels, and many had two or more, making sample contamination very likely. While coagulation-related issues are currently most prominent, increasing depth of plasma proteome coverage may replace platelet contamination as the most important source of error in the future. A corollary of the very large abundance variation of proteins introduced by quality issues is that it should further discourage pooling of samples. While this increases throughput, even a single contaminated sample can readily skew an entire batch.

Systematic bias introduced by imperfect sample handling or processing may lead to reporting incorrect biomarkers. Conversely, randomly distributed samples with poor quality will diminish overall statistical quality and may obscure true biomarker candidates.

The sources of quality issues are different kinds of variations in the pre-analytical processes, and we found platelet contamination during plasma harvesting to be one of the main culprits. Among the few previous studies, Hassis *et al* (2015) investigated different sample handling errors and concluded that only extreme conditions, such as delay in sample storage for 4 days, substantially changed the plasma proteome. However, proceeding with such extreme cases is rare, and quality issues are much more likely to originate from recontamination with whole blood after centrifugation during the plasma harvest or post-centrifugation times and resuspension of platelets, for instance. The comparison of 10 different blood sampling tubes showed that even seemingly minor differences in

**Table 1. Practical considerations to minimize systematic bias.****General instructions**

Avoid pooling of samples

Use plasma or serum exclusively, not a combination

**Sample collection**

Standardize blood collection and pre-analytical procedures (preferably same person collecting blood, centrifuge, sampling container, storage temperature, and time)

Centrifuge blood to generate plasma immediately

Centrifuge according to manufacturer's instruction

Harvest plasma immediately after centrifugation

Harvest the plasma starting from the top of the container and pool it before aliquoting

Discard the last 500  $\mu$ l of plasma to avoid contamination with platelets or use a second centrifugation step to generate platelet-poor plasma

Freeze samples immediately after harvesting

**Principal assessment of study sample quality**

When working with a new batch of samples from collaborators: run at least 10 test samples of each study group by mass spectrometry

Use quality marker panels to check for any indication of contamination

**Main study**

Continuously assess quality during the project to detect and avoid systematic bias (pre-analytics; mass spectrometric analyses)

Overall quality: report the number of contaminated samples

Systematic bias: report potential systematic bias

Check whether biomarker candidates are contained in the quality marker panels

Identification of several quality markers as biomarker candidates may be indicative of a study vector

If a quality marker is among the biomarker candidates, thorough validation is required

the sample handling devices like a pull vs. a vacuum deposition system can have a statistically significant effect on the measured proteome. Therefore, we want to stress the importance of strictly following standard operating procedures. We here provide general considerations for minimizing sample-related issues, ranging from immediate harvest of the plasma after centrifugation to discarding the lowest layer of plasma to avoid recontamination with platelets (Table 1). These recommendations update and extend general good laboratory practices as well as HUPO guidelines (Omenn *et al.*, 2005; Rai *et al.*, 2005). We also advocate that plasma samples are quality-checked by MS-based proteomics, at least for a representative subset. This is especially important for clinical studies but also for targeted single-analyte measurements, which by their nature are blind to the overall composition of the sample. Although it would be possible to determine contamination indices by multiplexed affinity-based methods, we recommend MS for this purpose because of its very high specificity and its unbiased nature. Furthermore, the proteomic depth needed to assess the quality is easily achievable even in rapid and economical measurements.

The concepts and methods put forward in this study could readily be adapted to other body fluids such as urine, saliva, or cerebrospinal fluid. This would require developing the appropriate contamination indices. Furthermore, the three quality marker categories are the largest but not the only ones. For instance, we imagine that similar experiments can be performed to gauge the effect of storage duration and temperature on the plasma proteome as it influences MS-based proteomics.

In conclusion, sample-related quality issues are clearly a concern for biomarker studies. However, we show here that they can be addressed rigorously and comprehensively by MS-based proteomics. As this technology continues to improve in throughput, depth, and robustness, we envision that it will be employed in routine clinical practice. Biomarker panels instead of single markers will be measured by MS-based proteomics as this takes advantage of its inherently multiplexed nature and allows the characterization of clinical conditions more comprehensively. These biomarker panels could routinely be extended with quality marker panels as introduced here, helping to establish biomarker-guided decisions in a wide variety of clinically important areas.

## Materials and Methods

### Samples for defining the three quality marker panels

All participants gave written informed consent for their participation in the Munich Study on Biomarker Reference Values (MyRef), which is registered under the local ethic number 11-16. All experiments conformed to the principles set out in the WMA Declaration of Helsinki and the Department of Health and Human Services Belmont Report.

To establish the quality marker panels, whole blood was harvested by venipuncture of 10 females and 10 males into commercial EDTA-containing sampling containers. The blood was centrifuged at 200 g for 10 min, and both the pellet and the supernatant were kept for further processing steps. The bottom layer of 500  $\mu$ l plasma was discarded to avoid contamination of the platelet-rich

plasma fraction with erythrocytes. The pellet was centrifuged at 2,000 g for 15 min, and the top layer containing plasma, the buffy coat, and 1 ml of erythrocytes were discarded. After adding 4 ml PBS containing 1.6 mg/ml EDTA, the suspension was centrifuged at 2,000 g for 15 min and the supernatant was discarded together with 500  $\mu$ l of the top layer of the erythrocytes. This step was repeated, and the pure erythrocyte fraction was harvested. We centrifuged the supernatant from the first centrifugation step containing plasma and platelets a second time at 200 g for 10 min and harvested the supernatant, which constitutes the platelet-rich plasma. This step was repeated, and we collected the supernatant and the platelet after centrifugation at 2,000 g for 15 min. The supernatant was centrifuged a second time at 2,000 g for 15 min to harvest platelet-free plasma by sampling only top layer of the supernatant, but discarding the bottom layer of 500  $\mu$ l. The platelets were washed twice by adding 4 ml PBS containing 1.6 mg/ml EDTA and centrifugation at 2,000 g for 15 min. The supernatant was discarded, and the pure platelet fraction was harvested.

For the serum and plasma comparison, blood samples from two females and two males were split into 18 samples each and serum and plasma were harvested after centrifugation at 2,000 g for 15 min.

To investigate the effects of different blood sampling devices on the blood plasma proteome, we invited 10 healthy individuals (five female and five males) and collected blood in the 10 different blood sampling devices (Table EVS). After collecting whole blood, it was incubated at room temperature for 30 min to allow coagulation in the serum tubes. The plasma tubes were also stored at room temperature for the same time, and the different tubes were centrifuged together. Afterward, 0.5 ml of plasma or serum was sampled from the top of the tubes.

To evaluate the platelet contamination in different layers of plasma after centrifugation, blood was collected in two different 9-ml S-Monovette EDTA-containing sampling containers (Sarstedt). The blood of one container was transferred to a 15-ml centrifugation tube without separation gel. Both containers were centrifuged at 2,000 g for 15 min. Plasma was harvested in nine volume fractions starting from the top layer in 500  $\mu$ l steps to the top of the buffy coat. The buffy coat itself was not touched, and a small amount of plasma (~200  $\mu$ l) remained on top.

### High-abundant protein depletion for building a matching library

We created a matching library and applied a consecutive depletion strategy, in which the top 6 and top 14 most abundant plasma proteins were depleted by using a combination of two immunodepletion kits, as described in ref. Geyer *et al.* (2016a). Briefly, the Agilent Multiple Affinity Removal Spin Cartridge was used for the depletion of the top six highest abundant proteins (albumin, IgG, IgA, antitrypsin, transferrin, and haptoglobin), followed by Seppro Human 14 Sigma immunodepletion for the 14 highest abundant proteins (albumin, IgG, IgA, IgM, IgD, transferrin, fibrinogen,  $\alpha$ 2-macroglobulin,  $\alpha$ 1-antitrypsin, haptoglobin,  $\alpha$ 1-acid glycoprotein, ceruloplasmin, apolipoprotein A-I, apolipoprotein A-II, apolipoprotein B, complement C1q, complement C3, complement C4, plasminogen, and prealbumin). Following depletion, we fractionated our samples using the high pH

**The paper explained****Problem**

New biomarkers are urgently needed in many health and disease contexts and mass spectrometry-based proteomics is a potentially powerful and promising technology for their discovery, as it can analyze the plasma proteome in a quantitative and specific manner. However, a systematic analysis of pre-analytical variations might obscure the discovery of novel biomarkers and has not been performed so far.

**Results**

We employ Plasma Proteome Profiling to discover three quality marker panels that report on the status of plasma samples with regards to erythrocyte lysis, platelet contamination, and partial coagulation. These panels can identify individual samples of poor quality and correct for systematic bias in biomarker studies. Moreover, they can be applied to evaluate whether a novel biomarker candidate is linked to one of the sources of contamination. We further provide sample preparation guidelines and an online resource to assess the overall sample-related bias in individual samples in clinical studies.

**Impact**

Quality issues due to erythrocyte lysis, platelet contamination, and partial coagulation might affect up to 50% of all biomarker studies as we showed by a literature survey of more than 200 published manuscripts. Our quality marker panels will prevent costly miss-assignment of potential biomarker candidates and support the discovery of promising biomarkers.

reversed-phase "Spider fractionator" into 24 fractions as described previously (Kulak *et al.*, 2017).

**Sample preparation: protein digestion and in-StageTip purification**

Sample preparation was carried out according to our Plasma Proteome Profiling pipeline as described in Geyer *et al.* (2016a,b) with an automated setup on an Agilent Bravo Liquid Handling Platform. In brief, plasma samples were diluted 1:10 with  $d_6H_2O$  and 10  $\mu$ l of the sample was mixed with 10  $\mu$ l PreOmics lysis buffer (P.O. 00001, PreOmics GmbH) for reduction of disulfide bridges, cysteine alkylation, and protein denaturation at 95°C for 10 min (Kulak *et al.*, 2014). Trypsin and LysC were added to the mixture after a 5-min cooling step at room temperature, at a ratio of 1:100 micrograms of enzyme to micrograms of protein. Digestion was performed at 37°C for 1 h. An amount of 20  $\mu$ g of peptides was loaded on two 14-gauge StageTip plugs, followed by consecutive purification steps according to the PreOmics iST protocol (www.preomics.com). The StageTips were centrifuged using an in-house 3D-printed StageTip centrifugal device at 1,500 g. The collected material was completely dried using a SpeedVac centrifuge at 60°C (Eppendorf, Concentrator plus). Peptides were suspended in buffer A\* [2% acetonitrile (v/v), 0.1% formic acid (v/v)] and sonicated (Branson Ultrasonics, Ultrasonic Cleaner Model 2510). Pools for each of the five sample types (whole blood, erythrocytes, platelets, plasma, and platelet-free plasma) were generated from the 20 individuals and prepared according to the procedure above. The peptides were fractionated using the high pH reversed-phase "Spider fractionator" into 24 fractions as described previously to generate deep proteomes (Kulak *et al.*, 2017).

**Ultra-high-pressure liquid chromatography and mass spectrometry**

Samples were measured using LC-MS instrumentation consisting of an EASY-nLC 1000 or 1200 ultra-high-pressure system (Thermo Fisher Scientific), which was coupled to a Q Exactive HF Orbitrap (Thermo Fisher Scientific) using a nano-electrospray ion source (Thermo Fisher Scientific). Purified peptides were separated on 40-cm HPLC columns [ID: 75  $\mu$ m; in-house packed into the tip with ReproSil-Pur C18-AQ 1.9  $\mu$ m resin (Dr. Maisch GmbH)]. For each LC-MS/MS analysis, about 0.5  $\mu$ g peptides were used for 45-min runs and for each fraction of the deep plasma data set.

Peptides were loaded in buffer A [0.1% formic acid and 5% DMSO (v/v)] and eluted with a linear 35-min gradient of 3–30% of buffer B [0.1% formic acid, 5% DMSO, and 80% (v/v) acetonitrile], followed stepwise by a 7-min increase to 75% of buffer B and a 1-min increase to 98% of buffer B, followed by a 2-min wash of 98% buffer B at a flow rate of 450 nl/min. Column temperature was kept at 60°C by an in-house-developed oven containing a Peltier element, and parameters were monitored in real time by the SprayQC software (Scheltema & Mann, 2012). MS data were acquired with a Top15 data-dependent MS/MS scan method for the construction of the library and BoxCar scans (Meier *et al.*, 2018) for the study samples. Target values for the full-scan MS spectra were  $3 \times 10^6$  charges in the 300–1,650 m/z range with a maximum injection time of 55 ms and a resolution of 60,000 at m/z 200. Fragmentation of precursor ions was performed by higher-energy C-trap dissociation (HCD) with a normalized collision energy of 27 eV. MS/MS scans were performed at a resolution of 30,000 at m/z 200 with an ion target value of  $1 \times 10^5$  and a maximum injection time of 120 ms. Dynamic exclusion was set to 30 s to avoid repeated sequencing of identical peptides.

**Data analysis**

MS raw files were analyzed by MaxQuant software, version 1.5.6.8, (Cox & Mann, 2008), and peptide lists were searched against the human UniProt FASTA database. A contaminant database generated by the Andromeda search engine (Cox *et al.*, 2011) was configured with cysteine carbamidomethylation as a fixed modification and N-terminal acetylation and methionine oxidation as variable modifications. We set the false discovery rate (FDR) to 0.01 for protein and peptide levels with a minimum length of 7 amino acids for peptides, and the FDR was determined by searching a reverse database. Enzyme specificity was set as C-terminal to arginine and lysine as expected using trypsin and LysC as proteases. A maximum of two missed cleavages were allowed. Peptide identification was performed with an initial precursor mass deviation up to 7 ppm and a fragment mass deviation of 20 ppm. The "match between run algorithm" in the MaxQuant quantification (Nagaraj *et al.*, 2012) was enabled after constructing a matching library consistent of depleted and all the undepleted plasma samples. All proteins and peptides matching to the reversed database were filtered out. Label-free protein quantitation (LFQ) was performed with a minimum ratio count of 2 (Cox *et al.*, 2014).

**Bioinformatic analysis**

All bioinformatic analyses were performed with the Perseus software of the MaxQuant computational platform (Cox & Mann, 2008);



Tyanova et al, 2016). For the global correlation analysis, proteins were filtered for at least 50% valid values in the weight loss study and the hierarchical clustering was performed using Euclidean distance. The weight loss study contained in total 28 proteins of the platelet panel, but after sorting for 50% valid values only 24 were left and all of them clustered in the platelet panel.

### Online platform for automated analysis of clinical studies

Our online portal is equipped with a user-friendly graphical interface that supports the most common web browsers, such as Google Chrome, Firefox, and Internet Explorer. For the front-end development, a Dash framework was used (version 0.27.0), which consists of a Flask server (1.0.2) that communicates with front-end React.js components using JSON, or JavaScript Object Notation, packets (a minimal, readable format for structuring data) over HTTP, or Hypertext Transfer Protocol, requests that work as request-response protocols between a client and server. Taking advantage of the full power of Cascading Style Sheets (CSS), every graphical element was customized: the sizing, the positioning, the colors, and the fonts.

The platform takes the results of the MS data processed by the MaxQuant software (Cox & Mann, 2008) from the proteinGroups table (to be extended to other formats). During the data uploading, the input file is verified through a combination of preliminary tests. We built a complex data structure using general Python libraries, such as NumPy, Pandas, and SciPy. Using three panels of markers for platelet contamination, erythrocyte contamination, and coagulation events in plasma samples, respectively, we identify samples affected by quality issues. Samples having at least 50% "valid values" (i.e. those with quantification results) are preprocessed by cleaning the data and prepare them for the subsequent visualization step.

### Data availability

The MS-based proteomic data have been deposited to the ProteomeXchange Consortium via the PRIDE partner repository and are available via ProteomeXchange with identifier PXD011749 (<https://www.ebi.ac.uk/pride/archive/projects/PXD011749>).

**Expanded View** for this article is available online.

### Acknowledgements

We thank all members of the Proteomics and Signal Transduction Group and the Clinical Proteomics Group for help and discussions and in particular Igor Paron, Christian Deimi, Alexander Strasser, and Gaby Sowa for technical assistance; Mario Orosi for help with the online resource; Nicolai J. Wewer Albrechtsen, Nils A. Kulak, Niels Skotte, and Martin Steger for discussion; and Jürgen Cox for bioinformatic tools. The work carried out in this project was partially supported by the Max Planck Society for the Advancement of Science, the European Union's Horizon 2020 research and innovation program with the MSmed project (no. 686547), and grants from the Novo Nordisk Foundation (NNF15CC0001, NNF15OC0016692) and the BMRF grant German Biobank Alliance (BMRF 01EY1711C).

### Author contributions

PEG designed, performed, and interpreted the MS-based proteomic analysis of patient plasma; wrote the paper; and generated the figures. PVT wrote the

manuscript and performed together with LN, SD, JBM, AK, MLB, and JB experiments and generated article text. DT and LMH designed experiments, drafted practical considerations for sample preparation, and worked on the article text. EV designed and established the interactive online resource. MM designed and interpreted the MS-based proteomic analysis of plasma, supervised and guided the project, and wrote the manuscript.

### Conflict of interest

The authors declare that they have no conflict of interest.

### For more information

- (i) <https://www.biochem.mpg.de/en/rd/mann>
- (ii) <https://www.cpr.ku.dk/research/proteomics/mann-group/>
- (iii) <http://www.plasma.proteomeprofiling.org/>

### References

- Aebersold R, Mann M (2003) Mass spectrometry-based proteomics. *Nature* 422: 198–207
- Aebersold R, Mann M (2016) Mass-spectrometric exploration of proteome structure and function. *Nature* 537: 347–355
- Albrechtsen NJW, Geyer PE, Doll S, Bojsen-Møller KN, Martinussen C, Torekov SS, Keilhauer E, Treit PV, Meier F, Holst JJ et al (2018) Plasma proteome profiling reveals dynamics of inflammatory and lipid homeostasis markers after Roux-en-Y gastric bypass surgery. *Cell Syst* 7: 601–612 e3
- Anderson NL, Ptolemy AS, Rifai N (2013) The riddle of protein diagnostics: future bleak or bright? *Clin Chem* 59: 194–197
- Assarsson E, Lundberg M, Holmquist G, Björkstén J, Thorsen SB, Ekman D, Eriksson A, Renzel Dickens E, Ohlsson S, Edfeldt G et al (2014) Homogenous 96-plex PEA immunoassay exhibiting high sensitivity, specificity, and excellent scalability. *PLoS One* 9: e95192
- Cox J, Mann M (2008) MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat Biotechnol* 26: 1367–1372
- Cox J, Neuhauser N, Michalski A, Scheltema RA, Olsen JV, Mann M (2011) Andromeda: a peptide search engine integrated into the MaxQuant environment. *J Proteome Res* 10: 1794–1805
- Cox J, Hein MY, Luber CA, Paron I, Nagaraj N, Mann M (2014) Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Mol Cell Proteomics* 13: 2513–2526
- FDA-NIH-Biomarker-Working-Group (2016) BEST (Biomarkers, EndpointS, and other Tools) Resource. Maryland: Silver Spring (MD): Food and Drug Administration (US); Bethesda (MD): National Institutes of Health (US)
- Ganz P, Heidecker B, Hveem K, Jonasson C, Kato S, Segal MR, Sterling DG, Williams SA (2016) Development and validation of a protein-based risk score for cardiovascular outcomes among patients with stable coronary heart disease. *JAMA* 315: 2532–2541
- Geyer PE, Kulak NA, Pichler G, Holdt LM, Teupser D, Mann M (2016a) Plasma proteome profiling to assess human health and disease. *Cell Syst* 2: 185–195
- Geyer PE, Wewer Albrechtsen NJ, Tyanova S, Grassi N, Iepsen EW, Lundgren J, Madsbæk S, Holst JJ, Torekov SS, Mann M (2016b) Proteomics reveals the effects of sustained weight loss on the human plasma proteome. *Mol Syst Biol* 12: 901
- Geyer PE, Holdt LM, Teupser D, Mann M (2017) Revisiting biomarker discovery by plasma proteomics. *Mol Syst Biol* 13: 942

- Gold L, Ayers D, Bertino J, Bock C, Bock A, Brody EN, Carter J, Dalby AB, Eaton BE, Fitzwater T et al (2010) Aptamer-based multiplexed proteomic technology for biomarker discovery. *PLoS One* 5: e15004
- Hassiss ME, Niles RK, Braten MN, Albertolle ME, Ewa Witkowska H, Hubel CA, Fisher SJ, Williams KE (2015) Evaluating the effects of preanalytical variables on the stability of the human plasma proteome. *Anal Biochem* 478: 14–22
- Herder C, Kannenberg JM, Carstensen-Kirberg M, Strom A, Bonhof CJ, Rathmann W, Huth C, Koenig W, Heier M, Krumsiek J et al (2018) A systemic inflammatory signature reflecting cross talk between innate and adaptive immunity is associated with incident polyneuropathy: KORA F4/FF4 study. *Diabetes* 67: 2434–2442
- Hoofnagle AN, Whiteaker JR, Carr SA, Kuhn E, Liu T, Massoni SA, Thomas SN, Townsend RR, Zimmerman LJ, Boja E et al (2016) Recommendations for the generation, quantification, storage, and handling of peptides used for mass spectrometry-based assays. *Clin Chem* 62: 48–69
- Kulak NA, Pichler G, Paron I, Nagaraj N, Mann M (2014) Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nat Methods* 11: 319–324
- Kulak NA, Geyer PE, Mann M (2017) Loss-less nano-fractionator for high sensitivity, high coverage proteomics. *Mol Cell Proteomics* 16: 694–705
- Liu CW, Bramer L, Webb-Robertson BJ, Waugh K, Rewers MJ, Zhang Q (2018) Temporal expression profiling of plasma proteins reveals oxidative stress in early stages of Type 1 Diabetes progression. *J Proteomics* 172: 100–110
- Meier F, Geyer PE, Vinreira Winter S, Cox J, Mann M (2018) BoxCar acquisition method enables single-shot proteomics at a depth of 10,000 proteins in 100 minutes. *Nat Methods* 15: 440–448
- Mischak H, Allmaier G, Apweiler R, Attwood T, Baumann M, Benigni A, Bennett SE, Bischoff R, Bongcam-Rudloff E, Capasso G et al (2010) Recommendations for biomarker identification and qualification in clinical proteomics. *Sci Transl Med* 2: 46 ps42
- Munoz J, Heck AJ (2014) From the human genome to the human proteome. *Angew Chem Int Ed Engl* 53: 10854–10866
- Nagaraj N, Kulak NA, Cox J, Neuhauser N, Mayr K, Hoerning O, Vorm O, Mann M (2012) System-wide perturbation analysis with nearly complete coverage of the yeast proteome by single-shot ultra HPLC runs on a bench top Orbitrap. *Mol Cell Proteomics* 11: M111 013722
- Omenn GS, States DJ, Adamski M, Blackwell TW, Menon R, Hermjakob H, Apweiler R, Haab BB, Simpson RJ, Eddes JS et al (2005) Overview of the HUPO Plasma Proteome Project: results from the pilot phase with 35 collaborating laboratories and multiple analytical groups, generating a core dataset of 3020 proteins and a publicly-available database. *Proteomics* 5: 3226–3245
- Qundus U, Hong MG, Tybring G, Divers M, Odeberg J, Uhlen M, Nilsson P, Schwenk JM (2013) Profiling post-centrifugation delay of serum and plasma with antibody bead arrays. *J Proteomics* 95: 46–54
- Rai AJ, Gelfand CA, Haywood BC, Warunek DJ, Yi J, Schuchard MD, Mehlig RJ, Cockrill SL, Scott GB, Tammen H et al (2005) HUPO Plasma Proteome Project specimen collection and handling: towards the standardization of parameters for plasma proteome samples. *Proteomics* 5: 3262–3277
- Scheltens RA, Mann M (2012) SprayQc: a real-time LC-MS/MS quality monitoring system to maximize uptime using off the shelf components. *J Proteome Res* 11: 3458–3466
- Schrohl AS, Wurtz S, Kohn E, Banks RE, Nielsen HJ, Sweep FC, Brunner N (2008) Banking of biological fluids for studies of disease-associated protein biomarkers. *Mol Cell Proteomics* 7: 2061–2066
- Schwenk JM, Omenn GS, Sun Z, Campbell DS, Baker MS, Overall CM, Aebersold R, Moritz RL, Deutsch EW (2017) The Human Plasma Proteome Draft of 2017: building on the human plasma PeptideAtlas from mass spectrometry and complementary assays. *J Proteome Res* 16: 4299–4310
- Skates SJ, Gillette MA, LaBaer J, Carr SA, Anderson L, Liebler DC, Ransohoff D, Rifai N, Kondratovich M, Tezak Z et al (2013) Statistical design for biospecimen cohort size in proteomics-based biomarker discovery and verification studies. *J Proteome Res* 12: 5383–5394
- Sun BB, Maranville JC, Peters JE, Stacey D, Staley JR, Blackshaw J, Burgess S, Jiang T, Paige E, Surendran P et al (2018) Genomic atlas of the human plasma proteome. *Nature* 558: 73–79
- Surinova S, Schiess R, Huttenhain R, Cerriello F, Wollscheid B, Aebersold R (2011) On the development of plasma protein biomarkers. *J Proteome Res* 10: 5–16
- Timms JF, Arslan-Low E, Gentry-Maharaj A, Luo Z, T'Jampens D, Podust VN, Ford J, Fung ET, Gammerman A, Jacobs I et al (2007) Preanalytic influence of sample handling on SELDI-TOF serum protein profiles. *Clin Chem* 53: 645–656
- Tyanova S, Temu T, Sinitcyn P, Carlson A, Hein M, Geiger T, Mann M, Cox J (2016) The Perseus computational platform for comprehensive analysis of (pro)teomics data. *Nat Methods* 13: 731–740
- Wild D (2013) *The immunoassay handbook: theory and applications of ligand binding, ELISA, and related techniques*, 4<sup>th</sup> edn. Oxford; Waltham, MA: Elsevier



**License:** This is an open access article under the terms of the Creative Commons Attribution 4.0 License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

## Acknowledgements

First of all, I want to thank my dear supervisor, Matthias Mann, for so many things. Thank you for convincing me, a rather confident Master student who wanted to go into industry after finishing my Master's thesis, to stay for my PhD. I can't compare myself now with myself finishing Master's degree, not only in the knowledge and experience I gained during this enjoyable PhD time, but also in my personal development. I was scared and at the same time honoured to be the first "renewed" bioinformatician in the department. I am extremely happy to see how things have developed in our bioinformatics team over 4 years and glad that I have had the opportunity to witness its growth. Thank you for your support along the way and for the nice atmosphere!

I would like to thank the rest of my thesis committee, Prof. Dr. Klaus Förstemann, Prof. Dr. Johanna Klughammer, Prof. Dr. Julian Stingele and Prof. Dr. Lucas Jae, for their critical reading of my thesis and their comments, and special thanks to Dr. Stefan Canzar for being part of my TAC and thesis committee and all the scientific and personal input.

Being in the department during my Master's thesis and my entire PhD work, I changed offices several times, meeting different nice people. So I am grateful to all the people I have shared an office with during my time in the department!

I would like to thank the people in my first office for the pleasant and welcoming atmosphere during my Master time. I am especially very grateful to Maria Tanzer for the pleasant chitchat about everything in the world and for the further successful work on my main PhD project!

I want to thank the timsTOF office, where I spent a lot of time during my PhD study. I hope, guys, that I did not freeze you out too much in the winter by opening the windows too often! ☺ Thanks to Florian Meier for the welcoming atmosphere and for the opportunity to collaborate on several Bruker projects. Thanks to Catherine Vasilopoulou, Igor Paron and Antonio for the endless jokes and enjoyable environment. I especially want to thank Andreas-David Brunner for being a nice roommate at first and later becoming a friend and a great collaborator in many of my projects, for being able to ask you any questions about mass spec and seeing you take a pen and piece of paper to draw something to explain the concept to me! It has been a great help and I really appreciate your patience!

Having been in the department for a long time, I have had the chance to get to know the "new generation" of the timsTOF office, and I would also like to thank all of you for your nice help with the Bruker data. I would especially like to thank my dear friend Patricia



Skowronek for her patience and diligence, for our teamwork on several projects and for the pleasant time we spent together as neighbors! I will never forget your tomatoes! ;-)

I am so grateful to our wonderful bioinformatics team! I was present when the team was set up and I am so pleased to see what nice people have joined the team. Everyone has been very helpful, and I am honoured to have had the opportunity to work with all of you guys at some point in time. I especially want to thank Isabell Bludau for the enjoyable time we had working together on several different projects and just sitting next to each other in the office and chatting! And, of course, I want to thank Sander Willems for many things. It has been such a pleasure working with you on projects and learning so much from you! I am happy that we have become colleagues and even friends, and I hope I will have the opportunity to work with you again one day!

To all the members of the Mann department, whom I have not mentioned so far, for creating such a thriving atmosphere of amazing science!

I want to thank all my friends for their tremendous support! Thank you, Tolga, for allowing me to complain about the difficulties of PhD and for finding understanding on your side. I especially want to thank my dear friend Natasha for our very long friendship and all that we had to overcome together!

I am eternally grateful to my whole family - my mother, sister and father - for supporting me and believing in my strength and success. Thank you, Mum, for taking a little girl's words seriously about moving to another country and achieving something to be proud of.

Finally, I would like to thank you, Joan. You are my support and my peace of mind! You have always believed in me and have taken my mind off all difficulties and worries. Thank you for always being there for me, my darling!