Integrative Bioinformatics Applications for Complex Human Disease Contexts

Markus Joppich

München 2021

Integrative Bioinformatics Applications for Complex Human Disease Contexts

Markus Joppich

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Markus Joppich
aus Bonn

München, den 21.05.2021

Erstgutachter: Prof. Dr. Ralf Zimmer
Zweitgutachter: Prof. Dr. Martin Hofmann-Apitius
Tag der mündlichen Prüfung: 26.11.2021

Contents

List of Figures

List of Tables

Zusammenfassung

Die Dissertation beschreibt Methoden zur Prozessierung von aktuellen Hochdurchsatzdaten, sowie Verfahren zu deren weiterer integrativen Analyse. Diese findet Anwendung vor allem im Kontext von komplexen menschlichen Krankheiten.

Neue Messtechniken erlauben eine detailliertere Beobachtung biomedizinischer Prozesse. Mit RNA-Sequenzierung (RNA-seq) wird mRNA-Expression gemessen, mit Hilfe von moderner single-cell-RNA-seq (scRNA-seq) sogar für (sehr viele) einzelne Zellen. Long-Read-Sequenzierung wird zunehmend zur Sequenzierung ganzer Genome eingesetzt. Mittels bildgebender Massenspektrometrie (IMS) können Proteine in Geweben räumlich aufgelöst quantifiziert werden. Diese Techniken bringen spezifische Herausforderungen mit sich, die mit neuen bioinformatischen Methoden angegangen werden müssen. Für die integrative Datenanalyse ist auch die Gewinnung von geeignetem Kontextwissen wichtig. Wissenschaftliche Erkenntnisse werden in Artikeln veröffentlicht, die über große Literaturdatenbanken zugänglich sind. Mittels Textmining können daraus Informationen extrahiert werden, z.B. miRNA-Gen-Interaktionen, die in eigenen Datenbank aggregiert werden um spezifische Fragen mit nachvollziehbaren Belegen zu beantworten. In Kombination mit experimentellen Daten bieten sich so neue Möglichkeiten für integrative Methoden.

Durch die Extraktion von Rohdaten und deren Vorprozessierung werden mehrere Datenquellen erschlossen, wie z.B. Literatur für Textmining von miRNA-Gen-Interaktionen (Kapitel 2), Long-Read- und RNA-seq-Daten für Genomics und Transcriptomics (Kapitel 4.2, 5) und IMS für Protein-Messungen (Kapitel 4.4). So dienen z.B. die poreSTAT und sequ-into Methoden der Vorprozessierung und Auswertung von Long-Read-Sequenzierungen [142]. In der integrativen (down-stream) Analyse werden diese (heterogenen) Datenquellen verwendet. Für die Bestimmung von Zelltypen in scRNA-seq-Experimenten wurde die cPred-Methode (Kapitel 4.2) erfolgreich im Kontext der SARS-CoV-2-Pandemie eingesetzt [228, 238]. Auch die robuste Pipeline `RoDE` fand dort Anwendung, die viele Methoden zur (differentiellen) Datenanalyse, zum Reporting und zur Visualisierung bereitstellt (Kapitel 6.1). Themen der Benutzbarkeit von (bioinformatischer) Software werden an Hand von praktischen Anwendungen diskutiert (Kapitel 3, [145]). Die entwickelte miRNA-Gen-Interaktionsdatenbank gibt wertvolle Einblicke in Atherosklerose-relevante Prozesse [144] und dient als regulatorisches Netzwerk für die Vorhersage von aktiven miRNA-Regulatoren in `RoDE` (Kapitel 6.1). Die cPred-Methode, `RoDE`-Ergebnisse, scRNA-seq- und IMS-Daten werden im 3D-Index Aorta3D (Kapitel 6.2) zusammengeführt, der relevante Datensätze durchsuchbar macht. Die diskutierten Methoden führen zu erheblichen Verbesserungen für die integrative Datenanalyse in komplexen menschlichen Krankheitskontexten.

Summary

This thesis presents new methods for the analysis of high-throughput data from modern sources in the context of complex human diseases, at the example of a bioinformatics analysis workflow. New measurement techniques improve the resolution with which cellular and molecular processes can be monitored. While RNA sequencing (RNA-seq) measures mRNA expression, single-cell RNA-seq (scRNA-seq) resolves this on a per-cell basis. Long-read sequencing is increasingly used in genomics. With imaging mass spectrometry (IMS) the protein level in tissues is measured spatially resolved. All these techniques induce specific challenges, which need to be addressed with new computational methods. Collecting knowledge with contextual annotations is important for integrative data analyses. Such knowledge is available through large literature repositories, from which information, such as miRNA-gene interactions, can be extracted using text mining methods. After aggregating this information in new databases, specific questions can be answered with traceable evidence. The combination of experimental data with these databases offers new possibilities for data integrative methods and for answering questions relevant for complex human diseases.

Several data sources are made available, such as literature for text mining miRNA-gene interactions (Chapter 2), next- and third-generation sequencing data for genomics and transcriptomics (Chapters 4.1, 5), and IMS for spatially resolved proteomics (Chapter 4.4). For these data sources new methods for information extraction and pre-processing are developed. For instance, third-generation sequencing runs can be monitored and evaluated using the poreSTAT and sequ-into [142] methods. The integrative (down-stream) analyses make use of these (heterogeneous) data sources. The cPred method (Chapter 4.2) for cell type prediction from scRNA-seq data was successfully applied in the context of the SARS-CoV-2 pandemic [228, 238]. The robust differential expression (DE) analysis pipeline `RoDE` (Chapter 6.1) contains a large set of methods for (differential) data analysis, reporting and visualization of RNA-seq data. Topics of accessibility of bioinformatics software are discussed along practical applications (Chapter 3, [145]). The developed miRNA-gene interaction database gives valuable insights into atherosclerosis-relevant processes [144] and serves as regulatory network for the prediction of active miRNA regulators in `RoDE` (Chapter 6.1). The cPred predictions, `RoDE` results, scRNA-seq and IMS data are unified as input for the 3D-index Aorta3D (Chapter 6.2), which makes atherosclerosis related datasets browsable. Finally, the scRNA-seq analysis with subsequent cPred cell type prediction, and the robust analysis of bulk-RNA-seq datasets, led to novel insights into COVID-19 [238]. Taken all discussed methods together, the integrative analysis methods for complex human disease contexts have been improved at essential positions.

Yeah, I can have that up and running in anywhere from 2 days to 2 months. Depends on like 200 things, some of which I don't even know exist yet.

The (Science-)Twitter Universe

Preface

Parts of the work described in this thesis have been performed within the DFG Collaborative Research Center (CRC) 1123 on *Atherosclerosis: Mechanisms and Networks of Novel Therapeutic Targets*. During that time I have been associated with the International Research Training Group (IRTG) 1123. My work within the CRC has not been limited to the project setting, but also involved regular meetings, or scientific conferences, with all members of the CRC. Subsequently, I have been working together with several collaboration partners from the CRC in the course of this doctoral thesis. In addition, I have supervised several practical courses and final theses as part of my teaching activity in the Bioinformatics Bachelor and Master programme at LMU Munich.

As part of the doctoral studies and my activity within the CRC, some work of this doctoral thesis has already been published in peer-reviewed journals. This applies to the atheMir method for text-mining atherosclerosis-relevant miRNA-gene interactions (Chapter 2.2) [144]. Also, the work on sequ-into [142] (Chapter 5.2) and bioGUI [145] (Chapter 3.1) has been published in advance. The cPred method (Chapter 4.2) and the robust differential expression pipeline (`RoDE` , Chapter 4.2) have already been applied in two publications [228, 238]. The papers are joint work and co-authored with collaborators, mostly with scientists contributing experimental data, samples or research questions. My contributions are the bioinformatics research question, the methods and the analysis of the experimental results. The author contributions for these publications have been outlined in the respective chapters and their appendix, and in the published papers according to the reporting standards of the respective scientific journal. The implementation of the `pIMZ` and Aorta3D frameworks has been started within the Bachelor thesis [232] of Margaritha Olenchuk under my supervision. The research questions and ideas for implementing both frameworks have been designed by me. The description of methods within these frameworks, which originate from this bachelor thesis, are included here for completeness and are indicated as being part of Olenchuk's bachelor thesis.

There exist two further publications which origin from my Master thesis [143] or cover work not contained in this thesis [259]. These do not contribute, with regard to content, to this doctoral thesis.

I can't be as confident about computer science as I can about biology. Biology easily has 500 years of exciting problems to work on. It's at that level.

Donald E. Knuth

1 Introduction

Major progress in the life sciences was made in recent years due to the introduction of new experimental techniques [300], new sequencing techniques [31, 165] and more multi-omics [147] data acquisition. Multi-omics data acquisition refers to the measurement of multiple omics data, e.g. the proteome and transcriptome, of the same sample. The branches of science known informally as *omics* are various disciplines in biology whose names end in the suffix -omics, such as genomics, proteomics, metabolomics, and glycomics[1]. Generating high-throughput data, be it sequencing-based, or in the area of proteomics, becomes a routine task. More multi-omics datasets are generated, which capture a biomedical sample through multiple different *omics* technologies simultaneously. Such multi-omics data analyses, in which data from multiple *omics* techniques are combined, are one example of integrative data analyses.

About 15 years ago, microarrays were routinely used for monitoring gene expression. Nowadays, microarrays are used with a decreasing frequency. Instead, they are replaced by sequencing-based techniques (Figures 1.2 and A.11), like it can be seen in the number of stored experiments in National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO) [22, 82],. The analysis of microarray data has already been challenging from a computer scientific and bioinformatics point-of-view, because several considerations for the analysis of large-scale data, but particularly also the interpretation, were required [78]. Microarrays are outdated nowadays, but with the number of high-throughput sequencing experiments per year increasing, the demand for, and the challenges of (sequencing) data analyses are still rising [113, 244, 321].

Commonly, scientific findings are presented to the community in the form of peer-

[1] see https://en.wikipedia.org/wiki/Omics

reviewed articles. More than 30 million abstracts are currently listed in NCBI PubMed[2], the primary literature index in the biomedical domain, each containing valuable information. Extracting knowledge from these texts, such that machine-interpretable data is generated, is within the domain of text mining. However, information extraction is challenging: the sheer amount of text requires efficient methods, and the biomedical setting, in which authors write about complicated relations in maybe even more complicated contexts[3], requires a specific understanding of the text. Hence, challenges, which need to be overcome, do not only relate to the pure finding of information, but also to dealing with the characteristics of biomedical literature, like the mixing of mathematical terms in text, and the usage of many ambiguous abbreviations. Overcoming these problems, biomedical literature can be a rewarding resource which must be exploited to aggregate existing knowledge.

Besides these bioinformatics challenges, the sequencing-based experimental data are also computationally interesting. Microarray data were usually in the hundred megabyte range per sample. Modern sequencing experiments easily produce several gigabytes of (compressed) data per sample, and require more compute-intensive steps, mainly in the pre-processing stages, than were needed for microarray data, such as the mapping of reads to the genome [321]. With sequencing becoming increasingly common in laboratories, the amount of data to process and interpret still rises. The thereby induced problems do not restrict to pure bioinformatics tasks, but to computer scientific problems in general: large data storages, which are easy to access [206], and efficient algorithms for processing on modern many-core or GPU architectures are needed [4, 252].

Due to the increase of high-throughput data from sequencing experiments, new computational methods for processing these data are developed, while many non-bioinformaticians already perform their own data analysis right after data acquisition. Meanwhile, bioinformatics has become intrinsic to almost every life science research project [17]. Bioinformatics evolved into a broad discipline and touches the areas of computer science, biology and data science. Problems no longer only belong to only one of these areas: bioinformatics is highly interdisciplinary. An important, but less frequently handled topic is the accessibility of data and software [271, 299]. There are many attempts in the biological and biomedical data domain to incorporate Findable, Accessible, Interoperable and Reusable (FAIR) data [328]. These principles are also increasingly applied to software [150]. Still, many methods and analyses do not adhere to these principles, making it even harder for non-computer experts to access computational methods. This is discussed as reproducibility crisis within the scientific community, because the problem is not limited to bioinformatics or computer science [20].

Considering that there are several resources of information in bioinformatics, be it experimental data, literature, or databases, combining all these resources can be of high relevance, importance and reward. This is what the bioinformatics domain of *integrative data analysis* is about. This thesis was started in the context of the collaborative research centre (CRC) 1123 on atherosclerosis. Much knowledge about relevant processes in the

[2]https://www.ncbi.nlm.nih.gov/pubmed/

[3]Context here refers to the experimental-, disease-, perturbation-, etc. conditions of an analysis.

development of this disease has already been gained [89, 91, 100, 331]. However, in order to identify new mechanisms in the genesis and progression of this disease, the use of already existing information is beneficial and motivates the use of integrative techniques within this area of a complex human disease. In the course of this thesis the COVID-19 pandemic emerged, which is caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Hence, the scientific community was massively interested in understanding the effects of COVID-19. Some methods developed in this thesis contribute to understanding this disease.

In Figure 1.1 a general bioinformatics analysis workflow of an integrative data analysis for transcriptomic data is shown. This workflow can generally be structured into two stages: a pre-processing stage, which primarily makes data sources available, and a *down-stream* analysis, which builds upon these data to derive specific results, e.g. using integrative analyses. A typical bioinformatics workflow starts with finding relevant data sources from which data can be extracted, such that user-friendly and interoperable algorithms and methods can be applied to these data. The obtained results are then *reported, visualized* and used for *data integration and knowledge discovery* tasks.

This work contributes to the scientific community by addressing each step of a general data analysis workflow: from finding data and exploiting it, making new technologies available for analysis, and presenting new methods for specific analyses. The structure of the thesis reflects the steps of an integrative data analysis (Figure 1.1). Chapter 2 presents methods for identifying miRNA-gene interactions using text mining and for using these data to interpret disease-relevant datasets. Chapter 3 is about accessibility and computability in bioinformatics, two aspects of the *FAIR* principles. In Chapter 4, two new data sources, single-cell RNA-seq (scRNA-seq) and MALDI-TOF-based imaging mass-spectrometry (IMS), are introduced, together with methods using data from these new technologies. Chapter 5 picks up the topic of accessibility but transfers it to a novel sequencing technology and applies it to a new complex human disease, the coronavirus disease 2019 (COVID-19). Finally, Chapter 6 addresses multi-modal and integrative data analysis. Within the following paragraphs, the topics presented in this thesis are brought in line with the typical data analysis workflow (Figure 1.1), which serves as a guide through this thesis.

Data Sources and Information Extraction

Literature Scientific literature is an important resource for any integrative analysis. The advantage of literature is that virtually any scientific finding is published in text. However, in order to extract this information, multiple steps are required.

At the beginning of any literature mining task stands the acquisition of the texts in a machine-readable format. However, due to the highly commercialized nature of scientific publishing, most full text documents are only available as PDF or HTML behind a pay-wall. Thus, methods for text extraction from PDFs are explored in Chapter 2.1. Furthermore, it is explored how well the structure of articles can be retrieved, and whether full texts contain more named concepts than abstracts only.

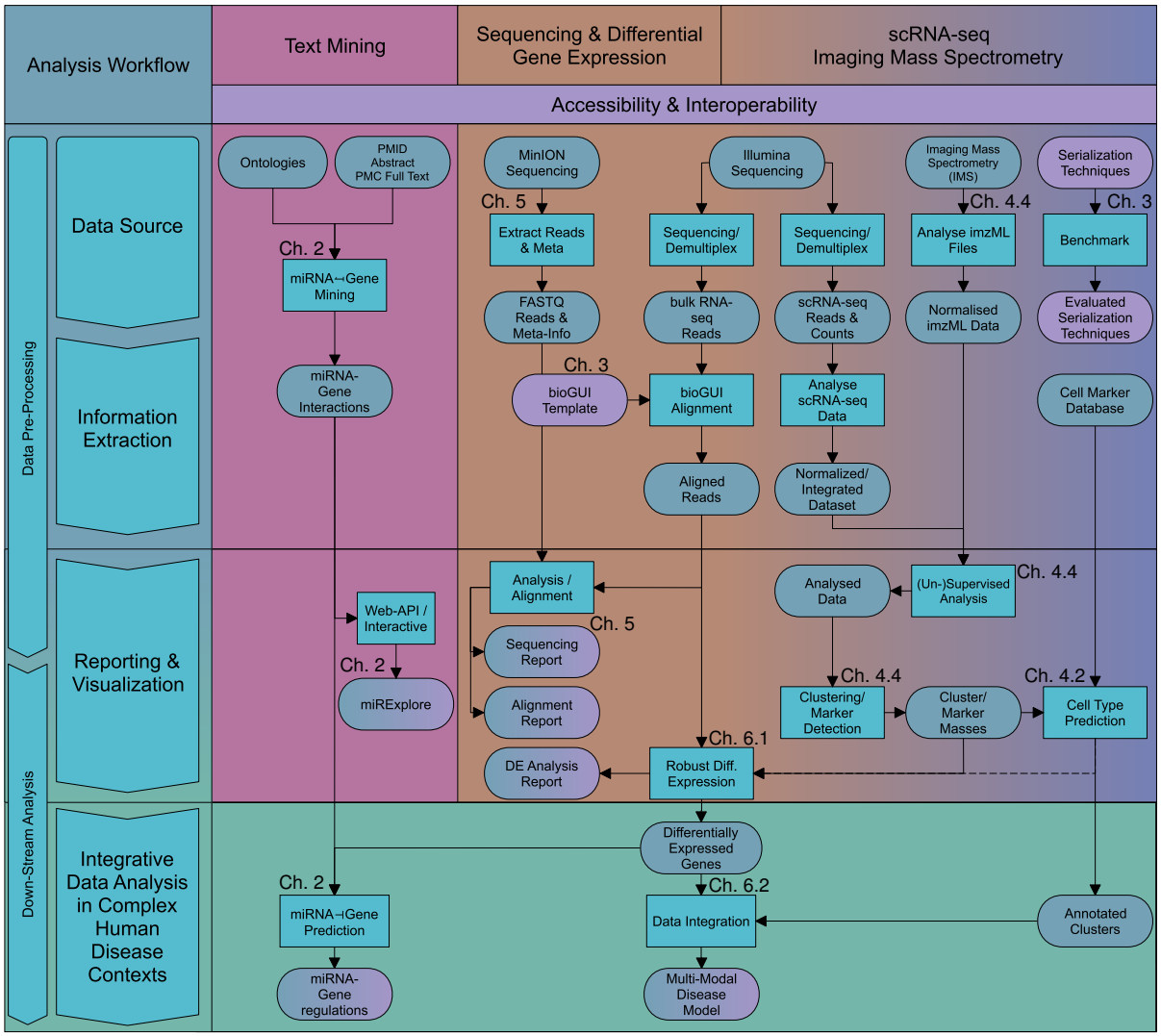


Figure 1.1: **Overview of the topics addressed in this thesis** *Information Extraction* methods are applied to texts, proteomics, genomics and transcriptomics data. *Accessibility and Interoperability* aspects for using such methods are discussed in the respective topic. Tools for the analysis and interpretation of extracted data are addressed under the topic of *Reporting and Visualization*. Finally, results are used for multi-modal analyses in the *Data Integration and Knowledge Discovery* step. All steps combined result in an *Integrative Data Analysis in Complex Human Disease Contexts*.

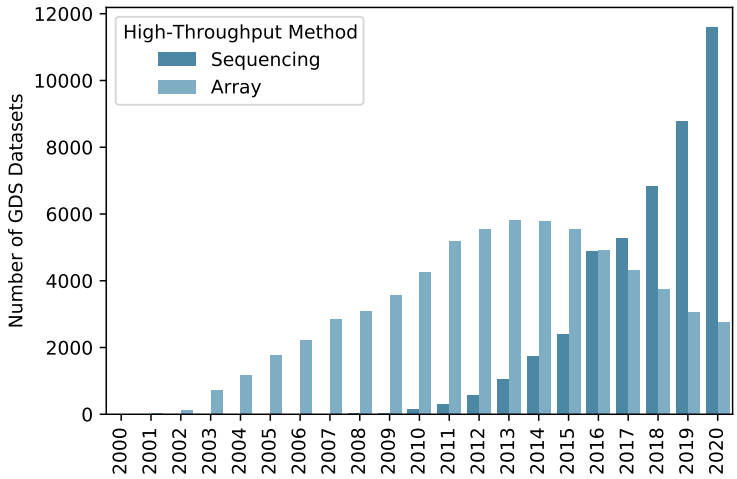


Figure 1.2: **Number of expression profiling experiments in GEO per year and per method**. Until 2013 the number of expression profiling by array experiments rose. By 2017, there are more sequencing experiments (for expression profiling) yearly deposited in NCBI GEO than microarray experiments. A still ongoing growth in sequencing experiments can be noticed.

In Chapter 2.2 an initial version of the anticipated text mining application for finding miRNA-gene interactions is evaluated and applied to the context of atherosclerosis. The advantages of extracting information from literature are described, namely retrieving a quite complete and comprehensive overview of the already existing knowledge.

Finally, Chapter 2.3 presents the miRExplore framework, which retrieves miRNA-gene interactions from biomedical literature. This framework is evaluated on a public benchmark, on which it shows better precision and recall than existing state-of-the-art software, and serves as one main data source for the miRNA-related integrative analyses presented in this thesis. Moreover, the miRExplore framework also touches topics of the reporting and visualization step, and proposes an integrative method to infer miRNA activity based on differential gene expression data. With this method it is possible to identify both interesting miRNAs and their gene interactions in a context-sensitive manner.

Sequencing Data Besides biomedical literature, sequencing data are an important ingredient for any bioinformatics analysis. Depending on the analysis and the task, several sequencing platforms are suitable. In this work the main focus for expression data will lie on the Illumina short read sequencing platform, a next-generation sequencing (NGS) technology. Using the 10X Genomics libraries[4], this technology is often used for scRNA-seq. In Chapter 4.1 and 4.2 methods related to the evaluation of scRNA-seq data are presented, while Chapter 6.1 deals with the evaluation of bulk RNA-sequencing (RNA-seq)

[4]https://www.10xgenomics.com/products/single-cell-gene-expression

experiments using the robust Differential Expression (DE) analysis pipeline RoDE, which can be combined with the text mining resource miRExplore for the prediction of active miRNA regulators (Chapter 2.3).

Third-generation sequencing (TGS) platforms are more suitable for an analysis in the domain of genomics, e.g. for genome assemblies (in remote places) [51] or meta-genomics [263], but also finds areas of application in neurosurgery [235] or pathogen detection [133]. The Oxford Nanopore Technologies MinION sequencing platform is an example of a TGS platform. Chapter 5 describes methods to assess TGS sequencing runs using the MinION platform, to access the acquired data and to use these data to detect specific sequences, while sequencing, with a focus on the usability of the software. This work is discussed at the example of the publicly available dataset of transcriptomic reads from a SARS-CoV-2-infected green monkey.

Proteomics / Imaging Mass Spectrometry For the final integrative analysis of a multi-modal atherosclerosis model, imaging mass spectrometry data is used. This experimental technique measures proteomics data spatially resolved and therefore allows interesting insights into the composition of tissue. In Chapter 4.4 methods for accessing this data type and performing several analyses are presented. A particular focus is set on FAIR software principles and usability, but also on comparative analyses using notebook-technology[5]. With the presented framework arteries with little and large atherosclerotic plaques were analysed, and the results are in good agreement with existing literature.

Accessibility, Interoperability, Reporting and Visualization

Recently it was found that bioinformatics web resources become unavailable at alarming rates soon after initial publication (with 20% of web resources being unavailable 4 years after publication) [152]. The topics of accessibility and interoperability are originally located in the domain of computer science, but become of increasing importance within bioinformatics. Among such topics are FAIR data and methods [150, 206, 299, 328]. Chapter 3.1 introduces the bioGUI framework which empowers non-computer experts to perform FAIR and repeatable analyses, while promoting the usability of bioinformatics tools. The second part of this chapter (Chapter 3.2) introduces tsxCount, which benchmarks the employability of hardware transactional memory for the bioinformatics problem of *k-mer* counting. With these results it becomes possible to choose the most suitable serialization technique for shared-memory parallel applications in a comparable biomedical setting.

Reporting and visualization often follows data extraction tasks. Tools for the extraction of MinION sequencing data, as well as IMS, also have integrated reporting and visualization features implemented. These are discussed in their respective chapters (Chapters 4, 5 and 6).

[5] E.g. Jupyter notebooks https://jupyter.org/ or R Markdown https://rmarkdown.rstudio.com/

Data Analysis

Differential Expression Analysis In the context of this thesis, DE analyses are employed to scRNA-seq data (Chapter 4.2), proteomic data (Chapter 4.4) and to bulk RNA-seq data as part of the robust DE analysis pipeline `RoDE` (Chapter 6.1). With a DE analysis, the key differences between two biological conditions can be determined. Not only the difference is quantified (as logarithmic fold-change (logFC)), but also the certainty for the difference being a true difference between the conditions is assessed (using (adjusted) p-values or q-values, depending on the method). Regularly such DE analyses are performed using one specific way of applying methods to the data, motivating a robust view on such analyses.

Creating new insights by using different data sources is the aim of any integrative analysis. The analysis of transcriptomic sequencing data is often an integrative task, since multiple resources are used during the analysis. This particularly applies to set enrichment analysis after a DE analysis, for instance. The presented DE analysis pipeline (Chapter 6.1) also allows to use the miRExplore database (see Chapter 2.3) and the therein described integrative miRNA regulatory prediction. The `RoDE` pipeline improves the generalizability of analyses by using external DE analyses for robust combination. This robust DE pipeline visualizes and reports the findings from several pre-processed inputs.

Integrative Analysis In Chapter 6.2 the Aorta3D framework for the multi-modal analysis of multiple data sources is presented. This framework allows using RNA-seq, scRNA-seq, microscopy and IMS data and combines these into a 3D model of the analysed tissue, e.g. an atherosclerotic aorta or artery. The Aorta3D framework serves as a spatial and 3D-index for atherosclerosis relevant data, with a unique 3D user interface.

The usage of text mining results for the prediction of miRNA-gene interactions in (robustly) analysed RNA-seq experiments, the prediction of cell types from scRNA-seq or IMS experiments, and the combination of such results in a 3D index are important building blocks, which are required for state-of-the-art data integration and knowledge discovery tasks (Figure 1.1), particularly in complex human disease contexts. With the current trends towards new measurement techniques using NGS and TGS, the ability to perform quality checks on such data, and to visualize the findings such that they can be assessed directly, is an important first step before an in-depth analysis can start. Only after this, further steps in the analysis should be performed, such as the creation of a fully integrative model of atherosclerotic tissue. The methods presented in this thesis make new data sources accessible, extract information from these data and allow a comparative view on the data, with the aim to make the respective complex disease contexts more understandable.

1.1 Current Trends in Bioinformatics

In contrast to microarray techniques, deoxyribonucleic acid (DNA) sequencing techniques and their application to transcriptomics allow answering multiple questions (Figures 1.2,

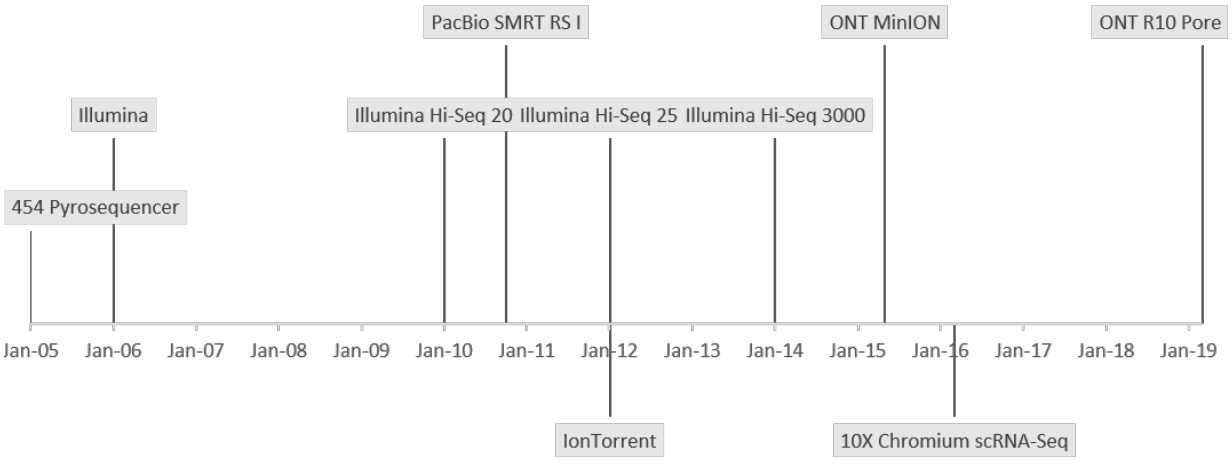


Figure 1.3: **Timeline of sequencing technology milestones** New sequencing techniques are published with increasing speed. Each release improves accuracy, cost per base, and resolution. Higher resolutions allow the analysis of complete transcriptomes and genomes, or closer looks into the regulation within (single) cells.

1.3) with just one experiment, e.g. using gene expression, alternative splicing [321] or gene fusion [90] analyses. Using microarrays it was only possible to measure the expression or presence of pre-defined sequences, where messenger RNAs (mRNAs) attached directly to short sequences, so called oligos. These arrays were designed before the experiment, and only sequences (e.g. mRNAs) which were considered at that time could be measured. Using RNA-seq technology it is now possible to measure sequences and thereby the mRNA levels directly, while also being able to identify new isoforms of genes, or changes in the base-pair composition of a gene. Moreover, using the more explorative approach of sequencing, novel features such as gene fusions [90] or alternative splicing [254] can be detected without first having to design the array accordingly. By increasing the number of sequenced reads[6], even low abundant isoforms can be detected. These advances mainly stem from short-read sequencing, or NGS, which is continually getting cheaper[7]. Nowadays, the costs for sequencing a whole human genome at 30x coverage are as low as $300 for a commercially available test[8].

However, with an increasing amount of data being generated, and new analysis methods emerging continuously, new problems arise. The re-analysis of published findings is often hampered. Many datasets are not available, but even if they were, the used methods are not published, not working and possibly also not accurately described. This was described as reproducibility crisis [20]. Nowadays increasing effort is put into making research data

[6]Also referred to sequencing depth.

[7]https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

[8]The 30x coverage package can be bought for $300 at Nebula Genomics Inc., https://portal.nebula.org/cart/nebula-30x, May 2021.

and software FAIR [150, 328]: findable, accessible, interoperable and reusable. For data, this is in most cases achieved by uploading the (raw) data to repositories such as NCBI's GEO [22, 82], SRA [3], or EBI's ArrayExpress [16] and ENA[173]. Services like FigShare[9], Zenodo[10] or MendeleyData[11] also allow providing data in an unstructured way. These services become increasingly popular, even though therein deposited data are not as findable as with GEO or SRA, for instance. At the software side, many bioinformatics analyses use standardized tools, hence a brief description of what was done seems to be sufficient for many authors. But not for all software all old versions are accessible, or are still usable on current computers. Providing containerized software, or environments with all required software, thus would improve both the accessibility and repeatability of analyses. However, the provisioning of such containers would also require substantial amounts of storage.

Even though it does not solve the above mentioned problems regarding software versioning, the chance to share analyses in the form of (interactive) notebooks should be pursued. This would allow for anyone to reproduce or repeat the analysis, given all software tools are accessible. Even though workflow systems can be seen critically, e.g. for not being flexible enough to react to non-standard analyses, they provide a repeatable and reproducible way of sharing data analyses. Regarding newly developed methods, it recently became popular to publish software (together with its source code) in large repositories such as GitHub, GitLab or BitBucket. However, this does not guarantee that the software is FAIR: quite frequently available software does not run or aborts with errors [200, 292]. These problems are tackled at various levels in this work. In order to perform an analysis, the required software must be installable and runnable on a computer system. Frequently, software only runs on Linux. With the *bioGUI* project not only the installation of required software gets a point-and-click endeavour, but most importantly, Linux software becomes easily usable on the Microsoft Windows platform by making software use the 'Windows Subsystem for Linux', completely managed by *bioGUI* and therefore hidden to the UNIX-inexperienced user. But FAIR data is not the only problem. Quite frequently, data is analysed using multiple tools. The one with the best outcome, or the tool which is easiest to use, is picked and its results are reported in scientific articles. A combination of multiple tools, applied to the same raw data, in order to get a consensus set of DE genes, is rarely performed. Such a robust DE analysis, including down-stream analyses like set enrichments, can be performed using the robust DE pipeline `RoDE` presented in Chapter 6.1. This pipeline also compares the down-stream results of the different analyses.

New experimental techniques are developed, frequently with new and better opportunities in understanding the biological behaviour. However, all of these induce new problems, such as the missing data problem in scRNA-seq analysis [122]. But, new technologies also solve common problems of previous generation technology. For example, with RNA-seq it was common to only have few replicates, because a larger amount of input-ribonucleic acid (RNA) was needed. Newer techniques like scRNA-seq do not require as much RNA,

[9] https://figshare.com/
[10] https://zenodo.org/
[11] https://data.mendeley.com/

enabling even measurements per cell. Such methods induce new problems, like missing data, low coverage or low sequencing depth, which require specific analysis methods. For low-replicate experiments thorough statistical models were needed. With many replicates, like single cells, the t-test becomes reasonably robust again [164] and receives new appreciation. Particularly in case of scRNA-seq analysis, most authors [183, 324, 329] are interested in the cell-composition of their sample: which cell type is present, and which fraction of all cells does it make up? These questions are answered using the results of the cell type prediction method cPred in Chapter 4.2.

The drive experienced in wet labs through sequencing technology however is not limited to transcriptomics, but is also felt in genomics or even metagenomics. Sequencing has become a routine task. However, the efficient storage and analysis of these data is a bottleneck [50, 233, 268]. On the one hand, efficient algorithms being capable of handling the massive amount of data are required. On the other hand, existing software must be easily available and usable. Even non-computer affine scientists need to use this software when performing their data analysis. A user-friendly execution of the required software is enabled through *bioGUI* (Chapter 3.1). Particularly in the fields of genomics, long-read sequencing has become very popular, due to its easy application to even small samples. The release of the Oxford Nanopore MinION sequencing device, no larger than a 2.5" SSD drive, has revolutionized sequencing: orbital microbes have been sequenced on the International Space Station [51], and with this small device, sequencing has become possible virtually everywhere. Combining this with an easy library preparation, MinION sequencing is quite popular in genomics. It is thus not surprising that it became a commonly used platform in the COVID-19 Genomics Consortium in the UK [306]. Reads originating from this device are bioinformatically of interest: basecalling, the transformation of the raw signal into nucleotide information, is an application of machine learning. But even beyond that, specific read aligners are required due to the technique's error rate. Due to these circumstances, an in-depth analysis of sequencing data is required after each run and before downstream analyses take place. A full stack analysis framework, from sequencing analysis, over alignment and differential analysis is provided by the poreSTAT package (Chapter 5.1). In contrast to other quality control frameworks, poreSTAT creates interactive figures, which can be thoroughly explored, and where annotations for single data points can be provided. A further advantage of the Oxford Nanopore sequencing technique is that the sequencing process can be paused or even stopped. After replacing the sample, sequencing can be continued on the same chip with the new sample (likely at lower fidelity, though). In case of a contaminated sample, being able to read out the data almost instantly also enables the early detection of unwanted sequences. This can help to improve sequencing results, by aborting the run, preserving the sequencing chip, and continuing with an improved sample. *sequ-into* is a tool to quickly align reads against a user-defined set of sequences, and evaluate which reads can be either marked on- or off-target sequences (Chapter 5.2, [142]). It therefore detects sample contaminations and allows an early decision on whether to continue the sequencing run, or to redo the library preparation.

Genome assembly is a common task which takes data from such long-read sequencing experiments as input. Many assembly methods make use of short k-long nucleotide fragments

of the reads, so called *k-mer*s. The resulting *k-mer* profiles are then used by the respective methods for evaluation. With the tsxCount (Chapter 3.2) project, several serialization paradigms for parallel memory access have been evaluated in the context of *k-mer* counting.

Not only the area of genomics and transcriptomics is driven by advances in technology, like sequencing, for instance. Proteomics, the subdomain about protein related research, also receives new opportunities due to better technology. One of the trends, which was first observed in proteomics, is to include spatial information into the analysis of samples: MALDI IMS [262, 297]. Raw information, mass spectra, are collected for a whole area, where for each measurement point (pixel) a mass spectrum (intensity value for each recorded m/z value[12]) is available. This trend was also recently transferred to scRNA-seq, which was awarded *Method of the year 2020* by Nature Methods [84]. With additional spatial information, not only questions regarding which masses are differentially expressed can be answered, but also where. Where are similar expression patterns on the tissue? This information can then be correlated with imaging techniques, for instance. Similar to scRNA-seq analysis, the key questions here are: which areas show a similar protein prevalence, which masses define these areas and which cell types or tissues are at that location. Current analysis tools are either commercial software and closed-source, use the R programming language or do not answer above questions. The `pIMZ` framework developed in this thesis allows a full stack analysis to these regards (Chapter 4.4), also adhering to FAIR software guidelines.

This availability of new analysis techniques also drives bioinformatics, particularly in the area of integrative bioinformatics. Here, several of these new techniques, and further resources, are combined, where suitable. In recent years, several new ideas and techniques were applied in so-called multi-omics analyses. Due to the sheer amount of data, and the wish to perform analyses in an unbiased way, these methods mainly rely on unsupervised machine-learning methods [213], and are frequently applied on scRNA-seq analyses [294]. The multi-omics factor analysis (MOFA) method [14] is one of the first in a row of such methods. For multiple samples, multiple factors are calculated, which are then used for decomposition of the samples, thereby allowing an automated inspection of the key differences between the samples. There are further methods which rely on canonical correlation analysis [304], like the DIABLO method [283]. Other integrative techniques, e.g. for the identification of cancer signatures, rely on network-based methods, and try to derive sets of edges as predictive marker [157]. Most work is performed on machine-learning techniques. The multi-omics late integration (MOLI) [278] contrasts MOFA by performing all predictions not on extracted features, but on the encoding subnetworks, thereby discretizing the feature factors as late as possible. The evaluation of such methods is hard, because no ground truth is known [207]. Yet, using small and large datasets, as well as totally unrelated ones, it is possible to see at least whether these differences are detected by such methods [207]. All these methods do not make use of specific prior knowledge — they are not context-sensitive. In combination with the text mining-based miRNA-gene interaction database (Chapter 2.2 and 2.3), a context-sensitive network analysis method for

[12] m/z is the mass-to-charge ratio [70, Chapter 11.7.1]

predicting actively regulating miRNAs from differential mRNA expression data is developed (Chapter 2.3). Using such a context-driven approach has the advantage, that the predictions are funded on existing knowledge, which can easily be checked for relevance. Reported results hence base on peer-reviewed findings. This approach, in contrast to data-driven machine-learning approaches, is more explicit in its predictions, and allows the backtracing of justifications.

The results obtained from sequencing experiments, be it from RNA-seq via `RoDE`, scRNA-seq, or IMS data, can contain important information about a disease: buried in huge lists of DE gene lists. In order to make such data available to researchers in a structured manner, e.g. filterable by disease progression or involved cell types, the Aorta3D project (Chapter 6.2) serves as 3D-index for such experiments.

Nowadays, many trends in bioinformatics are technology driven. New measurement techniques allow the combination of more information, from an increasing number of single entities. Hence, the domain of integrative bioinformatics, which combines many of these resources, can help to interpret this huge amount of data. This work selects, improves and applies methods and applications, which are relevant for context-sensitive data analysis in complex human disease contexts. In order to achieve this, several new data sources are made available, or existing ones are further exploited. In the end, the context in which observations are made, is a good estimator for which results can be expected, and is a good filter for irrelevant observations.

The following Chapters will present the methods developed in this thesis along a typical analysis workflow (Figure 1.1). Several data sources are made available: text mining (Chapter 2), scRNA-seq (Chapter 4.2), IMS (Chapter 4.4) and TGS (Chapter 5 and 5.2). These resources are integrated in multiple analyses, like the miRNA-gene regulatory prediction (Chapter 2.3), the bulk RNA-seq DE analysis pipeline `RoDE` (Chapter 6.1) and the 3D-spatial index Aorta3D (Chapter 6.2). With bioGUI and tsxCount (Chapter 3) a focus on accessibility and interoperability is set.

1.2 Relevant Data and Data Resources

Within this work multiple data sources are used to analyse specific disease-related conditions. Here, the most common data sources and their relevance in the context of this thesis are presented. Commonly used file formats are described in the Appendix (Chapter A.1).

Text Mining Most results from biological experiments are not published in an easily machine-interpretable format. Experimental data from low-throughput experiments, like (immuno-)blots, histologic staining, fluorescence markers or PCR results, are extremely important in the reporting of scientific findings. However, such data is usually only presented in figures and discussed in the text of scientific literature [59, 258]. Hence, the extraction of information from unstructured text, like journal articles, is of high importance in order to collect biomedical knowledge. Such collected data can then be combined with high-throughput data, which is usually analysed by bioinformaticians. The steps required for

information extraction from texts is in the domain of text mining.

MEDLINE is the prevailing online source for abstracts of biomedical research publications. It is maintained by the National Library of Medicine (NLM) and the National Institute of Health (NIH) and hosted at the National Center of Biotechnology Information (NCBI) [3]. MEDLINE contains over 30 million article abstracts which also include additional meta-information (July 2020). It is freely accessible and can be searched through the PubMed interface, but can also be downloaded in yearly releases in a standardized XML format (journal article tag suite (JATS)[13]). This allows the extraction of the unstructured text together with further meta-information like publication date, authors or journal. Open-Access articles are becoming increasingly popular. Thus, the number of full text articles available in the PubMed Central (PMC) database[14] continues to increase, with currently more than 600 000 open-access articles for download. These articles are partly available as JATS-formatted files. Non-open-access articles can be accessed by researchers from the publishers' websites and can be downloaded as PDF to a local hard drive, e.g. through citation managers, like Elsevier's Mendeley[15]. Considering that an abstract is considerably shorter than a full text, which may even be available as PDF, the computational effort needed to process full texts is much higher than just abstracts.

Even after obtaining the article texts, several problems need to be overcome. For instance, multiple researchers may use different terms, different vocabulary, for expressing the same meaning. Moreover, the name of an entity may change over the years: For instance, *C-C Motif Chemokine Ligand 2* has multiple accepted long form names listed as synonyms, among these are *Monocyte Chemotactic Protein 1* or *Small-Inducible Cytokine A2* [36]. For any computational analysis of such unstructured text it is important to map all these possibilities onto a common identifier, a common language which can be further used. Such common identifiers are, for instance, the official gene symbols (even though these change over time, too [40]). More generally, ontologies are used in several disciplines to structure information in a hierarchical way. Conceptually such ontologies are directed acyclic graphs (DAGs). A node within an ontology defines a term of the ontology, e.g. a specific concept. Each term usually has at least an ID and name. Any edge within this DAG then defines a special relationship between the connected concepts. Frequently further descriptive synonyms are annotated for each node, too, which can then be used for text mining. Using an ontology [111] has two pragmatic advantages: to facilitate communication between people and organizations and to improve interoperability between systems. Examples of ontology based controlled vocabularies are the Gene Ontology [48, 105], Disease Ontology [270] and the National Cancer Institute Thesaurus (NCIT) [117], which are used for text mining context-sensitive miRNA-gene interactions (Chapter 2).

Illumina Sequencing Analysis Using Illumina short read sequencing, which is the dominant NGS technique, the workflow from reads to expression values involves multiple

[13]https://www.niso.org/publications/z3996-2019-jats
[14]https://www.ncbi.nlm.nih.gov/pmc/
[15]https://www.mendeley.com/

steps [63]. The analyst starts with reads which are presented in a FASTQ-formatted file. For each read, which is of fixed length, also quality information are available. This quality information should give clues how confident the machine was that the respective base was measured correctly. However, particularly in downstream steps, this information is rarely used. After performing a quality control on these reads (e.g., quality per read position, or overrepresented substrings/*k-mer*s), the reads are aligned to reference sequences. These are usually reference genomes, like the GRCh38 (human) or GRCm38 (mouse)[16] assemblies with given genome annotation files [93]. This alignment or mapping process can take several hours depending on the tool used and the amount of reads. Particularly for transcriptomics, additionally the transcriptomic features (provided as genome annotation files) are required, as mRNA is spliced (which becomes visible as gaps on the genome) and some aligners need this information to be passed on. Popular choices for alignment tools are STAR [75] or HISAT2 [153]. The transcriptomic reference finally is required to determine how many reads have been found per gene or transcript during the counting step. Popular tools for this task are HTSeq [11] or featureCounts [184]. With the number of reads per gene or transcript the actual DE analysis between several conditions can be performed. This analysis is required for downstream analyses, such as gene set enrichments, etc. There exist several tools for the purpose of finding DE genes. Some of the more popular ones are DESeq2 [10], limma [255] or the EmpiRe-framework [7].

MinION Sequencing Analysis The Oxford Nanopore Technologies MinION sequencing device is one example of TGS. Compared to short read analyses, both the molecular process of MinION sequencing, and the analysis of data obtained from it, is fundamentally different [196]. The output from MinION sequencing is not a FASTQ file, but a *fast5* file (a HDF5-file following specific rules). Within this file, the raw measurements of the device are recorded. These raw measurements are interpreted on the fly (LiveBasecalling), or afterwards using several commercial or open-source methods. This has the advantage that these files can be reinterpreted at later times, when the algorithms used for the interpretation of the raw signal and transformation into sequences (basecalling) made significant improvements. The default basecalling programs for MinION reads are albacore or guppy[17]. Most methods in this domain rely on machine learning approaches [196].

After basecalling, the process of analysing MinION data is very similar to Illumina data. Most frequently MinION sequencing data are used for tasks in genomics, less frequently in transcriptomics. This is due to the lower throughput of the MinION sequencing, resulting in fewer reads per transcript or gene. DE analysis becomes problematic with only a few reads per gene because the established statistical models, which were originally designed for many short reads, can not be applied. With small counts the determination of fold changes is too imprecise. Two common choices for aligning TGS reads to reference sequences are Minimap2 [179] and graphmap [286].

[16] Both GRCh38 and GRCm38 assemblies are released by the Genome Reference Consortium https://www.ncbi.nlm.nih.gov/grc.

[17] Both only available to registered Oxford Nanopore Technologies (ONT) customers.

Imaging Mass Spectrometry Imaging mass-spectrometry (IMS) [297] probably is the most special measurement technique used in the course of this work. Instead of measuring mRNA, ion masses are measured in proteomics. While this concept is physically quite complex, the setup of IMS makes this slightly easier: for each detected mass (which here corresponds to the m/z value) an intensity is assigned. A complete overview of possible proteomic techniques is given by [230]. The general process during the bioinformatics analysis of IMS data is the following: from the machine vendor's software the measured spectra are exported into a standardized data format, imzML [269]. In the data used in this thesis, the spectra's masses were already binned. This means, all spectra have intensities for a common set of m/z values. After normalization of the binned spectra, they can be compared among the same measurement, but also among different measurements, if these were normalized with a suitable strategy. Since the single spectra were recorded with spatial information, there are many spectra available within one measurement. The position, where a spectrum has been recorded is also referred to as the pixel coordinate. The spectra can be clustered, only considering spectral similarity for all masses, or by making use of prior knowledge, e.g. about the structure which is observed, or relevant masses. All spectra from pixels of the same group can serve as replicates in order to identify common features of a group, such as marker masses after differential expression analysis. Using a reference, the masses can then be assigned to specific proteins, so that they can be related to transcriptomic analyses.

Biological Gene Sets The Gene Ontology (GO) [48, 105] probably provides the most commonly used controlled vocabulary and ontology — but is also annotated with gene symbols for each concept, and hence can be used to derive corresponding gene sets. Its ontology describes virtually any biologically relevant concept in its three domains: cellular component, molecular function and biological process. The deeper the term is within the ontology (that means the more distant from its root), the more specific the concepts within the ontology become.

A recent collection of (hierarchical) pathways is given by the Reactome database [136]. Within this database pathways consisting of multiple genes are listed by topic, for instance, immune system, signal transduction, or more common, disease. Within Reactome, pathways describe chemical reactions, or protein-protein interactions. To arrange the Reactome pathways within their cellular location GO terms are used.

Common analyses involving any biological gene sets in the context of DE analysis involve checks whether the collection of differential genes is enriched in one of the concepts of either GO or Reactome, for instance. In order to assess this enrichment, over-representation analyses using the hypergeometric test are performed [101], as well as more involved tests, like the gene set enrichment analysis (GSEA) [295].

miRNA Nomenclature and Resources Protein and gene-symbol nomenclature was in the beginning not fully deterministic. Specific consortia were established to name novel genes and proteins, and to update names and symbols of existing ones. The Human Gene

Name Consortium (HGNC) [36] or Mouse Genome Informatics (MGI) [41] are two of such consortia providing essential resources. With genomics becoming increasingly important in health care the need for standardizing gene naming and providing guidelines for the naming of protein-coding genes, non-coding RNAs or pseudo-genes was emphasized by HGNC [40].

In the early beginning of miRNA research, the nomenclature of the approx. 22nt long miRNAs was rather non-deterministic. This changed with the identification of even species-conserved miRNAs. Nowadays, miRNA nomenclature has become quite structured [110, 162, 163]. miRNAs are numbered sequentially and denoted in the form *hsa-mir-121*, meaning that this is the $121th$ human miRNA (the first three letters denote the organism). Mature miRNAs are written with capital R (miR-121). Each distinct precursor sequence and genomic loci expressing the same mature miRNA sequences are denoted as hsa-mir-121-1 and hsa-mir-121-2. Lettered suffixes identify closely related mature sequences. For example, hsa-miR-121a and hsa-miR-121b would be expressed from precursors hsa-mir-121a and hsa-mir-121b, respectively. Some studies identify two miRNAs from the same predicted precursor. The mature sequence of the dominant product has no extension, while the opposite arm sequence is denoted with an asterisk. More formally, the $5p$ suffix denotes a miRNA originating from the 5' arm of the precursor, while $3p$ denotes the one from the 3' arm (e.g. miR-142-5p or miR-142-3p).

There are several resources for miRNA research. Most important is the miRBase Sequence database and Registry [110, 161, 162, 163]. From there all (known) miRNAs are accessible, by name or sequence. It also provides a portal with relevant literature for specific miRNAs. Even though miRBase provides several tools for miRNA genomics, its focus is neither on text mining nor on providing integrative analyses. miRBase is a resource for listing miRNAs. In contrast, miRTarBase is a resource which focuses on integrative analysis and identifying miRNA-gene interactions from high-throughput sequencing experiments or small-scale experiments. However, miRNA-gene interaction predictions are left out. Databases containing predicted miRNA-gene interactions are TargetScan [2], miRWalk [80] or DIANA-TarBase [149], the latter also including experimentally validated interactions.

I define UNIX as 30 definitions of regular expressions living under one roof.

Donald E. Knuth

2

Text Mining Disease Specific Interactions

Most scientific findings are published in scientific articles, which contain almost endless amounts of knowledge. While regular articles shed new light into specific processes, scientific reviews try to give an overview over specific fields of interest. Delivering a good review requires a lot of effort, even for domain experts, and almost always a restrictive selection, assessment and compilation of the established scientific facts from existing literature. On the other hand, a review is maybe not as helpful for everyone, because the selection is too distant from the needs of the individual researcher, who wants to employ a review to set own research into context with the state-of-the-art.

In this chapter it is investigated whether an automated approach can make use of already published facts, and thus contribute to writing a compelling and up-to-date review more easily, allowing the user to obtain all established facts related to the reviewed field of interest. This is evaluated in the setting of miRNA-gene interactions. Many miRNAs have been already identified, and it is known that these small, 21-25-nt long small RNAs play a crucial role in many diseases and processes [26, 55, 139, 310, 338]. Thus, inherently, in this field long lists of findings have to be reviewed, and any of these findings can yield a relevant hypothesis for the question investigated by the researcher or measured via some high-throughput technique. Often researchers want to check whether their current finding is indeed new and surprising, and how it adds new facts to the established knowledge of the field. Thus, it is important to provide convenient access to facts, hypotheses, and the associated evidence in an as complete as possible manner.

Another challenge in any field of study is the dynamics of scientific progress. Findings are added continuously to the published literature. Performing incremental updates of reviews and the associated evidence becomes critical in order to achieve the above-mentioned goals. Otherwise, review articles quickly become outdated.

miRNAs have often been identified as important post-transcriptional regulators [211,

219] associated to various stages of complex human diseases, like diabetes[86] or cardiovascular diseases[140, 310]. For atherosclerosis research, in particular, a number of relevant miRNAs have been identified and brought into context[77, 135]. It has been found that miRNAs modulate the function of endothelial cells, smooth muscle cells and macrophages by controlling the expression levels of chemokines [118].

miRNAs are important regulators within complex human diseases, and therefore miRNA-gene interactions are a useful base to tackle a context-sensitive analysis (with the focus on finding new regulations). Predicting miRNA-gene interactions, which have already been confirmed in a broader context, from gene expression data would just be one interesting application for a context-sensitive miRNA-gene interaction database. The generated hypotheses could be used for further testing in animal models.

In this chapter methods for text mining miRNA-gene interactions with a context-sensitive approach will be designed. Along the analysis workflow (Figure 1.1), text mining uses specific data sources, but also methods for data extraction. Such methods are formulated and applied for the purpose of named-entity recognition (NER) in Chapter 2.1. A new strategy for the automatic creation of synonyms for ontology-derived synonyms (*inflating*) is proposed, as well as a benchmark of structured text extraction from PDF files. Using these resources, and a newly developed method for miRNA-gene interaction extraction, an independent resource for integrative analyses is constructed: a database of literature extracted miRNA-gene interactions (Chapters 2.2 and 2.3). Methods for miRNA-gene interaction extraction are established and benchmarked against other state-of-the-art methods. Using only the literature evidences, the knowledge of the field, e.g. regarding specific miRNA-gene interactions, can be explored. The new *Timeline* feature allows a direct comparison of literature evidences within the same selected context, e.g. for specific miRNA-gene interactions in a specific disease. And also the novel, data integrative prediction of actively regulating miRNAs from DE results, is an interesting application of the miRExplore database.

This work builds upon the in-house developed NER application *syngrep*[1], which was used in favour of other possible strategies.

2.1 Methods for Ontology-based Research in Structure-Extracted Documents (MORSED)

In this section, a text mining framework for mining miRNA-gene target relations will be set up. The used NER approach requires user-supplied synonym lists to identify words of interest (entities). Hence, it is important to know whether these synonym lists are already sufficient, or whether an automatic extension of the synonyms improves NER results. In order to improve the coverage of known miRNA-gene interactions, not only PubMed may be used, but also PMC full texts. Thus, another interesting question is, whether full texts improve the coverage of contextual information.

[1]Csaba, Gergely. Personal Communication. 2015-2020.

These questions are answered with *MORSED: Methods for Ontology-based Research in Structure-Extracted Documents*.

Scientific knowledge is usually published in text form and made available as PDF documents. This way, researchers are able to describe their work and share it with the community. From a bioinformatics perspective, there are two disadvantages of this approach that are addressed here: first, there is an increasing amount of publications added each year, making it hard to keep track of newly published resources and requiring automatic tools. Second, text publications are structured in itself, but automatic retrieval of the text (e.g. from PDFs) including its structure is hard.

Here, the MORSED framework describing methods for ontology-based research in structure-extracted documents is realized. With MORSED it is possible to search within a library of PDF documents for those documents which contain concepts the user is interested in. PDF documents can be automatically and structure-aware extracted. The therein identified sections can be classified to common sections, such as abstract, introduction, methods, results, discussion or references. This process is benchmarked regarding the ability to extract structures (document sections) as well as the ability to identify above classified sections from the extracted text. A method to increase (inflate) the number of synonyms for given ontology terms is proposed and evaluated. It is further assessed, whether the full text can support context-sensitive text mining in the sense that specific full text sections add relevant information, particularly information not present in the abstract. MORSED can be applied within an app and a web-based platform to perform the text extraction and structured-NER for custom PDFs in a user-friendly way.

Both, structured text extraction and section classification work accurately. The latter is correct in 99% of all cases. The remaining errors are a result of an incorrect PDF extraction, which makes it impossible to classify sections. Using inflated synonym lists increases the number of found terms per document up to 100%. This strategy proved to be context-sensitive, such that some contexts profit more from this strategy than others. Using the presented methods, it could be shown, that, depending on the input ontology, many synonyms are only found in other sections than the abstract. Particularly if the interest lies in methods and techniques used to measure biomedical entities, such information is often only contained in the methods section, and in no other sections.

With these methods a structure-aware text extraction can be performed, and sections are classified almost error-free. Moreover, it is shown that inflating the synonym lists improves the detection of synonyms with the applied NER method. Finally, by using the full text information, many synonyms, which are not contained in the abstracts, could be found. This makes the point, that whenever possible, text mining should be performed on the full texts.

Science evolves continuously, and the amount of released publications increases month over month. Keeping track of this never-ending flow of new information is hard. While there are numerous literature recommendation systems available, making the content of literature available through certain concepts (e.g. terms from ontologies) is not currently integrated in such tools. In order to tackle this hurdle, a system is proposed, which first takes a user's current literature, extracts its content in a structured form and finally searches

for occurrences of previously registered synonyms from, for instance, ontology terms. It can be observed that specific sections contain information which are not included in the abstract and thereby deliver a more complete picture of the document and its information.

Introduction

There is a vast amount of bio-medical literature currently available. The most famous resource for bio-medical literature is PubMed, which currently contains more than 30mio documents. But also PMC is constantly growing, with currently providing more than 650 000 open-access full texts. In addition, pre-prints are becoming more and more popular in life sciences and contain valuable information, even though they are not (yet) peer-reviewed. A lot of information is contained within these texts, and many structured collections (e.g. ontologies) of synonyms for summarizing this information exist. PubMed uses MeSH terms and NCBI Thesaurus [117] for this task, for instance.

These concepts cover all areas of interest. However, most life scientists probably only have few areas they are interested in for their daily research. Moreover, the structure or the concepts which are used by PubMed may not reflect the context domain these experts are in: the ontology might not be fine-grained enough, terms might be connected in different ways, or common abbreviations within the field might not be included.

Several tools exist, which allow a search of terms over all available open-access articles, like Textpresso Central [221] for general keyword queries. Recently, it has been shown that full-text articles have a gain in information, especially for determining protein-protein or gene-disease interactions [325].

The MORSED framework contains methods for named-entity recognition of (custom) ontologies in custom PDF-libraries in a structure-aware manner. The concept of MORSED differs from existing approaches, because it is not as specific as, for instance, ARMOUR, but allows more detailed searches than Textpresso Central, since the user can specify the ontology by himself, as well as the literature corpus to search in. In addition, MORSED focuses on full-text PDFs and extracts these structurally, using a customized[2] version of CERMINE[307]. Furthermore, it is analysed whether certain document sections are more valuable for specific ontologies than others.

Materials and Methods

Structured Text Extraction One of the necessities for NER is the provision of machine-readable text. For PDF files this implies the usage of specific structured text extraction methods, because not only the text must be recognized within a PDF file, which is already a challenge, but also section titles must be recognized, such that a relation between section titles and sentences can be established. In order to accomplish this task a modified version of CERMINE [307] is used. The modification is required to highlight the location of a specific term within the PDF. Additionally, the text extraction framework should avoid

[2]https://github.com/mjoppich/CERMINE

hyphenation, newlines and multiple white spaces wherever possible. Specific to scientific papers, abstract texts need to be processed in the same way as the main body. CERMINE has been parallelized in order to process large libraries of PDF files. Batches of PDF files are processed by distinct threads. The CERMINE package has been modified to cover these additional requirements and is available from `https://github.com/mjoppich/CERMINE`.

Synonym Localization Since a general NER approach is used, finding the named entities in a text fast is crucial for success. Thus, the syngrep[3]-implemented Aho-Corasick approach is used, leaving this task in linear time-complexity (length of strings, length of searched text and number of output matches). The used algorithm is able to search for inline abbreviation, but will only return the longest hit per position. Furthermore, small hits (e.g. abbreviations, gene symbols) are only accepted if they are a perfect match (including capitalization).

Synonym Creation Due to the NER approach, having a correct and complete list of synonyms is crucial. This is ensured by the synonym-inflation-strategy which was developed for this framework.

First certain synonyms are added (*replace* stage). For instance the term *oxygen level* would be duplicated into *oxygen concentration*. However, this step is context-sensitive. The user may submit more of such replacement tuples at run-time. In the second step spelling variants are created (*spelling* stage)[4]. British English terms are translated into American English terms and vice versa. For instance, the British English word *analyser* is converted into *analyzer*. In the third stage (*reverseform* stage) active descriptions are transformed into passive ones, and vice versa. For instance a *novelty test* is transformed into a *test of novelty*. Further default target words are *level, weight, interval, number, amount, ...*. Again the user may submit more of such target words. The fourth stage (*case* stage) fixes capitalization issues for the NER approach. In general, synonyms with capitalized first character are added. The fifth stage (*scoped* stage) changes suffixes of words. For instance, *-thalamo* is changed to *-thalamic*. Again the user may add further word pairs. Additionally, this stage modifies words such that concatenated expressions are merged together. For instance, *IL-1 secretion* is transformed into *IL1 secretion*. If a word ends on *test* or *task*, the version with a delimiting - is retained. In the final stage, plural versions of the words are added with a simple heuristic (*plural* stage). An *s* is appended if the word does not end on an *s*. Words ending on *y* are transformed with *-ies*.

Structured Extraction One of the main advantages of the MORSED approach is the ability to search by specific document sections. Hence, it is important to benchmark how well the section classification works. In this framework, sections are classified as abstract,

[3]Csaba, Gergely. Personal Communication. 2015-2020.

[4]Using the *Word list of UK and US spelling variants* by Words Worldwide Limited https://web.archive.org/web/20101223230739/http://www.wordsworldwide.co.uk/docs/Words-Worldwide-Word-list-UK-US-2009.doc.

introduction, methods, results, discussion or reference. All other or unclassifiable sentences for a document are assigned to the *other* section.

The classification of a document section to a text section is made using a keyword-match approach. Whenever a section title matches one of the synonyms for a section, the similarity is calculated using the token-sort- and partial-ratio (from the fuzzywuzzy[5] python package). A minimum similarity score of 75% must be achieved in order to directly classify a section. In a post-processing step, two further sanity checks are performed. If several sections with confidences below 90% are found between two matched sections, these lower-confidence sections are assigned to the previous 100% match. This bases on the observation that subsections sometimes are identified as main-section. However, this is only allowed if less than 5 unmatched subsections lie in-between the matched ones, or less than 8,000 characters (which resembles a large introduction). The second check is applied only to the first section. If this section is classified as *other* and is longer than 1,000 characters, it is classified as *introduction*. This is motivated by the observation, that frequently the introduction is not correctly identified as a section.

These rules for classification have been benchmarked on one training set (19 random documents from the pmc_athero collection), which was also used to fine-tune the above rules. An additional test set of 15 random documents from the same collection was then used to benchmark the final classification process.

The gold standard has been prepared in a way, such that for each sentence and each found section title the matching section classification was annotated (manually, 1 annotator). It was assessed whether the section title has been extracted in a way such that it is possible to assign a valid classification. If, for instance, a section was not recognized at all during text extraction, the classifier can not assign the correct section category based on the section name. This, allows to benchmark the structured text extraction process as well: any incorrectly extracted section is an error. Observed errors have been classified as *sec_title* error, if the section has been identified correctly, but no title was determined. An error is classified as *sec_spell*, if the section has been identified correctly, but the title contains a spelling error. More severe are *sec_struct* errors, which occur if a section structure is not identified correctly, e.g. a section is missed. The *subsec* error occurs, if a subsection is identified as main section. Finally, an *abstract* error is identified, if the abstract has not been detected correctly. The available section names and corresponding keywords are listed in Table 2.1.

Data The main evaluation is performed on a set of 1 609 PubMed Central full-text PDFs (pmc_athero). The texts of these PDFs were extracted using the modified structured text extraction method based on CERMINE. The resulting sentences were text mined on synonyms derived from the Gene Ontology (GO) [15, 47], the Measurement Method Ontology (MMO) [280, 284] and the Evidence & Conclusion Ontology (ECO) [56].

A second set of documents containing 2 978 documents (related to animal welfare) originated from an anticipated collaboration (allxml). These texts have been structurally

[5]https://github.com/seatgeek/fuzzywuzzy

Table 2.1: **MORSED section names and keywords** used for the classification of section titles.

Section	Keywords
abstract	abstract, title (assigned from text extraction)
introduction	introduction
methods	methods, materials, methodology, methodological
results	results
discussion	conclusion, future, discussion
references	references, literature
other	other

extracted using a different workflow, which is unknown to the author. This set has been text mined for concepts from the Animal Trait Ontology for Livestock (ATOL) [107].

Code Availability The source code for this project, including a web-service and the pdfAnnotate application, is available from `https://github.com/mjoppich/pdfAnnotate`. Some methods are already integrated into miRExplore directly (Chapter 2.3) and are available from `https://github.com/mjoppich/miRExplore`.

Results and Discussion

Named-entity recognition approach One of the specifics of any NER approach compared to machine-learning or natural language processing (NLP) approaches is that only known synonyms will be found. Therefore, it can not only be relied on the given synonyms in the ontology. The therein contained synonyms may not reflect all possible ways of expressing a respective term in a text. These have to be extended, or inflated, in order to become findable. The advantage of the chosen explicit NER approach is, that if a term is used in any text, and a matching synonym exists, it is likely found. Moreover, in contrast to machine-learning approaches, no large sets of training data are required. While there exist pre-trained machine-learning models, which contain the most common ontologies [171], this is not the case for custom or less-known ontologies.

Finally, the here developed synonym creation method increases the amount of synonyms approximately by factor 8 (Table 2.2), depending on the context/ontology (here: Gene Ontology[47]), where the fourth (case) and the sixth (plurals) stage add most new synonyms. The plural versions double the number of synonyms. Whether this has any impact can be evaluated on the number of found concepts per document (Figure 2.1, for ECO). It can be noticed that for most documents the number of actual hits is increased (about 50% on average). Because these mainly consist of plurals or capitalization, the number of uniquely hit synonyms is important. An increase in the number of uniquely hit synonyms (Figure 2.1b) of up to 100% is observed using the inflated synonyms. Having performed this comparison on multiple synonym lists it was noticed that the paraphrasing from the

Table 2.2: **Number of synonyms per stage for the Gene Ontology** [47]. The first stages only add few synonyms, while the latter, more general stages, add most synonyms.

Stage	Synonym Count	Rel. Increase (from original)	Rel. Increase
original	120,264	1.00	1.00
replace	120,412	1.00	1.00
spelling	130,980	1.09	1.09
reverseform	132,886	1.10	1.01
case	298,480	2.48	2.25
scoped	534,480	4.44	1.79
plurals	967,811	8.05	1.81
overall	967,811	8.05	-

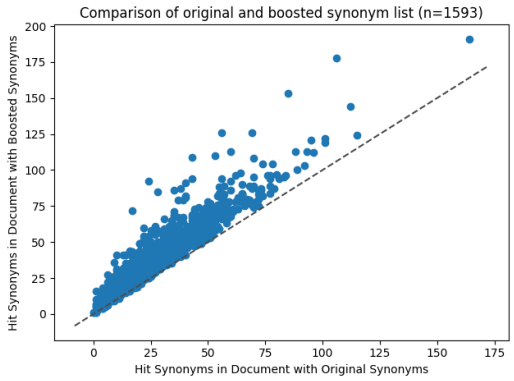
reverseform step was most useful. It was found that this step has less impact, for instance, with the Gene Ontology [48, 105] or Disease Ontology [270] synonyms. This is because the target words, e.g. test, level, temperature, etc., are manually curated lists of words. These must possibly be adapted to the given synonyms for best results.

It arises the question, which *stage* improves the finding of synonyms most. Two cases are distinguished: First those synonyms that were already found without inflation, and second, those synonyms, which are found due to synonym inflation. The found ECO synonyms in the pmc_athero dataset are discussed. Using the original synonyms, there are a total of 44 947 hits, of which most are direct hits. This is reflected by the fact that these hits are mostly (98%) explained by either the *case*, *plurals* or *original* category. Using the inflated version, additional 14 726 hits can be found. Most new hits are found using the *case* stage (67%) and the *plurals* stage (27%). Another 6% were found by synonyms added in the *scoped* stage. The *reverseform* and *spelling* stage add the fewest new hits. For GO synonyms in the pmc_athero test-set the inflation does not have that much effect. Only few hits have been identified additionally (2 443). Apart from the *case* (55.8%) and *plural* forms (32.3%), the *reverseform* (4.6%), *scoped* (4.3%) and *spelling* (2.9%) versions identify some additional hits.

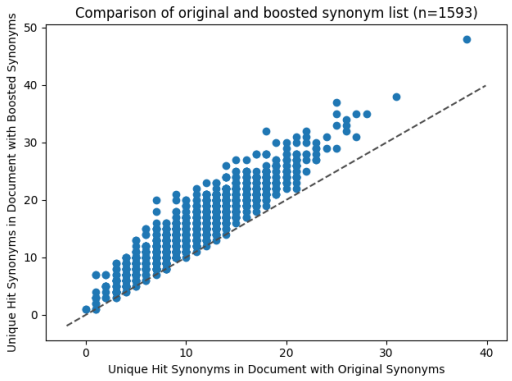
These results suggest that for the chosen NER approach, case-sensitivity and plural forms are most important. In fact, the case-sensitivity is of particular importance, since the chosen NER approach knows *common language* words which are only matched if the word was matched perfectly. This becomes a problem, if the desired terms are common words, like *Cancer* or *Heart disease* (Figure A.12).

From the above ECO results (Figure 2.1), as well as the ATOL result (Figure A.15), it can be seen that inflating the synonym lists can increase the number of found synonyms up to 100%. This effect, however, depends on the synonym list and thus is context-sensitive.

Section dependant analysis For many text mining applications only the abstracts of articles are considered. However, the question arises, whether this is sufficient or whether



(a) Number of NER hits per document with the original and inflated ECO synonyms.



(b) Number of unique NER hits per document with the original and inflated ECO synonyms.

Figure 2.1: **Comparison of using the original and inflated ECO synonyms** The amount of actual hits is increased by about 50%, the amount of uniquely hit synonyms can be increased by about 100%.

the full texts contain more information. Moreover, it is interesting to know whether the full text contains more recognized synonyms than the abstract and how much of these synonyms are unique for the full-text. Additionally, it should be explored whether the conclusion part of a scientific text contains the same synonyms as the methods and results part. If the overlap is large, then a combined view would be appropriate, otherwise, the conclusion part should be left out for analysis since this part might be too speculative.

As a first step, it needs to be checked whether the structure-aware extraction worked, and whether these results allow a classification to the identified sections. In Figure 2.2 it can be seen that most sentences can be classified to a section. Similarly, it should be explored how many complete documents can be found, and which parts are frequently missing. From the $1,607$ documents in total more than $1,119$ documents were complete (all sentences could be assigned to the 6 sections). About 252 documents have one section missing. This is sufficient for the next analyses.

Evaluating the section classification Using the test set two tasks can be assessed. It allows determining how well the structured text extraction worked, on a structural level. This then allows identifying and rating errors regarding the classification of the sections.

The structured text extraction does not work optimally. About half of all documents contain errors, in both the train and test dataset (Table 2.3). Taking a closer look at those errors, it can be noted that only few sections per document are affected. Less than 10% of all sections show unrecoverable errors. For instance, a missing section title can in most cases be imputed by its location. Likewise, the main section of a subsection can frequently be imputed during section classification. The extracted structure of the text, hence, is not perfect, but useful for further work.

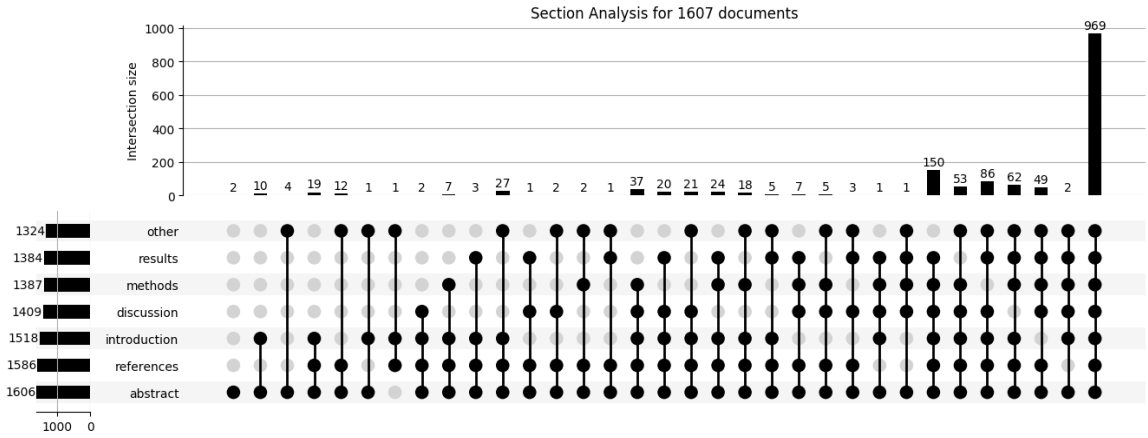


Figure 2.2: **PMC-athero-documents with recognized sections.** Most documents consist of all sections. Sometimes one section can not be determined. The number of documents with two and more not recognized sections is comparably small.

Despite the short-comings of the text extraction, it is interesting to see how well the section classification works in general. This analysis is evaluated on two levels, at the sentence and at the section level. These results are summarized in Table 2.4. The extraction and classification is able to assign nearly all sentences correctly, if there is no structural extraction error prohibiting this. For all documents and sections which are extracted correctly, the section classification works well.

Which section is important? In the further analysis only documents with at least 10 different synonyms in total are considered. This leads to a higher confidence with any relative evaluation.

It was investigated whether the original synonyms behave differently than the inflated ones. Therefore, it was checked which fraction of all synonyms in one document can be explained by synonyms found in the methods and abstract part for each document (Figure 2.3, ECO). While there is only little difference between the original and inflated versions of the ECO synonyms, it can be seen that the inflated synonyms improve the methods fraction more than the abstract fraction. The methods fraction distribution stays more or less constant, but the abstract fraction distribution is pushed towards lower fractions. The abstract, hence, is less important than, for instance, the methods section. This can be observed for the allxml dataset with ATOL (Figure A.16), too.

In a further analysis the section-unique synonyms distribution was looked at. These are synonyms which are only found in a specific section (Figure 2.4, ECO). It can be noticed that both ECO and MMO have more unique synonyms in the methods section than any other section. This is understandable, because both contexts assesses the type of measurement, which is less frequently discussed in abstract, introduction, results or discussion.

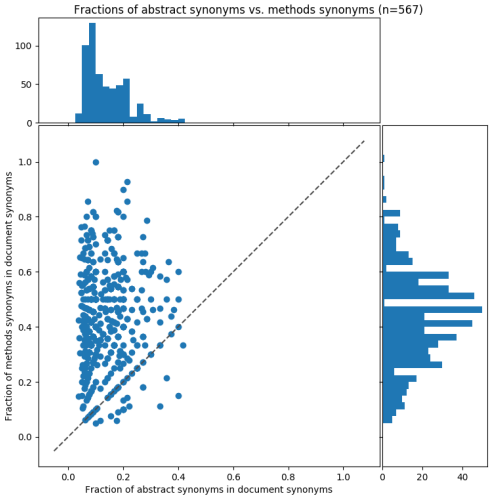
In Figure 2.5 the abstract/methods and discussion/methods sections are compared

Table 2.3: **Number of structure extraction errors per document ID** (DocID) and section pair (DocID, Section) in each dataset. About 15-18% of all (DocID, section) pairs show problems. Errors during structure extraction, which lead to unrecoverable errors in section classification make up 3-7%.

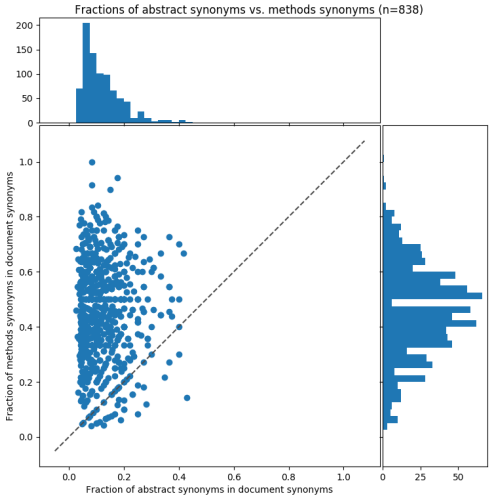
Structure Error	Training Data	Testing Data
Total number of documents	20	13
Total number of documents with structure error	10	8
Total number of structure errors	12	12
Error: Subsection as section title	10	9
Error: Incorrect section structure extracted	4	1
Error: Section without title	2	3
Error: Spelling error in section	1	0
Error: Abstract incorrectly extracted	4	2
Total number of (DocID, Section)	132	90
Total number of erroneous (DocID, Section)	21	16
% affected (DocID, Section)	15.9	17.7
% unrecoverable errors	6.6	3.3

Table 2.4: **Classification results on a per sentence and section basis** Only few errors are made, but some sections are not evaluated because of extraction errors which make a classification impossible.

Count	Training Data	Testing Data
Correct Sentences	6 595	4 035
Incorrect Sentences	1	3
Sentences with Struct Error	109	39
Total Sections	356	239
Correct Sections	356	235
Incorrect Sections	0	2
Sections with Struct Error	13	4

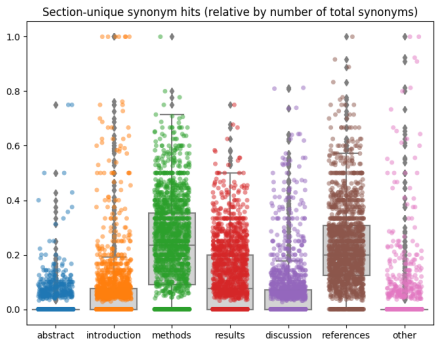


(a) Abstract versus methods synonyms for the ECO context. More synonyms are found in the methods part of a document.

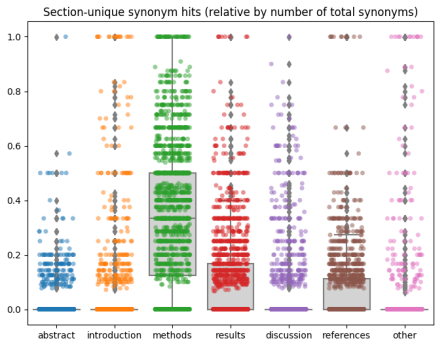


(b) Abstract versus methods synonyms for the inflated ECO context. More synonyms are found in the methods part of a document.

Figure 2.3: **Evaluation of section results (ECO)** In (a) the original ECO context has been evaluated. For many documents the methods synonyms make up more than 50% of all document synonyms. Using the inflated synonyms (b), this effect manifests itself. As a side effect, 271 more documents contain abstract and methods synonyms. The methods part often contains 40%+ of all synonyms, while the abstract synonyms usually contribute less than 30% of all synonyms.

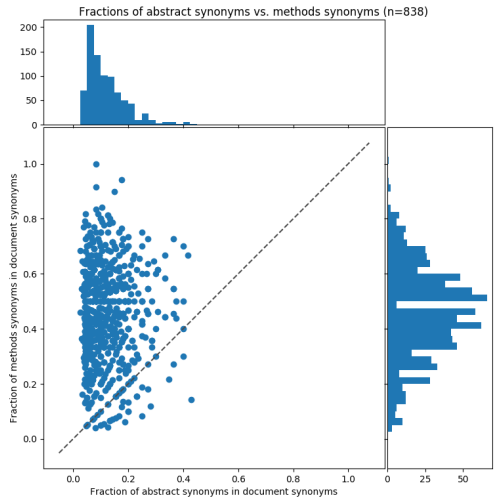


(a) Fraction of found unique synonyms per document and per section for the inflated ECO context.

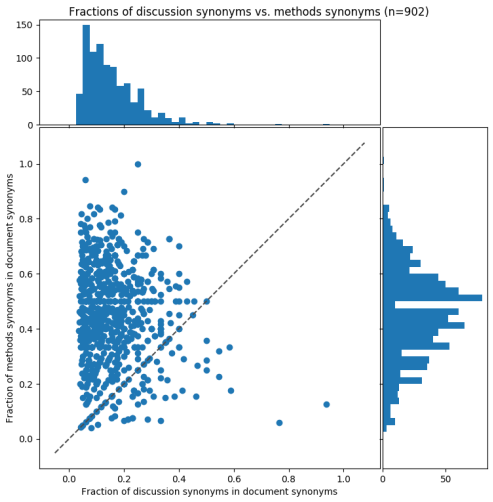


(b) Fraction of found unique synonyms per document and per section for the inflated MMO context.

Figure 2.4: **Per section analysis of ECO context**. In (a) the ECO context has been evaluated. Abstracts have few unique synonyms, while the methods sections have a considerable share in unique synonyms. (b) This effect becomes even stronger in the MMO context which focuses on measurement synonyms.



(a) Abstract versus methods synonyms for the inflated ECO context. More synonyms are found in the methods part of a document.



(b) Discussion versus methods synonyms for the inflated ECO context. The discussion rarely covers 30% or more of all document synonyms. More synonyms are found in the methods part of a document.

Figure 2.5: **Comparison of abstract and discussion synonyms**. Comparison of the abstract (a) and discussion (b) synonyms against the methods synonyms.

regarding their share in the whole document. The synonyms found in abstracts only make up 10-30% of all unique document hits, while the methods hits have a share of about 30%. This also holds for the conclusion. These results suggest that the methods synonyms usually make up at least 30% of all document synonyms, while abstract or conclusion synonyms make up only roughly 30% of all document synonyms, seldom more. From this it can be concluded that only using the abstract is not suitable for the identification of measurement strategies. Again, this depends a lot on the underlying ontology. This observation is made for the ATOL analysis (Figure A.17), too. However, such an effect can not be seen with GO (Figure A.18). Here, the abstract contains more found terms than the methods section, but less than the discussion, for instance. The synonyms from GO describe interpretations of (molecular) results, in contrast to measurement techniques. This explains the observed differences between the GO and ATOL, MMO or ECO results, which contain synonyms to describe techniques.

Taking all this together, synonym inflation improves the NER results well, but depends on the used ontology. Full texts should be used wherever possible: the remaining sections contain further interesting synonyms, which are hidden otherwise, but specific synonyms (e.g. ECO) are mostly not described in the abstract.

Conclusion

The MORSED framework contains methods for ontology-based research in context- and structure-sensitive extracted documents. It was demonstrated that in order to get a full picture of the mentioned synonyms in scientific literature it is essential to include all document parts. However, some sections, e.g. the methods section, are more important depending on the context that is searched for.

It could be shown, that some synonyms are less likely to occur in the most common considered part of a document: the abstract. Instead, it is important to search through the methods section if the interest lies in the applied methods. Likewise, the conclusion is not the place to look for such elements.

Additionally, an easy, rule-based classification of section titles is possible and works considerably well. This approach is only hampered by incomplete or structurally invalid extraction of text from PDFs.

When it comes to a NER approach, the list of synonyms is important. It could be shown, that using the inflation strategy more synonyms can be identified, than without, at least with the employed NER approach. In particular this was useful if the terms to be found contain regular language words, because then exact hits (plural, case-sensitivity) can be used to distinguish potentially false positive hits. But for some contexts also the paraphrasing of terms appears to be useful. However, this approach is highly context-sensitive and must be manually curated for each use-case.

Nonetheless, it can be concluded, that for a complete and comprehensive search in literature, full texts should be favoured — particularly if the *context*, that is surrounding terms from other synonyms, is of interest. It could be shown, that a section-aware search of full texts is important to find and identify all relevant terms. Failing to do so results in the loss of important information and incomplete scientific findings.

The MORSED app (Chapter A.3.1) allows extracting, query and show results from multiple contexts. It is easy to use and can support any life scientist in his/her daily research. The app itself is cross-platform compatible and allows researchers to easily navigate through their downloaded publications. A web-based PDF extraction, section classification and synonym search within the MORSED framework is implemented.

Taken all these findings together, it is evident that full texts provide more results and a better picture than using only the abstract. The advantage of using full texts is context-sensitive. For example, applied (measurement) techniques are less frequently mentioned in abstracts, than in the methods section of full texts. Full texts and section-aware text mining can thus improve finding the setting (or context) a specific document is about. Even contexts, which are used to summarize results, such as the Gene Ontology (GO) or the Disease Ontology (DO) [270], can profit from considering all sections of full texts and the synonym inflation, which is part of MORSED.

2.2 Context-Sensitive Text Mining for miRNA-gene Interactions (atheMir)

Context-sensitive text mining refers to the discovery of text mining results including their respective context, allowing analyses within the specific context. This context might be the mentioned organisms, cell types, diseases or processes. The challenge here is to correctly identify certain cell types, diseases or processes. While there exist ontologies for these data, e.g. Gene Ontology (GO) or the Disease Ontology (DO), the used NER text mining platform relies on specific synonyms, which are efficiently searched for within texts. The text extraction from biomedical resources or PDFs itself defines a further area of engagement, which was addressed in the previous section (Chapter 2.1), and yielded methods for the extraction of text as well as the usage of ontologies for NER.

In a first use-case study the developed text mining algorithm for finding miRNA-gene interactions will be applied on specific literature that has been associated with cardiovascular diseases. This method was a first test of the anticipated context-sensitive text mining approach. As such, the focus here was led on developing functional methods. Besides runtime considerations, this is the reason why only PubMed abstracts were considered at this stage.

With respect to the synonym lists needed for the NER approach, the implemented approach not only uses the entity term from the respective ontology, but also enhances these terms as presented in the previous section (Chapter 2.1). For instance, plural and paraphrased synonym versions are built and used. For example, *temperature difference* might be paraphrased as *difference in temperature*. Synonyms, which either resemble a common word, e.g. *and*, or occur too frequently, are filtered to improve the specificity of the NER approach.

With respect to miRNAs, a miRNA-parsing class has been developed, which allows parsing miRNA string representations and tries to detect respective elements in the systematic representation of miRNAs. Using this representation of a miRNA then allows to automatically generate all required and valid text representations for this miRNA (e.g. *miR-146∗*, *miR-146-3p*). This step enables the use of a NER approach for miRNA-mining.

With the ability to enhance existing synonyms, and to match a string to a specific miRNA, a first version of the anticipated context-sensitive miRNA-gene interaction mining approach has been developed. This version of the interaction mining system has been evaluated on two benchmark sets: existing reviews on miRNAs in atherosclerosis within a specific chemokine context as well as a more general one on endothelial cell inflammation. The regulative networks derived with the context-based approach of atheMir (atherosclerosis miRNAs) were of similar quality compared to expert curated ones, in parts even more complete. This does not devaluate the review's authors, but shows the benefit of the atheMir approach: easy access to context-sensitive data. Using the interactions determined by atheMir, combined with information from causal biological networks [35], new regulative hypotheses could be generated, like the roles of miRNA-124 and miRNA-126 in atherosclerosis [144].

Context-based text mining methods can massively influence and support reviews from

domain experts. While the information in atheMir is certainly not complete, it might be a good starting point for experts writing reviews, but also for researchers investigating highly specific hypotheses in certain contexts. For both use cases, atheMir facilitates easy access to individual and context-sensitive miRNA–gene interactions. Most importantly, it provides supporting evidence for each reported interaction. The presented context-based resource atheMir is a powerful method to explore miRNA-gene interaction hypotheses in atherosclerosis and beyond. The results have been published in [144]. The accepted publication is available as open-access online article `https://doi.org/10.1055/s-0039-1693165`. The author's contributions are listed in Appendix A.3.2.

2.3 miRNA-gene Interaction Mining (miRExplore)

The existing knowledge in miRNA-gene interactions is overwhelming, not only by size, but also by their origin. Various models for predicting miRNA-gene interactions exist [177, 224], and likewise several databases listing miRNA-gene interactions from such predictions — some of which combine predictions with experimentally verified interactions. Such experimentally verified interactions may be manually curated, or automatically derived from matching high-throughput experiments. Another vast number of miRNA-gene interactions has been verified in small-scale experiments, validating the interaction of a specific miRNA-gene combination in a specific experimental setting (context). Such results are likely only described in literature and can only be retrieved from such.

In Chapter 2.2 a miRNA-gene interaction mining approach has already been applied successfully to an atherosclerosis context. Now the focus is set on a broader methodology by mining full texts, improving the miRNA-gene interaction detection and providing integrative methods for using the text mining resource. The presented approach, miRExplore, collects information from public databases, recent literature (PubMed abstracts, PMC open-access full texts) and performs context-sensitive text mining on this literature. Curated sets of synonyms derived from domain specific ontologies for genes, diseases, species, cell-types, experimental contexts, functional classes, and pathways are used. For any found miRNA-gene interaction within a sentence, a sentence's structure, extracted using NLP methods, is used to classify the interaction. Finally, the PubMed abstract's, or the PMC document's, context is added to the found interactions. This enables users to search specifically for miRNA-gene interactions in a context relevant to them. The miRExplore approach is evaluated against the miRNA-gene interaction benchmark developed by Bagewadi et al. [19], and compared to methods presented by them, as well as to miRTex [177].

The miRExplore version improves the atheMir version regarding two points: first, it extends the (cardiovascular) disease context to a general context of all miRNA-gene interactions, including interactions found in full texts. Second, it improves the interaction detection and classification. Using a corpus of curated miRNA-gene relations [19], it was found that some interactions were not identified with the atheMir approach. This often occurred when the sentence structure has not been correctly determined, or because miRNA-enumerations were not correctly resolved. miRExplore resolves these problems and

improves the text mining and miRNA-gene interaction detection to these regards. It makes the found interactions available via a web-resource, for use with further applications by providing a programmatic access, e.g. for integrative data analysis.

miRExplore: A generalized miRNA-gene Interaction Text Mining Framework

More than 790 human and mouse miRNAs are currently known to be involved in diseases. More than 26 000 miRNA-gene interactions are annotated in humans and mice. Most of these interactions are canonical post-transcriptional regulations: miRNAs bind to the mRNAs of transcribed genes and induce their degradation, thereby reducing the gene expression of target genes. Thus, miRNAs are important regulators of complex human diseases.

While there are many databases for miRNA-gene interactions, retrieved from computational predictions or imputed from specific high-throughput data, most interactions are published in text form. Hence, retrieving (experimentally verified) miRNA-gene interactions from literature, abstracts and full texts, is an extremely important and useful task. miRExplore is a framework to mine miRNA-gene interactions, their regulatory direction as well as to analyse these data. Its interaction mining method is evaluated against well established benchmarks, and integrated analysis features are discussed.

The interaction between miRNAs and genes is extremely important in modern biomedical research. It is thought that miRNAs are key regulators in various diseases [137, 144, 239, 264]. As such, these may serve as a novel class of therapeutics [26, 261]. Having a broad understanding about the involved processes, and most importantly, the genes specific miRNAs target, is important. Most findings are still published in text form, and extracting such miRNA-gene interaction from published documents is of high importance in order to gain a complete picture of the miRNA-gene interaction landscape. Here, the miRNA-gene interaction extraction framework miRExplore is presented.

Introduction

miRNAs are small, non-protein coding, RNAs, which can post-transcriptionally regulate genes. Ameres et al. reviewed their generation, assembly and context-sensitive functional aspects [6]. The canonical mechanism works by binding to corresponding miRNA-binding sites on the regulated mRNA, leading to a decay of that mRNA. Thus, miRNAs regulate mRNA expression post-transcriptionally, potentially allowing a fine-tuned decay of specific mRNAs. miRNAs are thought to be highly relevant in specific diseases, or even play an important role in orchestrating these [264]. Because of this, miRNAs are discussed as novel class of therapeutics [26, 261]. Nonetheless, it is well known that miRNAs play a context-sensitive role [88], meaning that it depends on the specific circumstances (disease, cell type, etc.), whether a miRNA is expressed, available, and possibly whether it binds to certain (non-)canonical binding sites.

The importance of miRNAs is already well known in the community. There are several resources to explore the miRNA-gene-regulatory interactions known in many (model) organisms. More than 15 databases exist, which all collect miRNA-gene interactions. However, only one of these databases, miRTex [177], also considers contextual information around these interactions, such as mentioned diseases, processes, etc. While this contextual information is important, it is beneficial to regularly update such information. In 2019 alone, more than 5500 papers related to miRNA-gene interactions have been published and are available in PubMed (Figure A.21). The experimental databases, such as DIANA-TarBase [149] or miRTarBase [58] are regularly updated, but for most text mining resources this is unfortunately not true.

One requirement for text mining miRNA-gene interactions is that the actual text is available in a machine-readable format. Luckily, this is true for many article abstracts and full texts which are available from PubMed or PMC. For articles only available as PDF a suitable text extraction method was already presented in Chapter 2.1. The remaining challenge is the extraction of miRNA-gene interactions from the text. Finding biomedical entities has become a quite robust task, which usually is performed either by NER, machine learning (ML) approaches or more recently by applications of deep learning (which is a specific form of ML).

Here, a new method for miRNA-gene interaction mining is presented, which relies on pre-trained ML models for sentence dependency graph prediction, from which a NER- and rule-based approach detects and classifies miRNA-gene interactions. Comprehensible rules for both the retrieval of miRNA-gene interactions (mention-level) and their putative regulatory direction are deduced. This resource then serves as input for further integrative methods, including Timelines, miRNA regulatory prediction and set-based over-representation analyses.

Existing miRNA-gene databases There exist many miRNA-gene-target-centered databases. A list of such databases is compiled in Table 2.5. These can be generally categorized into *manually curated*, *predictive* or *experimental* databases, judged by their main content. Some formerly highly used databases have become unavailable, others emerged. Most of these databases have in common, that there is no context information available for the contained interactions. Even the single database, miRTex [177], which has this information available in general, is unfortunately not kept up-to-date, does not allow systematic queries by contexts and is not open source. Since miRNAs play a critical, but context-sensitive [88], role in many complex human diseases, knowing the context in which this role was observed is essential.

Materials and methods

Data Availability All source code is available from GitHub `https://github.com/mjoppich/miRExplore`. The database, miRExploreWeb and the API, is accessible from `https://rest.bio.ifi.lmu.de/miRExplore/`. miRExplore adheres to FAIR software principles because it is findable, accessible from the web, and interoperable because it relies on

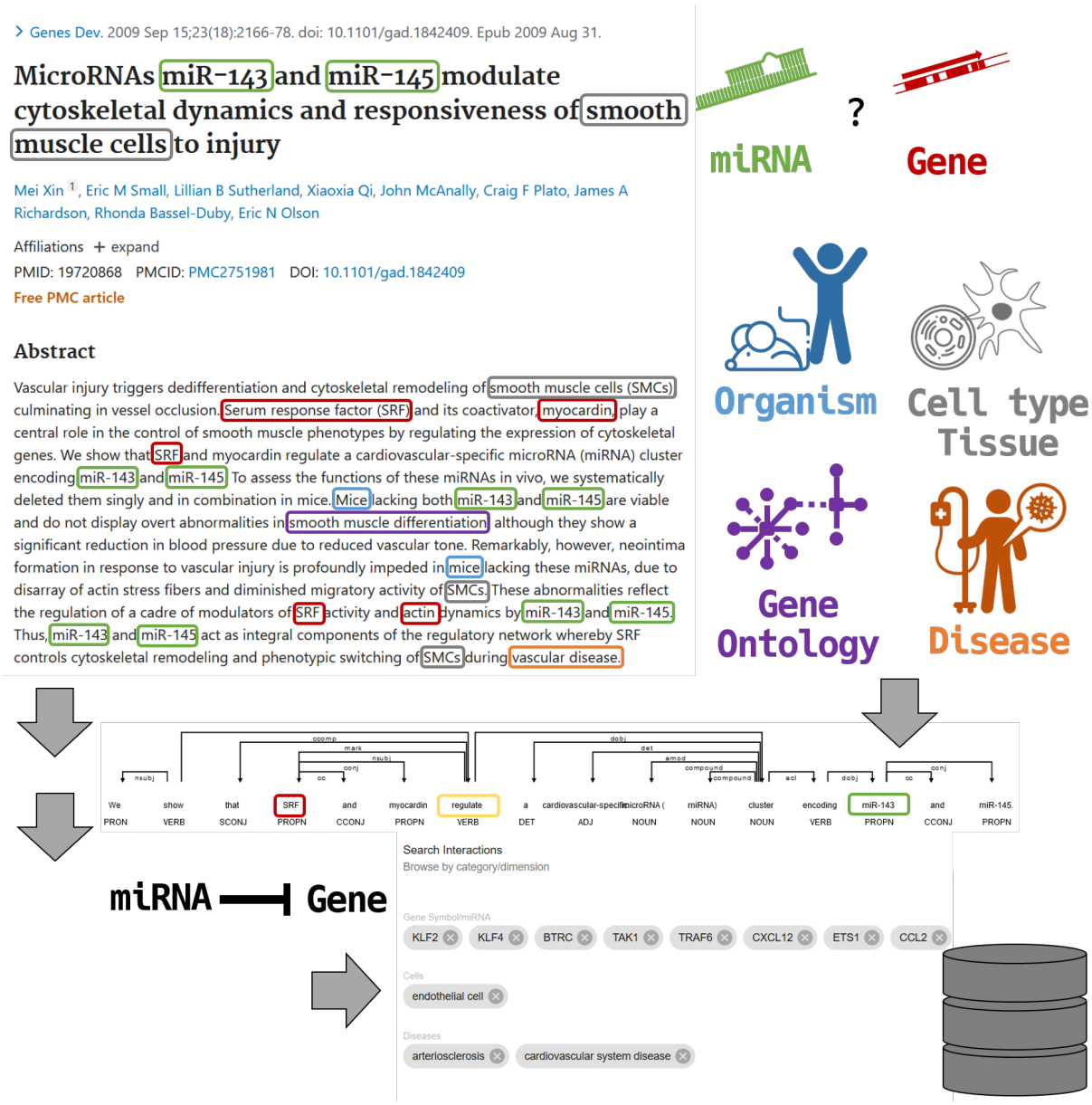


Figure 2.6: **The miRExplore framework** takes abstracts and full texts, identifies entities from several dimensions, extracts miRNA-gene interactions and classifies them. All collected information can then be queried in a database.

Table 2.5: **Existing databases for miRNA-gene interactions**. Databases are annotated by type of origin of interactions, as well as their last update and whether they provide context information.

Database Name	Type	Updated	Context	URL/Link
miRBase [163]	Curated			http://microrna.sanger.ac.uk/
MMIA [225]	Predictions			http://129.79.244.122/$\sim$MMIA/index.html
miRNAMap [126]	Predictions	2008	No	http://mirnamap.mbc.nctu.edu.tw/
microRNA.org [27]	Unavailable			http://www.microrna.org/microrna/home.do
miRDB [332]	Predictions	2020 2015 2008	No	http://mirdb.org/miRDB/
miRGator [57]	Prediction Experimental	2013	No (possible by dataset)	http://mirgator.kobic.re.kr
miRGen TarBase [149]	Prediction Experimental	2019	No	http://www.diana.pcbi.upenn.edu/miRGen.html http://diana.cslab.ece.ntua.gr/tarbase/
Argonaute [276]	Prediction Known	2006	No	http://zmf.umm.uni-heidelberg.de/apps/zmf/argonaute/
miRTarBase [128]	Experimental	2019 2017	No (provides PMIDs)	http://miRTarBase.mbc.nctu.edu.tw/
miR2Disease [138]	Unavailable			http://www.mir2disease.org/
miRecords [333]	Curated	2013	No	http://c1.accurascience.com/miRecords/
TargetScan [2]	Predictions	2015	No	http://www.targetscan.org/vert_72/
miRWalk [80]	Predictions	2018	No	http://mirwalk.umm.uni-heidelberg.de/
mESAdb [151]	Experimental	2011	By Dataset	http://konulab.fen.bilkent.edu.tr/mirna/mirna.php
magia2 [225]	Unavailable			gencomp.bio.unipd.it/magia2
Firefly Discovery Engine	Text-Mining	2020	No	https://www.fireflybio.com/portal/search
miRSel [224]	Text-Mining	2010	No	https://services.bio.ifi.lmu.de:1047/mirsel/
miRTex [177]	Text-Mining	2015	Indirectly	https://research.bioinformatics.udel.edu/miRTex/

miRTex can be queried by keywords, but there is no possibility to access all atherosclerosis relevant interactions programmatically, for instance.

common input formats for texts (JATS-formatted input from NCBI) or ontologies (OBO format). The miRExplore implementation is reusable in the sense that it can be replicated and used in different settings.

Finding named entities For the following miRNA-gene interaction detection it is required that all miRNAs and genes are found in one sentence. Concepts of relevant context-ontologies (e.g. GO) must be found in sentences of the same document. This is achieved using a NER approach. An initial version of miRExplore used syngrep[6] for this task.

More recently, syngrep was reimplemented with python. This ensures FAIR principles, and avoids problems due to invalid text encodings, because the handling of these is more straight-forward in python. This reimplementation is compatible with syngrep regarding the use of input and output formats. Like syngrep, an Aho-Corasick data structure[7] is used to store all synonyms. This enables a fast matching of synonyms within the query text (sentences from PubMed abstracts or PMC full texts). In addition to the requirements of NER, e.g. knowing all possible synonyms, inexact matching should be supported due to, for instance, the use of hyphenation in texts. Inexact matching is not directly supported by the Aho-Corasick data structure. It is thus implemented by transforming the input query such that all desired manipulations, e.g. collapsing white spaces or capitalization (e.g. case-sensitivity), are tested against the synonyms. Unlike the original syngrep implementation, the python-based version does not yet implement rules for the handling of abbreviations. The python implementation is parallelized using a fork-and-join pattern.

Dependency graph resolution The presented two-step approach relies on having accurate dependency graphs for the examined sentences. Such dependency graphs are predicted by specialized NLP frameworks such as spaCy[8]. The dependency graph is the result of the Part-Of-Speech (POS) tagging after parsing and tokenizing a document or sentence. For spaCy, this process is performed using specifically trained models. These models then tag and label each word (or: token) of a sentence (or: document) with (predictions for) several attributes, such as the POS tag and other tokens, on which the token itself depends on. The latter relation can be used to derive a tree structure of the analysed sentence, the dependency graph. A full description of all available features is available online from the spaCy documentation[9]. The tagging and labelling is performed by pre-trained models. Hence, this step is a prediction of attributes for a specific token. Because the final result defines the dependency graph, this process will here be referred to as dependency graph prediction. The effect of the distinct models has been evaluated on the miRExplore method. For a productive setting, AllenAI's scispaCy [227] sci-lg model is best used for dependency graph prediction, in combination with scispaCy's BioNLP model

[6] Csaba, Gergely. Personal communication. 2015-2020.
[7] https://github.com/WojciechMula/pyahocorasick/
[8] https://github.com/explosion/spaCy
[9] https://spacy.io/usage/linguistic-features

for entity classification in the respective rules. spaCy annotates tokens and their relations using the universal dependency annotations described online[10].

Finding micro-RNA-gene interactions Using the conjugation, shortest dependency path (SDP), compartment, context and entity (interaction) rules on a sentence's dependency graph prediction, it is decided whether a miRNA-gene co-occurrence is a valid miRNA-gene interaction (independent of the direction). The following rules are used for the miRNA-gene interaction extraction.

Conjugation Rule (Interaction) The conjugation rule is used to determine whether two entities are in the same conjugation, or not. Given a correct dependency graph, this is generally the case if the two entities (or any of their related words) are connected via a *conj* edge in the dependency graph. For all elements of *conj* edges, all related elements must be considered. Hence, for any word connected by *conj* edges, all dependencies connected by any of the following edge types are followed and collected: *case*, *amod*, *nmod*, *dep*, *appos*, *acl*, *dobj*, *nummod*, *compound*.

However, there are some conjugations which actually form an interaction and must not be rejected. For instance (Figure 2.7), conjugations, like *we observed a direct regulation between miR-124a and Cxcr4*, should be kept. Conjugations with proceeding *between* are kept for this reason. It was observed that there are many interactions mentioned between a miRNA or gene and a miRNA family or gene pathway. These are filtered out by checking that a conjugation may not contain a `[miRNA|gene]` *pathway* mention.

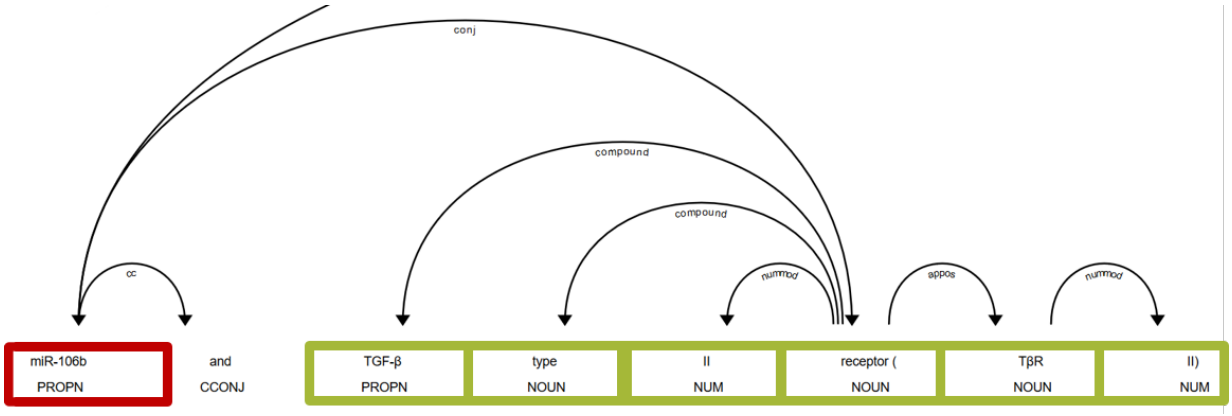


Figure 2.7: **Conjugation Rule Example** *miR-106b* and *TGF-β type II receptor* are in the same conjugation, resulting in no interaction.

SDP Rule (Interaction) The initial miRNA-gene interaction mining framework, atheMir [144], already used a rule concentrating on the intersection between the paths from entities to their respective root elements of the dependency graph. Indeed, using the

[10]https://spacy.io/api/annotation/

(shortest) path between two entities in the dependency graph has already proven to be useful for general relation extraction [249], and has since then been particularly applied to protein-protein interactions [127, 190]. This finding is used for miRExplore. The SDP contains those words, which connect two entities. Hence, it contains the important concepts between the two entities.

It is ensured that the SDP does not include a subject, which is not one of the two observed entities. In the example (Figure 2.8), *miR-17/92* and *Shh* are not in a valid relation, because *Shh* is a child of the actual subject *N-myc*. Thus, the interaction here is between *Shh* and *N-myc*, and *N-myc* and the miRNAs.

In order to exclude pathway related interactions no `<verb> <noun> pathway` is allowed within the SDP.

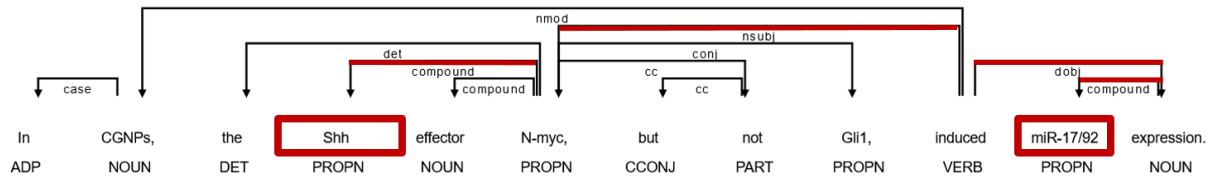


Figure 2.8: **SDP rule Example** *miR-17/92* and *Shh* are not in a valid relation, because *Shh* is a child of the actual subject *N-myc*. The interaction is between *Shh* and *N-myc*, and *N-myc* and the miRNAs.

Compartment Rule (Interaction) The compartment rule stems from the observation, that interactions between miRNAs and genes are commonly within a sentence or clause, and are directly connected by a verb. For sentences consisting of multiple clauses, it is quite uncommon that a valid interaction crosses or skips such a subclause. It is thus checked whether both entities are within the same subclause. In order to do this, the sentence is split into its subclauses, or compartments.

Given the following sentence, the four identified compartments are (listed directly below):

```
<e1>miR-200a</e1> was found to directly target beta-catenin mRNA,
thereby inhibiting its translation and blocking Wnt/<e2>beta-catenin</e2>
signaling, which is frequently involved in cancer.

0 [miR-200a was found to directly target beta-catenin mRNA]
1 [thereby inhibiting its translation]
2 [blocking Wnt/beta-catenin signaling]
3 [which is frequently involved in cancer]
```

Before the actual check can be performed, all subclauses must be found from within the dependency graph. Making use of the general English language flow, from left to right, the easiest possibility to extract these compartments is to split the sentence upon the encounter

of special words or dependencies. In general, a new compartment is formed at one of the following dependencies: *cconj*, *xconj*, *conj*, *ccomp*, *parataxis*, *advcl*, *xcomp*. However, certain compartments are too fine-grained for further relation extraction.

Depending on the observed token or dependency, further sub-rules are applied. Compartments may be separated by several tokens (particularly their POS or dependency). For separation by *VERB* (or *AUX*), the whole subtree will be considered subsequently. If the subtree starts with *because*, *through*, or some connecting verbs, it is not considered as a separate compartment. The same applies to compartments starting with nouns. If the found *VERB* is connected by a conjugation, it is ensured that the full conjugation is within the compartment. The compartment can be split by *amod*. For an *amod* dependency, the sentence must be split by *thereby*, *suggestive*, *while*, etc. A *VERB* which has an *acl:relcl* dependency must have split-words like *whereby* at the subtree start. Finally, a compartment can also be split by a *conj*, if it starts with a splitting word like *actually*. Some compartments unfortunately are not divided by verbs or conjugations. For any found compartment it is checked whether it contains a semi-colon or not. If so, the compartment is additionally split at the semi-colon.

After deriving all compartments, the final compartment check ensures that both miRNA and gene are contained within the same compartment.

Context Rule (Interaction) There exist several words which make a miRNA or gene entity not a target entity for the kind of relation that one is interested in. This rule is probably the most heuristic one in this framework and may need to be adapted for other use-cases, such as protein-protein interactions. For instance, the previously considered sentence

```
<e1>miR-200a</e1> was found to directly target beta-catenin mRNA,
thereby inhibiting its translation and blocking Wnt/<e2>beta-catenin</e2>
signaling, which is frequently involved in cancer.
```

is not about the *Wnt* or *beta-catenin* genes, but the related signalling pathways. Effectively, no direct miRNA-gene interaction is described here. In order to avoid the detection of such co-occurrences, for both gene and miRNA, it is checked whether certain words (like pathway, cells, family, mice, etc.) occur before or after the entity. If one of these words is found, the interaction is rejected. Here, the checks against miRNA-gene complexes are performed, as well as checks to determine whether the gene-entity is related to a cell or knockout. In order to achieve this, the closure around the gene-entity (x words before the entity and y words after it) is considered.

Entity Rule (Interaction) The entity rule is only active if an entity-aware model is supplied. In the context of miRExplore it was observed that the scispaCy BIONLP model performs well for biomedical entity detection. For each word of a document the model predicts from which known entity type a word might stem from. The BioNLP model contains by far the most biomedically relevant entity types of all scispaCy models, including

cell types, organisms, chemicals or tissues. The entity rule asserts that both the miRNA and gene entity of a suspected miRNA-gene pair are not confused with a cell type or an organism. If the model predicts the entities to be of such a type, the interaction is rejected.

Determining miRNA-gene regulation Besides the actual interaction, the direction of the interaction (be it miRNA regulates gene, or vice versa) as well as the direction of the regulation (up-/down-regulated) is of interest, too.

In order to classify found interactions regarding the direction of the interaction and regulation, again a rule-based approach is chosen. While it was heavily relied on the dependency graph prediction for determining relations, it has been observed that this prediction is incorrect in many details. Such details, however, might be essential for solving the problem of determining in which direction a regulation occurs. It was decided to not use the dependency graph as major analytical object of interest. Instead, this approach heavily relies on one property of the English language: the scrambling of words is uncommon. While many languages, such as German for instance, make heavy use of scrambling, in English the use of a pragmatic word order is much more tightened. Most sentences follow the Subject-Verb-Object scheme, mostly in this order [115]. Words, which associate with the start of a sentence, are usually found at the beginning of the sentence, and not somewhere else.

The following compartment, count, context-count, and final (regulation) rules thus operate on the stems of certain word-groups, such as the stem *up-regul* for any word in this family, like *up-regulating* or *up-regulator*. miRNA-gene interactions are classified by their direction. A miRNA regulating a gene is denoted as `MIR_GENE`, and a gene regulating a miRNA as `GENE_MIR`, respectively. The direction of a regulation may either be `DOWN`-regulating, `UP`-regulating or undetermined (`NEU`). For instance, a miRNA regulating a gene is classified as a `NEU`tral interaction, because no effect direction is directly given.

Compartment Rule (Regulation) The first check determines an interaction by trying to find specific context descriptions. In the given example sentence (Figure 2.9), first the compartment containing miRNA and gene is determined. Then it is searched for specific keywords, such as *negatively correlates* or *by targeting*. Depending on the found keywords and the order of miRNA and gene entity, the interaction direction is determined (`MIR_GENE`) as well as the direction of the regulation (`DOWN`).

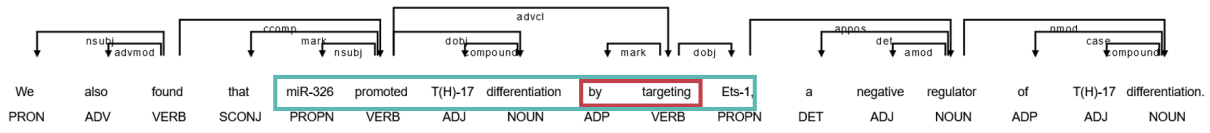


Figure 2.9: **Regulation Compartment Rule Example** The keyword *by targeting* has both miRNA and gene within its boundaries. The interaction is accepted as a `MIR_GENE`, `DOWN` relation between *miR-326* and *Ets-1*.

Count Rule (Regulation) If no direct associations are found, the focus has to be led on the stem-based approach described above. Here, for both the miRNA and the gene any stem within the proximity of the word is found, and its direction is counted. Neutral stems are ignored at this stage. In the example given in Figure 2.10, the searched proximity is marked in turquoise. For the miRNA one negatively associated stem is found (inhibition) and for the gene a positively associated one (increase). Then it is checked, whether the miRNA- and gene-stem counts point into opposing directions, that means miRNA negatively associated, and gene positively, or vice versa. If this is the case, a MIR_GENE DOWN regulation is predicted. Otherwise, the following context-count rule is considered.

On a side note: the red marked dependencies in Figure 2.10 show that the dependency prediction can be problematic. For instance the conjugation around the miRNA is not resolved correctly. While this does not hamper the interaction detection here (still same compartment, not same conjugation, just longer SDP), the regulatory detection could be influenced, because there is no relation between inhibition and miR-23a. This shows exemplarily why for this task the stem-based approach is more suitable.

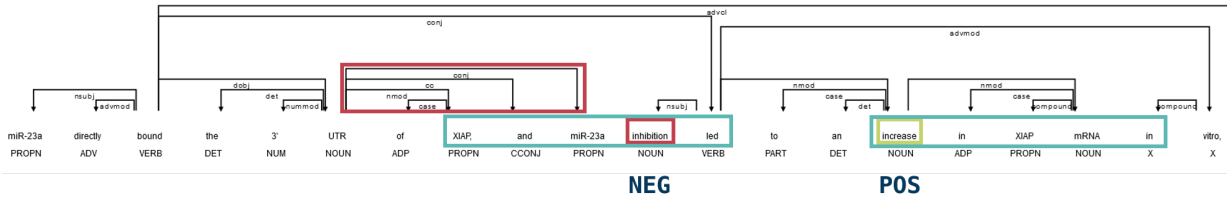


Figure 2.10: **Regulation Counts Rule Example** Within the boundaries of the miRNA and the gene all interaction keywords are determined and counted. If negatively associated action words are more prevalent, it is assumed that the respective miRNA or gene is negatively regulated. Here, a MIR_GENE DOWN-regulation is detected.

Context-Count Rule (Regulation) For the context-count rule, neutral stems are considered, too, and a proximity region around miRNA and gene (similar to the previous rule). Within the example region (Figure 2.11) one negatively associated stem (*Downregulation*) is observed. Given the order of the words, a `GENE_MIR`, `DOWN`-regulation is assumed. However, the word *by* close to the miRNA suggests a passive clause and thus a `MIR_GENE` interaction is reported in this case. The context-count rule not only counts any found stems, but considers the context around the entities, too, including passive and negated sentences.

Final Rule (Regulation) The final rule has to take care of any yet unassigned interaction. For this rule, all stems which are between both entities are considered. The most-frequently occurring stem direction is predicted. The interaction direction is determined by the order of the entities, and whether a passive clause is detected or not. In the example given in Figure 2.12, SRF stands before miR-143 in a non-passive sentence. Hence, the

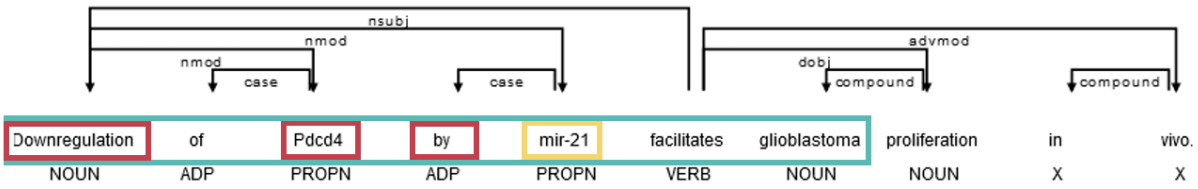


Figure 2.11: **Regulation Context-Count Rule Example** The principle is similar to the *Counts Rule*, but additionally NEUtral stems are considered.

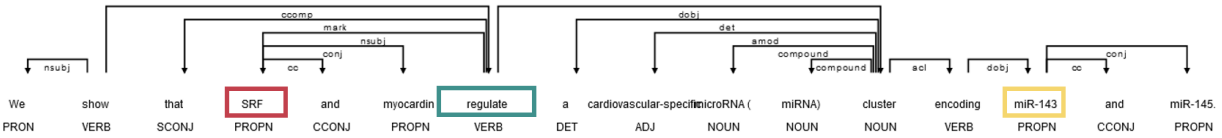


Figure 2.12: **Regulation Final Rule Example** The *Final Rule* is the last resort. All stems between both entities are considered, and the most-frequently occurring stem direction is predicted.

interaction direction is `GENE_MIR`. The found stem, *regulate* is neutral, because it is not known whether an up- or down-regulation is meant. Therefore, the predicted interaction is a `GENE_MIR` `NEU`tral interaction.

Benchmark For evaluating both proposed prediction methods, the benchmark provided by Bagewadi et al. [19] is used. This benchmark allows to compare the proposed method with other, already published methods, like miRTex [177]. The benchmark consists of a training dataset (201 documents, 397 interactions) and a test dataset (100 documents, 232 interactions). For 24 interactions in the test set (see Chapter A.3.3) this gold standard was adapted to reflect a common handling of gene or miRNA-pathway interactions, which will be referred to as modified benchmark. This modified benchmark has been extended as part of this work to enable the benchmarking of the interaction and regulation direction. Directions of interactions are annotated as `MIR_GENE` if a miRNA interacts with a gene, or `GENE_MIR` in the opposite case. The direction of the regulation may either be `UP`, `DOWN` or `NEU` for an up-regulation, down-regulation or undetermined/neutral regulation, respectively. General regulations (e.g. miRNA regulates gene) are annotated as neutral regulations, because no clear direction is given.

In addition, miRExplore was evaluated on the test dataset developed by the miRTex [177] authors.

The prediction results are compared to the gold standard. Predictions are classified as *true positive (TP)* if the predicted result is positive and the true condition is positive. Likewise, a *true negative (TN)* is a negative prediction for a true condition negative result. A *false positive (FP)* is a positive prediction of a true condition negative interaction. A *false negative (FN)* is a negative prediction of a true condition positive interaction, respectively. This allows the calculation of precision as the positive prediction value $\frac{\sum TP}{\sum TP+FP}$ and

recall/sensitivity as the true positive rate $\frac{\sum TP}{\sum TP+FN}$. In order to combine both precision and recall the F_1-score $= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision}+\text{recall}}$ is used.

Web-Portal There are two ways of accessing the found relations: miRExploreWeb and an API. Through miRExploreWeb it is possible to query the miRNA-gene interactions interactively and without programming skills. miRExploreWeb is developed using a TypeScript[11]/React[12]/MaterialUI[13]-stack and has a python flask-app as backend. From miRExploreWeb the user can query for specific miRNAs, genes or term names from all other supported dimensions. Results are shown in a tabular fashion, where for each interaction the user can retrieve all evidence, that is sentences for text mining, and related PubMed articles for experimentally supported, integrated databases like miRTarBase or miRecords. miRExploreWeb accesses the actual miRExplore web service via several HTTP POST queries. These can be accessed from separate clients, e.g. python requests, and thus provide an API for miRExplore. Most of the integrative features make use of this API. The server for fulfilling API requests is built in python using the Flask framework[14].

The web portal currently operates on data processed using the python-based syngrep version. The PubMed abstracts were downloaded in June 2020, the PMC full texts in September 2020. For both resources the JATS-formatted resources were utilized. Article meta information were extracted (e.g. date of publication) along the article's text. For this purpose the `medlineXMLtoStructure` text mining script is used for PubMed, and `medlineXMLtoStructurePMC` for PMC, respectively. Literature references are excluded in full texts. With the PubMed input each JATS-formatted file may contain an update for previously published articles, e.g. if articles are retracted or author details change. In such cases duplicate entries are removed for the PubMed data.

The web portal contains miRNA-gene interactions retrieved via the presented text mining approach. In addition, several public databases of experimentally validated interactions are included (miRTarBase [128], miRecords (Validated Target dataset) [333]), but also the DIANA-TarBase [149] resource with both, validated and predicted, interactions.

Timelines Using the additional meta-data from PubMed, such as publication date, authors and journal, relevant text mining evidences can be visualized as a timeline. Timelines can be created for any miRExplore result, and can thereby be filtered along the implemented dimensions.

Integrative Network Analysis Given a network of miRNA-gene interactions and differentially regulated genes (and possibly miRNAs), the integrative network analysis method outlined here is designed to find putative miRNA-regulators within this network.

[11] https://www.typescriptlang.org
[12] https://reactjs.org/
[13] https://material-ui.com/
[14] https://flask.palletsprojects.com/en/1.1.x/

Figure 2.13: **miRNA-gene Regulation Prediction** Given high-throughput experimental data, miRNA regulation can be predicted under the assumption that all regulation is mainly observed due to miRNAs down-regulating genes. Using rules 1-5, miRNAs are assigned a fold-change (UP or DOWN regulation) such that the regulation of as many as possible genes can be explained, while inducing the fewest possible inconsistencies (e.g. miRNA and gene down-regulated).

This prediction of active miRNAs is performed in 5 steps (Figure 2.13). Using a greedy approach, the amount of inconsistent regulations is tried to be minimized, while maximizing the amount of explained canonical (consistent) miRNA-gene regulations.

In the first step, all measured regulations are annotated. This means, that edges are annotated as consistent if the gene has the opposite regulation of the miRNA, otherwise the edge is annotated as inconsistent — implying a non-canonical miRNA-gene regulation. In the second step, any consistently regulated miRNAs are imputed. This means, if all targets of a miRNA are regulated into the same direction, the miRNA regulation can be annotated into the opposite direction. This does not induce any inconsistencies. In the third step, no clear assignments (that means without inconsistencies) are left. Hence, assignments must be made with a minimum of induced inconsistencies. Thus, miRNAs are imputed in the same way as in step two, but genes, which already have an explaining regulation, are not considered in the inconsistency count. All remaining unexplained targets must be regulated into the same direction. The fourth step considers only miRNAs where all target genes have another consistent regulation. Then the miRNA is imputed into the direction of the most frequent regulation in order to not induce any inconsistencies. The fifth step has to keep the balance between yet unexplained regulations, and the number of potentially induced inconsistencies. It is imputed into the direction of most consistently regulated new targets, thereby reducing the number of unexplained genes while adding the fewest inconsistencies.

In order to rank regulating miRNAs, the amount of consistently regulated mRNAs is compared with the amount of possible targets of a miRNA. This is done using the

hyper-geometric test, which is suitable for over-representation analyses [101]. Within this framework the *hypergeom.sf* function by SciPy [317] is used. This function takes as input the number of drawn successes x, the population size M, the number of successes in the population n and the sample size N. As drawn successes x, the number of putatively regulated target genes of a miRNA is used. The sample size N is the number of total target genes of the specific miRNA within the given context. Finally, the successes in the population n are all significantly regulated genes, and the population size M is determined by all measured genes. The result of this over-representation analysis can be used to identify miRNAs, which regulate more than the expected number of target genes. Such miRNAs might be of high interest, because these could be key regulators in the analysed dataset.

Data Analysis A public RNA expression dataset on T cells [156] was downloaded (GSE109735 (mRNA) and GSE109736 (miRNA)) and processed using the GEO2R approach involving limma [255] for differential gene expression analysis. Data analysis and visualization was done using EnhancedVolcano[15]. To determine differentially expressed genes, an adjusted p-value cut-off of 0.05 was chosen. For these genes, an over-representation analysis on the miRTarBase miRNA target sets has been performed (hyper-geometric test). Likewise, these genes served as input for the integrative network analysis.

For comparing the overlap between the measured and predicted miRNAs (miRTarBase and above network analysis), an adjusted p-value cut-off of 0.1 was used. miRNAs have been compared at the precursor level, meaning that miRNA number and precursor must match.

Using the measured, differentially expressed miRNAs, as well as the regulating miRNAs predicted from network analysis, over-representation analysis (hypergeometric test) was performed on miRNAs associated with Gene Ontology and Disease Ontology terms. For this, all context-relevant miRNA-gene interactions were retrieved for an ontology term, and the hypergeometric test was applied to identify terms with enriched miRNA sets. For an ontology term all children were retrieved. All documents containing any of these terms were retrieved by miRExplore in order to identify therein contained miRNA. However, only terms with less than 100 children were considered in the analysis, which excluded cancer-related diseases. The context was either left to include all information, or was restricted to *t cells* (cell ontology term ID: META:44).

Results and Discussion

The text mining method for the identification of miRNA-gene interaction employs a NER-approach. The general principles of this approach are described in [114] and are omitted here. The initial NER application, syngrep, was developed by Gergely Csaba[16] in C++, making it extremely efficient and enabling file-level parallelism. During the text mining of full texts it became apparent that this implementation has problems with the full text inputs. The

[15]https://github.com/kevinblighe/EnhancedVolcano

[16]Csaba, Gergely. Personal communication. 2015-2020.

need for a more robust and yet similar efficient implementation arose. This need is satisfied with a python-based re-implementation. The python version relies on the Aho-Corasick data structure, too. The used library internally has a C++ back-end. This ensures that performance is only affected slightly. This approach, however, does not (yet) implement all rules of syngrep, particularly related to the detection of abbreviations. Since the focus of miRExplore lies in the identification of miRNA-gene interactions, independent of the employed text mining strategy, this limitation was accepted. For a uniform appearance and processing of PubMed abstracts and PMC full texts, the web-server contains interactions identified through miRExplore in combination with the python syngrep re-implementation. The statistics are calculated with this version, too. The results presented for the timeline functionality and miRNA-gene imputation base on earlier text mining results derived from the initial syngrep version on PubMed abstracts only.

miRExplore is an integrative text mining framework for extracting miRNA-gene interactions. It consists of multiple parts, as shown in Figure 2.14. Outgoing from input texts, the (independent) NER phase delivers potential entity pairs for the interaction extraction (e.g. is a miRNA-gene co-occurrence an interaction?), which are then analysed regarding their regulation in the direction extraction. The found interactions, together with other miRNA-gene interactions from external databases, can then be used for integrative analyses, or be queried using a web-interface/API. All steps are modularized, enabling an interaction and direction extraction independently of the employed NER or NLP strategy.

Web-based access miRExplore provides a web-based user interface for quick, yet informative access. On this platform the user enters specific genes or miRNAs of interest for which miRNA-gene interactions should be retrieved. By specifying context terms, the displayed interactions can be filtered for evidences which are associated with the terms (context). These context terms may stem from ontologies of the included dimensions (currently: organism, cell type, gene ontology, disease). For all inputs, auto-completion is enabled.

For programmatic access, miRExplore has a JSON-enabled API built-in. Users may query the system along the same dimensions as mentioned above. Such requests must be sent via POST and contain a JSON-object which specifies the selection for each dimension. Because the retrieval of the textual evidence for many miRNA-gene interactions is slow, the user may specify whether the textual evidence is displayed.

Evaluating miRNA-gene mention detection The miRNA-gene interaction detection is a detection on the mention level (without any direction information). That means, that in this step it is decided whether two entities form a valid miRNA-gene interaction or not.

In a previous study, atheMir [144], it was found that the therein employed NER and rule-based approach works considerably well and performs as good or even better than field experts do — also in a context-sensitive way. For that approach a three-rule approach was used: for a valid interaction between two entities, both entities must not be within the same conjugation, but they must be connected by a verb, and they must share at least

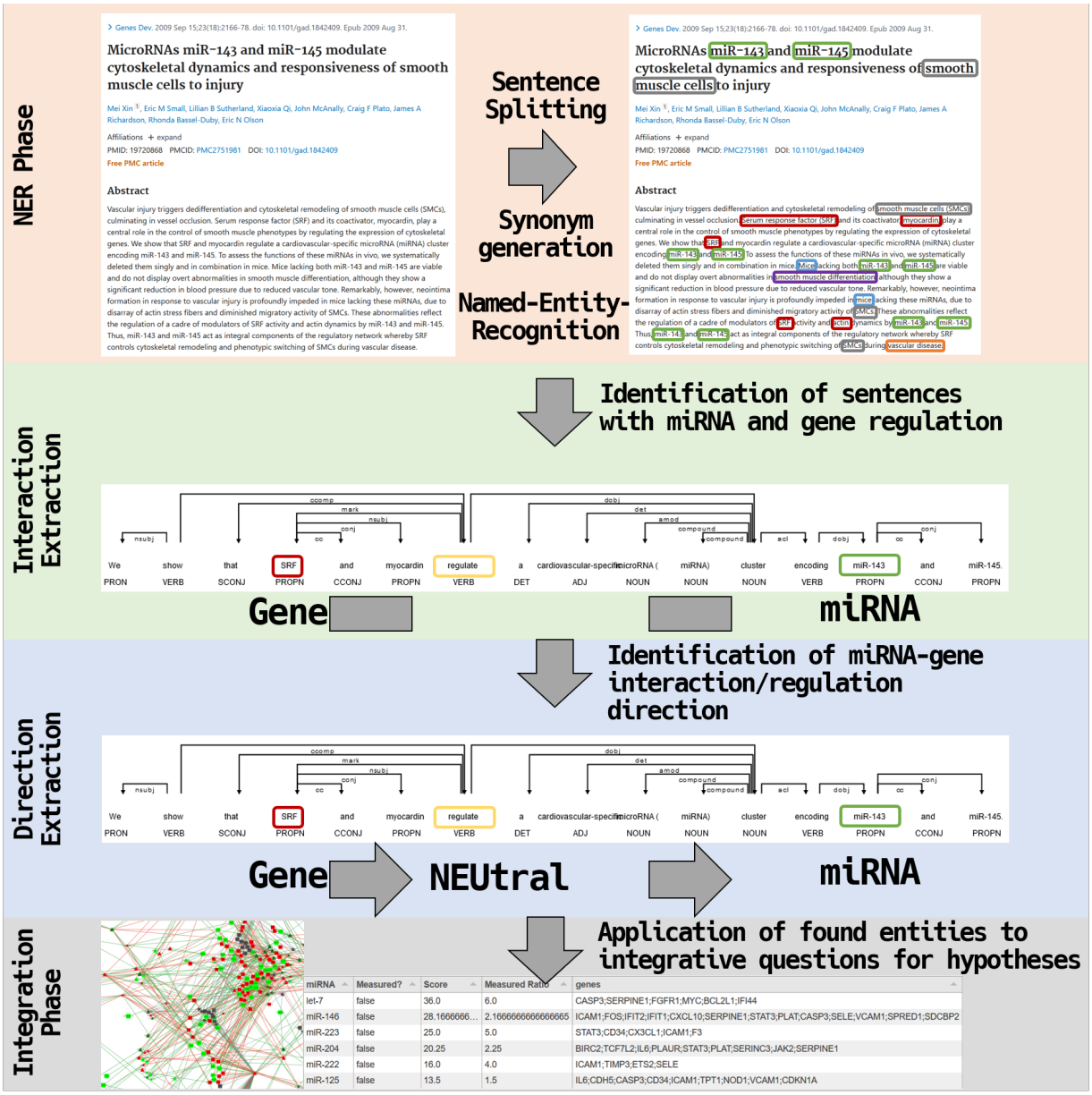


Figure 2.14: **miRExplore Text Mining Workflow** Sentences are extracted from publicly available literature. From specific resources, like the gene ontology, synonyms are created, which are searched in the sentences. After this NER phase, sentences with both miRNA and gene mention are processed. Such sentences are candidate sentences which might contain miRNA-gene interactions. After finding these interactions in the *Interaction Extraction phase*, the direction of the miRNA-gene interactions is determined in the *Direction Extraction phase*. The found interactions can then be used in integrative applications during the *Integration phase*, like the timeline module, or the network analysis.

one element of the paths from the root to the respective entities in the dependency graph. However, in some cases this approach did not deliver the desired results. While it was clear, that the approach in general was sufficient, it was decided to improve this approach with the rules described earlier. Using the training data from both benchmarks, five interaction rules have been developed.

A first evaluation of miRExplore's miRNA-gene interaction detection is performed on the benchmark created by the miRTex [177] authors. For this benchmark, first the NER approach needed to be executed on the relevant texts. Following this, an evaluation of the found interactions could be conducted. Due to the need of executing the NER approach an additional bias was introduced: not only the detection module would be benchmarked, but also the text mining approach. All interactions were curated to keep only those 367 interactions from 1548 sentences of 150 documents where both entities were identified by the text mining approach. For these interactions it was made sure that all relevant interactions were contained, e.g. if ambiguous gene symbols were mentioned. With these modifications the performance of the interaction detection was evaluated independent of the text mining performance. The miRExplore approach achieves a precision of 0.916 and a recall of 0.926 which summarizes in an F_1 score of 0.921. Unfortunately, there is no combined F_1 score reported for the whole benchmark set within the miRTex paper. By taking the fraction of documents per reported class, a weighted F_1 score of 0.915 for miRTex on the whole dataset can be inferred. Thus, miRExplore performs better than miRTex, with the strong remark that the original benchmark needed to be curated to fit the circumstances of this evaluation. The reported F_1 scores may not be directly comparable. But, these results suggest that the performance of miRExplore is comparable to that of miRTex on this benchmark.

Using the modified Bagewadi et al. benchmark, the performance of miRExplore and each combination of interaction detection rules has been evaluated on the training and test dataset using the F_1-score (Figures 2.15, A.22, Table A.1). Applying no rule (always accept) performs bad, similar to some single rules (only one rule applied). It is interesting to note that on the test set the SDP-only and entity-only predictions perform similarly bad, and even worse than the other single rules. For the entity-only rule this can be explained as such, that the rule alone does not make sense: it only checks whether the named entities are really no abbreviations of, e.g., cell lines. The SDP-only rule taken alone has the disadvantage that the dependency graph connects all words of a sentence, regardless whether they are in one subclause, or in different ones. Only in combination with further rules, e.g. the compartment rule, this check becomes more meaningful. The single rules are followed by the double-rules, which are followed by the triple-rules. Finally, the use of all available rules together delivers the best results. Hence, all rules are applicable and there is no single rule which could outperform all others. This also suggests that the single problems provided by the benchmark are considerably different.

It depends on the dependency prediction During the interaction extraction phase, miRExplore makes particular use of the dependency graph generated by spaCy[17].

[17]https://github.com/explosion/spaCy

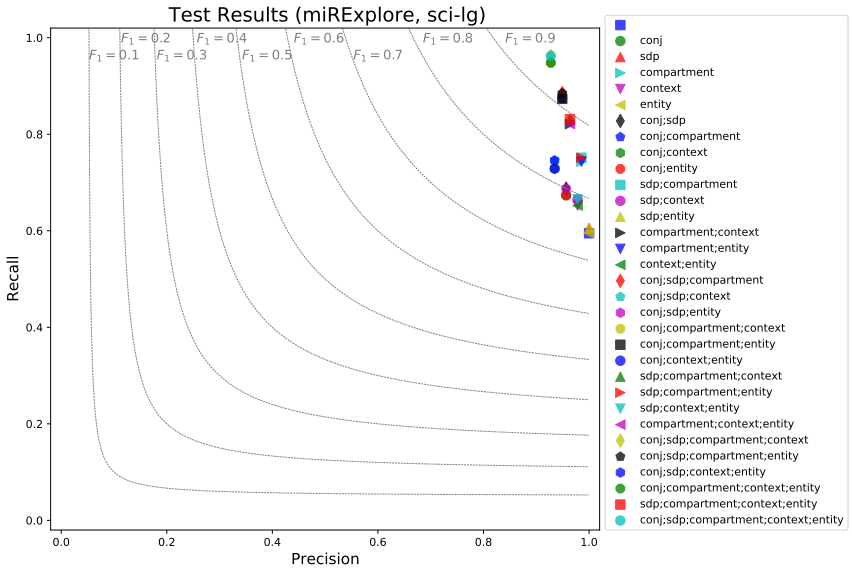


Figure 2.15: **miRExplore performance by rule (sci-lg model)** Corresponding values are shown in Table A.1. For each rule combination the miRExplore predictions are evaluated on the modified benchmark's test dataset. With more rules, results become better in both precision and recall.

All applied rules rely on correct dependency predictions, particularly the SDP rule. The entity rule depends on a correct prediction of the entity type. While the other rules rely on the dependency graph for recognizing conjugations or compartments, the acceptance of an interaction in the SDP rule directly depends on the dependency graph. Therefore, the question arises, how much does the success of this rule depends on the dependency prediction. Moreover, this might be an indicator on how much a ML approaches depend on adequate training data: these are meant to learn rules for dependency prediction from specific texts.

In the regular configuration, miRExplore uses the large scispaCy model (sci-lg) for dependency prediction. But there also exists a special model trained on the BIONLP13CG (BIONLP) corpus. Likewise, spaCy comes with a regular large model, trained on general texts, like newspapers or general literature. These additional models were evaluated on the test dataset (Figures 2.16, A.23, Tables A.2 and A.3).

The general observations made for the scispaCy large model remain valid for the other models: the more rules are added/applied, the better the predictions get. However, one specific observation for the SDP rule in the general spaCy-model is interesting: here the SDP rule performs worse than no rule. More interestingly, the remaining rules perform better if the SDP rule is not applied. This is not surprising: in the previous iteration of this text mining framework, atheMir [144], it was noticed that several incorrect interactions have been reported due to incorrectly resolved dependency graphs, particularly around biomedical words. When using a model not trained for such entities, the dependency

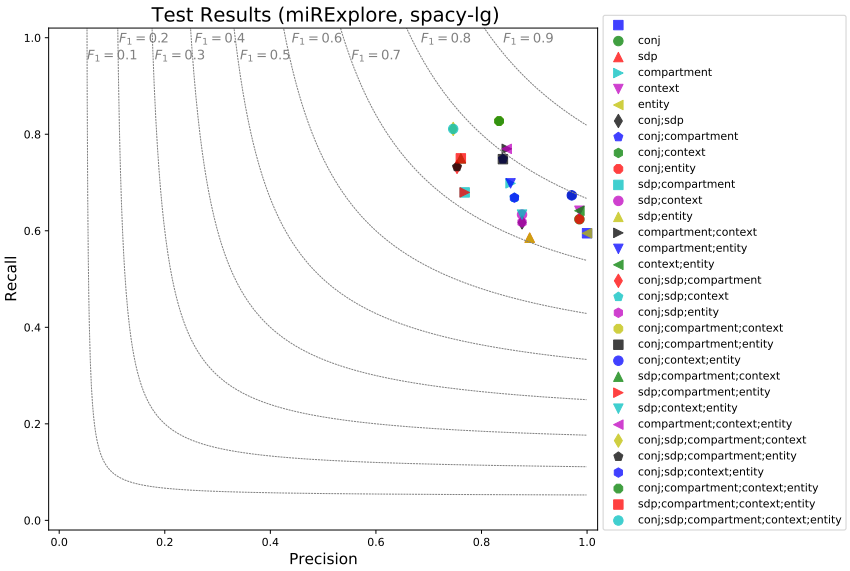


Figure 2.16: **miRExplore performance by rule (spacy-lg model)** Corresponding values are shown in Table A.2. In contrast to the evaluation in Figure 2.15, the default spaCy (large) model was used for dependency graph prediction. It can be seen that the results are worse than for the specific sci-lg model. Moreover, the SDP-only rule performs even worse than the *always accept* rule, at least in terms of F_1 score.

prediction delivers uninterpretable results, and the SDP rule will not operate as intended.

Looking at the absolute F_1 values it can be noticed that on choosing the wrong model, but applying the same rules, the score varies from 0.95 for the scispaCy large model (scientific texts) down to 0.78 with the spaCy large model (general texts). This highlights the major impact the used model for dependency prediction has on even a rule-based interaction detection approach. But moreover this highlights the need for well and correctly pre-trained models, in order to perform an accurate dependency and entity type prediction. Considering that the rule-based approach here resembles a fine-tuning of a pre-trained deep learning model, it can be argued that a badly pre-trained main model (e.g. pre-trained on different kind of text, not biomedical) can not be rescued by a good fine-tuning.

Comparing all methods While it could be seen that all rules combined perform considerably well, the key question is to analyse how well this rule-based approach works in comparison to other approaches and resources. Bagewadi et al. [19] provide a benchmark for identifying miRNA-gene interactions. This benchmark was already used by the miRTex authors [177] to compare their method as well as a re-implementation of the miRSel [224] method. It is thus used to compare the performance of miRExplore with these tools. In this benchmark, it was noticed that in the test set, 22 miRNA-gene co-occurrences were not annotated as interaction, where an interaction actually is described. Likewise, 3 interactions are reported, where only an interaction with a miRNA or gene family is reported, instead of

a direct interaction. This was changed in the modified benchmark for reasons of stringency. Moreover, this modified benchmark reflects better those interactions miRExplore is wanted to find. Using only this modified benchmark for comparison would be unfair regarding the other tools. Thus, the final miRExplore method was evaluated on both the original and the modified benchmark. The already reported results [19, 177] on this benchmark have been combined with the ones obtained for ReLeX and miRExplore (Figure 2.17, Table A.4). A simple, co-occurrence based analysis achieves an F_1 score of less than 0.5 [19], followed by the (general) relation extraction tool ReLeX [96]. ReLeX has problems with specific sentences, probably due to an incorrect dependency resolution, which builds upon old models of the Stanford parser [204]. It achieves an F_1 score of 0.6. The re-implementation of miRSel achieves a high recall with low precision (F_1 score of 0.71). The three-rule-based atheMir performs slightly better with an F_1 score of 0.75. miRTex ($F_1 = 0.87$) is a strong competitor outperforming all previous methods. Finally, the rule-based approach of miRExplore, using the scispaCy model (sci-lg) for dependency prediction, outperforms miRTex with an F_1-score of 0.88 (miRExplore/sci-lg). On the modified benchmark, miRExplore with the regular spaCy model ($F_1 = 0.78$, miRExplore/spacy-lg (mod.)) and the smaller BIONLP model ($F_1 = 0.82$, miRExplore/BioNLP (mod.)) achieves similar results to atheMir. Strikingly, miRExplore, using the sci-lg model, achieves an F_1-score of 0.95 (miRExplore/sci-lg (mod.)), delivering excellent performance on the miRNA-gene interaction detection task.

miRTex is rule-based, like miRExplore. But it shares a design decision with ReLeX: it requires a trigger word for any miRNA-gene interaction. This was avoided in miRExplore. Countably infinite suitable trigger words exist in literature, making it impossible to find and enumerate them all (as the context rule shows, too). Instead, more use of the dependency graph is made with miRExplore. While the dependency graph has problems of its own (and thus should not be used for determining the direction of a regulation), it is good enough to identify whether two entities are in some kind of relation or not. Particularly the dependency graph can be used to reject entities which occur together in a conjugation, or in separate sub-clauses. Thus, having a correct dependency graph is crucial. Modern dependency predictions have become quite reliable, even on unknown sentences and words. Being trained on actual biomedical literature improves their performance such that it can be used reliably to derive interactions.

Evaluating miRNA-gene regulation detection After having identified suitable miRNA-gene interactions, the question arises whether these interactions are outgoing from the gene or the miRNA, and how the induced regulation is directed. Using the extended Bagewadi benchmark, the four rules presented in Section *Determining miRNA-gene regulation* were designed according to the training data. These rules make use of two observations: (1) while the dependency tree is able to identify whether miRNA and gene are not interacting, it is too coarse to identify the direction of a regulation. (2) Scrambling of words is very uncommon in the English language, making the determination of the interaction direction possible using word stems where possible, instead of relying on the dependency graph.

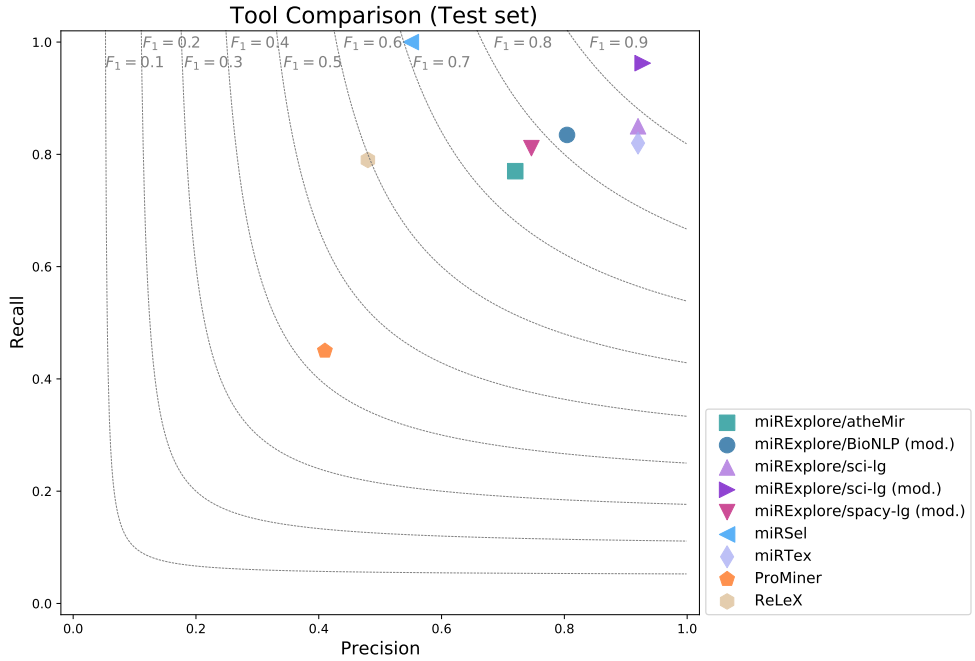


Figure 2.17: **miRExplore Performance Comparison** on the Bagewadi et al. benchmark (test set). All tools perform better than a simple co-occurrence approach (ProMiner). miRSel and ReLeX perform well, but range behind miRExplore. miRExplore improves its performance in contrast to atheMir, which can be explained by the usage of more suitable rules. The performance of miRExplore is better than the one of miRTex. miRExplore using the scispaCy large model (sci-lg) achieves an F_1 score of 0.88 on the original benchmark. On the modified benchmark miRExplore achieves an F_1 score of 0.95 using the sci-lg model (miRExplore/sci-lg (mod.)).

Again, the single rules and any combination of rules have been compared (Figure 2.18, Table A.5). This comparison is performed on the modified Bagewadi et al. [19] benchmark, which was extended to incorporate interaction and regulation directions. For the no-rule case it is always predicted that the miRNA regulates the gene down. It is not unexpected that the return rule alone performs worst: this rule is only applied if none of the other rules was applied, other cases should have been handled by previously checked rules. In contrast to the interaction detection, here the rules are incremental and cannot be seen individually. Nonetheless, with more added rules, an improved F_1 score can be observed. Again, all rules taken together perform best with an F_1 score of 0.93. For miRTex such a detailed evaluation has not been performed.

Database of miRNA-gene interactions from text mining In the previous sections two methods for interaction and direction extractions have been presented. These methods were applied to all found entities in the PubMed abstracts and PMC full texts. Only miRNA-gene interactions, which are described in the same sentence, are considered in

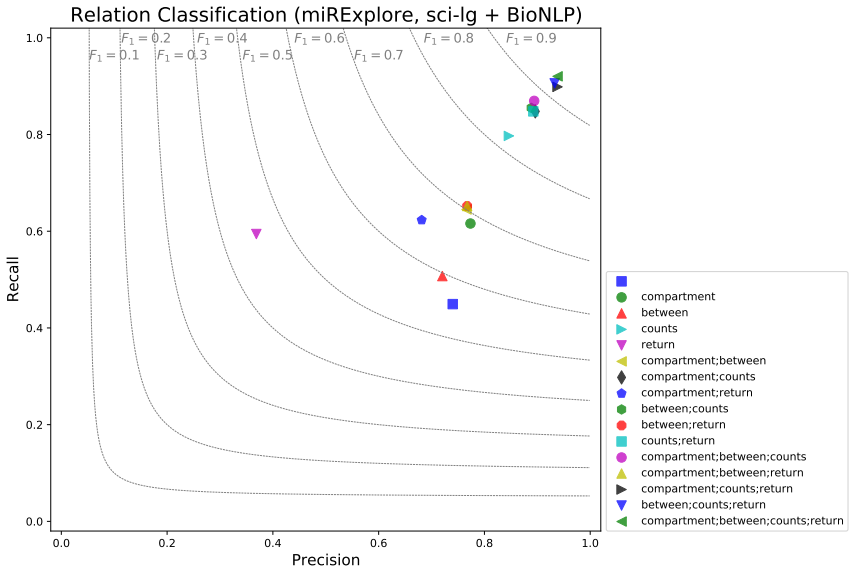


Figure 2.18: **miRExplore Interaction Direction Evaluation** Corresponding values are shown in Table A.5. The extended and modified Bagewadi benchmark is used to evaluate the prediction of miRNA-gene interaction and direction predictions. While the previous checks focused on detecting an interaction, here it is checked which entity is the regulator into which direction. Using no rule (and always predicting `MIR_GENE`, `DOWN`), performs quite well. Some rules, taken out of sequence, perform worse. In general, with more rules the prediction results improve continuously.

this approach. All found interactions are saved in a text file and serve as input for the miRExplore web service.

Using the miRExplore pipeline, many miRNA-gene interactions could be discovered. An overview of all found interactions, their type and their source is given in Table 2.6. Because miRExplore not only contains text mining results, but additionally includes experimentally verified results from other major databases, such as miRTarBase [128], miRecords [333] and partly DIANA-TarBase [149], a comparison at the precursor level between these resources is possible. The result of this comparison is shown in Figure 2.19. The overlap between the single databases is low. Particularly the large databases, which partially rely on (computational) prediction or the automatic evaluation of high-throughput experiments, like miRTarBase and DIANA-TarBase, have many unique hits, even though there are more than 40 000 interactions common between miRTarBase and DIANA-TarBase. Unfortunately, there are many unique text mining results for PubMed abstracts (miRExplore/PMID) and PMC full texts (miRExplore/PMC). Both resources have a large overlap of more than 10 000 interactions, but a majority of hits remains unique. This can be explained by the fact that not all PubMed articles with miRNA-gene interactions are available as full text on the one hand. On the other hand, within the full text more information about other relevant interactions may be named. While this explains the existence of unique PubMed

Table 2.6: **Overview of text mining-based miRNA-gene interactions** More than 50 000 interactions are identified from text resources. However, certain interactions are recorded as both neutral and repressing regulation and are counted twice.

Selection	Abstracts	Full texts
Total documents	31 091 532	677 520
Total documents with gene mention	8 373 762	674 582
Total documents with miRNA mention	59 471	13 078
Total documents with miRNA-gene interaction	34 466	10 152
Number of different genes	6 932	6 256
Number of different miRNAs (precursor level)	1 470	1 257
miRNA <> gene (interaction)	53 791	54 499
miRNA −‖ gene (gene repression)	23 936	18 501
miRNA − > gene (gene induction)	10 650	12 512
miRNA −− gene (interaction)	32 585	34 514
gene −‖ miRNA	5 070	7 278
gene − > miRNA	4 600	7 824
gene −− miRNA	12 684	16 792

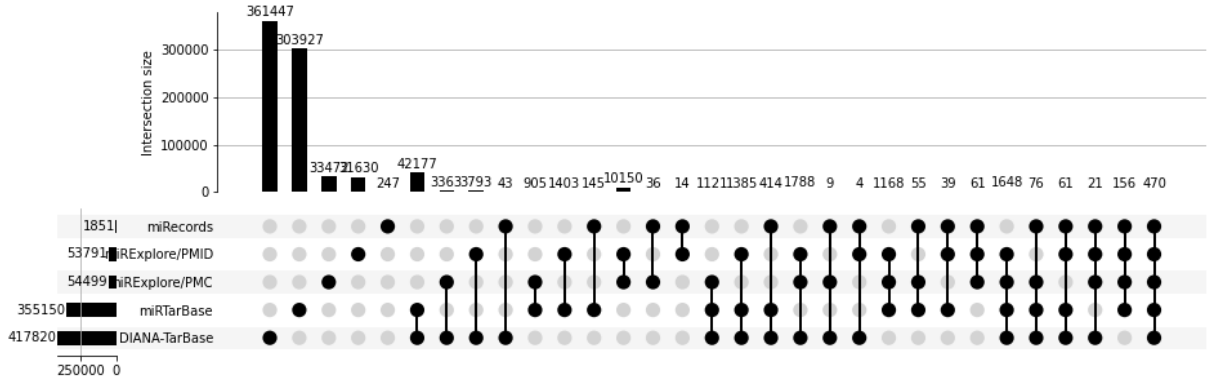


Figure 2.19: **miRNA-gene Interactions of Integrated Databases** A comparison of the in miRExplore integrated databases. Except miRecords, a manually curated database, most miRNA-gene interactions are unique per database. The overlap between the predictive databases (miRTarBase, DIANA-TarBase) is relatively small.

and PMC interactions, a further look into this issue may be needed. It is nice to see that the smallest of all integrated databases, miRecords, achieves a very high overlap with all other databases. This is anticipated, at least for literature-mined interactions, as only validated interactions are contained within the miRecords data set. Such interactions can be expected to be published in literature, too.

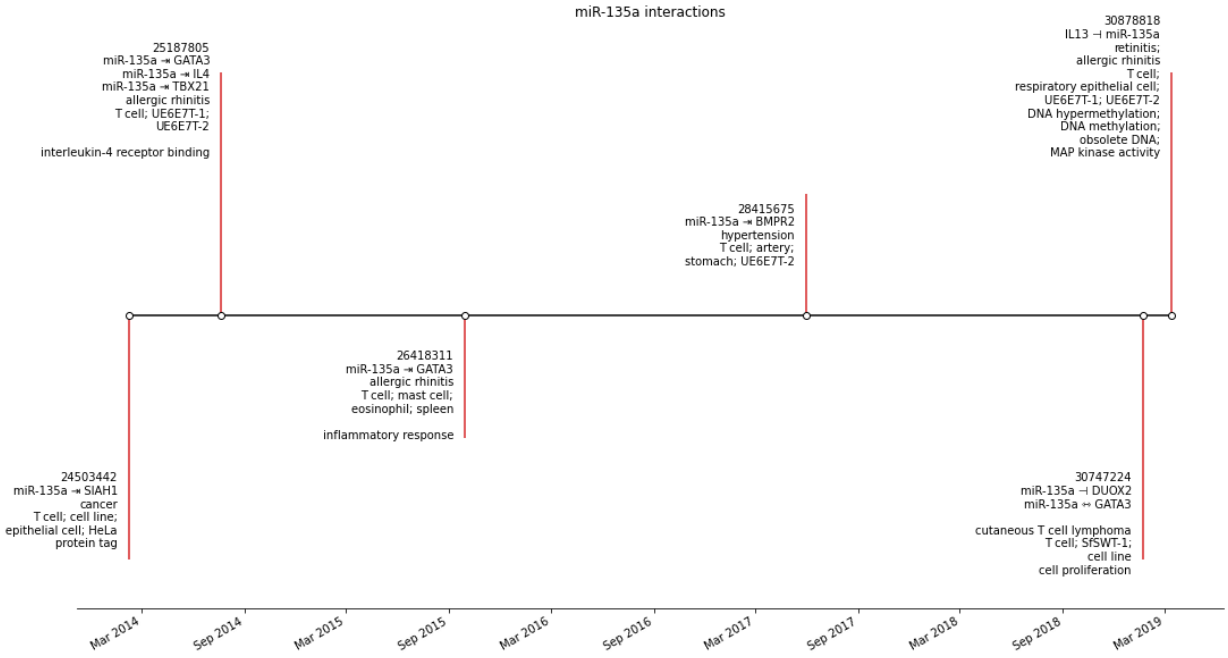


Figure 2.20: **Recorded miR-135a interactions in T cells over time** A comparison of the miRExplore interactions on miR-135a within T cells. Most interactions are recorded in cancer, and other diseases, mainly in an inflammatory context, are listed.

Timelines For any miRNA-gene interaction it is interesting to know, in which *contexts* this specific interaction has already been identified in. However, some researchers might be more interested in all interactions of one specific miRNA, in a specific context, or which interactions are already known for a specific gene. Using the Timelines module (Figure 2.20), it is possible to create graphical answers to these questions, e.g. in which context the specific miRNA-gene interaction has already been looked into, and when. Results from miRExplore are queried, interpreted and plotted. Due to its connection to miRExplore (via its API), the *Timeline* feature can restrict the query such that only interactions within a specific context are returned. As a result of the example query for miR-135a interactions in T cells, it can be seen that the miRNA plays roles in cancer, inflammation and allergic rhinitis.

Integrative Network Analysis The miRNA-gene interactions contained in miRExplore are of high relevance, because the miRExplore system can be queried via its POST-API for miRNA-gene interactions, with particular focus on the contextual information. This can be useful in the interpretation of high-throughput experimental data, e.g. obtained from RNA-seq or microarray analysis. If the context of such experiments is given, it can, for instance, be predicted, which miRNAs might be active regulators.

With most high-throughput data, the problem is that short RNAs are not measured. Particularly miRNA expression is seldom recorded. Hence, the challenge here is: given the

effects, which miRNAs lead most probably to these effects? In this case the assumption is made, that all regulation is outgoing from miRNAs. Whilst this assumption does not reflect the truth, it is suitable enough for hypothesis generation and later experimental validation.

Through the integrative network analysis, which fetches context-sensitive miRNA-gene interactions from miRExplore, hypotheses for likely regulation can be obtained. In general, this relies on the assumption, that, if a miRNA is actively regulating mRNAs, many of its targets will be regulated in a canonical way. This means, if more miRNA targets are regulated into the same direction as expected, this gives a hint for an active miRNA involvement. This hypothesis can be tested using the hypergeometric test.

Here, this method is applied to a dataset of T cells [156], which is publicly available at GEO (GSE109735 (mRNA) and GSE109736 (miRNA)). Not only regular mRNA levels have been measured, but also miRNA expression using microarray. The dataset consists of antigen-naive CD4+ T cells from spleens of sham-treated mice, Th2 cells after polarization and recruitment to the inflamed tissue (early Th2), and cells from chronic inflamed tissue (stable Th2) which were isolated after an Ovalbumin-induced allergic airway inflammation from long tissue. For this purpose, both mRNA and miRNA expression of the naive cells were compared against the chronic inflamed tissue, stable Th2 cells. A total of 65 miRNAs were significantly down-regulated, and 373 additional miRNAs were down-regulated. A total of 9 miRNAs were significantly up-regulated, and 276 additional miRNAs were up-regulated. A total of 245 mRNAs were significantly down-regulated, and 11,598 additional mRNAs were down-regulated. A total of 562 mRNAs were significantly up-regulated, and 18,572 additional mRNAs being up-regulated.

In the miRTarBase miRNA target set over-representation analysis, seven miRNAs (miR-124/155/1276/4766/7705/669k/669h*) were enriched for up-regulated DE genes at an adjusted p-value cut off at 0.1. These represent possibly down-regulated miRNAs. A total of 10 miRNAs (miR-124/miR-155/miR-1/miR-218/miR-1276/miR-4766/miR-487a/miR-669k/miR-669h*) were enriched if both up- and down-regulated genes are considered.

Integrative network analysis was performed as described above. The context was chosen to include only miRNA-target interactions, which have been found in documents associated with *t cells* (META:44). From the integrative network analysis, a total of 6 miRNAs were predicted to be up-regulated (no significant prediction), compared to 96 down-regulated predictions. A total of 37 miRNAs were significantly (adjusted p-value < 0.1) over-represented and predicted to be actively regulated. The results are summarized in Table 2.7. From the 37 significant predicted miRNAs, 8 were found in the experimental data (of which 95 miRNAs fulfilled the p-value cut-off at 0.1), too. The remaining predicted miRNAs are either not recorded by the micro-array analysis or lack a specific precursor in contrast to the micro-array data. The overlapping miRNAs are highlighted in bold in Table 2.7.

While an overlap of only 8 miRNAs first seems a little low, the miRExplore resource allows checking whether these 8 miRNAs at least describe the experimental outcome well. Hence, it was analysed by an over-representation analysis whether this set of miRNAs (or the thereby induced documents) describe a disease or process (in terms of GO) context. For this, for each ontology term its related miRNAs were retrieved from miRExplore and

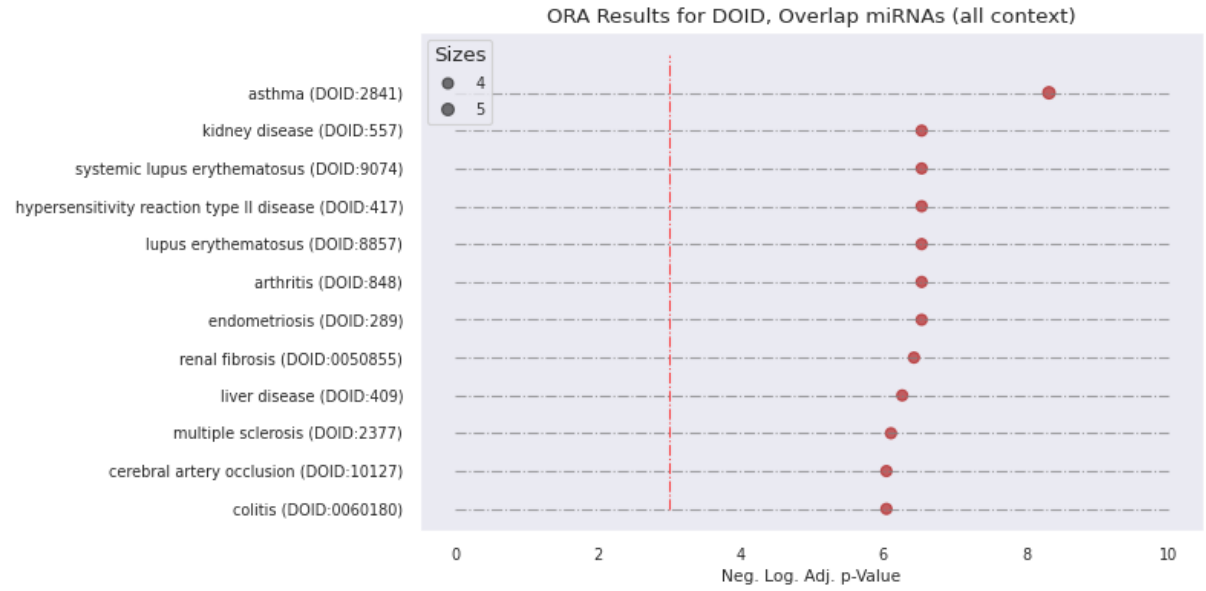


Figure 2.21: **miRNA Over-representation in DOID Terms.** Enriched disease ontology terms for the 8 overlapping miRNAs. Using the overlapping differentially regulated miRNAs, it was possible to independently identify the experimental context of the data: (allergic) asthma. The further significant diseases have in common that they frequently involve an inflammatory response.

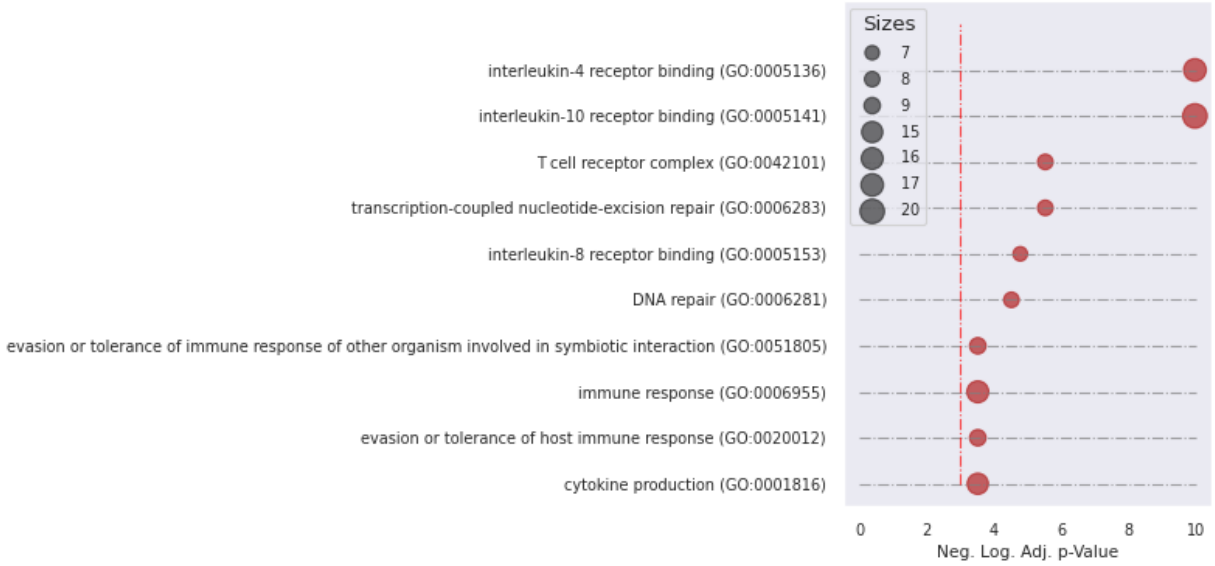


Figure 2.22: **miRNA Over-representation in GO Terms** Enriched GO terms for the predicted miRNAs (T cell context). Since the experiment deals with chronic inflammation due to allergic asthma, the immune-related GO terms are expected, matching with published knowledge regarding the cell type of the experiment. Even the DNA repair mechanisms match existing knowledge.

used for the over-representation analysis (Figure 2.21). This analysis correctly identifies an asthma disease as most enriched term for this set of overlapping miRNAs. It is noteworthy that an association to *hypersensitivity reaction type II disease (DOID:417)* is significant. Typically, allergic reactions are considered to be of type I hypersensitivity. However, atypic allergic reactions are usually classified as a type II immune response [44].

The same analysis was performed for the measured and predicted miRNAs (Figure A.25) separately. While the results are not this clear for both the predicted and measured miRNA sets, both have asthma among the top 12 significant terms. In general, the terms enriched for the miRExplore predicted miRNAs are more similar to the ones of the overlap miRNAs than those of the measured miRNAs, which also do not include allergy related terms.

Making use of the context information available in miRExplore, the context was restricted to *T cell*-related relations only. An enrichment on GO terms for all predicted active miRNAs was performed (Figure 2.22). It is interesting to see that both Interleukin 4 and 10 receptor binding GO terms are found. Both processes are known to be involved during a stable allergic asthma inflammation (IL4 [345], IL10 [272]). Particularly, the IL4 activity is interesting, as this is a key element of the anticipated type II immune response [85]. The IL10 activity can be expected, as it is known to suppress the Th2 response. The further associated GO terms match the experimental context, too. These are mostly related to an immune response, suggesting, that the predicted miRNAs can modulate this response. Both DNA repair associated GO terms seem to break this immunity pattern: however, it has already been found that Th2 cells play an important role in repairing DNA damage due to allergens in asthma [104]. Without the ability to focus on processes relevant in T cells, the results are different and less relevant, as higher order GO terms (like base pairing) are reported. Focusing on specific contexts is essential for hypothesis building, and also reflects the nature of context-sensitive miRNA regulation.

Conclusion

In this section the miRNA-gene interaction framework miRExplore is introduced. The miRExplore framework provides methods for extracting miRNA-gene interactions and several integrated analyses. Moreover, it is possible to integrate further resources into miRExplore, such as miRTarBase or miRecords, enhancing these resources with the additional context information, if evidence documents are provided.

The miRNA-gene interaction extraction has been benchmarked on a public benchmark and thus is comparable to other existing methods. While the miRExplore interaction mining outperforms all other methods, it becomes apparent that rule-based interaction mining platforms stand and fall with accurate dependency graph predictions. When using model-based methods for dependency graph creation, it is important to use a model trained on the same type of text as used with the prediction: biomedical literature.

As part of the integrative methods within the miRExplore framework, the Timeline module and the integrative miRNA-gene regulatory prediction are presented. Using the Timelines module it is easy to explore the history of certain miRNA-gene interactions, or interactions for specific genes or miRNA. It is easy to see in which specific contexts,

Table 2.7: **Integrative network analysis results.** Predicted active miRNAs ordered by significance value from the over-representation analysis on the context-sensitive (T cell) set of target genes. Even though miR-155 has many targets, it has several inconsistently regulated targets. miRNAs highlighted in bold were differentially expressed in the reference miRNA dataset.

miRNA	Pred. Direction	Target Genes	Inconsistent Targets	Context Targets	adj. P-value
miR-155	DOWN	21	4	191	5.12e-05
miR-15a	DOWN	6	0	21	0.0015
miR-16	DOWN	7	0	36	0.0032
miR-7	DOWN	3	0	6	0.0102
miR-214	DOWN	6	1	35	0.0102
miR-135a	DOWN	3	0	6	0.0102
miR-410	DOWN	2	0	2	0.0129
let-7e	DOWN	3	0	8	0.0139
miR-146b	DOWN	4	0	16	0.0139
miR-200c	DOWN	3	0	8	0.0139
miR-125b	DOWN	5	0	30	0.0165
miR-126	DOWN	4	0	18	0.0165
miR-1	DOWN	3	1	10	0.0185
miR-302c	DOWN	2	0	3	0.0185
miR-181a	DOWN	6	0	47	0.0185
miR-197	DOWN	2	0	3	0.0185
let-7a	DOWN	3	0	11	0.0224
miR-212	DOWN	2	0	4	0.0288
miR-142	DOWN	3	0	16	0.0592
miR-27a	DOWN	4	0	30	0.0592
miR-424	DOWN	2	0	6	0.0592
miR-150	DOWN	5	1	47	0.0630
miR-20a	DOWN	3	0	19	0.0776
miR-22	DOWN	3	0	19	0.0776
miR-31	DOWN	3	0	20	0.0856
miR-4739	DOWN	1	0	1	0.0856
miR-340	DOWN	2	0	9	0.0856
miR-26a	DOWN	3	0	21	0.0856
miR-1275	DOWN	1	0	1	0.0856
miR-4736	DOWN	1	0	1	0.0856
miR-95	DOWN	1	0	1	0.0856
miR-1248	DOWN	1	0	1	0.0856
miR-135	DOWN	1	0	1	0.0856
miR-337	DOWN	1	0	1	0.0856
miR-195	DOWN	2	0	10	0.0929
miR-27	DOWN	2	0	10	0.0929
let-7b	DOWN	2	0	10	0.0929
miR-1246	UP	1	1	4	0.3548
miR-491	UP	1	0	3	0.3548
miR-30a	UP	1	1	8	0.4451
miR-24	UP	1	4	14	0.5345
miR-223	UP	1	5	45	0.8898
miR-146a	UP	1	9	70	0.8898

such as disease, cell type, tissue or biological processes, a miRNA-gene-interaction is active. Moreover, this feature allows checking whether a certain interaction within a specific context is actually a new finding, or whether it was already reported by other publications.

Finally, the integrative network analysis allows the prediction of active miRNA-gene regulations from mRNA transcriptome measurements only. Matching small RNA expression data is not required. This is helpful, because there are many datasets publicly available, where no miRNA expression was measured. In the presented use-case, it was possible to identify the specific setting of the experiment, from only the predicted miRNA-gene interactions. Key processes have been successfully reproduced.

The miRExplore framework is open-source, adheres the FAIR principles and is publicly available on GitHub. There are several jupyter example notebooks showcasing analyses which can be performed using miRExplore.

The miRNA-gene interaction field is emerging, with the number of miRNA-gene interaction-related papers rapidly increasing every year. This framework contributes to the highly innovative field of miRNA research by enabling scientists to validate miRNA activity, and to relate own findings with existing knowledge.

2.4 Conclusion

In this chapter the extraction of useful, i.e. structured and context-sensitive, information from unstructured (raw) test has been demonstrated. Methods for text extraction with ontologies and PDFs are derived, applied and evaluated in the first part (Chapter 2.1). This includes a technique for increasing the detection rate of synonyms, which were originally extracted from ontologies. By inflating these synonyms, increasing the number of named entities which are searched for by a factor of 8, more concepts could be found in the text than otherwise. While these changes are partly needed specifically due to the employed NER approach, some methods, like the reverseform or scoped modifications, are useful for general NER approaches. It could be verified that the presented text extraction method from PDF files works with satisfactory results, and that even a structured text extraction works almost error free. Frequently only PubMed abstracts are considered for text mining tasks. It could be seen that these abstracts, however, do not contain the full information, with regard to found and identified terms, in contrast to full texts. Depending on the ontology, the concepts are not distributed uniformly over all sections of a paper. This, however, can often be explained by the topic of the ontology: measurements are more frequently named in the methods sections than elsewhere. These findings are important for the creation of the context-describing synonym lists used in both atheMir and miRExplore.

In the second section, the initial version for mining miRNA-gene interactions was presented: atheMir (Chapter 2.2). Even though only PubMed abstracts were considered for text mining, the results could be used to write a scientific review about miRNA-gene interactions in the context of atherosclerosis [144]. Moreover, the found interactions were more complete than those of domain expert reviews. Nonetheless, it was observed that the established rules build a solid base for possible improvements, with the target to perform

better miRNA-gene interaction mining than other state-of-the-art methods.

Improving the rules used for interaction mining, a new version of the miRNA-gene interaction mining was developed: miRExplore (Chapter 2.3). miRExplore not only improves the miRNA-gene interaction mining, but also contains several integrative features for miRNA-gene relations. The miRExplore database makes it possible to generate Timelines of specific miRNAs, genes or interactions. It can easily be checked whether there is already an existing publication on the same miRNA, within the same context. While not directly used within miRExplore, the PDF text extraction enables further potential use-cases for this feature: fact checking new manuscripts. While this analysis is of interest for any discipline, it obviously has the potential to raise discussions regarding first publication: it is possible to see who reported a specific finding first, and in which context. Besides the Timelines feature, miRExplore integrates with the robust differential expression pipeline `RoDE` (Chapter 6.1). Based on DE results, miRExplore can predict actively regulating miRNAs, using a greedy approach minimizing inconsistent regulations, within specific contexts on a routine basis. From this context-sensitive prediction of either up- or down-regulated miRNAs, it is possible to reconstruct the context of a specific dataset. The processes identified through the predicted miRNAs were shown to describe the regulatory activity on the gene level well, and thereby help to find possible hypotheses of regulated mechanisms.

In the context of this thesis, the miRExplore resource provides the starting point for integrative analyses, which can even lead to well funded hypothesis to follow up. The higher precision and sensitivity of miRExplore over existing methods, as well as the integration with context information, makes miRExplore a valuable resource for many analyses, like Timelines or miRNA-gene regulatory predictions.

The purpose of computing is insight, not numbers.

Richard W. Hamming

3

Accessibility and Interoperability in Bioinformatics

Bioinformatics is one of the disciplines in science that relies massively on an interdisciplinary setting [97]. In 2013 the editorial of *Briefings in Bioinformatics* emphasized that 'Bioinformatics is central to biology in the 21st century' [267]. However, the topics of accessibility, usability and interoperability only play a small role in bioinformatics methods development, although these are the key for interdisciplinary research [30]. The problems of accessibility and usability of bioinformatics software have been taken up in the first section of this chapter (Chapter 3.1). The topic of interoperability is discussed at the example of a benchmark of serialization techniques in the setting of a *k-mer* counting strategy (Chapter 3.2).

Improving the accessibility and computability of bioinformatics tools and workflows (Figure 1.1), respectively, is highly important. Without access to bioinformatics resources, no data analysis can be performed. Making bioinformatics accessible to more scientists makes bioinformatics data analysis a more versatile and more widely employed tool. Likewise, more efficient bioinformatics tools allow a higher throughput of data analyses, preferably adhering to FAIR principles.

With the amount of bioinformatics analyses in the domain of sequencing experiments continually increasing, the computability of results is brought into focus due to the sheer amount of data to process. While the low-level performance analysis of certain programs or libraries is not the key area of bioinformatics, many problems in bioinformatics can only be tackled by understanding performance bottlenecks and improving these. For instance, many bioinformatics programs, particularly in the area of genomics, rely on parallelism, mostly in the form of shared-memory parallelism. With the advent of new sequencing technologies, like Oxford Nanopore MinION sequencing, the interest in *k-mer* counting

has risen again. *k-mer*s are a useful ingredient to genome assembly [68, 160] and read error correction [32, 143]. More recently *k-mer*s were used for genome indexing tasks [155]. While commonly locks in the form of mutexes are used for serializing the access to certain data, new hardware features are available. These are assessed in the second section of this chapter (Chapter 3.2).

3.1 Accessibility of Bioinformatics Software (bioGUI)

The accessibility of bioinformatics applications, also by researcher from other domains, is an important topic, because otherwise many researchers are limited in their choice of tools [97]. Unfortunately, the limited availability and accessibility of many bioinformatics resources [152, 201] and the existence of many data formats does not improve the user-friendliness of bioinformatics tools in general. Moreover, even for a single data format, like the FASTA format, multiple definitions exist (see Appendix A.1). As a matter of fact, bioinformatics becomes more and more important in every-day biological work. This is underlined by the advance of the Oxford Nanopore Technology sequencing platform. Providing measures to overcome the gap between usage of the portable sequencer and bioinformatics tools, the shift from command-line (CL)-only application to graphical applications is important.

With bioGUI, an application for making command-line interface (CLI) applications accessible via a graphical user interface (GUI) is presented [145]. *bioGUI* has two modes of operation. First, by providing install modules it allows installing selected software in Linux, macOS and Microsoft Windows. On Windows, bioGUI makes use of the Linux-environment 'Windows Subsystem of Linux', which emulates (almost) a full Linux on Windows. The second mode of operation is the execution of CLI applications via the *bioGUI*, its GUI. This works by providing an XML-based description of the graphical inputs for all necessary arguments, such as file inputs, text or number inputs, combo-boxes, etc. From these inputs, the command-line arguments are assembled using a Petri-net-like method. The general workflow of bioGUI is shown in Figure 3.1. A list of all available (install) templates of *bioGUI* is available in Table 3.1. Particularly for long read sequencing many modules exist, with a focus on tools required for genomics (assembly). In addition, the analysis of RNA-seq data is supported with the read aligner hisat2 [153] and the read quantification method featureCounts from the subread package [184].

In a user-study it was evaluated, whether bioGUI improves the usability of bioinformatics applications. Participants were asked to install a common bioinformatics tool, graphmap [286], and use this tool to align reads to a given reference. It could be shown that using bioGUI, the installation and usage of bioinformatics tools becomes (significantly) easier.

The accepted publication is available as open-access online article `https://doi.org/10.7717/peerj.8111`. The author's contributions are listed in Appendix A.4.1. bioGUI is available from GitHub `https://github.com/mjoppich/bioGUI`.

Table 3.1: **bioGUI: Available Templates and Install Modules** Install modules (starting with *Install* in the module name column) allow an automatic installation of the software by bioGUI on Windows (using Windows Subsystem for Linux(WSL)), Ubuntu or macOS. Several applications relevant to NGS and TGS workflows are available, such as read alignment, read quantification and assembly.

Module Name	**Task**	**Install Module**	
		WSL & Ubuntu	**macOS**
First Time macOS Setup	Initialization	-	✓
First Time Ubuntu/WSL/apt-get Setup	Initialization	✓	-
Install Ballgown v1.0.1 [240]	NGS transcriptomics	✓	
Install Bowtie1 [168]	NGS	✓	
Install Bowtie2 v2.2.9 [167]	NGS	✓	✓
Install bwa v0.7.17 [178])	NGS	✓	✓
Install canu (github, [160]	Assembly	✓	
Install featureCounts [184]	NGS transcriptomics	✓	✓
Install glimmer302b [67]	Genome Annotation	✓	
Install graphmap [286]	Long Read Sequencing	✓	✓
Install albacore (pip wheel, ONT)	Long Read Sequencing	✓	
Install guppy (linux tar.gz, ONT)	Long Read Sequencing	✓	
Install hisat2 [153]	NGS transcriptomics	✓	✓
Install hmmer-3.1b2 [326]	Sequence Analysis	✓	
Install jellyfish-2.2.6 [203]	NGS	✓	
Install minimap2/miniasm/racon	Assembly (long-read)	✓	✓
Install MS-EmpiRe [7]	NGS transcriptomics	✓	✓
Install PureSeqTM [320]	Sequence Analysis	✓	
Install rMATS-3.2.5 [279]	NGS transcriptomics	✓	
Install rnahybrid [251]	Sequence Analysis	✓	✓
Install RSEM v1.3.0 [176]	NGS transcriptomics	✓	
Install samtools-1.3.1 [180]	NGS	✓	✓
Install SPAdes v3.13.0 [21])	Assembly (hybrid)	✓	✓
Install StringTie v1.3.0 [240]	NGS transcriptomics	✓	
Install Top Monitor (ssh example)	Technical Demo	✓	✓
Install Trimmomatic v0.36 [33]	NGS	✓	
Install wtdbg2 [260]	Assembly (long-read)	✓	x
Template Circlator [131]	Assembly	✓	✓

Tools marked with ✓ provide an install module for the operating system of the respective column.

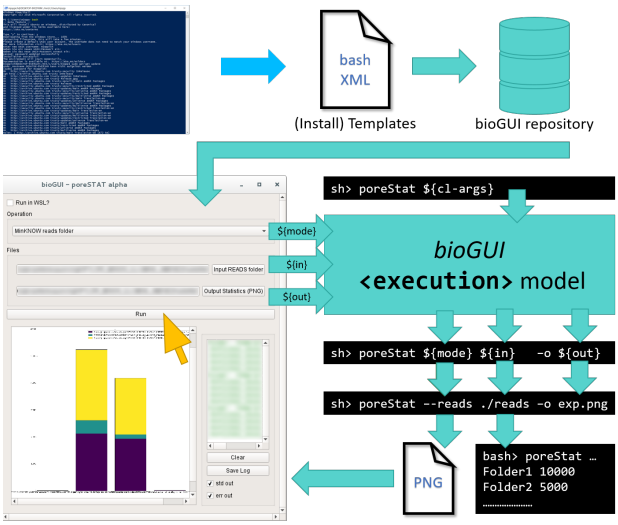


Figure 3.1: **bioGUI modes of operation** Transitioning from the CLI to a GUI, bioGUI provides install modules for the easy installation and usage of bioinformatics applications in its bioGUI repository. From the respective user inputs, bioGUI assembles the command-line arguments and runs the CLI application, visualizing the output again in the GUI.

3.2 Transactional Memory for Entity Counting (tsx-Count)

*K-mer*s are one of the smallest entities in any sequence-based analysis. They are frequently used in the area of genomics as ingredient to genome assembly [68, 160] and read error correction [32, 143], but also as an important ingredient to genome indexing [155]. Even though the topic of *k-mer* counting is very old, it recently has observed attention by researchers from Harvard University[1] and Johns Hopkins University [155].

In principle, the task of *k-mer* counting is straight forward: for a sequence of letters, every k-long subsequence is formed and counted. Every seen *k-mer* is counted by incrementing the respective counter by one. In order to avoid the allocation of space for unseen *k-mer*s, for instance, a hash-map can be used. However, the problem about *k-mer*s in reality is, that they are not distributed uniformly, neither in transcriptomic or genomics reads, nor in the genomes themselves. For reads of the *S. cerevisiae* transcriptome generated by a MinION sequencing device it can be seen that most *k-mer*s actually occur only a few times or once (Figure 3.2). For larger datasets and higher values of k, even more *k-mer*s which only occur once would be found, while some *k-mer*s appear more often. It is thus not practicable to allocate the same amount of storage space for each *k-mer*, particularly for those occurring only a few times, it would be a waste of memory.

Li et al.[1] benchmark several counting strategies, e.g. hash-maps and distinct tools like jellyfish [203]. It is known that *k-mer* counting tools can well profit from multi-threading in

[1] https://github.com/lh3/kmer-cnt/

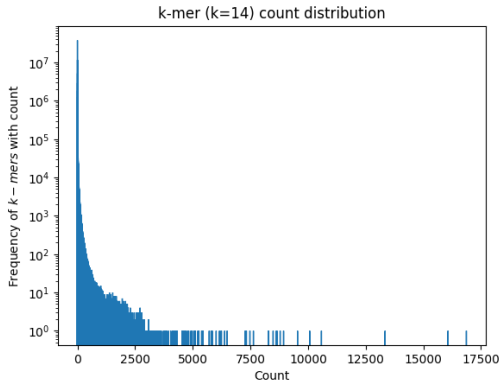


Figure 3.2: ***k-mer* histogram** of all 14-mers of the *S. cerevisiae* reads from accession SRR5989373.

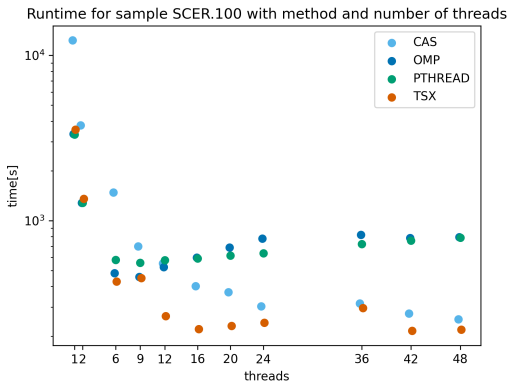


Figure 3.3: **tsxCount run-times for the full dataset (SRR5989373)**. Experimental run-times for the *Saccharomyces cerevisiae* dataset (full dataset, 2x Intel(R) Xeon(R) Silver 4214 CPU with 12 cores and 24 logical processors, each, `OMP_PROC_BIND=spread`).

order to speed the counting up and achieve useful run-times. The particular long run-times for counting *k-mer*s stem from an effort to keep the amount of used memory small. However, when running in parallel, serialization becomes an issue, as otherwise the result will not be exact, or, the application will not terminate. Assessing which impact the serialization has, and which serialization technique is most favourable for this task, was done in the tsxCount project.

With tsxCount it is investigated how hardware transactional memory (TSX) can be used as serialization technique for counting genomic entities, *k-mer*s. The TSX strategy is compared against no locks (SERIAL), pthread-mutex (PTHREAD), OpenMP-locks (OMP) [65] and a compare-and-swap (CAS) implementation. The real-life-problem benchmark is large enough for a significant workload and includes both biases introduced by the sequencing technology (error rate), and the transcriptomic sample origin (poly-A ends at reads, leading to high counts for few *k-mer*s).

Given the differences between the PTHREAD and OMP implementations (Figure 3.3), it can be noted that the hinted-lock implementation (OMP) is more performant, possibly due to already using hardware transactions internally. However, the difference, in general, is neglectable.

For our application, OMP (using speculative locks) and TSX serialization have been the most performant implementations on few threads. With an increasing number of threads TSX becomes advantageous compared to OMP locks, and CAS shows increasing efficiency, which can be seen by its linear speed-up, even at a high number of threads.

Using only few threads, an OMP lock-based serialization technique can be preferred. Not only because OMP is at least as fast as TSX, but the small advantage of TSX in

time efficiency is considerably offset by the experienced difficulties during implementation, debugging and its lower portability. If more threads are intended to be used, or if in general a linear speed-up is required, TSX and CAS are useful choices. Of all tested serialization techniques, CAS has the most constant speed-up. But having high initial costs, CAS only pays out with more than 24 threads, and can not overtake TSX. However, TSX is not available on all CPU platforms. The TSX implementation is not platform robust regarding cache misses and heavily depends on the available caches of the processor. Hence, given that the OMP-approach is much easier to implement and faster or as fast for fewer threads, the OMP lock-based serialization wins also for reasons of interoperability and availability on all platforms.

In summary, the main question one has to answer before choosing a serialization technique, is on which platform the given software will be run. If an application can be tailored for a specific computer architecture, which supports TSX, the TSX implementation can be regarded as favourite, as it performs fast in general and can develop its full potential through fine-tuning (which was not performed for this benchmark). For a general purpose software, the usage of TSX is hardly possible, because only some Intel CPUs support TSX. For software which is intended to be run with many threads, the CAS approach can be useful, as it delivers an excellent speed-up, with very high initial costs though. If the target is a regular workstation, with an average CPU and thread count, lock-based approaches remain the favoured serialization technique, also being the most interoperable choice.

The results are currently submitted to a journal and are in revision. The full manuscript, with detailed information on the employed methods, can be found in the Appendix A.4.2.

3.3 Conclusion

In this chapter two applications in the area of accessibility and computability of bioinformatics tools have been presented. In Chapter 3.1 bioGUI was introduced. The main focus of bioGUI is the accessibility of bioinformatics software. There are many great bioinformatics applications available, but any non-computer affine scientists can not, or only with a high burden, access these applications, even though they are distributed at no cost. Instead, many scientists rely on closed-source applications to perform their bioinformatics or statistical analyses. With bioGUI a framework to make open-source software more accessible was developed. Using bioGUI the task of installing and executing software becomes easier, both for professionals and non-computer-affine scientists. In addition, bioGUI contributes to FAIR bioinformatics. With bioGUI's ability to make use of the 'Windows Subsystem for Linux', Linux applications become available on a Windows host, increasing the accessibility and interoperability of the software. By saving filled out bioGUI templates, repeatable analyses are promoted.

In Chapter 3.2 tsxCount was presented, which benchmarks several serialization techniques for parallel applications: a topic of interoperability and computability. tsxCount compares several serialization techniques in the setting of a classical bioinformatics exercise, counting *k-mer*s. The results show that mutex/lock-based approaches clearly limit the

scalability of such programs. While this can already be seen when using only few threads (after 8 threads for OMP/PTHREAD), the TSX technology pushes the limit slightly further to 16 threads. The only lock-free approach, CAS, achieves an ideal speed-up over all threads. However, this achievement must be put into perspective: it still is the slowest method in the observed setting. On the other hand, the fastest method, TSX, does not adhere to the FAIR principle of interoperability, because specific Intel hardware is required. Using the OpenMP locks hence seems to be a good trade-off between interoperability and speed, at least as long as the application is not run on too many threads, generally.

The results presented in this chapter contribute to the accessibility and interoperability of bioinformatics applications, and thereby to the FAIR principles. Since bioinformatics not only has to care about new methods, but also about how to make efficient use of hardware resource for such methods, both bioGUI and tsxCount contribute to this question on different levels: bioGUI aims at the top layer of accessibility, the user, while tsxCount focuses on interoperability on the machine level. With bioGUI, more scientists can use actual bioinformatics software on any operating system. The results presented in the tsxCount section focus at both the interoperability and parallelization from a developer's perspective. The results from the tsxCount project help a developer to better judge which serialization method is suitable for a specific project, considering both the expected speed-up and interoperability of his software.

I don't know anything, but I do know that everything is interesting if you go into it deeply enough.

Richard P. Feynman

4

Single Cell Analysis and Imaging Mass Spectrometry

In Chapter 2 information has been extracted from peer-reviewed journal articles. As presented in the description of a general bioinformatics workflow (Figure 1.1), making data sources available and using these to extract data, is a fundamental step for any data analysis workflow. In this chapter work on extracting information from two different experimental techniques is presented: single-cell RNA-seq (scRNA-seq) and imaging mass-spectrometry (IMS). Even though these two experimental techniques are very different, with the first measuring gene expression on the transcription level, and the other protein expression, both have in common that they are emerging technologies, which are increasingly used in biomedical experiments [12, 76, 83, 287]. Moreover, due to the underlying analysis methods, both methods face the problem of missing data [166, 281]. Both techniques, however, capture similarly large entities in many single measurements and can profit from many replicates.

The first new data source, made accessible within the research unit bioinformatics, is scRNA-seq, and is introduced in Chapter 4.1. The cell type prediction method cPred, which was developed for the analysis of scRNA-seq data, is discussed in Chapter 4.2. The second technique, MALDI IMS, is described in Chapter 4.3. The `pIMZ` framework for the analysis of IMS data is introduced and applied in Chapter 4.4.

4.1 Single Cell Analysis

Background

scRNA-seq experiments have emerged as an important type of sequencing experiments, and are currently performed with many high-impact publications [186, 329]. The number of performed and in GEO deposited scRNA-seq experiments is increasing each year (Figure A.11). The scRNA-seq technique is on the verge to replace RNA-seq experiments, which used to sequence many cells and their RNA together. The major advantage of scRNA-seq, e.g. using the 10X Genomics[1] protocol, is that it can deal with low amounts of input RNA for sequencing, unlike traditional approaches, which require much more input RNA. This ultimately enables sequencing of the RNA from only one cell. This creates new problems, such as the missing value problematic: no detected reads for a specific gene may result from no expression, non-abundant expression or even technical artefacts. However, the advantages are major: the transcriptome of many single cells is detected, allowing to see where the cells differ, and maybe draw conclusion of the composition of the sample. Each cell, which behaves similar to other cells, could be seen as a replicate for this cell group. With many replicates, statistical methods, e.g. for differential gene expression, can gain a higher (statistical) power.

With the outbreak of the Coronavirus Disease 2019 (COVID-19) in early 2020, scRNA-seq became an important measurement technique to understand this disease [9, 34, 248, 334]. Many experiments on a single cell level have been performed to understand the disease, and the consequences of the actual virus infection, for instance for the immune system. Particularly the possibility of scRNA-seq to detect the absence, or reduction of specific cell types, or specific genes in specific cell populations, brought particular insight into the progression of this disease [183, 238, 329].

In the following, the general workflow for single cell analysis is described, along with a new method to identify the cell type of specific cell populations or clusters from differential expression data. As part of this work, single cell analysis was established in the research unit bioinformatics at the LMU, best practices were developed and several scRNA-seq datasets were re-analysed to gain confidence with the analysis itself and the cell type prediction. This resulted in the publication of a first scRNA-seq analysis paper [228], another one currently in submission [238] as well as others currently in preparation.

Analysis Workflow

A typical scRNA-seq analysis workflow (Figure 4.1) starts with the molecular library preparation[2]. Each cell is put into a droplet, in which gel beads-in-emulsion, so called

[1] https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger

[2] https://assets.ctfassets.net/an68im79xiti/4UZmzpmHGn09BRRIIyagG2/56766ac8e3488f9d66b0275c907751ac/CG000206_ChromiumNextGEMSingleCell3_v3.1_CellSurfaceProtein_Rev_D.pdf

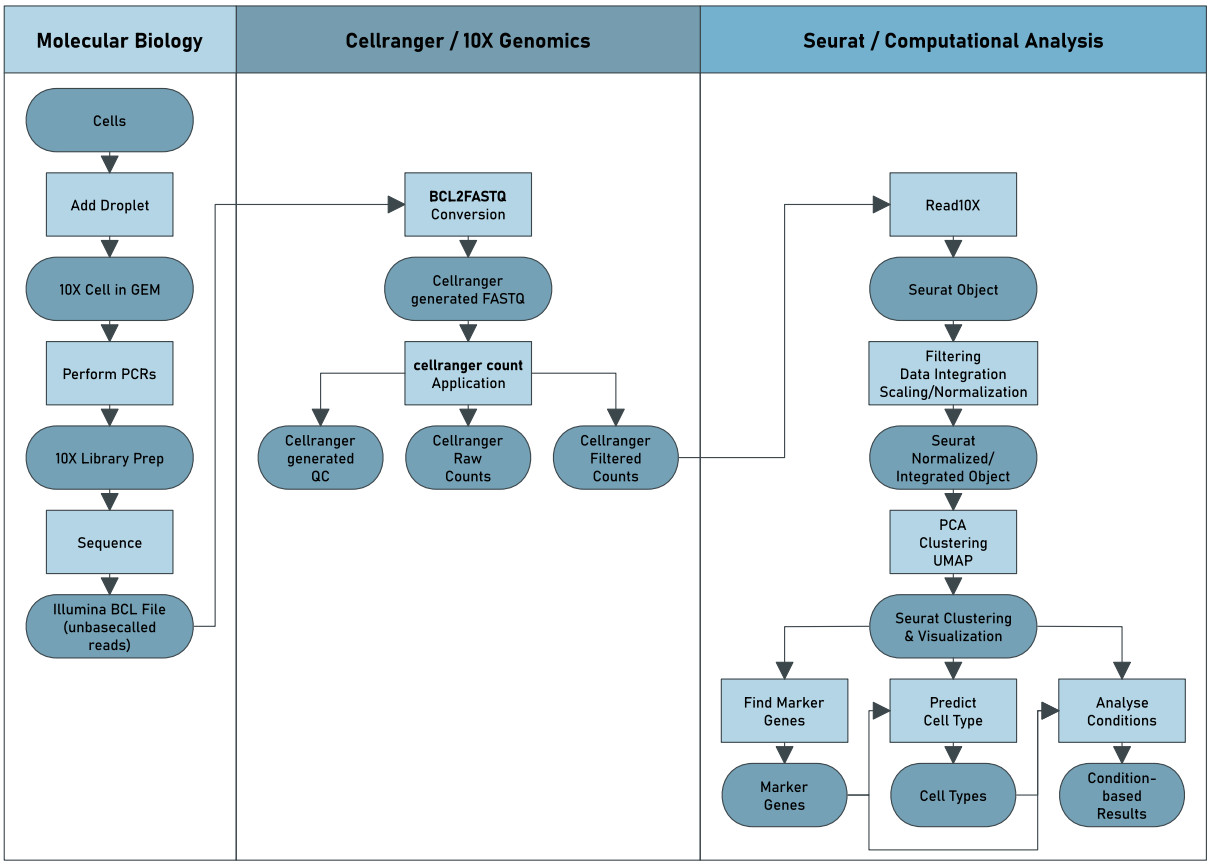


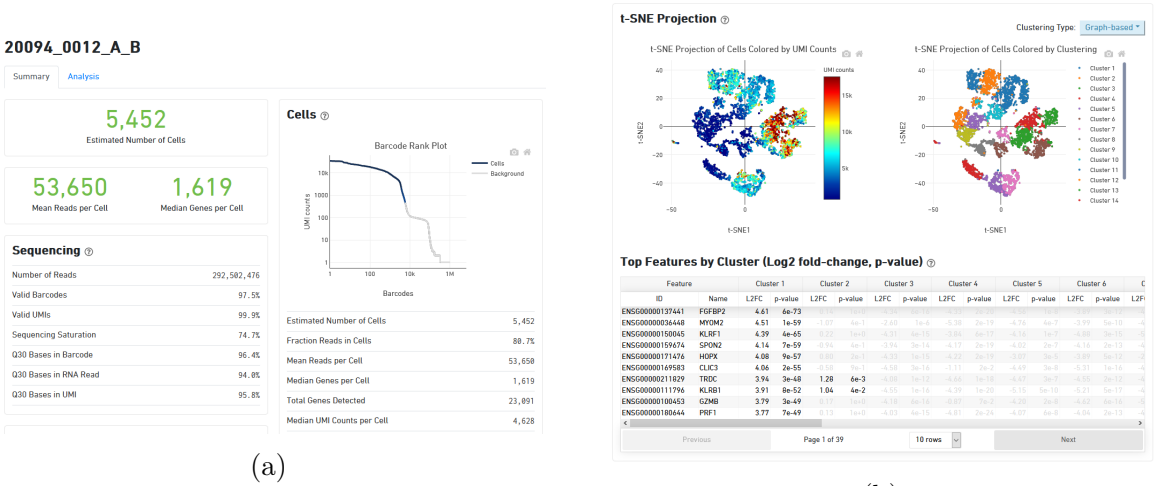
Figure 4.1: **Typical scRNA-seq Analysis Workflow** The typically performed steps within a scRNA-seq experiment are listed, beginning with the retrieval of the cells and finishing with the computational analysis. Typically, all steps on the molecular level and the cellranger level are performed according to the 10X protocol, whilst computational methods and method development start with the ready count matrices from cellranger.

GEMs, are contained. Each bead has multiple anchors, to which the mRNAs or antibodies, which are contained in the cell, can bind. All beads within one droplet have the same barcode sequence such that the cell can later be identified by this barcode. Each bead has multiple anchors, which consist of the same barcode, to which a Unique Molecular Identifier (UMI) is attached, which allows an identification of the specific mRNA which bound to this anchor. This way it is clear how many distinct mRNA were originally detected. Each bead also has multiple anchor types. There is one type which captures mRNA using a poly(dT) tail, which binds to the poly-adenylated ends of mRNAs. Another anchor is designed for specific RNA fragments on antibodies, which allow an additional hashing of the cells. This is useful if several samples are sequenced in the same sequencing run. After sequencing, the raw data from the Illumina sequencing device are interpreted by the cellranger[1] software suite, which performs the conversion of the raw data (BCL format) into universally interpretable FASTQ format, following further counting steps. In the first step,

three FASTQ files are created, one containing the cell barcode and UMI, one containing the Illumina sample index, and a third one, containing the sequence which was bound to the anchor. In the *cellranger count* stage, these three files are interpreted and count matrices are created, where the number of different reads per cell and per gene is saved. During the quality control (QC) stage, cellranger sorts out cells which do not have enough mRNA-reads captured. Additionally, it evaluates several parameters from the read mapping step, for instance the amount of reads mapping to the transcriptome.

A typical cellranger QC report is shown in Figure 4.2. First, the quality features (mostly mapping related) are reported (Figure 4.2a). Using the graphics from the Barcode Rank Plot, the number of detected and valid cells can be detected. The typical shape for this figure is like a bent leg: first the drop over the y-values (UMI count) is low, then, at the knee, the drop becomes quite deep, and recovers at the foot. This specific shape can be used to tell whether there have been problems with the handling of the cells. For example, if the drop, here at 10.000 cells, is not detectable, this may hint at burst droplets and intermixed cells, which could render the whole experiment useless. Not shown here, and depending on the protocol, this HTML QC report contains statistics on an additionally executed antibody capture. In Figure 4.2b additional analyses are automatically performed. The user can query and visualize the expression levels of specific cell clusters, which were found using k-means clustering. Usually, the data are displayed using an additionally performed t-SNE [311] visualization, whereas nowadays UMAP [210] is used in a productive environment. As already mentioned, using the Barcode Rank Plot, it is possible to estimate which barcodes represent valid cells, and which contain spurious information only, and hence can be regarded as background. The output from cellranger provides all barcodes and counts, as well as a filtered version, with only the valid barcodes, or cells, included. In most cases, the filtered data is to be used for further analysis using the scRNA-seq analysis framework of choice.

The general computational analysis workflow takes the expression matrix or matrices from cellranger, undertakes additional filtering steps (e.g. for mitochondrial or ribosomal fractions), integrates multiple datasets and finally scales and normalizes expression values. On these, dimensionality reduction is performed using principal component analysis (PCA). The principal components are used for clustering the cells and deriving a 2D-embedding for visualization (e.g. using UMAP or t-SNE). After this step, more detailed analyses can be executed, such as the identification of marker genes for each cluster, which can then be used for cell type prediction. But also in-depth differential analyses of specific clusters are often performed.



(a) (b)

Figure 4.2: **Cellranger QC report** of a scRNA-seq experiment. In (a) the mostly mapping related quality features are reported. Using the graphics, the number of detected and valid cells can be detected. This tells whether there have been problems with the handling of the cells, e.g. if the drop, here at 10.000 cells, is not detectable. In (b) additional analyses are automatically performed. By default, cellranger clusters cells by their expression values using a k-means clustering or t-SNE.

4.2 Cell Type Prediction from Expression Data (cPred)

Introduction

Of major interest in any scRNA-seq analysis workflow is the identification of cell populations. As part of a typical scRNA-seq analysis (see Figure 4.1), all identified cells are clustered. It is then assumed, that all cells within a detected cluster belong to the same population of cells, e.g. one cell type. In order to determine the cell type for such a cluster, it is necessary to derive actively expressed genes, which describe the cluster, so-called marker genes. From these marker genes, the cell type of a specific cluster can then be predicted. This can, for instance, be done using gene signatures. Gene signatures are, in general, lists of genes. For cell type prediction, for instance, this could be lists of genes present and (highly) expressed in specific cell types. Using an over-representation analysis it can be checked for which gene list, the marker genes are most over-represented [101]. The corresponding cell type of this list is then predicted for the cluster.

In bioinformatics, there are two common tasks which are performed with respect to gene signatures: (1) the creation of gene signatures [236], and (2) the usage of gene signatures in applications, ranging from the detection of, for instance, specific cancer types [187], over the prognosis of disease outcome [346] to the prediction of how effective a specific treatment might be [49]. Over-representation analyses are commonly used for the evaluation of such gene signatures. Using, for instance, the hypergeometric test, it can be checked

whether selected genes are over-represented in a gene signature, compared to all measured genes [46]. However, these approaches, which are also implemented by the DAVID web service [69], STRING DB [212] or PathCase [87], have one problem in common: they compare gene sets with gene sets and thus rely on thresholds which control both gene lists' lengths. Common thresholds are applied to the absolute fold-changes, or the (adjusted) p-value. These thresholds are of high importance, because it is known that the overlapping probability depends on the length of a gene list [98]. Given ranked gene sets, a correlation test (like Spearman's correlation) can yield better results than the hypergeometric test [242, 295]. Additionally, there are other statistical tests possible, or scoring-based analyses, like DOSE [341] or GSEA [295]. The probably most famous database of gene (expression) signatures is MSigDB [185, 295]. Here, signatures for specific processes, miRNA targets, transcription factor targets, cancer or immunologic phenotypes are available. However, no cell type specific signatures are contained in MSigDB. Gene signatures are commonly used in cancer genomics. But the way they are applied, in the sense that sets of marker genes can predict the survival rate of cancer, needs to be rethought. It was found that most random gene expression signatures are significantly associated with breast cancer outcome [313].

Nonetheless, the use of gene signatures for the prediction of specific conditions receives a revival with the advent of scRNA-seq techniques. During the analysis of such data, clusters of cells are produced, in which all cells show a similar expression pattern. Hence, the question is posed, what kind of cells are these? In order to answer this question, several tools have been developed by the community. The Single-Cell Signature Explorer [243] takes lists of genes (e.g. KEGG [148] or other gene signatures) as input and scores these against all cells of the dataset, by putting the amount of marker gene UMI-counts into relation with all expressed UMI counts of that cell. This way a cell type is assigned to each cell. For a cluster of cells, the cell type can then be derived, e.g. by the most frequent cell type within the cluster. In contrast to this approach, which takes existing gene signatures to estimate how well a signature matches a specific cell, Torang et al. [308] present a method to obtain new cell type signatures from scRNA-seq data. Using an elastic-net logistic regression approach they identify gene signatures for immune cells. The particular problem with immune cells is that even though they derive from a common progenitor cell, the individual cells differentiate into distinct cell types. Due to their common ancestor, it is extremely difficult to discriminate the individual cell types. The authors provide a classifier and gene signatures which can be applied on custom data. Another method to assign cell types to whole clusters within scRNA-seq analysis is presented by the authors of PanglaoDB [94]. The authors build for each cluster and each possible cell type a score, which consists of the expression of a marker gene, but also weights the specificity of the gene for this cell type (derived from other experiments) and the frequency of this gene appearing in cell type signatures. This, so called, down-weighting of overlapping genes [301] is meant to improve gene set analysis. By applying their algorithm to a whole database of (automatically evaluated) scRNA-seq experiments delivers also an incrementally updating database of (possible) marker genes for specific cell types. Some are canonical, curated, markers, others are predicted. Marker genes are listed by cell type and tissue, allowing the identification of not only the cell type, but also the tissue the cells are from. The

implementation of this method, however, is not available to the public. More recently the SingleR method for cell type identification was published [13]. For each cell (or cluster) the gene signature lists are ranked, compared with a reference and sorted out until only two possible cell types remain. Instead of using all genes, genes are restricted to a set of variable features between all cells. SingleR is similar to PanglaoDB in not relying on fixed sets of markers, like those published in literature, or listed in text books. The authors rely on reference datasets where cells were annotated manually. These reference datasets are then compared to the actual data in order to assign cell types.

There are few databases which focus on the provision of marker genes for cell types. Among these are, as already presented, PanglaoDB, and CellMarkerDB [343]. CellMarkerDB lists marker genes, which were found for a particular cell type (which can be accessed via an ontology) in a specific experiment. For each dataset it is annotated whether it is from a cancer or normal tissue, which is useful and important. Finally, there exists a commercial service, CellKB[3], which integrates several experiments and makes them available by organism, cell type, publications or even disease. This platform, however, does not offer the possibility to download signatures without a paid subscription.

While there are multiple resources for obtaining cell type predictions for scRNA-seq data, these methods rely on manually curated reference datasets, focus on a specific framework to be used in or are limited to a specific experimental type. In this work a method for scRNA-seq cell type prediction, cPred, is developed, which differs from existing tools. cPred is independent of the used analysis framework, but integrates with Seurat [43], scanpy [330] and `pIMZ` (Chapter 4.4). It integrates PanglaoDB and CellMarkerDB databases for marker genes such that users must not first create or find own sets of cell type specific genes or respective reference data sets. As such, it operates independently of the experimental type, and works for both scRNA-seq marker gene lists, and for IMS marker masses lists. This method was successfully applied in a recent publication *Vascular neutrophilic inflammation and immunothrombosis distinguish severe COVID-19 from influenza pneumonia* [228] and the submitted manuscript *Protective immune trajectories in early viral containment of non-pneumonic SARS-CoV-2 infection* [238].

Methods

Extracting Marker Genes Marker genes are extracted for each cluster, for instance, in scRNA-seq data using Seurat's *FindMarkers* function, which compares all genes for all cells of the active cluster with all other cells (background). The comparison test can be user defined, but the t-test works generally well. All other arguments are set to their default values (e.g. only test genes that are expressed in a minimum of 10% of cells, and have at least a logarithmic fold change (logFC) difference of 0.25).

All identified marker genes (regardless of their p-value or expression level) are then annotated with expression data. For each *marker* gene in each cluster the number of cells within this cluster and the number of cells expressing this gene in this cluster are annotated.

[3]https://cellkb.combinatics.com

For the gene expressing cells the five number summary (minimum, 0.25-quantile, median, 0.75-quantile, maximum) and the mean are calculated for the gene's expression values. This statistic is available both for the cluster cells and the background cells. The results for all clusters are then merged into a single data frame which can be exported to a tab separated file, which serves as input for the actual cell type prediction.

Data Sources and Features cPred can automatically retrieve cell type gene signatures from two sources. The default choice is PanglaoDB [94], whose regularly updated gene signatures for cell types are downloaded, decompressed, saved and read in, extracting all required values. The other source is the CellMarkerDB [343], which can be used for the cell type prediction using the `-cellmarkerdb` flag. In order to allow for reproducible predictions, both databases are downloaded and stored, and will only be overwritten if the user wishes to update the cell type gene signatures.

By default, the top 10 predictions for each cluster are printed. This number can be adjusted by the needs of the user. In order to improve the usability, e.g. the `-seurat` flag generates *R* code which can be used to annotate the found cell types directly in the *R* scRNA-seq data analysis.

In order to restrict the prediction to organs or tissues of interest, the user can specify a positive list of tissues or organs to consider exclusively.

Predicting Cell Types The prediction of the cell type for a specific cluster is achieved using a weighted sum. For each cell type j, and for each expressed gene g in a cluster k, which happens to be a marker gene (it will be named *accepted* gene), a gene score $\mathrm{GS}_{j,k,g}$ (Equation 4.1) is calculated. $\mathrm{meanSens}_{j,g}$ refers to the sensitivity with which the gene g is expressed in the cell type j, $\mathrm{meanSpec}_{j,g}$ refers to the respective specificity. Contributing to this score is also the prevalence of the gene in the cluster, $\mathrm{CP}_{k,g}$ — a measure that is deducted from the cluster annotation, namely the number of cells which express the gene and the total cells in the cluster. The average expression $\mathrm{avgExpr}_{g,k}$ relates to the mean expression of the specific gene in the cluster. The importance of the gene g in all reference clusters is defined as $\mathrm{impRC}_g = \frac{1.0}{1+log_2(|\{j|g \in MG_j \forall j\}|)}$.

The gene score is summed up over each accepted gene of a cluster such that the $\mathrm{totalScore}_{j,k}$ (Equation 4.2) for all significant marker genes MG_k of cluster k can be determined.

$$\mathrm{GS}_{j,k,g} = \mathrm{meanSens}_{j,g} \cdot \mathrm{avgExpr}_{g,k} \cdot \mathrm{CP}_{k,g} \cdot (1 - \mathrm{meanSpec}_{j,g}) \cdot \mathrm{impRC}_g \tag{4.1}$$

$$\mathrm{totalScore}_{j,k} = \sum_{g \in MG_k} \mathrm{GS}_{j,k,g} \tag{4.2}$$

$$\mathrm{clusterScore}_{j,k} = \mathrm{totalScore}_{j,k} \cdot \frac{\mathrm{accUnique}_{j,k}}{\mathrm{allUnique}_j} \cdot \frac{\mathrm{accGenes}_{j,k}}{\mathrm{allGenes}_j} \tag{4.3}$$

The final cluster score (Equation 4.3) expresses the score for cluster k being of cell type j, where $\mathrm{accUnique}_{j,k}$ refers to the accepted unique genes for cell type j in cluster k, and

allUnique$_j$ to all unique genes of that specific cell type. Furthermore, accGenes$_{j,k}$ is the number of accepted genes for cell type j in cluster k, and allGenes$_j$ is the number of all marker genes for cell type j.

Datasets for Evaluation The cell type prediction of cPred is evaluated on four gene expression tables from two experiments. The first scRNA-seq experiment originates from human and mouse atherosclerotic aorta and can be accessed through NCBI GEO [22, 82] accession GSE131780 [329]. The analysis was conducted for each species separately, as well as using an integrated (combined) approach, resulting in three distinct expression tables. The second experiment is within the COVID-19 context of immune cells isolated from bronchoalveolar lavage fluid (BALF) and can be accessed through GEO accessions GSE145926 [183] and GSE128033 [217].

All datasets were downloaded from GEO and processed with Seurat [43, 293]. After integrating the single datasets (human and mouse for GSE131780, all patients for GSE145926 and GSE128033), the data were normalized and scaled with default parameters. After PCA and neighbour-finding the results are visualized using UMAP [210]. Using the described marker gene extraction method, marker genes were extracted and used for cell type prediction.

While cPred can make use of several resources for cell type prediction, the primary source is PanglaoDB, because cell types are annotated with their tissue residency, and with their sensitivity and specificity of being a marker gene for each cell type in each tissue. The presented results were calculated using the PanglaoDB cell markers retrieved in April 2020. For the purpose of cell type prediction for the human/mouse atherosclerotic plaque dataset, only cell types within expected tissue types *Smooth muscle, Vasculature, Connective tissue, Immune system, Heart, Epithelium* were considered. Likewise, the context for the BALF dataset consisted of *Immune system* and *Lung* cells.

For the comparison with SingleR (version 1.3.8, [13]), the celldex (version 0.99.1, [13]) dataset ImmGenData (IGDT, [120]) and HumanPrimaryCellAtlasData (HPCA, [191]) were used. Particularly the ImmGenData reference set is specific to immune cells, while the other reference set additionally contains more distant or (here) irrelevant cell types. Both datasets were identified as the best matching ones, containing all expected cell types. Cell type prediction with SingleR was conducted in both supported modes: per cluster and per cell. Cluster-level predictions are derived by assigning the most frequently observed cell type within a cluster.

Availability cPred is available from GitHub `https://github.com/mjoppich/scrnaseq_celltype_prediction` including documentation. cPred adheres to the FAIR principles because it is findable and easily accessible due to its documentation. The software is interoperable because common tab-separated files serve as input. It is reusable because its source code is publicly available and easily extendable.

Results and Discussion

The presented scRNA-seq cell type prediction application (cPred) consists of two steps. The first step generates expression data from a Seurat [43] or scanpy [330] scRNA-seq object. The main objective here is to aggregate all relevant information and to condense all necessary information into a single one. Specific focus was led on a fast computation. Particularly for datasets with numerous cells the aggregation of the per gene statistics can take some time, if the sparse, column indexed, expression matrix is not handled correctly. With large datasets the transformation of the sparse expression matrix into a dense matrix is no option due to memory restrictions. Hence, the matrix is transformed into a data frame which allows fast filtering of gene expression values. In the second step, the actual cell type prediction using the cPred method is made. The cPred method follows the general principles of a weighted sum scheme, similar to the one presented by PanglaoDB [94]. However, there are several differences in the way the weighted sum is calculated. In particular, the cPred weighted sum approach makes more use of gene down-weighting [301] in the calculation of the $\text{clusterScore}_{j,k}$. The principle of gene down-weighting refers to designing gene weights such that genes appearing in few gene sets are emphasized, while genes that appear in many gene sets are penalized. In the cPred method, gene down-weighting is implemented in the $\text{clusterScore}_{j,k}$ calculation by the two multiplied fractions for accepted unique marker genes, and found marker genes. These fractions are multiplied with the $\text{totalScore}_{j,k}$, because, only a high overlap of unique marker genes, or only a high overlap of all marker genes, is not sufficient for a cell type assignment. Both of these conditions should be met.

cPred is easy to use. For the creation of the gene expression per cluster predefined functions are available. The expression values are arranged such that the following prediction step can directly take place. Likewise, after the prediction, scripts are provided as output such that the user can directly transfer the result of the prediction back into the Seurat session. The actual prediction tool automatically downloads the required databases and makes them accessible. The user does not have to first arrange files, nor does the user have to care about finding or even creating databases or annotated reference datasets first. Cell type prediction becomes a copy and paste endeavour.

The resulting cell type prediction for the joint human and mouse dataset of atherosclerotic aorta is visualized in a UMAP (Figure A.34). In the following, the results from the presented cell type prediction method cPred are compared to the gold standard (Table 4.1). The assignment was made according to the published marker genes given in the original literature [329]. For two clusters the original author did not provide a cell type (clusters 14 and 20). Cluster 14 is found to be CD68_{low} and CD11c^{+} while also being positive for FLT3. It can thus be assumed, that these cells represent dendritic cells [42]. However, given the broad range and high similarities between macrophages and dendritic cells, the assignment remains difficult. The other unnamed cluster is cluster 20, which is only present in the human cells. This cluster expresses APOD, DCN and LUM. The latter two are identified as fibroblast markers in heart tissue by Muhl et al. [220]. Tsukamoto et al. describe that APOD may be part of a protective response of myocardium (e.g. fibroblasts) to vessel [309]. This cluster is thus assigned the fibroblast cell type. Comparing the cell type predictions

of the combined dataset to the gold standard reveals two key results: the agreement with the human and mouse datasets is high, and most cell types are correctly assigned. Cell types are assigned correctly for 18 of the 24 clusters for the combined (human and mouse) dataset. In one case (cluster 2), fibroblasts are predicted instead of smooth muscle cells. For cluster 11, cPred predicts smooth muscle cells, where originally pericytes were annotated. While pericytes are the secondary cell type prediction for that cluster, one notes an ACTA2 up-regulation in this cluster, which clearly hints at a vascular smooth muscle cell origin: ACTA2 is a marker for smooth muscle cell differentiation [256]. The cell type of this cluster thus may not be fully clear. Cluster 19 is originally annotated as neurons, however, the cPred prediction of adipocytes seems plausible as well. Fibroblasts can migrate into adipocytes [5, p. 1228], and it is known that adipocytes can exhibit fibroblast-like behaviour [141]. For cluster 21 cPred predicts either NK cells or gamma delta T cells. Both are similar to the originally annotated T cells. The epithelial cells in cluster 22 are not identified correctly. Here, endothelial cells are predicted. The annotation for cluster 16 is of major interest, since this is the cluster with the newly identified fibromyocytes, according to the original authors. Fibromyocytes are cells which develop from a contractile smooth muscle cell towards a fibroblast phenotype [329]. cPred predicts fibroblasts or chondrocytes for this cluster. Indeed, chondrocytes are already known to occur in atherosclerotic lesions, being involved in the calcification of atherosclerotic plaque [28]. Moreover, it is known that fibroblasts can differentiate into chondrocytes and vice-versa [5, p. 1228]. Likewise, smooth muscle cells can differentiate into chondrocytes [28, 124], too. With this information it does not seem unlikely that chondrocytes can express characteristics of both, fibroblasts and smooth muscle cells. The cell population the original authors describe as fibromyocytes could well be chondrocytes, with smooth muscle cell origin. This would match with the GO process the authors identified being up-regulated in this cell population, *Osteoblasts/clasts and chondrocytes in RA (rheumatoid arthritis)*. This brief literature research poses at least the question, whether fibromyocytes are distinct cell types, or whether it only describes a (trans-)differentiated cell of the connective tissue. Overall, the non-matching predictions at least identify closely-related cell types, and the available data, combined with literature research, does not rule out the predicted cell types.

A similar evaluation was performed for the single cell immune landscape of bronchoalveolar lavage fluid (BALF) of COVID-19 patients [183]. In contrast to the previous analysis, fewer distinct cell types are expected, but mainly lymphoid or myeloid derived cells. The UMAP representation of the data for the BALF dataset is shown in Figure 4.3, split by disease stage.

For this evaluation the prediction results from cPred and SingleR are compared (Table 4.2). A prediction is classified as exact if the predicted cell type matches the observed one. An approximative match is assigned if the predicted cell type does not match the actual one exactly, but the prediction is somewhat plausible, e.g. if monocytes are predicted for a macrophage cluster. Finally, more distant predictions are classified as incorrect.

In this benchmark the results from cPred with PanglaoDB gene markers are compared against SingleR predictions on a per-cell and per-cluster basis using the HPCA and IGDT reference datasets. Cluster-wise predictions in the per-cell prediction run were determined

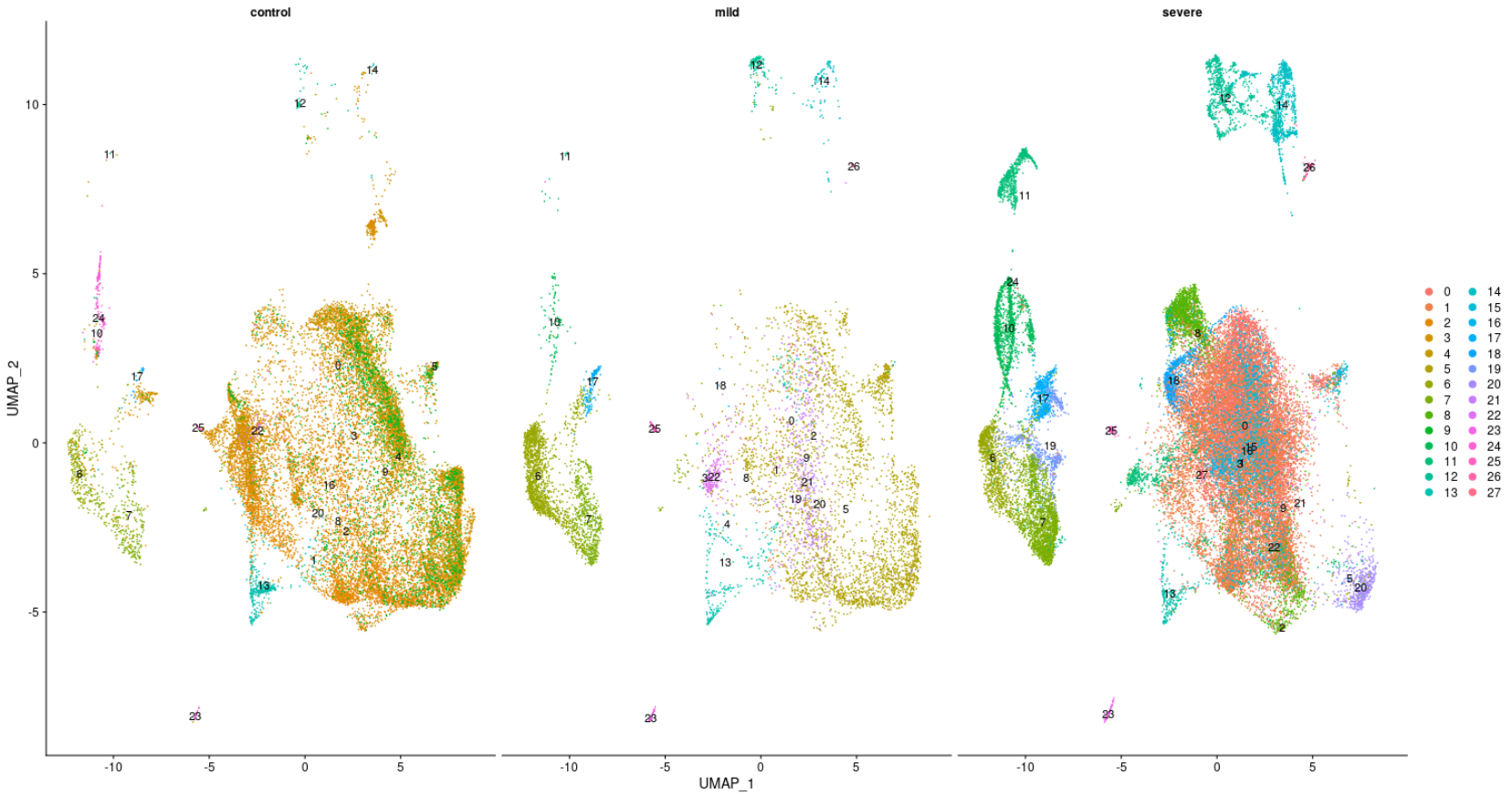


Figure 4.3: **cPred: UMAP for GSE128033 dataset (scRNA-seq analysis of BALF)** split by disease stage (control, mild, severe). A total of 28 clusters were detected, with a majority of the cells being in one large cluster (on the right), which represents cells of myeloid origin or macrophages (according to [329]).

by majority-vote of all cluster cells.

In general, the cell type prediction with cPred shows useful results, both using the context-sensitive version and the global one. The context-sensitive cPred prediction achieves 15 exact matches and thereby achieves a better result than SingleR. The (global) prediction using no context definition assigns 14 clusters correctly, like the best SingleR prediction using the IGDT reference on cluster-level predictions. Focusing on the context-sensitive evaluation it must be noted that for 7 of the 13 inexact assignments, the cells were annotated as macrophage in the original paper, but monocytes were predicted using the cell type prediction. For these clusters macrophages were predicted as secondary prediction. In clusters 10 and 24 gamma delta T cells are predicted instead of regular T cells. In cluster 16 neutrophils are predicted instead of macrophages. Two assignments were more severely incorrect. In cluster 14 mast cells were predicted, which are annotated as ciliated epithelial cells in the original dataset. In cluster 27 neutrophils were predicted, but plasma cells are annotated. An analysis of the scoring shows that for instance for cluster 27, IGHG4 is given as distinct marker for plasma cells. However, this gene is listed as marker for B cells in Panglao DB, hence plasma cells were down-weighted. Particularly immune cells have only few unique marker genes due to their ancestry. Then such a shared gene enhances the effect of the down-weighting strategy. It is surprising though that the original paper does not report any undifferentiated monocytes. Some predicted monocyte clusters could thus actually be monocytes, considering that the assignment in the original paper was performed using the CD68 antigen, which is expressed on all monocytes, macrophages, neutrophils, basophils and large lymphocytes according to literature [222]. In depth analysis of expressed marker genes suggests a monocytic origin for clusters 3-5: $CD74^+$, $CD81^+$ (4+5) or $CD9^+$ (3+4). It is interesting to see differences between the cPred predictions with and without context filtering. These results are an effect of down-weighting, because cPred calculates cell type-specific marker genes on only the selected tissue types. Considering approximative matches, both cPred predictions perform better than SingleR, achieving up to 86% correctly assigned clusters. The context-sensitive cPred prediction performs best, both for approximative matches and exact ones.

The results obtained through cPred are compared to one of the recent competitors, SingleR [13]. In comparison with the results of SingleR, cPred predictions show a higher variability in cell types, even among the quite similar cells of myeloid origin. For instance, SingleR seems to have problems in distinguishing T cells from (differentiated) monocytes in this setup, which has not been a problem for cPred. On the contrary: cPred predicted various monocytic successor cell types for what should be macrophages. A total of 14 clusters were assigned incorrectly by SingleR on the IGDT data set with per cluster predictions. This is worse compared to cPred, but in a similar range. Taking also approximate matches into consideration, cPred takes the lead with 24 approximatively correct assignments. The rather poor performance of SingleR on this dataset might be explainable by biases induced due to the use of other sequencing techniques in contrast to the reference datasets. Particularly immune cells are rather hard to handle and to sequence, potentially inducing additional biases to the data, ultimately making datasets not easily comparable. Because SingleR relies on reference datasets, only therein contained cell types can be predicted. When

using cPred it is possible to restrict the search space to cell types from specific tissues only, allowing a context-sensitive prediction.

Therefore, cPred has several advantages compared to SingleR and other state-of-the-art software. First, it does not require raw data for cell type prediction, but can be used directly on derived expression data. The calculation of these data/gene lists is done for any scRNA-seq experiment that is collected by PanglaoDB, and that output can directly be used. No further compute-intensive steps are required, as compared to SingleR, for instance. With cPred a context-aware exploration of the data can be performed, simply by defining the context, e.g. in terms of tissue types to consider in the analysis. cPred does not require specific reference data, but operates on gene lists, which are directly available from two resources, Further resources can easily be integrated.

Conclusion

Cell type prediction is an important task within any scRNA-seq experiment. It gives researchers a clue which cell types might be of interest, or behave abnormally. Still, identifying cell types in general is not an easy problem. Most times abnormally small p-values are calculated for marker genes, genes which define a specific cluster. Going through these lists manually and searching for specific markers is extremely unwarranted work. Working with predictions, which can be eventually checked much easier, helps a lot. It is important to have a solid prediction tool available, which not only tries to predict the correct cell type, but also delivers the reasoning. cPred has the ability to print the top n predictions, and for each prediction the accepted unique or cell type-specific genes are printed. This makes it easy to curate the predictions manually, helping with the decision on whether a prediction is correct, or should be refused. For one analysed dataset, consisting of about 60 000 cells and 28 clusters (BALF dataset), the prediction takes approximately one minute and is bound on the number of clusters. With cPred an initial cell type prediction is done fast. For both publications where cPred was already applied [228, 238], only little manual curation at the progenitor level was needed.

The cPred predictions evaluated on both datasets matched well, with differences to the GOLD standard usually being on the progenitor or successor level. cPred performs better than SingleR on the BALF dataset. In fact, it could be learned that already differentiated cells may change their differentiation again, but often keep genes associated with previous differentiations expressed. This irritates prediction methods, but with cPred it can be understood due to the open reasoning of the tool. However, even with manual curation the cell type could not be clearly derived from expression data alone, and possibly only additional histological analyses can shed light onto the actually observed cell type.

It is extremely important to note that cPred works on differentially expressed marker genes for specific clusters. This opens new ways for using cPred, because it does not matter where these marker genes came from, whether from a bulk RNA-seq experiment with sorted cells, a scRNA-seq experiment, or possibly even proteomics data. The latter will be exploited in the next section.

Table 4.1: **Cell type prediction for human/mouse atherosclerotic plaque** single cell data. Empty fields origin from cell types not detected, or with only few cells. Predictions across human, mouse and the combined data set are consistent. Predictions deviate where human and mouse predictions are not the same, and one cell source is more prevalent than the other. The gold standard was inferred from the marker genes determined by the original authors [329].

#	Human	Mouse	Combined	Gold standard
0	Fibroblasts	Fibroblasts	Fibroblasts	Fibroblasts
1	Smooth muscle cells	Smooth muscle cells	Smooth muscle cells	Smooth muscle cells
2	Smooth muscle cells	Fibroblasts	Fibroblasts	Smooth muscle cells
3	Macrophages	Macrophages	Macrophages	Macrophages
4	Smooth muscle cells	Smooth muscle cells	Smooth muscle cells	Smooth muscle cells
5	Endothelial cells	Endothelial cells	Endothelial cells	Endothelial cells
6	Fibroblasts	Fibroblasts	Fibroblasts	Fibroblasts (2)
7	T cells	T cells	T cells	T cells
8	Fibroblasts	Fibroblasts	Fibroblasts	Fibroblasts
9	Fibroblasts	Fibroblasts	Fibroblasts	Fibroblasts
10	Pericytes	Pericytes	Pericytes	Pericytes
11	Smooth muscle cells	Smooth muscle cells	Smooth muscle cells	Pericytes (2)
12	B cells		B cells	B cells
13	Plasma cells		Plasma cells	Plasma cells
14	Dendritic cells	Dendritic cells	Dendritic cells	Dendritic cells (original Cluster 11)
15	Macrophages	Macrophages	Macrophages	Macrophages
16	Fibroblasts	Chondrocytes	Chondrocytes	Modulated cells
17	Endothelial cells	Endothelial cells	Endothelial cells	Endothelial cells
18	Endothelial cells	Endothelial cells	Endothelial cells	Endothelial cells (2)
19	Endothelial cells	Fibroblasts	Adipocytes	Neurons
20	Fibroblasts		Fibroblasts	Fibroblasts (original Cluster 18)
21	NK cells	Gamma delta T cells	Gamma delta T cells	T cells
22		Fibroblasts	Endothelial cells	Epithelial cells
23	Mast cells		Mast cells	Mast cells
Exact (abs. / rel.)	19 / 0.83	14 / 0.70	**18 / 0.75**	
Approx (abs. / rel.)	20 / 0.87	15 / 0.75	**19 / 0.79**	

Table 4.2: **Cell type prediction for BALF immune cells** The cPred predictions are mostly consistent with the gold standard. For many cells, presumably tagged as macrophages in the original publication, the monocyte state is predicted.

		cPred		SingleR/HPCA		SingleR/IGDT	
#	**Gold**	**PanglaoDB**	**PanglaoDB /w context**	**Cluster**	**Cell**	**Cluster**	**Cell**
0	Macrophages	Monocytes	Monocytes	Keratinocytes	Neutrophils	Macrophages	Macrophages
1	Macrophages	Monocytes	Macrophages	Keratinocytes	Neutrophils	Macrophages	Mast cells
2	Macrophages	Dendritic cells	Macrophages	Epithelial	Macrophages	Macrophages	Basophils
3	Macrophages	Dendritic cells	Monocytes	Epithelial	Astrocyte	Macrophages	Mast cells
4	Macrophages	Dendritic cells	Monocytes	Epithelial	SMCs	Macrophages	Mast cells
5	Macrophages	Kupffer cells	Monocytes	Macrophages	Macrophages	Mast cells	Basophils
6	T cells	T cells	T cells	T cells	T cells	Macrophages	Macrophages
7	T cells	T cells	T cells	Gametocytes	T cells	Macrophages	Macrophages
8	Macrophages	Macrophages	Monocytes	Keratinocytes	Macrophages	Macrophages	Mast cells
9	Macrophages	Macrophages	Macrophages	Keratinocytes	Macrophages	Macrophages	Mast cells
10	T cells	T cells	Gamma delta T cells	Neuroepithelial	Erythroblast	Macrophages	Macrophages
11	Plasma cells	Plasma cells	Plasma cells	Gametocytes	B cells	Macrophages	Macrophages
12	Ciliated Epithelial	Ependymal	Ciliated cells	Epithelial	Epithelial	Macrophages	Epithelial
13	mDC	Dendritic cells	Dendritic cells	Gametocytes	B cells	Macrophages	Macrophages
14	Ciliated Epithelial	Epithelial	Mast cells	Epithelial	Epithelial	Epithelial	Epithelial
15	Macrophages	Monocytes	Macrophages	Keratinocytes	Neutrophils	Macrophages	Mast cells
16	Macrophages	Macrophages	Neutrophils	Keratinocytes	Neutrophils	Macrophages	Epithelial
17	NK cells	NK cells	NK cells	Keratinocytes	NK cells	Macrophages	Macrophages
18	Neutrophils	Monocytes	Monocytes	Keratinocytes	Neutrophils	Macrophages	Macrophages
19	T cells	NK cells	T cells	T cells	NK cells	Macrophages	Macrophages
20	Macrophages	Kupffer cells	Macrophages	Macrophages	Macrophages	Macrophages	Mast cells
21	Macrophages	Monocytes	Monocytes	Dendritic cells	Macrophages	Macrophages	Basophils
22	Macrophages	Monocytes	Monocytes	Epithelial	Neuroepithelial	Macrophages	Macrophages
23	B cells	B cells	B cells (memory)	B cells	B cells	Macrophages	Macrophages
24	T cells	Dendritic cells	Gamma delta T cells	Neuroepithelial	MEP	Macrophages	Mast cells
25	pDC	pDC	pDC	B cells	B cells	Macrophages	Macrophages
26	Secretory Epithelial	Alveolar type II	Alveolar type II	Epithelial	Epithelial	Epithelial	Epithelial
27	Plasma cells	Plasma cells	Neutrophils	Gametocytes	Gametocytes	Macrophages	Epithelial
	Exact (abs. / rel.)	14 / 0.50	**15 / 0.54**	8 / 0.28	14 / 0.50	14 / 0.50	5 / 0.18
	Approx (abs. / rel.)	22 / 0.79	**24 / 0.86**	8 / 0.28	14 / 0.50	14 / 0.50	5 / 0.18

4.3 MALDI Imaging Mass Spectrometry

Imaging mass-spectrometry (IMS) is an emerging sub-discipline of regular mass spectrometry (MS). While IMS can not yet deliver the same throughput and resolution as currently seen by regular LC-MS/MS, its advantage is the conservation of spatial information: it is recorded at which location specific spectra have been recorded, and subsequently it can be determined at which location specific peptides, lipids or proteins have been detected. In general (see schematic in Figure 4.4), the to be measured, solid, sample is fixated on a sample holder and coated with a matrix layer. This matrix aids in laser analyte desorption and ionization [226]. The size of the measured pixels depends on the ablation area of the laser beam as well as the distance between measurement points. The desorbed ionized singly charged molecules are then reflected and guided to the mass analyser for recording the spectrum.

While regular bulk-measurements do not provide spatial information, these data are still important to complement the spatial resolution measurements. In fact, such bulk measurements allow a deeper understanding of proteomic dynamics than IMS alone.

IMS is an emerging technology to capture both proteomic data and spatial information. This combination allows the correlation and combination of imaging data (e.g. from microscopic images) and spectral information. Hence, it is possible to draw conclusions on the correlation of specific masses and cellular or tissue structures. IMS data are getting used more frequently in a wide area of applications. Schulz et al. demonstrate the use of this technique in drug development and pharmaceutical research [273]. More specifically, Spraker et al. describe several ways how IMS can be used in natural product discovery [287]. For instance, IMS helped to discover a novel bioactive lipopeptide produced by the plant pathogenic bacterium *Ralstonia solanacearum*, which can cause morphological shifts in fungi [288, 289]. On the topic of chemical heterogeneity across bacterial colonies, Pessotti et al. used MALDI-IMS to visualize metabolites by small assemblages of bacterial cells and could thus show how these can differentially produce metabolites in response to local chemical gradients [241]. Similarly, Schleyer et al. showed that bloom-forming algae perform a metabolic shift towards odd-chain fatty acid lipids during viral infection [266]. Finally, Patterson et al. used IMS to produce a three-dimensional model of lipids in atherosclerotic plaques. These examples show the broad area where IMS can be a useful technique to resolve spatial processes.

While there are many areas where IMS can be applied to, toolboxes for the analysis of MSI data are rare. There exist several stand-alone software like Mirion [234] or the MATLAB-based MSiReader [29], which allow an analysis using specific applications with a GUI. On the other hand there exists a predefined workflow for IMS analysis [92] for the popular Galaxy server [102]. Bemis et al. presented an R package for the statistical analysis of IMS data [23], and Veselkov et al. [315] implement a python-based analysis suite for efficiently processing of hundreds of IMS datasets. Unfortunately this analysis suite is not further maintained. Many of these toolboxes focus on the calculation of principal components to perform any analysis on this reduced space. Such methods are collected by Verbeeck et al., providing a collection of existing unsupervised learning methods [314].

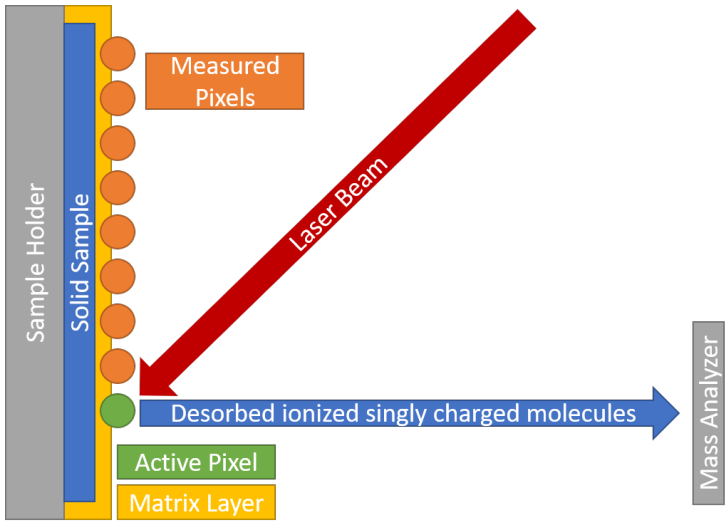


Figure 4.4: **MALDI-TOF IMS schematic** A laser beam hits the solid samples covered with the matrix layer, desorbs ionized singly charged molecules and guides these to the mass analyser for the measurement of the spectrum for the active pixel.

4.4 A Framework for Imaging Mass Spectrometry Data Analysis (`pIMZ`)

Imaging mass-spectrometry (IMS) is a new technology in the area of proteomics, and is increasingly used for broad biological and clinical applications. It allows the simultaneous measurement of hundreds of analytes within a specific m/z range together with their spatial distribution. Particularly the varying m/z range, which allows the measurement of analytes such as lipids, peptides or whole proteins, makes it a versatile and powerful measurement tool (the preparation of the sample may depend on the performed measurement). The most common measurement technique for IMS, and the one applied in the context of this work, is MALDI (matrix-assisted laser desorption/ionization) in combination with a time-of-flight (TOF) mass analyser. Since IMS experiments characterize the chemical composition of biological samples (e.g. tissues) at spatial resolution, their application area ranges from botany samples [287], over clinical research [273] to forensic investigations [169, 170].

Two typical goals of the bioinformatics analysis of IMS data are image segmentation, which partitions a tissue into regions of homogeneous spectral profiles, and image classification, which assigns locations of the tissue to pre-defined classes, based on their spectral profiles [23, 245]. Commonly these two tasks are combined: first the tissue is partitioned into regions of homogeneous spectra, which are then assigned to pre-defined classes. However, this task sounds more simple than it is: due to the large and complex nature of the datasets, but also due to the biological and more so technical variation, these tasks are quite difficult. As with any spectrometry based method, particularly the variation in the intensity of the measured spectra has to be taken care of. Frequently, measured spectra

suffer from m/z shifts or require baseline correction in the preprocessing. While there exist several closed-source commercial solutions for the analysis of IMS data, there is no open-source package, which goes beyond this clustering step. Most packages finish with the identification of specific masses, but do not allow comparing multiple IMS datasets, or assess differential masses to detect the measured proteins. These, however, are important for detecting differences in complex human disease contexts.

In order to overcome this challenge, pIMZ[4] is presented: an open-source, python-based framework for a fast, integrated and convenient analysis of IMS data. pIMZ differs from existing software packages such that it is open-source, is not within the R universe but is python-based and specifically designed for IMS analysis. In contrast to commercial tools, such as SCiLS Lab (SCiLS[5]), flexImaging (Bruker[6]), HDI (Waters[7]) and TissueView (AB Sciex (out of production)[8]), pIMZ is open-source. Regarding existing R packages for IMS data analysis, like Cardinal [23], MALDIquant [103] or the recently published rMSIproc [250], pIMZ is going further than spectral segmentation and classification, allowing a comparative view within the spectra of one sample and between spectra from other samples or measurements. It also integrates methods which create a better understanding of the data, e.g. by annotating gene names to masses, or predicting cell types for identified clusters. pIMZ differs from existing python packages like pyBASIS [315] in such that it promotes an interactive, (jupyter) notebook-based analysis, similar to that of scRNA-seq frameworks like Seurat [294] or scanpy [330]. It must be added that at the research unit bioinformatics pyBASIS could not be started due to runtime exceptions.

Moreover, pIMZ allows the researcher to perform a high throughput analysis of the data, but can additionally be used to explore the data in detail. This allows to explore the researcher's degree of freedom during the performed analysis. By providing these methods, researchers can directly estimate the reliability of important single results and thus estimate the robustness of the results, e.g. for developing further hypothesis or performing distinct experiments.

pIMZ enables a python-based analysis of IMS data, outgoing from data in the imzML format [269] and performing differential and integrative analyses of one or many regions. The most important steps are shown in Figure 4.5. After IMS data extraction from the imzML file, first the raw data must be normalized. With the normalized spectra, cluster analysis can be performed. pIMZ offers both supervised and non-supervised methods to create a clustering. In addition to classic approaches, pIMZ employs strategies known from scRNA-seq analysis: dimensional reduction via UMAP [210] and clustering via HDBSCAN

[4] pIMZ stands for python-based Imaging Mass Spectrometry, where the latter is abbreviated by MZ, which stands for m/z, the mass-to-charge ratio for which intensities are measured in mass spectrometry experiments.

[5] https://scils.de/

[6] https://www.bruker.com/products/mass-spectrometry-and-separations/ms-software/fleximaging/overview.html

[7] https://www.waters.com/waters/de_DE/HDI-Imaging-Software/nav.htm?locale=de_DE&cid=134833914

[8] https://sciex.com/products/mass-spectrometers/tof/tof-systems/tof/tof-5800-system

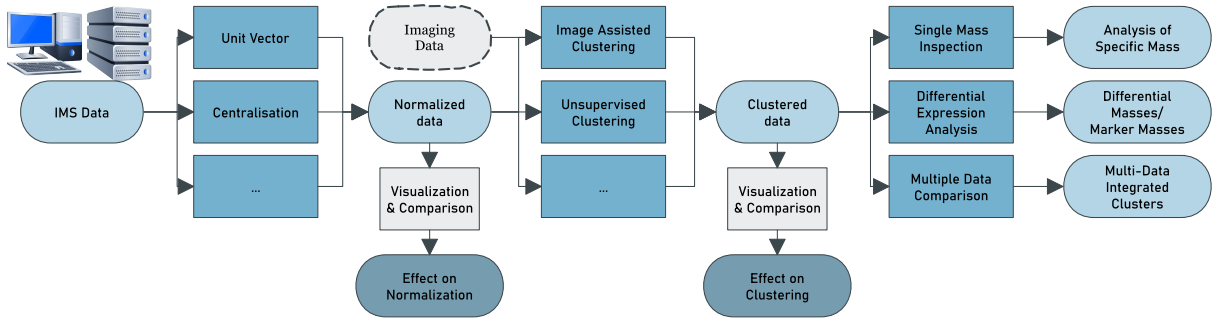


Figure 4.5: **pIMZ pipeline and analysis steps** This flow chart displays the steps available within pIMZ to get from the raw IMS data in imzML format to differential masses and marker masses. On this path, multiple visualizations and comparisons of sub-results are possible and supported by pIMZ.

[45]. Finally, pIMZ offers many visualization possibilities and means to explore single masses. Using DE analysis, high-throughput analyses of many masses can be performed at once.

Materials and methods

Nomenclature Within the following description a specific terminology is used. One input imzML-file corresponds to a measured slide. If there are multiple, unconnected areas of measurement on a slide, the single areas will be referred to as regions. In a biomedical setting, there will be multiple sections on a slide, which correspond to these regions, respectively. After clustering, the regions may be subdivided into clusters, of which some can be classified as background. In general, the background cluster will be assigned ID 0.

Data Input and Processing Wherever possible, the pIMZ framework relies on already existing and maintained packages. A list of used packages is presented in Table 4.3.

Using the *IMZMLExtract* class imzML-files can be loaded and regions extracted. To enable this, first all regions within an imzML-file must be identified, which is achieved using the image labelling function provided by SciPy [317]. From this labelling it can be determined which pixels belong to each region, as well as which dimensions each region has. This is used to extract the region as a numpy [116] 3D array, with the m/z values in the third dimension. The x,y coordinates refer to the position of the pixel in the measured area. Any unmeasured pixel (e.g. because a non-rectangular area was measured) is replaced by a 0-vector. For peak binning and peak calling, pIMZ relies on the capabilities of the ms_peak_picker library.

[9] https://github.com/mobiusklein/ms_peak_picker
[10] https://www.globus.org/

Table 4.3: **List of libraries used in pIMZ** pIMZ internally uses several already published libraries for several purposes, like loading imzML-files, perform numeric operations or statistical tests.

Library	Purpose
pyimzml	Load imzML-files [269].
skimage	Imaging methods [312].
numpy	Numeric operations [116].
ms_peak_picker	Identify m/z peaks if no common m/z bins were used [9].
ctypes	Include C++ library for parallel calculations (part of python).
matplotlib	Visualize results [132].
biopython	Calculate theoretical protein masses [61]
diffxpy	Perform differential expression tests [330].
scipy	Clustering algorithms [317].
umap	Dimensionality reduction for clustering [210].
hdbscan	Find clusters [209].
dabest	Visualize detailed mass intensities per cluster [123].
globus_sdk	Access to globus [10] enabled data storage, e.g. HuBMap [130].

Normalization Within the pIMZ toolbox several normalization methods are implemented: (1) divide by maximal intensity within spectrum, (2) divide by maximal intensity within region, (3) divide by maximal intensity within all regions and (4) by unit vector. More sophisticated methods, which are mentioned as best-practice in the pyBASIS framework [315], are supported by pIMZ, too. These are the (5) *intra_median* and (6) *inter_median* normalization. For the *intra_median* normalization, the *median fold change* procedure described by Veselkov et al.[315] is followed: 'each mass spectrum is normalized to its median fold change ion current by dividing each peak intensity within a mass spectrum by the median fold change between all peak intensities of the same spectrum and the reference spectrum, which typically is chosen to be median profile'. Likewise, the *inter_median* normalization is implemented: 'Each dataset is normalized to its median fold change ion current by dividing each peak intensity of a dataset by the median fold change between the peak intensities of the same dataset profile and reference dataset profile' [315].

Clustering Currently, pIMZ supports 9 different unsupervised clustering techniques. These include UPGMA, UPGMC, WPGMA, WPGMC, WARD and k-means clustering methods (implemented by SciPy [317]), which take as input the pairwise similarity matrix calculated via the pIMZ C++ module. The C++ module is integrated into python using the ctypes package and operates on a numpy matrix. Using OpenMP [65] `parallel for` shared-memory programming, all pairwise cosine similarities are calculated. There are other clustering methods implemented, which do not rely on the similarity matrix, but perform a dimensionality reduction using UMAP [210] first. The actual cluster-finding, e.g. using

the density based clustering method HDBSCAN[45] or WARD, is then performed on this reduced space.

Differential Expression analysis In order to generate a map from protein ID to expected mass, first the genomic annotation is read using the `gffutils` package [11]. Then, for each transcript its RNA sequence is determined, which is translated using `biopython` [61]. From the protein sequence, the `molecular_weight` function calculates the protein's mass once using the regular protein weights as well as using the monoisotopic weights. It is free to the user which variant should be used for mass-to-protein assignments.

The differential expression analysis is conducted either via the methods provided by diffxpy [330] or via nlEmpiRe[12] and operates on the intensity values of the masses of the compared regions.

Data Access Input data for `pIMZ` are usually provided in the imzML-format. Additionally, `pIMZ` has a module to access data which is stored via Globus [13], a platform for the storage of huge research data. Using the `HubMAPDownloader` data stored in Globus, e.g. data provided by the HuBMAP consortium [130], can be accessed easily. Currently, there are 13 different MALDI IMS data sets published with HuBMAP[14].

Data The presented use-case data were obtained from the Soehnlein lab[15]. These data stem from sectioned arteries of mice harvested at different time points. All mice underwent the same high-fat diet and developed atherosclerosis.

The measurement of the slides was performed using MALDI-TOF with single charged proteins ($z = 1$). Thus, the m/z value equals the protein mass in Dalton (Da) and is used interchangeably here[16].

Availability The `pIMZ` framework is available online `github.com/mjoppich/pIMZ` or via the python package manager pip `https://pypi.org/project/pIMZ/`. By providing `pIMZ` through these resources it becomes easily findable. Through it extensive documentation, available online `https://pimz.readthedocs.io/en/latest/`, `pIMZ` becomes easily accessible. The framework is interoperable by using a common input file format, and providing common outputs, which can serve as input for other tools, like cPred (Chapter 4.2). By providing the source code and the documentation, `pIMZ` also becomes easily reusable and extendable. `pIMZ` thus adheres to the FAIR principles.

[11] https://github.com/daler/gffutils

[12] Csaba, Gergely. Personal Communication. 2019.

[13] https://www.globus.org/

[14] Data Accessed 2021/02/07. https://portal.hubmapconsortium.org/search?mapped_data_types[0]=MALDIIMSpositive&entity_type[0]=Dataset

[15] Söhnlein, Oliver. Personal Communication. 2020.

[16] Lahiri, Shibojyoti. Personal Communication. 2019.

Results and Discussion

First the general idea of the pIMZ framework is presented, together with the most important features. In the second part of this section the functionality of pIMZ is demonstrated at the example of the atherosclerotic mice arteries use-case.

Package Overview The pIMZ framework consists of four distinct functional parts. The first part is the IMZMLExtract class, which implements any functionality related to the input. This includes reading data from the imzML file, including the normalization of the input data. The second part, implemented in the SpectraRegion class, provides all operations on a single region, e.g. clustering and differential analysis. The third part complements the second part: a C++ library which performs the similarity calculations in parallel for performance reasons. The fourth and last part of functionality is the comparison of multiple regions, which involves normalization and differential analysis methods, made available through the CombinedSpectra class. All analysis steps are accompanied by relevant plotting capabilities.

Data Input Using the IMZMLExtract class imzML-files can be loaded and regions extracted. During the provision of the region spectra, the user can demand to recalibrate the spectra to the null-line. This assures that the minimal measured intensity is 0.

In general, pIMZ assumes that all spectra have common m/z values. For data sets where this is not true, a method to call peaks and bin m/z values is provided, which relies on the ms_peak_picker library. Furthermore, there are means to shift the spectra on the m/z-axis in order to achieve a better 'alignment' of the peaks. In order to support lipid-data, pIMZ supports the subtraction of a mean or media spectra from all spectra of a region. This might be useful if the used coating of the sample was measured in the spectra itself.

In order to assess the quality of a dataset, as well as to gain a first overview of the sample, the IMZMLExtract component allows to detect IMS regions, visualize detected regions and extract spectra from these regions. In addition, IMZMLExtract provides multiple spectra-wide plots like the total ion current.

Clustering Having the normalized spectra for a region obtained, further processing can take place. This is performed by the SpectraRegion component of pIMZ. The SpectraRegion provides means to calculate pair-wise spectra distances, cluster spectra using several unsupervised methods or from supervised data, visualize and filter the obtained clusters, calculate and visualize representative cluster spectra, intra- and inter-cluster differences as well as plot the intensities for specific masses using DABEST plots [123]. In addition, the SpectraRegion can orchestrate the calculation of differential comparisons.

After obtaining the initial clustering, it is important to fine-tune the results. Within pIMZ, the cluster ID 0 is reserved for background clusters. Hence, using the SpectraRegion component, clusters which are in the border regions, which only consist of a few pixels (so-called singletons), or which are fully surrounded by other clusters (so-called islands), can

be assigned to the background. Such filtering is particularly needed if the actual sample is embedded in a specific tissue, e.g. liver.

Differential Expression Using a utility script `pIMZ` can calculate the theoretical masses of all protein coding transcripts. These can be used by the `ProteinWeights` component, which allows the mapping of masses to gene or protein names. A protein name is assigned to a (differential) mass, if the distance between the marker mass and any available mass of the protein is less than a given threshold. This threshold accounts for potential post-translational modifications. The required protein to mass table can either be derived from the gene annotation files on a purely computational base, or from experimental data (LC-MS/MS).

Similar to the processing of scRNA-seq data, the `SpectraRegion` allows the calculation of marker masses: masses of a cluster (or a set of clusters) which are differentially compared to a specific background (e.g. all other spectra, or all but the embedding tissue). Using the EmpiRe package[17], a comparison of clusters using empirical distributions of the spectral intensities can be performed. Even though this package is originally designed for bulk LC-MS/MS measurements, it allows to consider each IMS pixel as a single replicate. Like with single cell data, the major problem with these tests is the number of replicates (=pixels), which slows computation down. The similarities between IMS and scRNA-seq data, a high replicate count and missing values, allow the use of the python package *diffxpy* for differential analysis, which was originally developed for scRNA-seq analysis within scanpy [330]. It provides further differential tests, like a t-test or a rank sum test.

Multiple Data Comparison Usually multiple IMS measurements are performed at once. Being able to compare these measurements thus is an important feature for `pIMZ`. With the `CombinedSpectra` class `pIMZ` implements this possibility. This component arranges several comparative visualizations, but most importantly makes several `SpectraRegion` objects comparable: the distinct clusters from the `SpectraRegion` are aligned (e.g. by similarity of the median spectra), and intensity values are normalized along multiple measurements using the *inter_median* normalization as alread described. It additionally provides means for differential expression between the single `SpectraRegion`s.

Demonstration of the `pIMZ` framework at the example of atherosclerotic mouse arteries: a use-case

The functionality and applicability of `pIMZ` is discussed alongside the use-case of an atherosclerosis relevant data analysis. The histological microscopy images of the areas analysed by IMS are shown in Figure 4.6. The only difference between the samples is the time point at which the mice were harvested (8am vs. 12pm). Nonetheless, it can be seen that regions 0 and 4 (Figure 4.6ab) show a higher similarity to each other than regions 1 and 5 do (Figure 4.6cd), which are, again, quite similar to each other. This matches the

[17]Csaba, Gergely. Personal Communication. 2019-2020.

experimental layout: regions 0 and 4 stem from the same mouse and time point (8am), and regions 1 and 5 likewise (12pm). Nonetheless, it can be observed that regions 0 and 4 (Figure 4.6ab) show more intimal thickening (a thicker artery wall) than the other regions, which suggests a more advanced athero-progression in regions 0 and 4. Even though the mice were treated in the same way, such high biological variance is not totally unexpected[18].

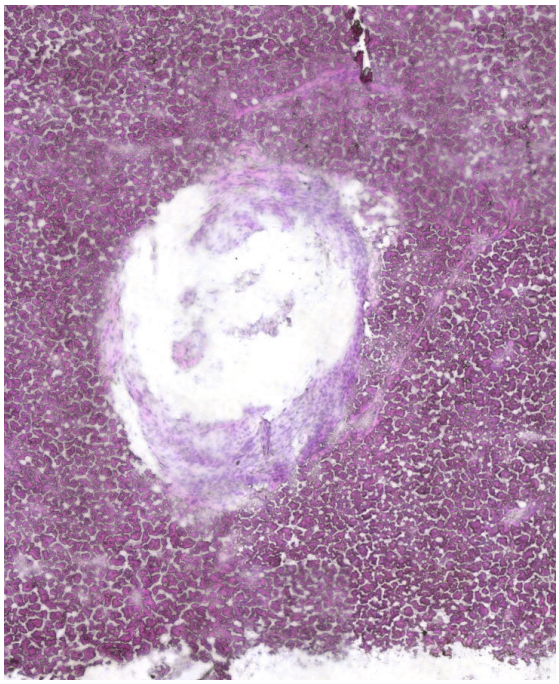
Loading Data Upon receiving the required input files (`imzML` description and `ibd` binary data file), the `IMZMLExtract` class can be used to load a measurement file and list all regions. The whole measurement layout can be displayed to identify the single regions with their respective IDs (Figure 4.7). As part of this use-case the four left regions (regions 0, 1, 4 and 5) are of interest. These correspond to the stained microscopy images shown in Figure 4.6.

As part of a quality control, the maximum peak location (m/z value) may signal a consistent shift in the data. For region 0 the maximal peak is usually around 4000m/z, with the exception of a few pixels (Figure 4.8a). In order to see whether the correct region has been measured, the total ion current, the sum over all intensities, can be helpful. The density of high values is higher in the middle of the shown region, surrounded by a circle of low intensity pixels (Figure 4.8b). This circle may show the boundary of the measured artery.

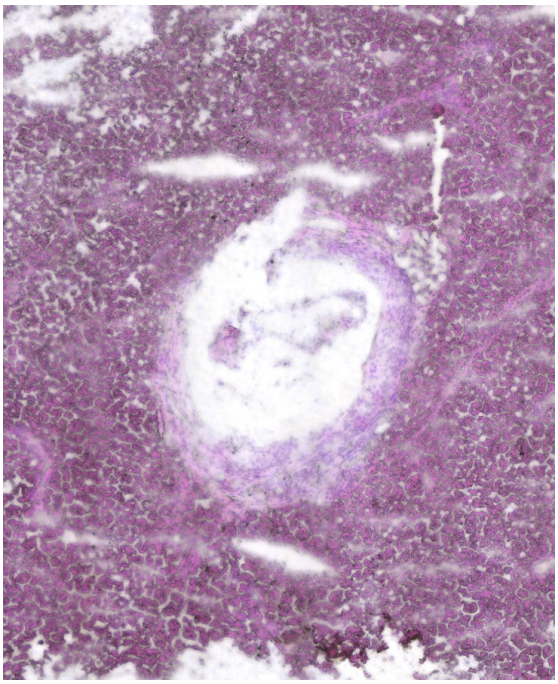
Normalizing Data After confirming that the measured region contains an object of interest, the processing of the input spectra continues with normalization. As a default processing, first the spectra are normalized using the *intra_median* and *inter_median* normalization technique. In the unnormalised data (Figure 4.9a) it can be seen that the average fold changes in comparison to the region's median spectra varies by pixel location. After normalization, the median fold changes of all shown pixels are set to a fixed level (Figure 4.9b). These normalization techniques were chosen because Veselkov et al. [315] describe them as most robust. The `pIMZ` framework implements further normalization techniques in addition.

Clustering a region After normalization, the input spectra can be used to build a `SpectraRegion`. The `SpectraRegion` orchestrates the clustering and differential analysis of a region and offers two ways of clustering: either from the UMAP-embedding or using pair-wise cosine similarities. For any UMAP clustering first the 2D embedding is calculated, on which clustering methods can be executed. This can either be a density-based method (HDBSCAN) or a hierarchical clustering (WARD). Using `pIMZ` both the 2D-embedding and the resulting clustering can be displayed (Figure 4.10ab). After the initial clustering further fine-tuning should be applied in order to gain more physiologically relevant structures (Listing 4.1). As a first step, background clusters can be merged. These clusters are recognized by being in the corners of the measured region, here within the 5x5 corner pixels. Then clusters which consist of only one pixel are removed (singletons), followed by

[18] Söhnlein, Oliver. Personal Communication. 2021.

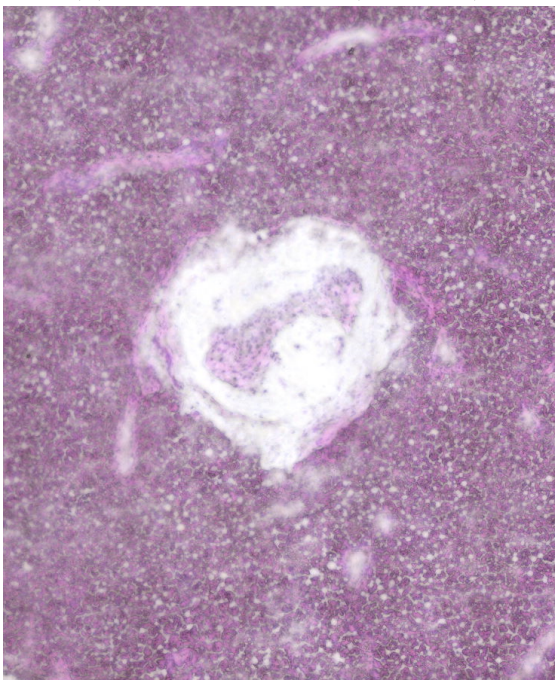


(a) Slide D, Region 0 (top left)



(b) Slide D, Region 4 (top right)

(c) Slide D, Region 1 (bottom left)



(d) Slide D, Region 5 (bottom right)

Figure 4.6: **`pIMZ` HE stained test data** The haematoxylin and eosin (HE)-stained sections used for testing `pIMZ`. Differences in the atherosclerotic plaque between the top slides (a and b) in contrast to the bottom ones (c and d) can be seen, particularly in the vessel wall (outside).

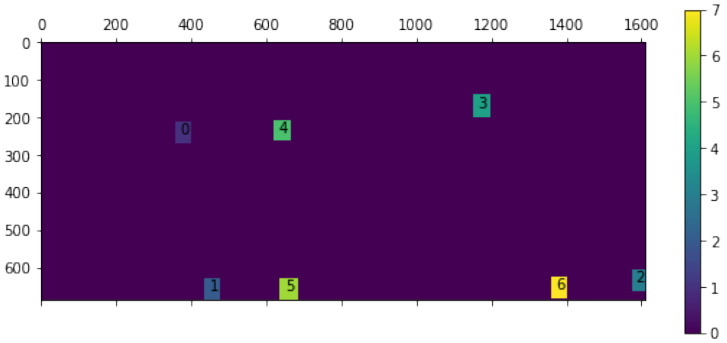
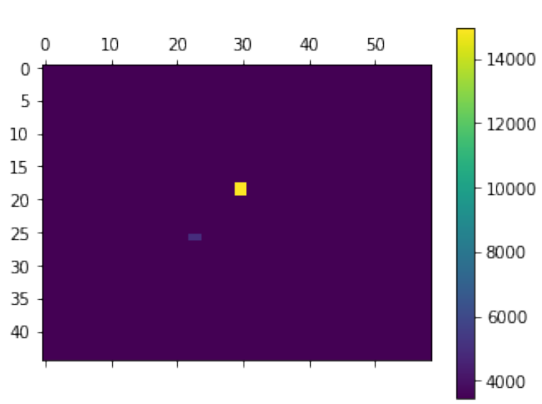
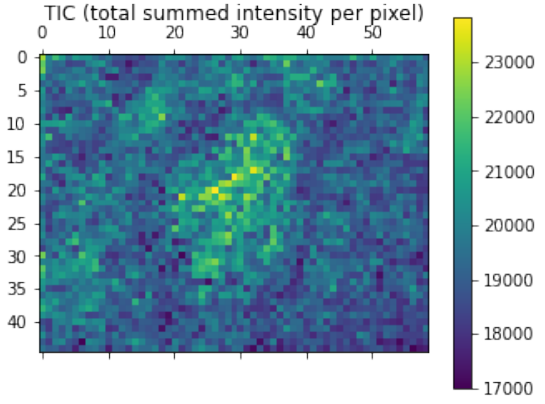


Figure 4.7: **pIMZ loading data (slide layout)** From the loaded `imzML` file all measured regions are listed. The background is unmeasured, and for all measured regions their IDs are shown within the region (0-6).

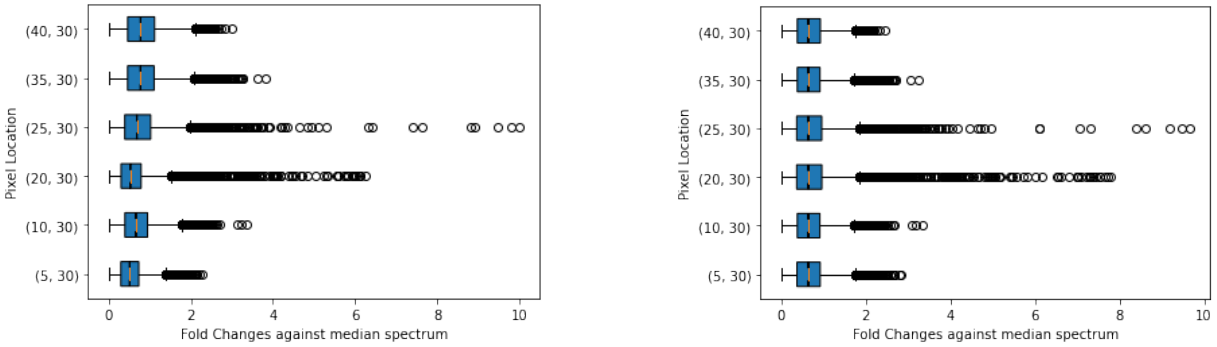


(a) Slide D, Region 0: Max Peak m/z



(b) Slide D, Region 0: Total Ion Current (TIC)

Figure 4.8: **pIMZ loading data (maximum peak and TIC plots)** From the maximum peak's position (a) it can be seen whether there is a systematic shift within the spectra. Here, no such systematic shift can be observed. From the total ion current (TIC, sum over intensities of all masses (b)) pixels with large signals can be identified. In the best case, the shape of the analysed object becomes visible. For the artery it can be seen that the inner parts have a high TIC, while the border/vessel wall shows a low TIC.



(a) Slide D, Region 0: Unnormalized.

(b) Slide D, Region 0: Inter-normalized

Figure 4.9: **pIMZ loading data (normalization plot)** For 5 points from within the artery the m/z intensity fold-change against the median spectra of the whole region is aggregated as box plot. It can be seen that without normalization (a) the points' median fold-change varies, whereas after inter-normalization (b) all fold-changes have a similar median fold-change.

isolated regions (islands). Finally, cluster 10 is identified of likely being background and thus manually added to the background (Figure 4.10bcd).

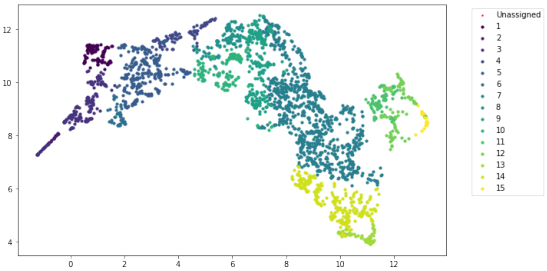
For the hierarchical clustering approach the calculation of all pairwise similarities is needed. This is accomplished by the `cIMZ C++` library for fast and parallel (OpenMP [65]) calculation of all pair-wise cosine similarities. Using the OpenMP shared-memory approach ensures a faster result than using numpy and allows for real multi-threaded calculations. Using a hierarchical clustering like WARD ensures that exactly k (here: 15) distinct clusters are formed (Figure 4.11a). After post-processing the clustering using the same filtering techniques as described for UMAP (Listing 4.1), the final clustering is created (Figure 4.11b). Differences between the UMAP and WARD clustering can easily be spotted, however, the general structure resembles each other in both approaches.

For the remaining use-case, the WARD clustering results are used. After obtaining the clustering results, the similarity of the single clusters can be calculated based on the median spectra of each cluster. This similarity can be visualized using a regular heatmap (Figure 4.12a). Clusters 0 and 8 are the most unsimilar ones. This makes sense as the background is liver tissue, in contrast to the artery sample. Cluster 8, in the centre of the artery, could contain infiltrates from the blood, increasing the heterogeneity. The similarity between clusters 13 and 15 is interesting, yet expected, as this seems to describe the wall of the artery (but also includes some plaque area). The within cluster similarity (Figure 4.12b) can be visualized by aggregating the similarities between all pixels of a cluster. Particularly for all non-background clusters there is a high similarity between all cluster-pixels, with a median similarity above 0.95.

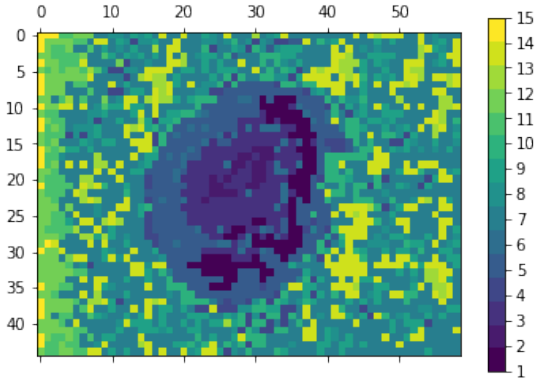
Single Mass Analysis With `pIMZ` single masses can be analysed (Figure 4.13). This is relevant, if the user is interested in one (or few) specific masses. As already noted, cluster 8

Listing 4.1: Python commands for filtering clusters after initial clustering and for fine-tuning the clustering.

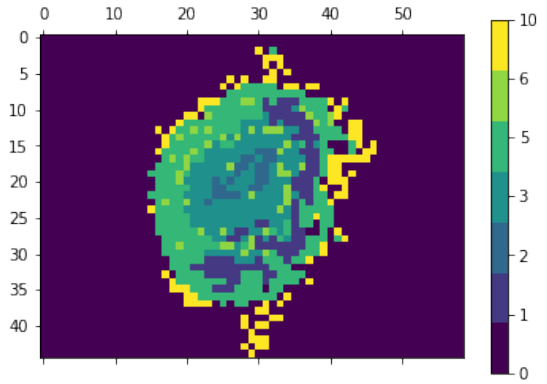
```
#filtering clusters
spec.filter_clusters(method='merge_background', bg_x=5, bg_y=5)
spec.filter_clusters(method='remove_singleton')
spec.filter_clusters(method='remove_islands')
#setting cluster to background
spec.set_background(10)
```



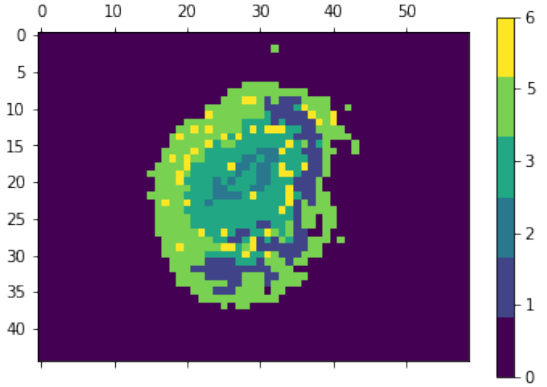
(a) Slide D, Region 0: UMAP embedding.



(b) Slide D, Region 0: HDBSCAN clustering

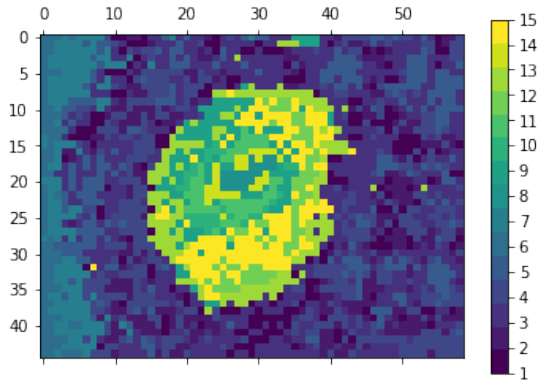


(c) Slide D, Region 0: Clustering after filters.

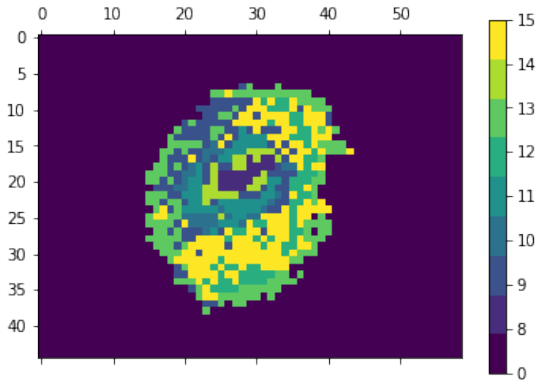


(d) Slide D, Region 0: Final clustering

Figure 4.10: **pIMZ clustering (UMAP clustering)** pIMZ supports UMAP for dimensional reduction and subsequent density-based clustering (a). The resulting segmentation within the original region (b) can be plotted. Additional filtering steps can be used to remove background clusters (c). Manual curation is supported. Here cluster 10 is assigned to the background manually (d).

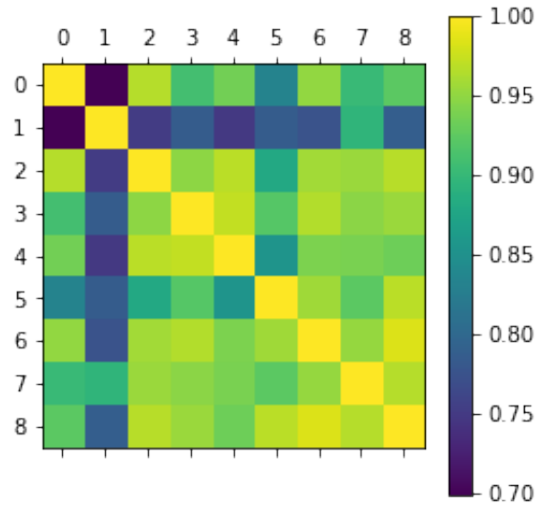


(a) Slide D, Region 0: Hierarchical clustering on pairwise similarity (WARD).

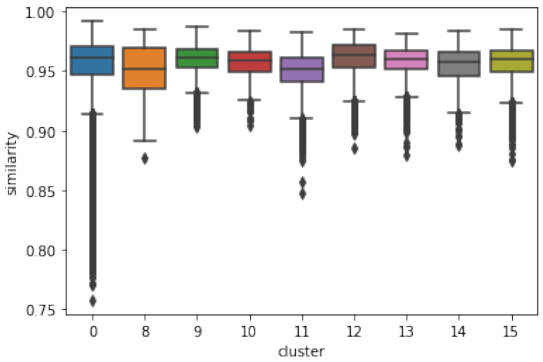


(b) Slide D, Region 0: Final clustering.

Figure 4.11: **pIMZ clustering (hierarchical clustering)** `pIMZ` supports several hierarchical clustering methods. Using WARD clustering on the pairwise similarity values (a) a first segmentation is calculated. Using several filters (Listing 4.1), background clusters are identified and merged (b).



(a) Slide D, Region 0: Cluster similarity.



(b) Slide D, Region 0: Within cluster similarity.

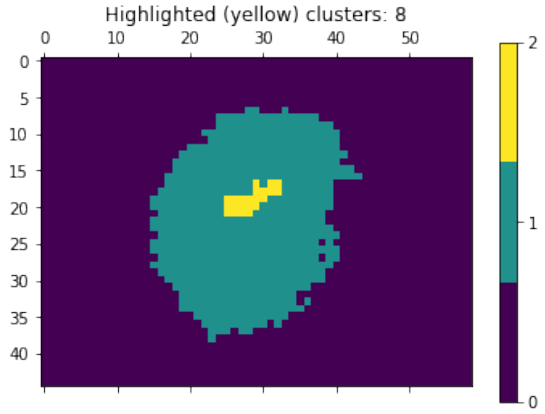
Figure 4.12: **pIMZ clustering (cluster similarity)** `pIMZ` allows the calculation and visualization of the similarity between clusters (a), based on the consensus similarity of the median spectra per cluster. In order to understand whether the clusters contain similar spectra, the within cluster similarity plots all pair-wise similarities per cluster (b).

is of special interest, because it shows a high dissimilarity from the other artery-related clusters.

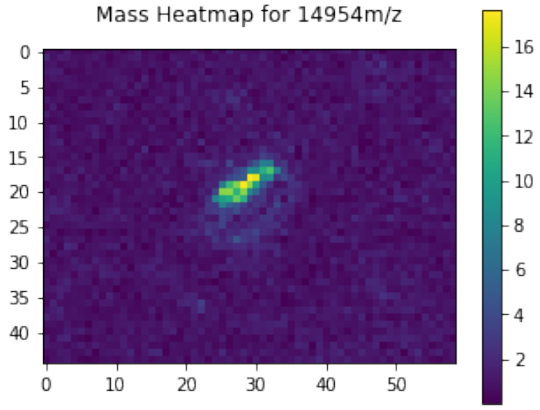
From recent COVID-19 related research it is known that Ifitm3 plays an important role in the immune response of peripheral blood mononuclear cells [275]. The Ifitm3 protein has a mass of $14954.18Da$. Upon looking at the mass heatmap of this specific mass (Figure 4.13b), it can be expected that Ifitm3 is found in cluster 8 of region 0 (Figure 4.13a). Using the `mass_dabest` function the intensity values per cluster are plotted as boxplot (Figure 4.13c) and as DABEST plot (Figure 4.13d), which allows a better inspection of the actual effect size. Indeed, the Ifitm3 prevalence is highly increased for cluster 8, but cluster 14 already shows a slightly increased Ifitm3 intensity. This matches the expectation.

Differential Expression Analysis With the existing methods it is possible to check the intensity values for a specific mass. However, there are many genes which have multiple associated masses due to multiple protein-coding transcript isoforms. Moreover, the user might not be interested in just one mass, protein or gene, but in many. A mapping from protein mass to gene is important for an efficient interpretation of any differential results in high throughput IMS datasets. For pIMZ, the `ProteinWeights` class provides such functionality. The `ProteinWeights` class takes either a tab-separated file with protein-name, gene name and associated mass as input, or can also read SDF-formatted files, which are particularly known in lipidomics. As already elaborated earlier, the reported masses can either stem from measured experimental data, or from theoretical protein mass predictions. Particularly in the latter case it might be possible that there exist multiple proteins stemming from different genes with the same (or a similar) mass. The `ProteinWeights` class thus contains a method to analyse such mass collisions (Listing 4.2). By default, and in this use-case, the mapping allows an error of $2m/z$ for each mass: for each mass (or m/z-value), any gene having an associated protein within $2m/z$ of this mass, is assigned to it. This threshold is meant to correct for post-translational modifications or the use of isotopes, as well as machine accuracy. The mapping file contains masses for 7 280 distinct protein names within the measurement range from $2990m/z$ to $30012m/z$. One protein (name) can have multiple annotated masses due to multiple isoforms. With the original threshold of $2m/z$, a total of 6 316 these proteins has a mass collision with another one. With a lower threshold ($1m/z$) the amount of affected proteins can be reduced to 4 881 proteins, and less than 2 collisions on average for all proteins. Yet one has to keep this inaccuracy in mind when interpreting high-throughput results, because the differential analysis is performed for each m/z value. Protein names are only assigned afterwards, based on the potentially ambiguous mapping from expected masses. This ambiguity can only be resolved by using experimental data, such that it is known which proteins are detectable in the sample. Still, even this approach is not guaranteed to remove all ambiguities.

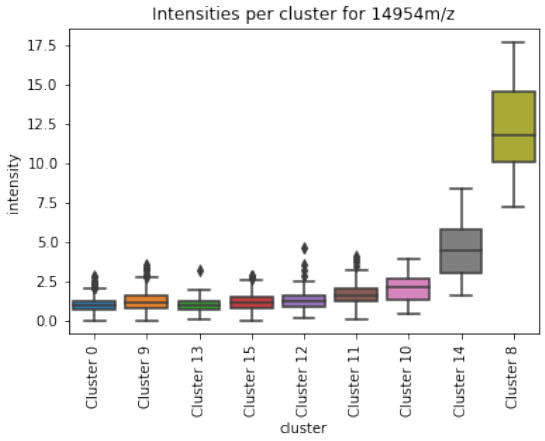
The calculation of all marker proteins is one of the key-features of `pIMZ`. With just one single call `pIMZ` will calculate all marker proteins and masses for a specific `SpectraRegion`, similar to Seurat's FindMarkers function [43]. A marker mass is similar to a marker gene in scRNA-seq: a mass which is differentially regulated in a specific cluster in



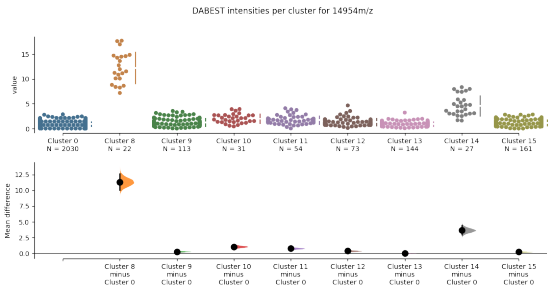
(a) Slide D, Region 0: Cluster 8 highlighted



(b) Slide D, Region 0: Mass heatmap 14 954 m/z



(c) Slide D, Region 0: Box plot of intensities for mass 14 954 m/z



(d) Slide D, Region 0: DABEST plot of intensities for mass 14 954 m/z

Figure 4.13: **`pIMZ` exploring data (single mass plots)** (a) `pIMZ` can highlight a specific cluster (here cluster 8) from the segmentation. (b) In the `mass_heatmap` a heatmap of the per-pixel intensities of a mass is visualized. (c) Using the `mass_dabest` function, a box plot of the per cluster intensities for a mass can be created, (d) together with a DABEST plot for effect size estimation.

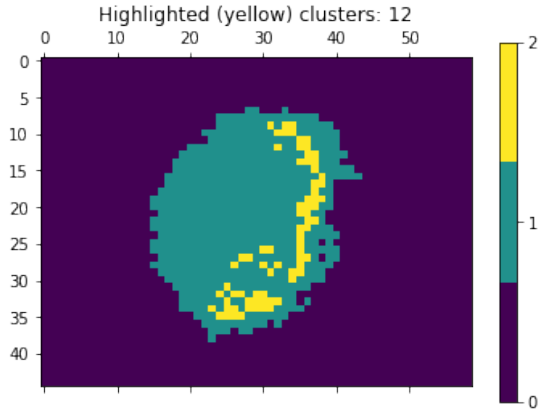
Listing 4.2: Protein mass collisions for theoretically derived protein masses with 2m/z and 1m/z thresholds.

```
pw_theo.print_collisions(maxdist=2.0, print_proteins=False)
ProteinWeights  INFO:          Number of total proteins: 7280
ProteinWeights  INFO:            Number of total masses: 10181
ProteinWeights  INFO: Number of proteins with collision: 6316
ProteinWeights  INFO:         Mean Number of collisions: 2.87
ProteinWeights  INFO:       Median Number of collisions: 2.0
ProteinWeights  INFO: Proteins with collision: [('Mbp', 30), ...]

pw_theo.print_collisions(maxdist=1.0, print_proteins=False)
ProteinWeights  INFO:          Number of total proteins: 7280
ProteinWeights  INFO:            Number of total masses: 10181
ProteinWeights  INFO: Number of proteins with collision: 4881
ProteinWeights  INFO:         Mean Number of collisions: 1.90
ProteinWeights  INFO:       Median Number of collisions: 2.0
ProteinWeights  INFO: Proteins with collision: [('Mbp', 19), ...]
```

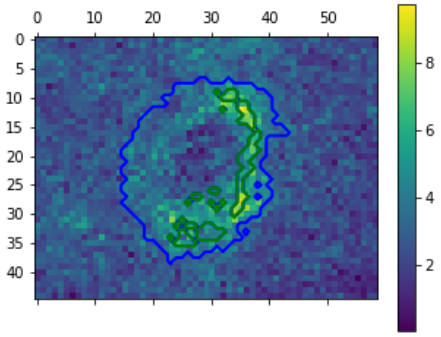
contrast to all other clusters. The user can decide whether the background cluster should be considered for this calculation. For `pIMZ` this method can simply be called by using the `find_all_markers` function. This function takes as input a `ProteinWeights` object in order to annotate a gene symbol to each differential mass. Since `pIMZ` offers the use of several statistical tests for the DE analysis, the user should also specify which tests should be used. In this example only the t-test was used.

For easier screening of the DE results, and to enable easy sharing with collaborators, the DE results can be exported as HTML-file using the `export_deres` function. This creates one HTML-file for each performed comparison (e.g. for each cluster), which consists of an overview of all regions (Figure 4.11b) as well as the selected regions (e.g. one cluster for which the marker genes are calculated, Figure 4.14a). The differential expression results are presented in a javascript-enabled table which allows sorting and filtering (by string or numeric value) of all entries. The generated table can be exported to a tsv-formatted file directly from the HTML web-page. Additionally, for each DE entry, a modified mass heatmap is shown, which shows the boundaries of the selected background (blue line) as well as the targeted pixels (green line), which here is cluster 12 (Figure 4.14b). It can be clearly seen that the shown mass, $15.271 m/z$, which only matches with a protein of the Il11 gene, is well up-regulated in cluster 12. Including these mass heatmaps in the shareable report allows an easier assessment of the results, and enables the user to identify false-positive results, e.g. stemming from the embedding or insufficient clustering.

After having calculated all marker proteins, the DE results are summarized in a single data frame, which can also be exported in various formats. This allows the integration of previously described methods, such as the cPred cell type prediction (Chapter 4.1). The cell type prediction is an interesting analysis which can be performed on the marker proteins. With the lower abundance of proteins, and the lower sensitivity of the MALDI



(a) Slide D, Region 0, Cluster 12: Selected pixels (yellow).



(b) Slide D, Region 0: Mass intensity (15269.878m/z) per pixel with background delimited by the blue line and selected region by the green line.

Figure 4.14: **`pIMZ` exploring data (DE analysis results)** After performing the DE analysis, `pIMZ` can export the results into an HTML table with sortable and filterable columns. For each identified mass, its annotated gene, the direction of regulation, the significance value as well as mean and median expression in the selected region and background (here clusters 8-15, except 12) are shown. (a) For cluster 12, (b) Il11 is upregulated (logFC 0.66) with an average intensity of 6.

IMS in contrast to scRNA-seq, the chance that cell type defining genes are detected is lower than for scRNA-seq. However, such marker proteins, cell type defining proteins, are expected to be highly expressed, which slightly increases the chance to be detected by IMS. There are various possibilities available for cell type prediction. The marker proteins can be calculated including or excluding background pixels. Cell types can be predicted via cPred on all available cell types, or only on expected cell types (context-sensitive). For the presented results (Table 4.4) the marker masses were calculated including the background, but the context was set to immune system cell types, and particularly leaving out *liver*. Liver tissue was only used as embedding of the aorta, and is not of interest for the further analysis. Because proteins originating from the liver, like ApoC1 (originating from liver and macrophages [95]) or ApoA2 (second most abundant protein of HDL, atheroprotective [146, 73]), play a crucial role in atherosclerosis, these must not be evidence for liver cells. Many liver-specific genes are lipid related and thus detectable in atherosclerotic plaque regions, which by definition has a pathogenic lipid-content [134].

The HE staining (Figure 4.6) shows white areas in the inner part of the artery. Yet, there are marker proteins for these regions found, and cell types predicted. This could be an artefact in the samples, where the tissue broke during sample preparation, leaving a protein smear on the slide. The predicted cell types for these regions (8, 9 and 14), could match such a scenario: B cells could originate from the blood stream, smooth muscle cells

and fibroblasts from the fibrous cap and basophils, adipocytes and macrophages from inside the plaque area. Fibroblasts play a crucial role in wound healing [193] and can thus be well expected at this location, too. Clusters 10 and 11 are adjacent to cluster 8, which explains why remains of smooth muscle cells can be observed. The co-localization of dendritic cells and adipocytes in the anticipated plaque area makes sense, as monocytes leave the blood, infiltrate the plaque and differentiate into antigen-presenting dendritic cells [229]. The cPred cell type prediction thus adds important information to the DE results and helps to find an interpretation of the results.

Comparative Integration of Multiple Regions The integration of multiple single data objects into one combined object, allowing inter-region comparisons is one of the key analyses in pIMZ. This integration of multiple `SpectraRegion` is accomplished by the `CombinedSpectra` class, which takes a dictionary or list of multiple regions as input. These regions must first be processed as described in the paragraphs above, in order to derive an initial clustering. Before any differential analyses on the combined spectra can be performed, some preprocessing steps should be undertaken. These steps are shown in Listing 4.3. A name for use in the combined object is assigned to the single region objects in lines 1-6. After normalization (line 10), the combined object compares the single regions' clusters (line 12), plots a heatmap of similarities (line 13) and derives 8 new clusters from the original ones (line 15), which can be visualized (line 16, Figure 4.16).

As a first step of integration, the single regions are normalized using the previously described *inter_median* normalization technique regarding the median (or mean) region spectra. Since all regions stem from arteries embedded in liver, it was assumed that the liver embedding does not change between the samples. Hence, the liver embedding is a suitable cluster on which the normalization factor can be calculated. After this normalization, the median fold-change between the region's background reaches 1 (Figure 4.15).

After normalization, the regions' clusters can be compared. As each region should consist of one artery, at different disease stages, there should be physiological elements which are in common, and where the clusters should be similar. The similarity comparison is calculated via the cosine similarity on the clusters' median spectra. The clustering itself is performed via the WARD algorithm. The resulting dendrogram is dissected to create 8 clusters (Figure 4.16). It can be seen that region 0 and 1 (slided_0 and slided_1) show a high structural similarity, but regions 4 and 5 (slided_4 and slided_5) are more heterogeneous. The background liver tissue clusters into the same new group (group 2), which shows, that this embedding is similar across regions. The suspected plaque areas in slides 4 and 5 are assigned the same new group (group 1). It is interesting to see that even slide 1 has a small plaque area, while this can not be observed in slide 5. With the inter-normalized intensity values, cross-region DE analyses can be performed.

Comparative: Whole Artery As a first comparison, the whole arteries from regions 0 and 1 are compared. This means that for all masses, all pixels from region 0 (Figure A.35a, yellow area) are compared with all pixels from region 1 (Figure A.35b, yellow area).

Table 4.4: **Cell type prediction in `pIMZ`** Using the cPred method (Chapter 4.1), cell type predictions for each cluster can be performed. The prediction is executed on the protein names inferred from the marker masses with tissues limited to: Connective tissue, Vasculature, Heart, Skeletal Muscle, Smooth muscle, Immune system, Blood, Epithelium. Immune system hits, smooth muscle and vasculature hits are expected. *Liver* was used as embedding only and is thus discarded.

#	Cell type	Score	Cell type specific marker	Cell type marker
0	Mast cells;Immune system	0.28	SLC31A	IL4, RGS13, PCP4L1, ...
0	Basophils;Immune system	0.12	DEFB40	CCL9, IL4, CCL4, ...
0	Basal cells;Epithelium	0.07	HEBP2	HEBP2, S100A14, RAB38
8	Macrophages;Immune system	9.99	CD209F, HILPDA, TYROBP	TREML4, CCL9, CCL6, ...
8	Smooth muscle cells;Smooth muscle	5.57	SEC24D, SSPN, MSRB3	SSPN, CRABP1, MSRB3 ...
8	Fibroblasts;Connective tissue	4.01	GREM1, ADAMTS10, MDK	GREM1, MGP, CLEC3B, ...
9	Adipocytes;Connective tissue	0.67	APOC1	APOC1, MT2, METRN
9	Basophils;Immune system	0.31	DEFB40	CCL4, DEFB40, IL4
9	Smooth muscle cells;Smooth muscle	0.29	SEC24D	SEC24D
10	Smooth muscle cells;Smooth muscle	6.14	SEC24D, MSRB3	CRABP1, RGS5, MSRB3, ...
10	Platelets;Blood	3.04	ATP8A1, MPIG6B, GPX1, OST4	RGS13, GPX1, NRGN, ...
10	Monocytes;Immune system	2.42	LY6C2, IFITM3, RHOC, MGMT	LY6C2, IFITM3, MGMT, ...
11	Dendritic cells;Immune system	3.47	IFITM6, CADM1	IFITM6, CR2, H2-DMA, ...
11	Smooth muscle cells;Smooth muscle	2.14	SEC24D	SEC24D, RGS5, PCP4L1, CRABP1
11	B cells naive;Immune system	0.96	TCL1	ADAM28, CR2, TCL1, MS4A2
12	Dendritic cells;Immune system	2.94	IFITM6, PDCD1LG2, CADM1	LST1, IFITM6, CD180, ...
12	Adipocytes;Connective tissue	2.43	APOC1	APOC1, DBI, CCL11, ...
12	Monocytes;Immune system	2.10	LY6C2, IL1RN, RHOC, MGMT	LST1, RHOC, CD40, ...
13	B cells naive;Immune system	0.11	TCL1	TCL1
13	Platelets;Blood	0.04	ATP8A1	ATP8A1
13	Nuocytes;Immune system	0.03		IL13
14	Macrophages;Immune system	2.90	HILPDA, TYROBP	CCL9, CCL6, CCL3, ...
14	B cells memory;Immune system	1.71	IL7	MS4A2, IL7, CD180, ...
14	B cells naive;Immune system	1.60	TCL1	MS4A2, TCL1, CD180, ...
15	Dendritic cells;Immune system	1.25	IFITM6, CADM1	LST1, IFITM6, CD180, ...
15	B cells naive;Immune system	0.72	TCL1	TCL1, CD180, CD37
15	Basophils;Immune system	0.67	DEFB4 0	CCL9, IL4, CCL4, ...

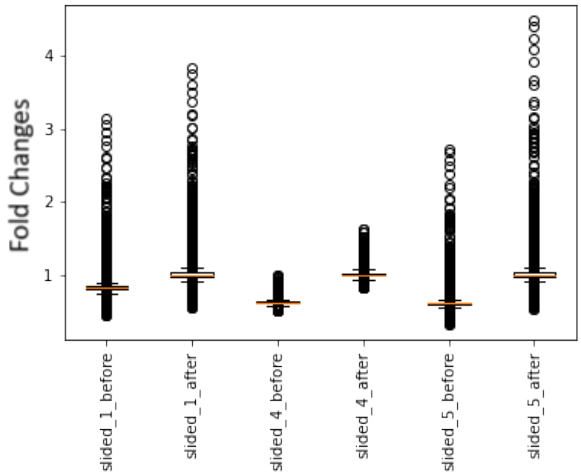


Figure 4.15: **pIMZ comparative (inter sample normalization)** Using the `CombinedSpectra` object, multiple `SpectraRegion` objects can be combined. This requires a normalization of the intensity levels, which is achieved using the *inter_median* normalization on the regions' backgrounds. After normalization, the median fold-changes between the region and the reference region (here: slided_1) is 1.

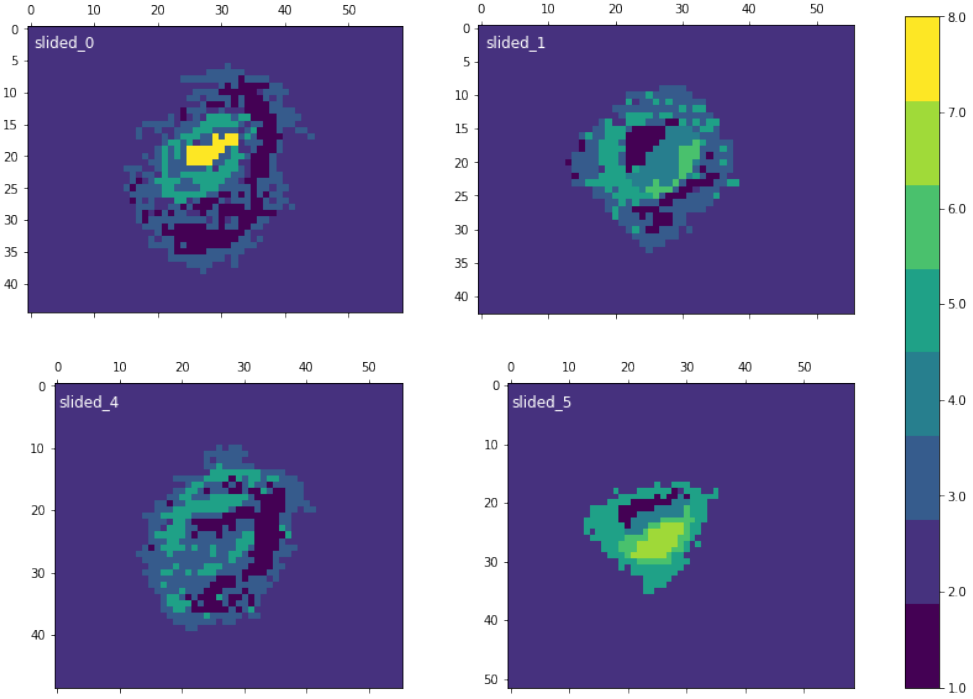


Figure 4.16: **pIMZ comparative (common segments)** From the normalized data, the consensus spectra for each region and cluster are calculated and compared pair-wise using the cosine similarity. With the WARD algorithm, the region-cluster-pairs are clustered and assigned new labels such that 8 new clusters are created. It can be seen that region 0 and 1 show a high structural similarity, but regions 4 and 5 have more heterogeneous clusters.

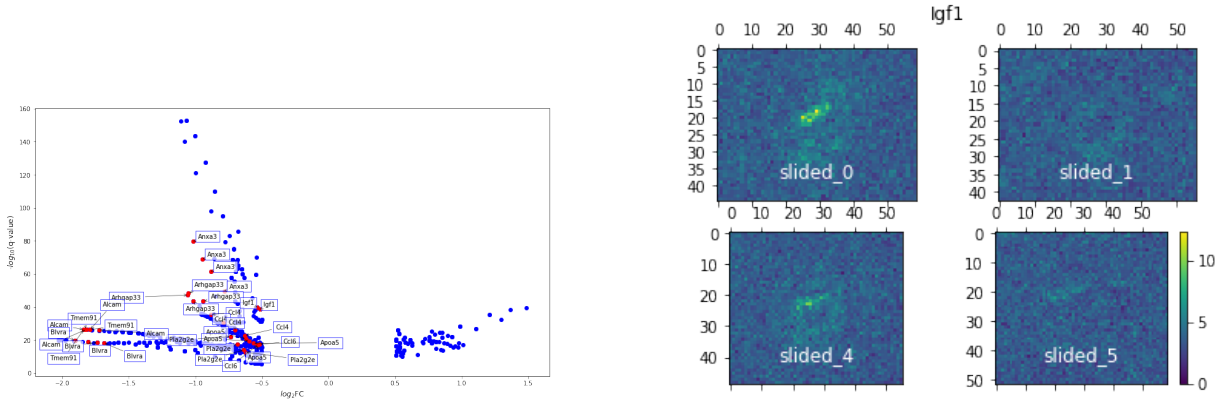
Listing 4.3: Python commands required to create and normalize the CombinedSpectra object for differential questions.

```
slided_0.name = "slided_0"
slided_1.name = "slided_1"
slided_4.name = "slided_4"
slided_5.name = "slided_5"

combSpec = CombinedSpectra({0: slided_0, 1: slided_1,
4: slided_4, 5: slided_5})
# normalize regions against each other
combSpec.get_internormed_regions(method="median")
# calculate common segments
combSpec.consensus_similarity()
combSpec.plot_consensus_similarity()

combSpec.cluster_concensus_spectra(number_of_clusters=8)
combSpec.plot_common_segments()
```

With the idea in mind that region 0 shows a late stage atherosclerotic artery, while region 1 shows an earlier stage, this comparison should give the difference between early and late stage atherosclerosis. After performing the differential test (t-test), a total of 368 m/z-values are significantly (q-value < 0.05) regulated, with 62 masses being up-regulated (higher intensities in region 1) and 306 masses down-regulated (higher intensities in region 0) (Figure 4.17a). Among the 264 DE proteins is Igf1, Insulin-like growth factor I, which is more prevalent in regions 0 and 4, mostly in the suspected blood stream area, but also within the suspected plaque areas a higher density of high intensities can be seen (Figure 4.17b). Indeed, this confirms the observations which Steffensen et al. [290] describe in their review of Igf1 in atherosclerosis: Igf1 originates from the liver and is transported via blood, from where Igf1 binds to the Igf-binding protein 3 to cross the endothelial cell barrier and is kept in the interstitium of the artery wall.

Besides the Igf1 case, there are further interesting differential proteins. For instance Anxa3 is more abundant around (but not within) cluster 8 of region 0, but not present in region 1. In the context of this analysis, this area might correspond to scattered or ripped plaque. The higher abundance in late-stage atherosclerotic plaque has, for instance, already been found by Goicuria [106, Table 9]. The presence of Arhgap33 within the suspected late-stage atherosclerotic plaques, but mostly not within regions 1 and 4, is of interest, too. Little is known about Arhgap33 in literature, and particularly nothing about its connection in atherosclerotic plaque. Arhgap33, however, is associated with vesicular trafficking within the human insulin signalling pathway[19] and thereby might be related to the Igf1 case.

[19] https://www.wikipathways.org/index.php/Pathway:WP481



(a) Slide D, Region 0 vs Region 1: Volcano plot.

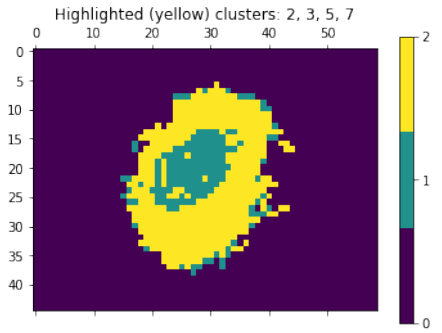
(b) Slide D, Regions 0,1,4,5: Igf1 intensity.

Figure 4.17: **pIMZ comparative all clusters results (Region 0 vs Region 1)** Using `pIMZ` and the `CombinedSpectra` a differential analysis (t-test) between region 0 and region 1 was performed. The resulting average log-fold-changes and (neg. log.) significance values (q-value) are shown as volcano plot. (a) Several genes of interest are highlighted due to their known interactions in atherosclerosis. (b) Among the differentially regulated proteins is Igf1, which shows a down-regulation in region 1.

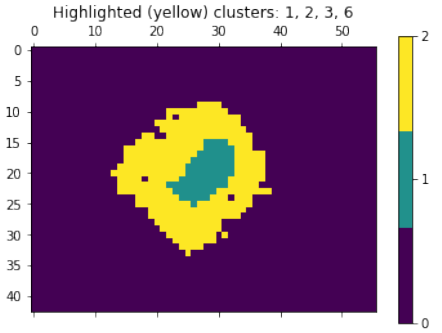
Comparative: Artery Wall After analysing both arteries in total, this second comparison looks into the difference of the artery walls of regions 0 and 1. Subsequently, the clusters for region 0 (Figure 4.18a) and region 1 (Figure 4.18b) were chosen such that the inner part of the artery remains unselected. Hence, the comparison is performed on the actual vessel wall as well as possible plaque areas.

After performing the differential test (t-test), a total of 259 m/z-values are significantly (q-value < 0.05) regulated, with 46 masses being up-regulated (higher intensities in region 1) and 213 masses down-regulated (higher intensities in region 0) (Figure 4.19a). Among the 186 differentially regulated genes is Ccl4 (Figure 4.19b). The importance of Ccl4 in atherosclerosis was recently assessed by Chang et al. [52]. The authors state that 'in ApoE knockout mice, CCL4 antibody treatment reduced circulating interleukin-6 (IL-6) and tumor necrosis factor (TNF)-α levels and improved lipid profiles accompanied with upregulation of the liver X receptor. CCL4 inhibition reduced the atheroma areas and modified the progression of atheroma plaques, which consisted of a thicker fibrous cap with a reduced macrophage content and lower matrix metalloproteinase-2 and -9 expressions, suggesting the stabilization of atheroma plaques'. This observation fits the analysed data, because regions 1 and 5 show less severe plaques. Another chemokine, Ccl6, shows a very similar pattern, for which it is known to be a macrophage chemoattractant and to promote immune cell activation and recruitment [62].

The Phospholipase A2 Group IIE (Pla2g2e) shows a similar pattern, however is additionally detectable in the regions 1 and 5, but to a smaller extent. The participation of Pla2g2e in atherosclerosis is already suggested by the analysis of other phospholipases [265].

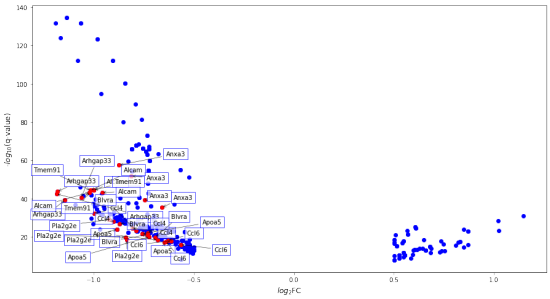


(a) Slide D, Region 0: Selected pixels (yellow).

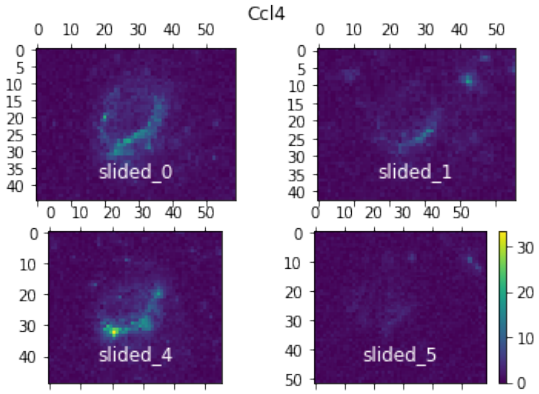


(b) Slide D, Region 1: Selected pixels (yellow).

Figure 4.18: **`pIMZ` comparative wall clusters (Region 0 vs Region 1)** Using `pIMZ` and the `CombinedSpectra` a differential analysis across multiple `SpectraRegion` objects is possible. Here, all non-background pixels (shown in yellow) of Region 0 (a) are compared with Region 1 (b).



(a) Slide D, Region 0 vs Region 1: Volcano plot.



(b) Slide D, Regions 0,1,4,5: Ccl4 intensity.

Figure 4.19: **`pIMZ` comparative wall clusters results(Region 0 vs Region 1)** Using `pIMZ` and the `CombinedSpectra` a differential analysis (t-test) between region 0 and region 1 was performed. The resulting average log-fold-changes and (neg. log.) significance values (q-value) are shown as volcano plot. (a) Several genes of interest are highlighted due to their known interactions in atherosclerosis. (b) Among the differentially regulated proteins is Ccl4, which shows a down-regulation in region 1.

With respect to lipid-related proteins, Apoa5 shows a higher abundance in the suspected plaque areas in contrast to the suspected early stage arteries. The relevance of Apoa5 in atherosclerosis has recently been highlighted by Chow et al., who state that Apoa5 '*helps to deliver atherogenic particles to the arterial wall*'[60], suggesting a pro-atherogenic effect.

Conclusion

The IMS analysis package `pIMZ` is presented and applied to a use-case from the atherosclerosis context. One advantage of `pIMZ` over other frameworks is its usability from within python notebooks in a scRNA-seq-like fashion. `pIMZ` offers multiple ways of normalizing the input spectra, and provides several means for clustering the pixels. The pair-wise pixel similarities are calculated in parallel using the cosine similarity. A data-driven analysis can be performed after clustering the spectra, like it has been demonstrated in the use-case. User-specified clusterings, as a form of a supervised clustering approach, can be used in `pIMZ`, besides several unsupervised methods. Another focus of the `pIMZ` framework is an easy export of DE results. Together with the DE results, additional data which allow a fast verification of the DE results, such as mean expression in the selected clusters and the background, are exported. The results are easily sharable via interactive HTML reports, and allow the export of the results into a tab-separated format. `pIMZ` follows FAIR software principles. It is findable via GitHub and pip. Its accessibility is ensured through continuous integration. A well documented API ensures a high interoperability, which is further increased by providing a Docker image. `pIMZ` is reusable because it is citable and archived via Zenodo, and analyses can easily be shared through jupyter notebooks.

Another feature of `pIMZ` is the ability to analyse multiple measured regions in an integrated analysis using the `CombinedSpectra` class. After normalization of all regions, a comparison between one or multiple regions is possible. As part of this analysis, a suspected late-stage atherosclerotic artery is compared against a less severe atheroslerotic artery. The DE results confirmed several known atherosclerosis related differentially regulated proteins, like Igf1 and Ccl4. Less frequently studied targets could be revealed, like Anxa3, Pla2g2e and Apoa5, where the latter are associated with lipid-related processes.

With `pIMZ` a new class of high-throughput data can be analysed. The interesting feature of IMS is the spatial resolution, allowing conclusions about the localization of specific proteins. This is of high interest for building a 3D model of atherosclerosis, which is discussed in Chapter 6.2.

4.5 Conclusion

In this chapter two emerging measurement methods have been presented in combination with respectively developed frameworks for comprehensive analyses. scRNA-seq is a single cell resolved version of RNA-seq. This stands in contrast to bulk RNA-seq, which is a technique to measure a mixture of cells, in one bulk measurement. For measuring protein levels, the MALDI-TOF technique acquires IMS measurements: a pixel-wise capturing of proteomic data. Both techniques, scRNA-seq and IMS, allow for a more detailed look into the transcriptomics and proteomics of tissue than their traditional counter-parts would.

With scRNA-seq analysis a new paradigm in the analysis of sequencing data was introduced: the analysis is typically conducted in interactive environments, where analysis

code is executed in small chunks: using (so-called) notebook technology[20]. This allows the inspection of the transformed data after almost every step. The use of this technology is interestingly independent of the chosen analysis platform: *Seurat* in R and *scanpy* in python follow this paradigm. In Chapter 4.1 the general analysis workflow for scRNA-seq data is summarized. This thesis introduces a novel method for fast and context-sensitive prediction of cell types from cluster expression values: cPred (Chapter 4.2). This method was evaluated on two datasets. With the second dataset, the cPred prediction were additionally compared to the recent competitor SingleR. The cPred cell type predictions are better than those of the competitor, particularly using the context-sensitive query. cPred relies on a database of known cell type specific marker genes. In contrast to SingleR this method has the advantage that it can be used with a very large number of cell types simultaneously. Moreover, not relying on expression patterns, it does not matter how the required expression values were measured. Indeed, applying this method to IMS data yields reasonable results. With cPred it becomes easy to generate an initial overview of the contained cell types in a new dataset. This framework was successfully applied in two COVID-19 related projects [228, 238].

The MALDI-TOF measurement technique (Chapter 4.3) is often applied in the context of IMS. This measurement technique is in such unique as it not only captures spectra of the sample, but can keep track of the location where the spectra were taken. This allows for an analysis of data not only on the spectra-level, but also on its spatial resolution. In contrast to existing scientific packages, the developed `pIMZ` framework (Chapter 4.4) concentrates on usability (a concept already highlighted in Chapter 3.1), while not restricting the user in its freedom to analyse the data. The focus of `pIMZ` lies in providing a framework for an in-depth DE analysis of IMS data, even from multiple samples at once. Samples from the public domain can be very interesting, as these can often serve as an additional baseline. In order to easily access such data, `pIMZ` can directly access data from the HuBMAP consortium, which is meant to provide IMS data for many human organs, including lung and vasculature [130]. At the use-case example of the atherosclerotic arteries it could be seen how well the cPred cell type prediction method (Chapter 4.1) can be applied to IMS data. Unlike most other IMS analysis frameworks, `pIMZ` focuses on the analysis of multiple samples, allowing DE analyses within a single sample, or over multiple samples. The `pIMZ` framework has been developed as part of this thesis and its usage and workflow is demonstrated on samples with atherosclerotic plaque.

As shown above, a workflow for processing scRNA-seq data and cell type prediction was presented, together with a framework for the analysis of spatial IMS data. The integration of both scRNA-seq and IMS data will be further elaborated in Chapter 6.2. The future will most likely see a combination of both approaches: the possibility to perform single cell spatial transcriptomics and proteomics. Most recently, Stickels et al. claim to have 'highly sensitive spatial transcriptomics at near-cellular resolution' established [291]. In fact, spatial transcriptomics was awarded the method of the year 2020 by *Nature methods* [84]. Spatial transcriptomics analyses can already be performed using commercially available protocols[21],

[20] E.g. Jupyter notebooks https://jupyter.org/ or R Markdown https://rmarkdown.rstudio.com/
[21] https://www.10xgenomics.com/spatial-transcriptomics/

and are becoming increasingly common for biomedical analyses [18, 25]. Likewise, IMS is applied at even larger scale in order to generate more complete pictures of the proteomic landscape of whole tissues, making use of integrative methods [246]. To these regards, `pIMZ` offers a framework for the analysis of IMS data. For a multi-omics spatial analysis, the `pIMZ` framework can be combined with the Aorta3D project, which is discussed in Chapter 6.2.

Perhaps thinking should be measured not
by what you do but by how you do it.
Richard W. Hamming

5

Third Generation Sequencing Data Analysis Frameworks

Sequencing data are one of the primary resources for many bioinformatics tasks, as already pointed out in the introduction (Chapter 1) to this work. In contrast to the older microarray technique, increasing amounts of sequencing data are produced every year. Due to the required handling, sequencing data are usually analysed in the domain of a bioinformatics workflow. Several distinctions between sequencing data should be made. Foremost, the sequenced material plays an important role. In general, one distinguishes between the sequencing of genomic material, or transcriptomic material, respectively. While genomic sequencing reveals information at the DNA level (e.g. new genomes, mutations, chromatin binding), transcriptomic material reveals information at the RNA level, and hence allows conclusions about how much of a specific sequence is present, and available for protein translation. Another use-case at the DNA level is meta-genomics, where many DNA fragments of (mostly) unknown origin are sequenced, for which it is then tried to identify therein contained species and their genomes. The MinION sequencing technology developed by Oxford Nanopore Technologies has been used to improve the accuracy of single nucleotide polymorphism (SNP) genotyping in complex polyploid plant genomes, where even low-coverage long-read sequencing achieves superior genome alignments [197].

In general, there are several sequencing technologies available. As already described in Chapter 1, microarrays were frequently used for gene expression analysis at the RNA level. However, with the advent of next-generation sequencing (NGS) (in contrast to traditional Sanger sequencing), microarrays are deemed too inflexible. The most prominent sequencing technology of the NGS era is the one provided by Illumina. In contrast to NGS stands single-molecule real-time (SMRT) and nanopore sequencing. These technologies form the third era of sequencing technology and are often referred to as third-generation

sequencing (TGS) techniques. All sequencing technologies have their own specifics. Thus, it is important to have specialized analysis platforms for these techniques.

It was possible to obtain sequencing data[1] from an Oxford Nanopore MinION sequencing device, which falls into the category of TGS. Early in the development of the MinION device there were two different sequencing strategies: 1D and 2D reads. For 1D reads, the molecule is read through the measurement pore once, 2D reads were processed such that using a hairpin-connector at one end, the reverse complement was sequenced, too. During basecalling, the information from both strands were then combined for a better basecalling result. With newer sequencing kits the 2D-sequencing option, however, has been discontinued, because read quality and identity improved in the 1D case. Nowadays, the Oxford Nanopore devices only generate 1D reads at continuously improving error rates. Due to the above-mentioned specifics and differences to NGS, mostly a higher error rate and variable read length of the TGS technology, specific tools for the processing of TGS and Oxford Nanopore reads are required.

In this chapter, the two developed frameworks for the analysis of TGS sequencing data are discussed. The first framework, poreSTAT (Chapter 5.1), focuses on the usage of new *data sources* and *information extraction*, but additionally performs *reporting and visualization*. This framework mainly serves as a quality control tool, but provides interactive, javascript-based, plots for reporting and visualization. The second framework, *sequ-into* (Chapter 5.2), spans the bridge between *Information Extraction* and *reporting and visualization* by providing highly accessible means for generating a summarized data report, including visualizations, from the actual sequencing data. The gene quantifications from poreSTAT may serve as input for the `RoDE` pipeline (Chapter 6.1), which completes the general analysis workflow with *reporting and visualization* and *data integration and knowledge discovery* aspects. In terms of the general workflow of data analysis (Figure 1.1), these frameworks cover almost the full analysis stage.

5.1 A Framework for MinION Sequence Analysis (poreSTAT)

Sequencing a genome or transcriptome has become a standard procedure in wet-labs worldwide. NGS techniques have revolutionized biological and biomedical research in many areas and even lead to new treatment procedures such as personal genomics and personalized medicine [216]. This is only possible due to competitive sequencing costs.

The advantage of NGS for sure is its cost: at costs of less than 1ct per base it is very cheap. However, the downside of NGS is its inability to sequence long stretches of DNA (e.g. more than 500bp) reliably. While this must not be bad for all applications, it was recently shown that there exist genomic and transcriptomic areas which are relevant to specific diseases, yet are usually not covered using Illumina NGS [81]. Here, TGS techniques based on single-molecule real-time (SMRT) and nanopore sequencing come into game: using these

[1] Luisa F. Jiménez-Soto. Personal Communication. 2017.

techniques it is possible to overcome the above-mentioned problems. These techniques allow the sequencing of very long stretches of DNA, with recorded single sequences of up to 1 million bases [274]. With nanopore sequencing, a DNA strand is recorded while it transits through a nanoscopic pore. The fluctuations in electric current at the pore are recorded and later translated into sequences (so called basecalling).

With the Oxford Nanopore Technologies MinION the first commercially available device using nanopores for sequencing single-stranded DNA molecules has been presented several years ago. Nowadays, MinION sequencing is becoming more popular for several reasons. The sequencing device is particularly cheap (just a few thousand dollars) and very mobile. This allows the collection and sequencing of samples even at remote areas [51]. Especially the information gain, for genomics, of long reads often pays off the slightly higher price per base compared to Next-generation sequencing technologies like Illumina. Even the downside of Nanopore sequencing, a relatively high error rate, improves with each release of a new sequencer version, reporting up to around 95% read accuracy with current R9 release, and more than 99% accuracy with the newer R10 release[2].

In contrast to Illumina sequencing, the MinION outputs the raw signals for each read, in an open format. This allows to reprocess the measured signals at any time, and multiple different methods for transforming these signals into read sequences (basecalling) exist. Some examples are the MinION bundled albacore and guppy[3], but there exist open-source methods from within academia, like Chiron [305], too. However, the MinION delivered tools for basecalling perform considerably perfect [327].

While the possibility to apply several methods on the raw data is unique to MinION sequencing, it creates another level of noise and possible variability, in contrast to existing sequencing technologies. Thus assessing and quality checking the received data is more important than with, for example, Illumina sequencing. There exist several tools to assess MinION sequencing datasets [194, 196]. Their main purpose is to expose the basecalled sequences from FAST5 files (by read-type, for instance) as well as to assess the quality of these reads per type. The R package poRe [322] has been published in 2014 and enables researcher to access FAST5 files from the programming language *R*. It allows measuring and plotting several statistics. Poretools [188] is one of the oldest tools and is implemented in python. It generates static plots in *png* format and allows extracting reads from FAST5-files. NanoOK [172] summarizes several statistics and plots into reports and can create a PDF ouput for a dataset. However, these reports are not interactive and do not allow a comparative view of multiple runs. Finally, HPG pore [302] is the most recent tool for analysing FAST5 files. It is implemented in Java and uses Hadoop for an efficient parallelization. However, the output is limited to a text report and static images. Except NanoOK, all tools can be regarded as deprecated, because current multi-FASTQ files are not supported any more.

Here, poreSTAT, a python-based framework for summarizing MinION sequencing

[2]https://nanoporetech.com/about-us/news/r103-newest-nanopore-high-accuracy-nanopore-sequencing-now-available-store

[3]Both albacore and guppy are only available to ONT customers via their community site (`https://community.nanoporetech.com`).

experiments and creating interactive reports at each stage (after basecalling, alignment and DE) is presented. As a main advantage over existing tools, poreSTAT combines a read analysis of sequenced data, an alignment analysis and a DE analysis, all integrated into one framework. For easy sharing of the reports, interactive HTML reports are provided. poreSTAT is an efficient python framework which does not only extract read data from FAST5 files, but also analyses the raw sequences and aligned sequences. An extension for poreSTAT allows comparing multiple runs in a DE analysis (Chapter 6.1). By supporting interactive reports, poreSTAT makes it easy to understand the results, and trace outliers. poreSTAT finds its niche in-between existing tools particularly in its ability to create interactive plots combined with its reporting system, and its extension for performing an alignment analysis, read quantification and differential gene expression tasks.

The above-mentioned tools have been compared regarding their functionality by Tarraga et al. [302]. The table has been extended, showing where poreSTAT innovates this field (Table 5.1). While poreSTAT implements all essential features and statistics the other tools provide, poreSTAT is the only tool to support modern multi-FAST5 files, prepare an interactive report and to perform DE analysis right away.

Methods

Sequencing analysis For the analysis of Oxford Nanopore/MinION data, several steps are required. However, all steps have in common that Oxford Nanopore FAST5 (see Chapter A.1) files serve as input. The FAST5 files can either contain one or many sequenced reads.

Using several modules, information about the contained reads can be collected. These modules are described in Table 5.2. The most important ones are *seq* for read extraction, *info* to generate an information file for further analysis steps and *summary* to create a summary report for all supplied read files. The *summary* module combines all single analyses like the *occupancy* or *yield* plot.

All these modules implement a single interface. Thus, for the summary, each read is handed to each of the modules only once. This ensures that the I/O is minimized by loading and iterating through each read file only once. At the same time, the structure and distribution of reads over multiple files allows a fork and join parallel scheme. This scheme is explained in the Performance Considerations paragraph.

Alignment analysis The alignment analysis is less dependent on the original reads and can purely be performed using a read alignment file, if read sequences and qualities are left in the file. A read-type specific analysis can be performed if a read-information file from the sequence analysis stage is available.

The alignment analysis exploits OpenMP parallelism using the task-paradigm[4] [65]. By a performance analysis of the alignment analysis library it was found that due to the high amount of CIGAR elements per read, the analysis takes a lot of time. Particularly this code was subsequently optimized.

[4]http://www.openmp.org/wp-content/uploads/openmp-4.5.pdf

Table 5.1: **Comparison of available MinION quality control tools**. poreSTAT is compared to HPG Pore [302], NanoOK [172], poRe [322] and poretools [188]. Initial comparison by Tarraga et al. [302].

Feature	poreSTAT	HPG Pore	NanoOK	poRe	poretools
Multi-FASTQ support	✓	-	✓	-	-
Extract FASTQ	✓	✓	✓	✓	✓
Extract FASTA	✓	✓	✓	✓	✓
Organize FAST5 files into folders[1]	✓	-	✓	✓	-
Create tar files of runs [1]	-	-	-	-	✓
Organize results into folders	✓	✓	-	✓	-
Plot yield	✓	✓	✓	✓	✓
Plot squiggle	✓	✓	✓	✓	✓
Extract run stats	✓	✓	✓	✓	✓
Read length histogram	✓	✓	✓	✓	✓
Read length (max., avg., min)	✓	✓	✓	✓	✓
Mean read quality	✓	✓	-	✓	-
Nucleotides content: count and %	✓	✓	-	✓	✓
%GC	✓	✓	-	✓	-
Plot frequency- %GC	✓	✓	-	✓	-
Plot per base sequence content	✓	✓	-	✓	-
Read quality histogram	-	✓	-	✓	-
Reads per channel histogram	✓[2]	✓	✓	✓	✓[2]
Nucleotides per channel histogram	✓	✓	✓	✓	-
Report	✓	✓	✓	✓	-
Interactive figures in report	✓	-	-	-	-
Gene-level quantification	✓	-	-	-	-
Alignment analysis	✓	(✓)[4]	✓	-	-
DE analysis[3]	✓	-	-	-	-

[1]: This becomes obsolete with multi-read FAST5 files.
[2]: poreSTAT and poretools return the occupancy of pores, not the reads per channel.
[3]: As part of the RoDE pipeline discussed in Chapter 6.1.
[4]: Available only through the HPG Aligner [303].

Table 5.2: **poreSTAT sequencing analysis tools**. poreSTAT has several sequencing run analysis tools implemented. Most important are *seq* for read extraction, *info* to generate an information file for further analysis steps and *summary* to create a summary report for all supplied read files.

Tool	Description
`info`	Creates a read information file with important meta information for each read
`expls`*	Prints a brief summary (number of reads, average length, ...) for each run.
`occupancy`*	Plots the occupancy of all pores.
`seq`	Extracts the read information from FAST5 format into FASTQ format.
`timeline`*	Plots the sequencing yield for each time bin.
`nucleotides`*	Plots the distribution of nucleotides for each run.
`qual dist`*	Plots the quality distribution over all reads.
`qual`*	Aggregates the quality per position over all reads.
`histo`*	Length histogram of all reads per run.
`yield`*	Plots a yield plot for each run.
`demangle`	Distributes fast5-files into run-specific folders.
`kmer`*	Prints top *k-mer*s found in a run's reads.
`squiggle`	Creates a squiggle plot for specific reads.
`summary`	Creates a summary report making use of the marked (*) tools.

A crucial part in the alignment analysis module is the determination of read counts and coverage per feature. This is done based upon the `HTSeq GenomicArray` data structure [11], which is used to count reads per genomic position. To determine the read-count per feature, the count for a feature is increased by one if the read overlaps the feature with at least one base. In order to calculate the coverage per feature the number of reads per position are summed up and divided by the feature length.

Interactive Reporting System poreSTAT allows to assess the quality of MinION sequencing experiments similarly to other available tools (Table 5.1). One of the advantages of poreSTAT is the creation of interactive plots with HTML output, while regular static plots are supported in addition. A custom version of the *mpld3*[5] package is used to achieve this. This modification was necessary to correctly visualize rotated tick labels, to allow adjustments of the figure size, and to account for offline usage. This package converts *matplotlib* plots to interactive plots in *HTML* using *d3.js*[6]. In addition, *mpld3* supports plugins, such that data displayed in the plot can be annotated with custom information (e.g. general read information). Using *mpld3* to transfer matplotlib plots to interactive figures has the full advantage of the matplotlib stack: plots can be displayed directly by matplotlib, extracted to several image formats via matplotlib or exported to HTML via

[5]https://github.com/mjoppich/mpld3
[6]https://d3js.org/

mpld3. However, especially for generating reports, relying on *d3* is a bottleneck, since a sequencing experiment can have many reads. *mpld3* uses *d3* in *SVG* (scalable vector graphics) mode, which adds each displayed data point (e.g. a read) as a single Document Object Model (DOM) element in the browser. Plots with many single data points create many DOM elements, pushing the browser's rendering capabilities to the limits. This can, for instance, occur if many reads are analyzed. Hence, the poreSTAT package uses plots based on kernel density estimation where suitable, e.g. if many entries are possible (Figures 5.8 and 5.9).

Performance Considerations The poreSTAT sequencing analysis tools all implement the ParallelPSTReportableInterface class. This ensures that all tools work in the same way, enabling the use of a fork-and-join pattern for the summary report: chunks of multiple reads are read in, processed, and the master process aggregates all results. The summary tool only needs to forward the chunks to all included tool-process functions, and finally aggregate the results from all tools for all chunks.

For alignment statistics, a C++ library was implemented, which collects all necessary information in parallel (OpenMP [65]) and transfers this information back to python for visualization. The C++ library directly reads the alignment file via htslib/samtools [180], and distributes chunks of alignments to worker threads using the OpenMP task construct[7] [65]. For each primary read alignment, statistical descriptors such as read length, histograms of CIGAR-codes, etc. are collected. The C++ library is embedded into poreSTAT using the ctypes module.

Data The poreSTAT sequencing and alignment analysis is demonstrated on a recent dataset, which analyses the green monkey transcriptome after infection with the SARS-CoV-2 virus (SRR11350376). The analysis was conducted using the NCBI SARS-CoV-2 reference genome with assembly `ASM985889v3`, as this taxonomic ID was linked with the sequencing data. This dataset allows to explore all features of poreSTAT, and is used in the *sequ-into* case study (Chapter 5.2), too.

Results and Discussion

poreSTAT Performance Considerations Even though python natively supports threading, only one thread can manipulate data at a time, which does not improve the runtime of an application much. This is, because true multithreading in python is not possible due to the global interpreter lock (GIL)[8]. This lock must be acquired by a thread in order to operate on any python object. The only way to overcome this restriction is to fork the current process, and work then in parallel: a multi-process architecture. This, however, comes at the cost of a high inter-process communication (IPC), since objects reside in the memory of one python interpreter process and need to, at last at the end, be merged

[7] http://www.openmp.org/wp-content/uploads/openmp-4.5.pdf
[8] https://en.wikipedia.org/wiki/Global_interpreter_lock

in order to obtain the final results. Depending on the task performed, this merge-step involves large data, and thus a lot of IPC. Forking, however, requires all subprocesses to load all required information (such as text databases, etc.). Due to the GIL, python does not implement the concept of shared-memory processing, like it is known from OpenMP [65].

Nonetheless, the tools implemented for the poreSTAT sequencing analysis make use of a fork-and-join pattern, by implementing the `ParallelPSTReportableInterface` class. While this comes at the above described costs, it becomes feasible because the sequencing analysis tools require little to no external data (no databases, etc.), and the single, incremental results are rather small in memory size and well serializable, reducing IPC costs. Using the parallel library interface, the runtime behaviour of the poreSTAT sequencing tasks is well improved.

For the input-chunk creation, two levels of parallelism are implemented: the folder-level parallelism is most suitable to older MinION runs, where single-read FAST5 files are stored in folders of $1,000$ reads. For multi-read files, each file contains about $4,000$ reads. Hence, one file is the perfect level of granularity needed for efficient parallelization.

For the alignment analysis, an alignment file, possibly several gigabytes large, must be processed. Within the alignment file, all alignments are processed in the same way. Using the previously described parallel python library has the disadvantage, that all alignments must be loaded by one process, and are then serialized in order to transfer these to the other processes. Finally, the results must be serialized again, and be transferred back into the main process. This involves substantial serialization costs for IPC, and thus should not be used. In order to circumvent the multi-threading limitations of python, C/C++ libraries, which can easily share data with python, can be used. For data sharing, multiple python modules are available, of which the `ctypes` module was used here (because it is integrated directly into python). In short, on the python side it requires C/C++-compatible data objects (e.g. lists, numpy-arrays, etc.), which can be accessed via pointers on the C/C++ side. For poreSTAT, the C++ library should already handle the data read-in, hence only a list of file-paths needs to be made accessible. This C++ library was implemented to calculate all required alignment statistics in parallel (OpenMP [65]). The result is then transferred back to python for visualization. The C++ library has to rely on data structures (vectors of `struct`s, for instance) which can be interpreted by python. Since all data reside in the memory of the main python interpreter process, no IPC is required. Only specific memory locations must be interpreted correctly, which is known from C/C++ as *casting*. The results are organized in the poreSTAT alignment analysis tool such that these become available to the summary creation and plotting.

poreSTAT Sequencing Analysis The sequencing analysis is used to get a first overview of the sequencing data. While the wet lab scientist can rely on the (live) output from MinKNOW during sequencing, the bioinformatician needs to assess all relevant information at a different time. Moreover, it became common practice to re-basecall the FAST5 files in order to gain better accuracy, or to use more reads. A re-assessment of all reads is required

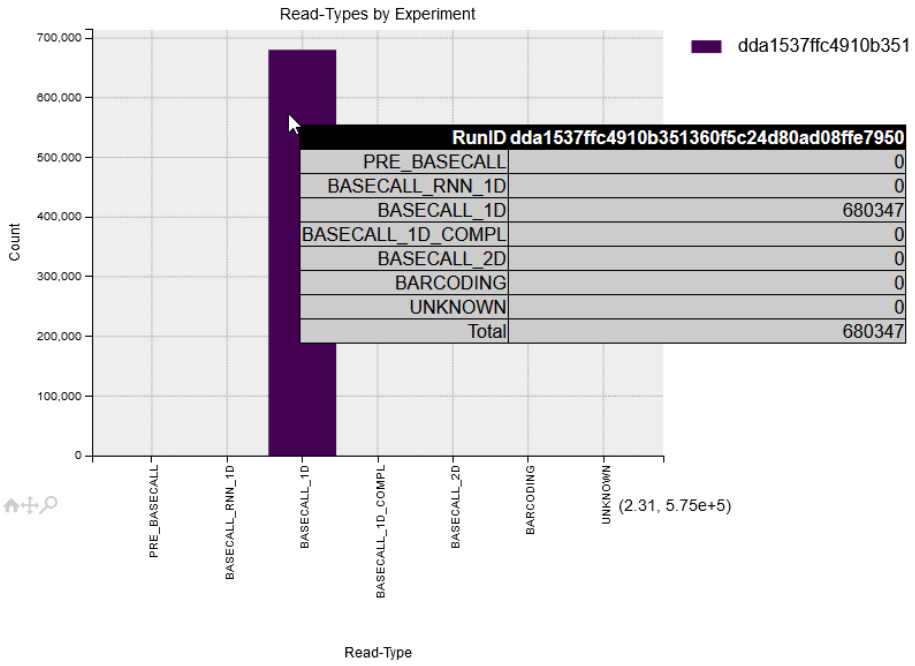


Figure 5.1: **poreSTAT basecalling summary** showing the number of reads sequenced per experiment, divided into the specific read-types. For each read, the highest ranked read-type is reported. Only BASECALL_1D and BASECALL_2D reads have basecalled sequences. A mouse-over overlay shows the summary per experiment.

before any analysis should take place. The poreSTAT sequencing analysis performs such a re-assessment.

The first analysis of the sequencing analysis is a report on the identified read types (Figure 5.1). With earlier Oxford Nanopore products, besides 1D reads, there were 2D reads. While these are mostly deprecated nowadays, at the time poreSTAT was developed, these provided better accuracy. With advances in the basecalling strategy, fewer reads with no basecalled result exist or are sorted out earlier.

The following topics are discussed at the example of the previously mentioned transcriptomic sequencing experiment on an SARS-CoV-2 infected green monkey (SRR11350376). It should be noted that due to the nature of mRNAs, no (ultra-)long reads are expected. The majority of the reads should be in the 2000nts range.

As already shown (Figure 5.1), poreSTAT first reports the found read types for each processed sample. These might be 1D, 2D, $1D^2$ or Barcoding (Figure 5.1). To further evaluate the sequencing process, the number of reads and their length per pore are evaluated (Figure 5.2). This allows the user to find out whether following sequencing runs can be performed using the same chip again, or not. Air bubbles on the chip can be detected using this plot: wherever an air bubble was, no or only few reads have been sequenced. The implication for future runs would then be that the chip is damaged at that location and may not generate any further reads at these pores.

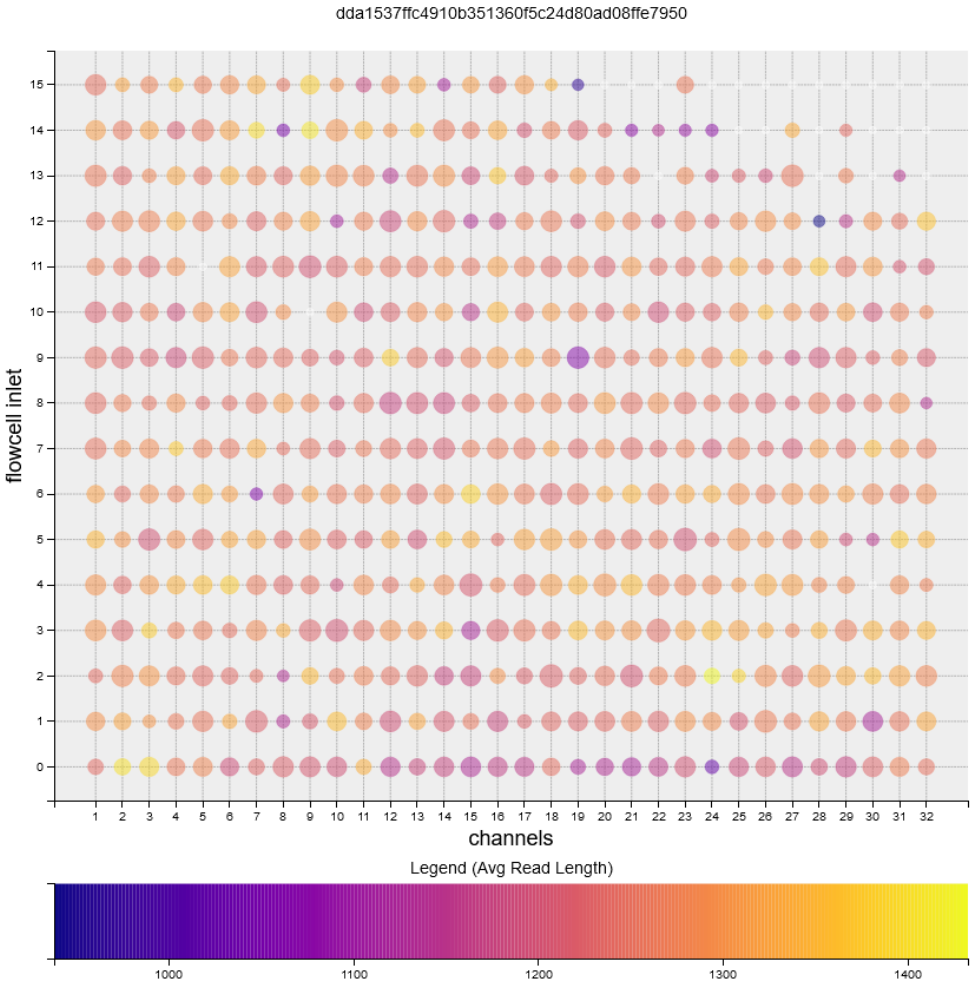


Figure 5.2: **Interactive pore layout plot** For each pore on the MinION device, the amount of reads as well as their average length and median length are shown as a dot plot. The colour of a dot (each representing a pore) shows the average read length, the width of the circle symbolizes the number of reads gained through this pore.

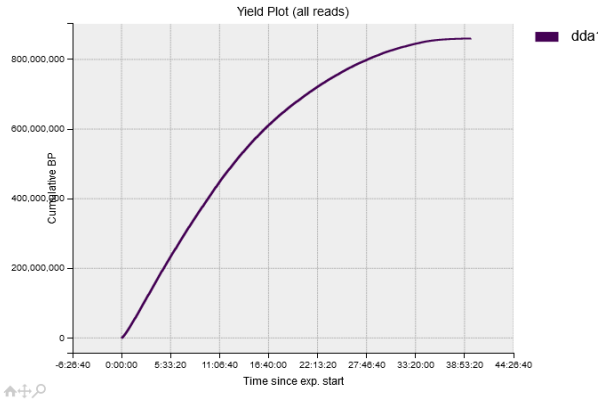


Figure 5.3: **poreSTAT yield plot** The cumulative histogram of sequenced bases over time is an important measure. It can be used to find out whether the input amount of DNA or RNA was a limiting factor, or whether the sequencing should have been continued for more reads.

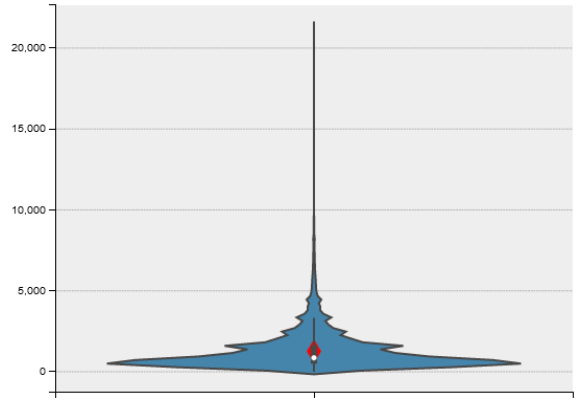
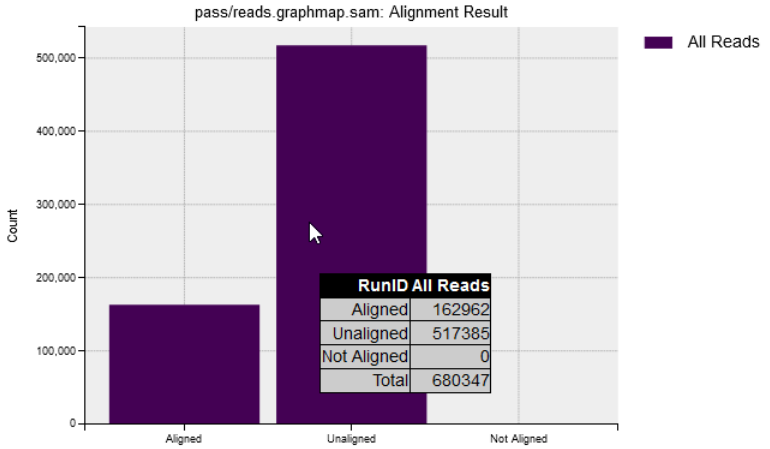


Figure 5.4: **The read length distribution plot** can be used to determine whether the size selection before sequencing worked. This may be used to determine whether (prokaryotic) RNA was correctly processed: if too much DNA was contained, the mean or median sequencing are longer.

Another interesting statistic is the total yield or yield per read-type, which is shown in a plot of time against yield (in bp) (Figure 5.3). Using this plot it can be determined whether and when a significant drop in sequenced bases or reads occurred. In future sequencing runs such a drop could then be avoided by either taking more input material right away, or by reloading the chip after a specific time (which is possible with MinION sequencing, in contrast to NGS). Important for both transcriptomic and genomic sequencing is the read length distribution, which is evaluated by poreSTAT (Figure 5.4). This distribution gives information about observed read lengths. In experiments with known expected read lengths, this statistics evaluates the library preparation and size selection phase. Very long reads in a transcriptomic sequencing project might, for instance, stem from DNA contamination. Particularly for genomic sequencing, the read length distribution is already an estimator of the quality: a high fragmentation will make any assembly process much harder.

In addition to the mentioned plots and analyses, poreSTAT can evaluate the quality distribution, the average quality by position and the k-mer distribution. The latter is, again, interesting particularly for genome sequencing, as this is an estimator of genome coverage [125]. With respect to the performance of *k-mer* counting, several serialization techniques for this task were evaluated in Chapter 3.2.

poreSTAT Alignment Analysis With the reads exported into the FASTQ format, additional analysis steps can follow. These may range from genome assembly over (differential) gene expression analysis. While genome assembly is performed to derive a new reference, for (differential) gene expression the reads (which stem from transcriptomic mRNA) are often



(a)

Alignment Statistics

Show 10 entries Search:

Name	Value
Aligned Identity	77.8257
Avg GC content of Aligned Read Sequence	0.4704
Avg GC content of Reference Sequence	0.3797
Deleted bases per 100 aligned bases	4.7194
Identical bases per 100 aligned bases	77.8257
Inserted bases per 100 aligned bases	2.1661
Overall Identity	36.4637
Substituted bases per 100 aligned bases	22.1743

Search Name Search Value

Showing 1 to 8 of 8 entries Previous 1 Next

(b)

Alignment Feature Statistics

Show 10 entries Search:

Name	Value
Featuretype Coverage (CDS)	0.9774
Featuretype Coverage (exon)	0.9784
Featuretype Coverage (gene)	0.9784
Featuretype Coverage (start_codon)	0.0011
Featuretype Coverage (stop_codon)	0.0012
Featuretype Coverage (transcript)	0.9784
Featuretype Covered By Reads (CDS)	0.7839
Featuretype Covered By Reads (exon)	0.7833
Featuretype Covered By Reads (gene)	0.7833
Featuretype Covered By Reads (start_codon)	0.3333

Search Name Search Value

Showing 1 to 10 of 12 entries Previous 1 2 Next

(c)

Figure 5.5: **poreSTAT alignment overview** (a) reporting how many reads aligned, did not align or were not considered for alignment (e.g. with no sequence), (b) reporting several quality control metrics. (c) poreSTAT can report to which genomic features the reads were aligned.

aligned against a reference genome, in a splice-aware manner. This alignment is performed using specific tools, like graphmap [286] or minimap2 [179].

After aligning the reads to a reference, the poreSTAT alignment report (Figure 5.5) can help to understand how good the retrieved reads match the reference sequence. Besides alignment rates and general alignment statistics (Figure 5.5b), it contains an analysis of the covered feature types (if a gene annotation was given, Figure 5.5c). This information is very interesting for transcriptome sequencing as it shows how many genes are detected. It is particularly helpful to quickly determine whether all regions of the genome have been sequenced, or for transcriptomic data to see whether polyA-selection or rRNA depletion worked. In the shown example (Figure 5.5), about 98% of the reference sequence is a transcriptomic feature. Of these 98%, about 80% are covered by reads, suggesting that indeed mostly transcriptomic data was obtained.

The substitution statistics (Figure 5.6) is helpful for identifying methylation effects [282]. Such methylations are detected by a careful analysis of the electrical current signals measured by nanopore-based sequencing devices. If regular basecalling is performed only, modified bases, such as methylated ones, will be incorrectly basecalled and create a substitution pattern in this report. Therefore, a table to spot such single nucleotide polymorphisms is of interest for the user. Here, this analysis reveals that the basecalling produced many undetermined bases (N), suggesting to re-do the basecalling.

Finally, a length distribution of all CIGAR codes over all reads helps to identify the mismatch rate (Figure 5.7). For transcriptomics it would be interesting to check for clipping effects. The latter information are further enriched by a *k-mer* analysis of the sequences before mismatches, insertions or deletions (data not shown). In this example, `I`nsertions and `D`eletions are quite low, while `M`atched regions are about 20 nucleotides long. Strikingly, `S`oft-clipping removes in average several hundred nucleotides per read. Such a behaviour should not occur and must be looked into further.

The GC content, the fraction of guanins and cytosines (GC), is often an important measure, as it varies among genomes of different species (e.g. *S. cerevisiae* 38%, *A. thaliana* 36%, *P. falciparum* 20%). Thus, the GC content of aligned reads is of interest and is reported as part of the alignment statistics (Figure 5.5b). For the studied SARS-CoV-2-genome, the expected transcriptomic GC content is about 38%, however, an average GC content of 47% is reported. This may point at a bias regarding a specific transcriptomic region, or incorrectly aligned reads of other origin. Relating the GC content to the read length thus may already give first insight whether the long reads originate from the expected organism, or are spurious alignments (Figure 5.8).

The alignment identity ($\frac{\text{exact matches}}{\text{reference length}}$) tells how well the aligned reads match the reference (Figure 5.9). For this example it can be seen that for short reads (less than 1500bp) the alignment identity is less than for longer reads. In fact, the alignment identity remains highest for reads ranging from about 1500bp to 8000bp. An alignment identity of about 58% suggests that 42% of the reference sequence are not identical with the read (e.g. due to deletions in the read, clipping or mismatches).

There are further alignment quality measures implemented. These mostly focus on more

Substitution Statistics (read -> ref)

Show 10 entries Search:

Substitution	Abs Count	Rel Subst
N -> T	29999781.0000	0.8428
N -> C	1580149.0000	0.0444
N -> A	985776.0000	0.0277
A -> T	817526.0000	0.0230
C -> T	636748.0000	0.0179
G -> A	387435.0000	0.0109
A -> G	311415.0000	0.0087
N -> G	286371.0000	0.0080
A -> C	202453.0000	0.0057
G -> T	177529.0000	0.0050

Search Substitution Search Abs Count Search Rel Subst

Showing 1 to 10 of 16 entries Previous 1 2 Next

Figure 5.6: **poreSTAT substitution statistics** shows how frequently particular substitutions occurred in the alignment. This information is used to see whether there is a general bias in substitutions.

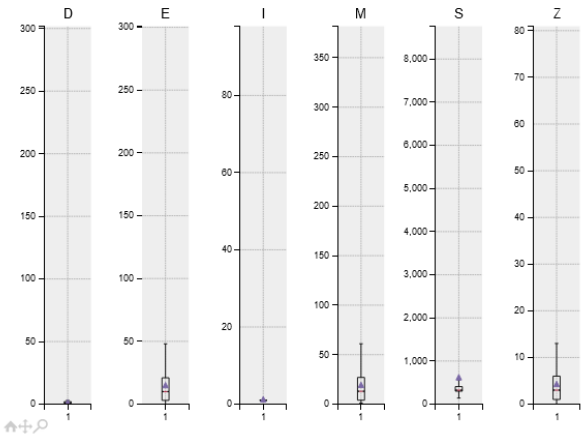


Figure 5.7: **poreSTAT CIGAR evaluation plot** shows for all CIGAR elements the respective length distributions (in bp). The reads were soft-clipped (code S) about 300bp in average. The general aligned regions (`M`atches) show a similar distribution like the special CIGAR code `E` (exact matches). Within the aligned regions, the distribution of inexact matches (`Z`) indicates that few errors are made in any matching region.

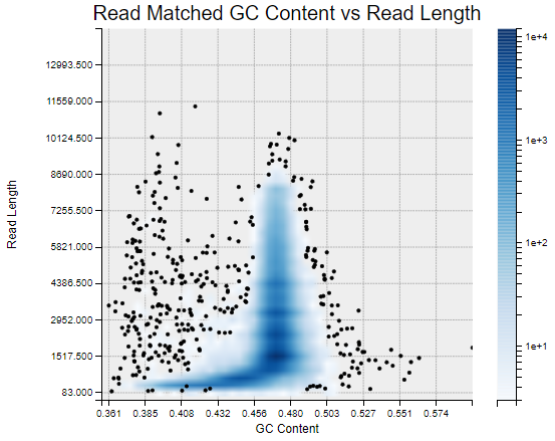


Figure 5.8: **poreSTAT read length vs. GC content plot** The aligned long sequences have a GC content of roughly 46.8%. This information can be useful to judge any GC bias or whether the reads stem from the desired organism.

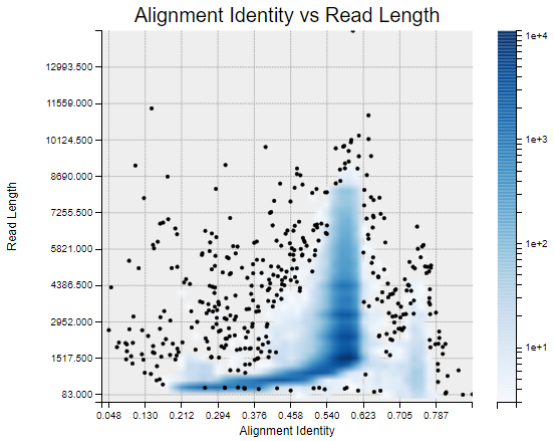


Figure 5.9: **poreSTAT read identity quality vs. length plot** against the read length. The accuracy can be evaluated by checking how well particularly the long reads align.

specific concepts than the just mentioned measures. For instance, the relation of *Read Identity vs. Read Length* is similar to the presented *Alignment Identity vs. Read Length*, but may not be as helpful to identify cases of clipping. Several measures make use of the *alignment quality* as reported by the mapping tool, or the sequence quality (reported from the sequencer).

poreSTAT Differential Expression Besides the sequencing and alignment analysis, poreSTAT can be used for DE comparisons. The poreSTAT DE pipeline `RoDE`, which is also useful for bulk RNA-seq in general, will be introduced in Chapter 6.1.

Most DE tools rely on read counts, which is a concept from NGS platforms, where the reads have a fixed length. Reads from TGS techniques are of variable length, but their analysis often is still forced into the fixed-length scheme to make use of bioinformatics methods developed for DE analysis from NGS data. poreSTAT thus supports the counting of reads per feature, e.g. per transcript. With the example data presented here, an odd distribution of feature counts can be observed, where one transcript gained the majority of read counts (Figure 5.10). The massively expressed transcript, the nucleocapsid phosphoprotein of the virus (ENSSASG00005000005), is known to be abundantly expressed in coronavirus in general[296], however, the lack of reads for all other transcripts seems surprising.

Strikingly, this observation explains the GC content discrepancy, because the highly expressed transcript has a GC-content of 47.22%, which comes close to the reported GC content of all aligned sequences. Still, this result warrants a closer look at the alignment (Figure 5.11). It can be seen that many reads of the abundantly expressed region end

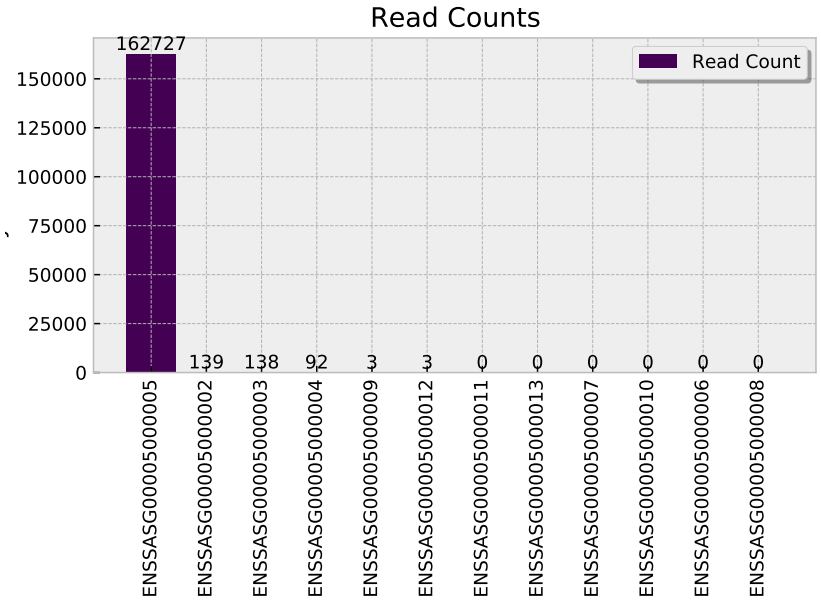


Figure 5.10: **poreSTAT Read Counts** for the discussed example data. One transcript (ENSSASG00005000005) has massively more reads than all other transcripts.

abruptly, which is a phenomenon of clipping. This may suggest that the strain linked with the sequencing data may not be the actual strain which was used for infection. This will be further analysed in Chapter 5.2.

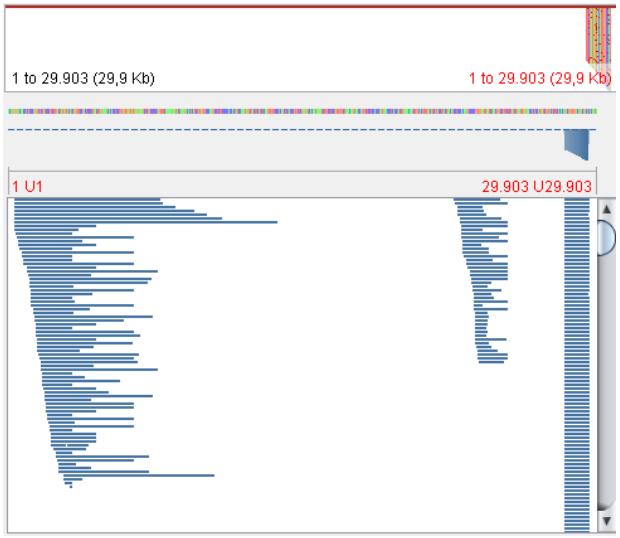


Figure 5.11: **Tablet alignment visualization** for the discussed example data. One transcript (at the end of the sequence) has massively more reads than all other transcripts. The sharp boundary of the rightmost covered region is surprising compared to the characteristics on the leftmost area. This area might correspond to the high soft-clipping activity observed.

Conclusion

In this section the poreSTAT analysis framework for TGS data (Oxford Nanopore MinION) was presented. With poreSTAT, both the exploration of read-level quality metrics, and alignment-level analyses are possible. Using poreSTAT it is possible to prepare the data for down-stream analyses.

The advantage of poreSTAT over other existing software is its compatibility with current multi-FAST5 read files, its analysis of read-level and alignment-level data as well as its interactive plotting ability. The poreSTAT read counts can directly be used by the integrative DE pipeline presented in Chapter 6.1.

With the presented analysis first the general sequencing run can be evaluated. With the pore layout plot, any bias from the pores, e.g. air bubbles, can be identified. Using the yield plots, it can be further checked whether more input material (DNA or RNA) is needed for better usage of the sequencing chip. The alignment-level analysis shows how well the sequenced reads match the anticipated organism. Coverage for multiple feature types can be assessed, besides more general alignment statistics. With the help of these metrics it was possible to find out early in the analysis, that not the whole sequenced virus genome is equally well covered. Using the correct virus genome for alignment, the key alignment statistics are improved considerably (Appendix A.6.1).

5.2 Online-Analysis of MinION Sequencing Data (*sequ-into*)

The MinION sequencer by Oxford Nanopore Technologies turns DNA and RNA sequencing into a routine task in biology laboratories or in field research. For down-stream analysis it is often required to have a sufficient amount of target reads. Especially prokaryotic, bacteriophagic or viral sequencing samples can contain a significant amount of off-target sequences in the processed sample, stemming from human DNA/RNA contamination, insufficient rRNA depletion, or remaining DNA/RNA from other organisms (e.g. host organism from bacteriophage cultivation). Such impurity, contamination and off-targets (ICOs) block read capacity, require to sequence deeper and longer. In comparison to NGS, MinION sequencing allows to reuse its chip after a (partial) run. This allows further usage of the same chip with more samples, even after adjusting the library preparation to reduce ICOs. The earlier the ICOs of a sample are detected, the better the sequencing chip can thus be conserved for future use. *sequ-into* requires few resources and is a user-friendly cross-platform tool to detect ICO sequences from a predefined ICO database in samples early during a MinION sequencing run. The data provided by *sequ-into* empowers the user to quickly take action to preserve sample material and chip capacity.

sequ-into was initially developed in the course of the iGEM 2018 competition by the iGEM Team Munich 2018. Active development of the base application was performed by Margaritha Olenchuk and Julia Mayer under the supervision of this thesis' author. After the competition, *sequ-into* received several important changes to fit into the framework of this thesis: the analysis and reporting works in an incremental, online fashion such that only new data must be processed. In contrast to the original application, this required a redesign of the whole backend. Instead of relying on the result from an asynchronous system call, like in the initial version, an asynchronous HTTP request to a locally spawned server has to take place. This implicates that routines for the active polling of the server's state must be included. After the alignment, the application plots new figures: an UpSet [175] plot (Figure 5.12), showing which reads become aligned to which reference sequences, as well as a summary pie chart visualizing the overall alignment results.

With the recent events regarding the outbreak of SARS-CoV-2, it is interesting to see whether *sequ-into* can be used to detect reads originating from the virus which is responsible for the current pandemic. Using the publicly available dataset SRR11178051 from NCBI SRA, this ability can be tested. The whole genome sequencing of samples from COVID-19 patients was performed using an Oxford Nanopore MinION device. For the analysis using *sequ-into*, one genome per identified genus was added to the *sequ-into* reference genomes. The reference genome for each genus was determined to be the first actual species mentioned in the respective Wikipedia[9] entries. For SARS-CoV-2 the NCBI reference genome with assembly `ASM985889v3` was used. The reads were then loaded into *sequ-into* for processing. From the UpSet-plot (Figure 5.12) it can be seen that reads stemming from SARS-CoV-2 are the majority of the reads, more than 35%. Since the

[9] https://www.wikipedia.org/

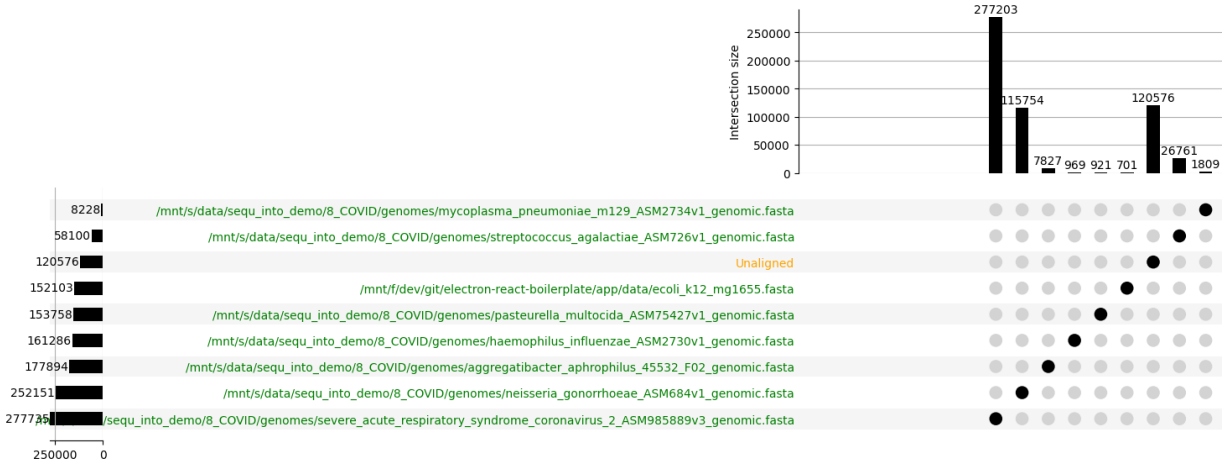


Figure 5.12: **UpSet-plot of uniquely aligned sequences in SRR11178051** The full visualization contains all set combinations, which are omitted here for space reasons. Specifically the genomes with low unique counts share sequences with other genomes.

UpSet-plot reveals all other combinations of multi-mapped reads, it can be seen that about 120.000 reads align simultaneously to at least 3 distinct genomes (*Pasteurella m.*, *Haemophilus i.* and *Aggregatibacter a.*; data not shown). Unaligned reads are a category of its own, because there is a difference between a read being aligned to a specific off-target, or being unaligned to any supplied reference. Particularly with MinION reads it is quite frequent that reads are generated, which do not match any (known) genome. Even blasting these reads against the database of all known nucleotide sequences does not yield any result. Such reads probably are an artefact of blocked pores, or incorrect basecalling. Nonetheless, even though such reads are no on-target reads, they are also no off-target ones: they are just not aligned, and hence displayed as such.

With the previous analysis working well, another publicly available experiment, `SRR11350376`, was analysed. This experiment analyses the (viral) RNA taken from a SARS-CoV-2-infected *Chlorocebus sabaeus* (green monkey). Again *sequ-into* was set up to consider SARS-CoV-2, SARS-CoV-2 transcripts (NCBI reference genome `ASM985889v3`) as well as the green monkey transcriptome. With the alignment results from Chapter 5.1 in mind, the *SARS-CoV-2/Australia/VIC01* genome (NCBI accession `MT007544.1`) was added, as this strain is mentioned in the sample data's title (but not annotated as (sequenced) organism). The green monkey can be regarded as off-target sequence, because particularly viral RNA was targeted by the experiment. From the pie chart (Figure 5.13) it can be seen that of the 680 347 reads, 199 175 reads align to on-target sequences: the SARS-CoV-2 genome or transcriptome. There are 473 ambiguous reads which align to both on- and off-target sequences. About 20% of the reads remain unaligned. In total, 49.2% are off-target reads, originating from the host organism itself. With this information it can easily be determined whether the sequencing run can be considered a success or not. More interesting is the upset plot (Figure 5.14). Using the information from this plot, it can be seen that most

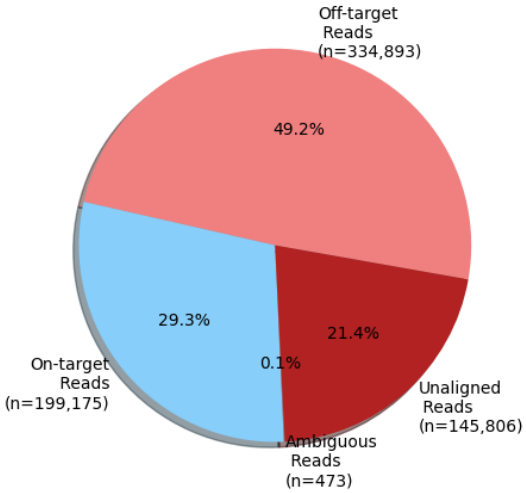


Figure 5.13: **Pie-Plot of ICO sequences in SRR11350376** It can be seen that a majority of the reads align to the off-target sequences of the green monkey transcriptome. About one third of the reads aligns to any of the on-target SARS-CoV-2 sequences, and about 20% remain unalignable.

Figure 5.14: **UpSet-plot of aligned sequences in SRR11350376**. Of all on-target reads, most reads align to all SARS-CoV-2 sequences. About 13,000 reads only align with the full genomic data, which is not unexpected due to the nature of viral genomes.

viral reads align to the genomic and to the transcriptomic sequence. Only few reads align only to the genomic virus references (ASM985889v3_genomic.fasta and MT007544.1.fasta).

Given the FAST5-files of the experiment, even more information can be extracted using *sequ-into*: a timeline analysis of the off-target rate can be calculated, because the FAST5-files also contain timestamps for the reads' sequencing time. Hence, the aligned ratio to the reference viral transcriptome can be analysed over time (or, as shown here, over the number of sequenced reads; Figure 5.15). It is interesting to see that throughout the sequencing run the viral ratio increases from initially 23% to more than 27% in the end, meaning that there originate more reads from the virus towards the end.

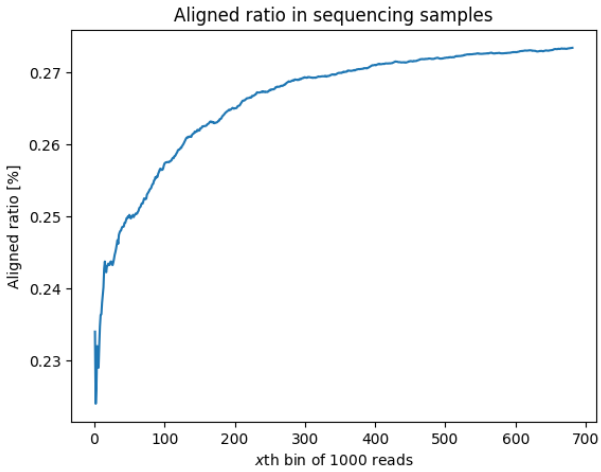


Figure 5.15: **Ratio of viral sequences in SRR11350376**. It can be noticed that the fraction of sequences aligning to the reference virus genome increases in time with the number of sequenced reads.

Taking up the results from the poreSTAT analysis, a difference between the two SARS-CoV-2 genomes (ASM985889v3_genomic.fasta and MT007544.1.fasta) can be seen. The reference genome has about $123kb$ less aligned in contrast to the specific strain (MT007544.1.fasta). The bad performance of the sample in the discussion of the poreSTAT alignment analysis (Chapter 5.1) could stem from alignments against an unsuitable reference. Upon using the correct reference genome, the alignment statistics and read counts improve to expected levels (Figure A.36).

These two analyses show that *sequ-into* can not only deliver insights into the relevance of the sequenced material for scientific questions, but may suit as quality control tool within a clinical setting, quantifying specific pathogenic sequences — using an online algorithm, and directly while sequencing. It is not necessary to wait until the data is acquired and analysed, but the analysis can take place directly while sequencing, allowing for shorter turn-around times in case of problematic results, e.g. low on-target rates.

The accepted publication is available as open-access online article `https://doi.org/10.1016/j.csbj.2020.05.014`. The author's contributions are listed in Appendix A.6.2.

5.3 Conclusion

In this chapter the poreSTAT data analysis framework for TGS data has been introduced. Particularly TGS profits from a number of unique analyses in order to understand the sequencing quality. With the poreSTAT sequencing analysis the sequencing quality can be assessed. The focus is set on whether the sequencing run delivered useful and interpretable output. The sequencing quality can be evaluated during the experiment, also allowing conclusions on whether the initial basecalling performed by the sequencing software is

sufficiently well. In the *poreSTAT Alignment Analysis* a more in-depth look into the alignment to specific reference sequences can be made. Not only is the alignment evaluated on a feature level, but additionally on a sequence level, too, considering substitutions and GC contents. Using this information, it was possible to note and correct irregularities in the initial analysis of the presented use-case.

Using the *sequ-into* approach, the observation of the incorrect reference genome in the *poreSTAT Sequencing Analysis* use-case could be manifested. With *sequ-into* an online and incremental analysis of Oxford Nanopore reads, even at sequencing-time, can be conducted. This analysis can be started and controlled using an interactive GUI which is cross-platform compatible. This GUI controls the underlying application server which itself supervises the analysis. Particularly for bacterial genomes *sequ-into* can be run even on regular laptop computers. On more powerful computers even the human genome is unproblematic. The focus of *sequ-into* is to determine which organisms are contained in the sequenced sample, with the requirement that the target organism (or a close relative) must be supplied. With this information *sequ-into* can be used to understand how many reads match each organism.

In this chapter two methods and applications for the analysis of TGS data have been presented and discussed in the context of SARS-CoV-2. They build the fundament for further downstream analyses, like the `RoDE` pipeline (Chapter 6.1). The presented methods in this chapter are another outcome of this thesis, which contributes with methods to assess the sequencing and alignment quality in a user-friendly and interactive way, here, for TGS data. These results pave the road to robust DE analyses.

Knowledge isn't free. You have to pay attention.

Unknown

6

Integrative Data Analysis in Complex Human Disease Contexts

Integrative Bioinformatics is a subdiscipline of bioinformatics, which focuses on the data integration for the life sciences. Data integration involves combining data residing in different sources and providing users with a combined, an *integrated*, view of these [174].

In Chapter 6.1 the robust DE analysis pipeline `RoDE` (*ro*bust *DE*) is introduced. `RoDE` builds upon the quantification results from the poreSTAT framework (Chapter 5.1), but can also be used with any count matrices, e.g. gene-level quantifications from bulk RNA-seq experiments. The `RoDE` pipeline contributes to the stages of *reporting and visualization*, but also to the *data integration and knowledge discovery* stage in the general bioinformatics workflow model (Figure 1.1). High-throughput count data are first assessed regarding their quality and replicate consistency, and then used for subsequent DE analyses to derive DE genes robustly. These results then serve as input for set enrichment-based evaluations (e.g. which pathways are up-regulated?), or for the miRNA-gene regulatory prediction (Chapter 2.3). Using these resources, both NGS and TGS data can be evaluated, processed and used for robust DE analyses. The use-case of hypothesis generation is promoted by both the emphasis on using robust results for regular set enrichment methods, and the integration of the miRNA-gene regulatory predictions. The latter combines the DE results obtained from high-throughput experiments, and combines these with results obtained from text mining methods (Chapter 2).

In the last section of this thesis, the data integrative project Aorta3D is presented. The previously presented methods were already data integrative, like the *cPred* cell type prediction (Chapter 4.2) using marker gene databases, `pIMZ` allowing the use of multiple datasets (Chapter 4.4), and the robust DE pipeline `RoDE` (Chapter 6.1) making use of several processing techniques. The Aorta3D project integrates results from these methods

into one project. With respect to the general workflow of data analysis (Figure 1.1), Aorta3D integrates the various frameworks already presented into one resource to obtain an integrative view on a specific context (e.g. atherosclerosis). Such a view allows an easier evaluation of the acquired results, connects multiple data sources, and hence promotes *knowledge discovery* through *data integration*.

6.1 A Robust Differential Expression Pipeline (**RoDE**)

Bioinformatics data analysis is a rewarding and yet sometimes frustrating experience: using bioinformatics data analysis existing hypotheses can be verified, or new ones can be reasoned, eventually leading to striking discoveries. However, many bioinformatics data analyses are neither replicable nor reproducible, making it hard to understand many analysis results, despite replicability and reproducibility being values of utmost importance in science [231, 237]. Scientific results are called reproducible if the same data and same analysis executed by different scientists yields the same results. If different data, but the same analysis, lead to the same results, this is called replicability. Robustness refers analysing the same data, with different analyses, leading to the same result. And finally, a result is generalizable, if different data and different analysis lead to the same result. This connection is displayed in Figure 6.1.

The analysis of RNA-seq data with respect to finding which genes' expression differs between multiple conditions is a common task practised on a routine basis [63]. However, most frequently the acquired data is analysed following a very strict analysis path: one tool and setting is used for aligning the reads with the genome, one tool and setting is used to quantify gene expression and a further tool and setting is used to evaluate whether a specific gene is significantly differential between two conditions or not. Such a sequence of tools with given settings is often called a *pipeline* or *workflow*, as data is fed into the first step, intermediate results serve as input for the following step, creating a complete analysis with the last step.

Due to the sheer amount of data analyses executed in bioinformatics, these practices have become a problem in bioinformatics and biomedical analyses, but also beyond: many results are neither reproducible nor replicable. 80% of all questioned researchers acknowledged a reproducibility crisis in their respective fields [20]. By just using one pipeline, and one result per step, as described above, many analyses are neither robust nor generalizable. While the use of just one specific software for each step is common practice and acceptable, using a combination of multiple instances of tools and settings can deliver better performance. This multitude of results can then be used to eliminate the variance of the different methodological approaches. Accepting only DE results found by all methods, may reduce the output (e.g. differentially expressed genes), but also increases the reliability of the found results. A pipeline, which uses several analysis methods on the same input data, and combines these results instead of choosing just one result to continue with, is named a *robust* pipeline. Here, the *robust* pipeline for DE analysis of RNA-seq data, `RoDE`, is presented.

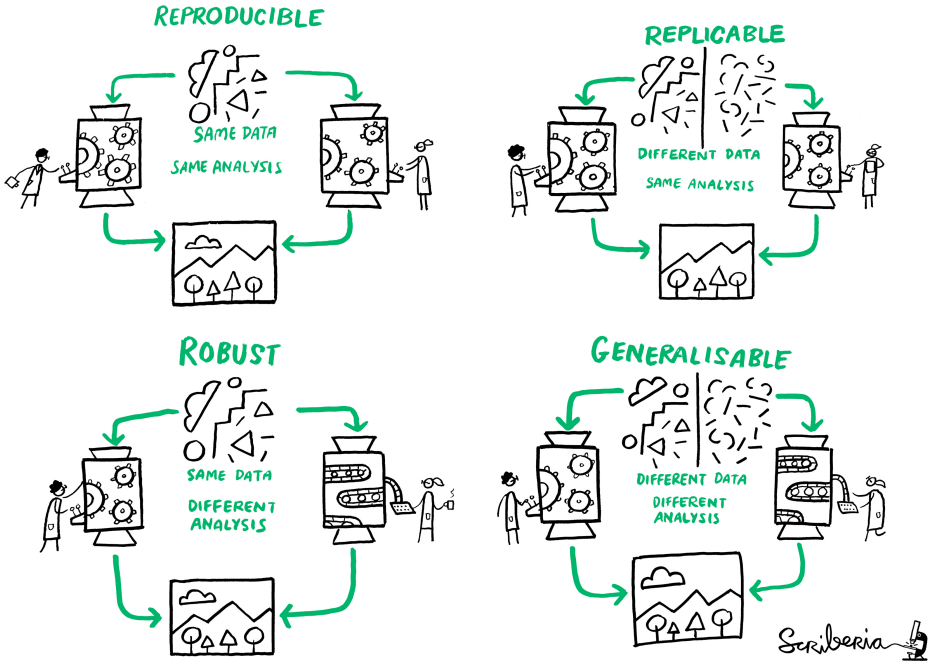


Figure 6.1: **Reproducibility, replicability, robustness and generalisability explained** Often repeatability is named as an additional important factor, which can have two meanings: reproducibility or replicability. Image created by Scriberia[1].
[1] https://zenodo.org/record/3695300

Introduction

RNA-seq has become a regular analysis method in biomedical research. With sequencing costs per base still going down[1], and numbers of RNA-seq experiments still rising, methods for the stream-lined analysis of RNA-seq data are needed.

There are many approaches to stream-line analyses. The bioinformatics community puts a lot of research and effort into workflow systems. One of the biggest workflow systems in bioinformatics is the Galaxy project [1, 102], which integrates not only specific workflows, but additionally offers to generate plots and to analyse data in detail. The nextflow workflow system [74] is a more general workflow system, but has many applications in the bioinformatics domain. Despite these rather general workflow systems, more specific ones exist. The Maser platform [154] is specialised in NGS-data analysis, from RNA-seq to ChiP-seq. But there are further RNA-seq analysis pipelines available like RAP [64], NARWHAL [38] or CSI NGS [8]. While each of these pipelines focuses on a specific new part, like data integration from cloud resources, running in the cloud, or preparing data for following steps, all of these follow the one-step-one-tool-one-setting approach. There exist predefined pipelines for RNA-seq data analysis in nextflow from the DolphinNext framework [342].

[1]https://www.genome.gov/sequencingcostsdata

This includes an RNA-seq module running multiple aligners and quantification methods. However, this workflow stops with the provision of multiple quantifications, and does not provide a robustly combined result. In addition to real workflow system, there exist python-based RNA-seq DE pipelines, like pySeqRNA[79], which was recently presented at ISMB2020. At that time, the pipeline was not (yet) publicly available, hence it is unknown how pySeqRNA relates to `RoDE`. The authors of pySeqRNA do not mention a robust view on the input data on their poster, but they focus on the analysis of different read counting strategies, such as uniquely mapped reads or multi-mapped ones.

The here introduced robust DE analysis pipeline `RoDE` starts exactly after the gene-expression quantifications, and applies multiple DE tools to the single results, combines these and allows adding further DE results, derived elsewhere, for a robust combination. With `RoDE` it becomes possible to create a robust DE workflow, which allows generalizing gained information by combining further differential results. A use-case, which compares RNA-levels of stable and unstable human atherosclerotic plaque sections, serves for discussing the `RoDE` pipeline.

Methods

The robust DE pipeline `RoDE` consists of multiple steps which are summarized in Table 6.1 and Figure 6.2. Each of the pipeline's steps, starting with DE analysis, adds specific analyses to the output report. The first steps are of general interest as these perform a quality control of the obtained samples. Most steps of the DE analysis are performed for each input count matrix, e.g. from different aligners or gene quantification methods. The step deriving robust DE results from the different DE methods is the core part of this robust pipeline. The output from this step serves as input for the enrichment analysis, where, again, the effects of the different possible robustly combined results can be analysed.

In this section, first the input data are described, followed by the used workflow mechanisms. Finally, the method to robustly combine multiple results is described.

Data The use-case data are publicly available from NCBI SRA [173] as accession `PRJNA493259`. The project consists of eight human samples, runs `SRR7905615` to `SRR7905622`. These samples, from fresh human carotid plaques obtained at carotid endarterectomy in 4 symptomatic patients were dissected in stable and unstable regions based on macroscopic appearance. The unstable regions were characterized as visible zones of plaque rupture. For these samples total RNA was extracted and sequenced (paired-end, unstranded).

Data Preparation The data was downloaded from SRA via EMBL ENA and trimmed using bbduk (Version 38.87[2]). Following trimming the samples were mapped using STAR (v2.7.6a [75]) and HISAT2 (v2.1.0 [153]) against the human reference genomes obtained from Ensembl [336] (release 101). Gene expression was quantified on the primary gene

[2]https://sourceforge.net/projects/bbmap/

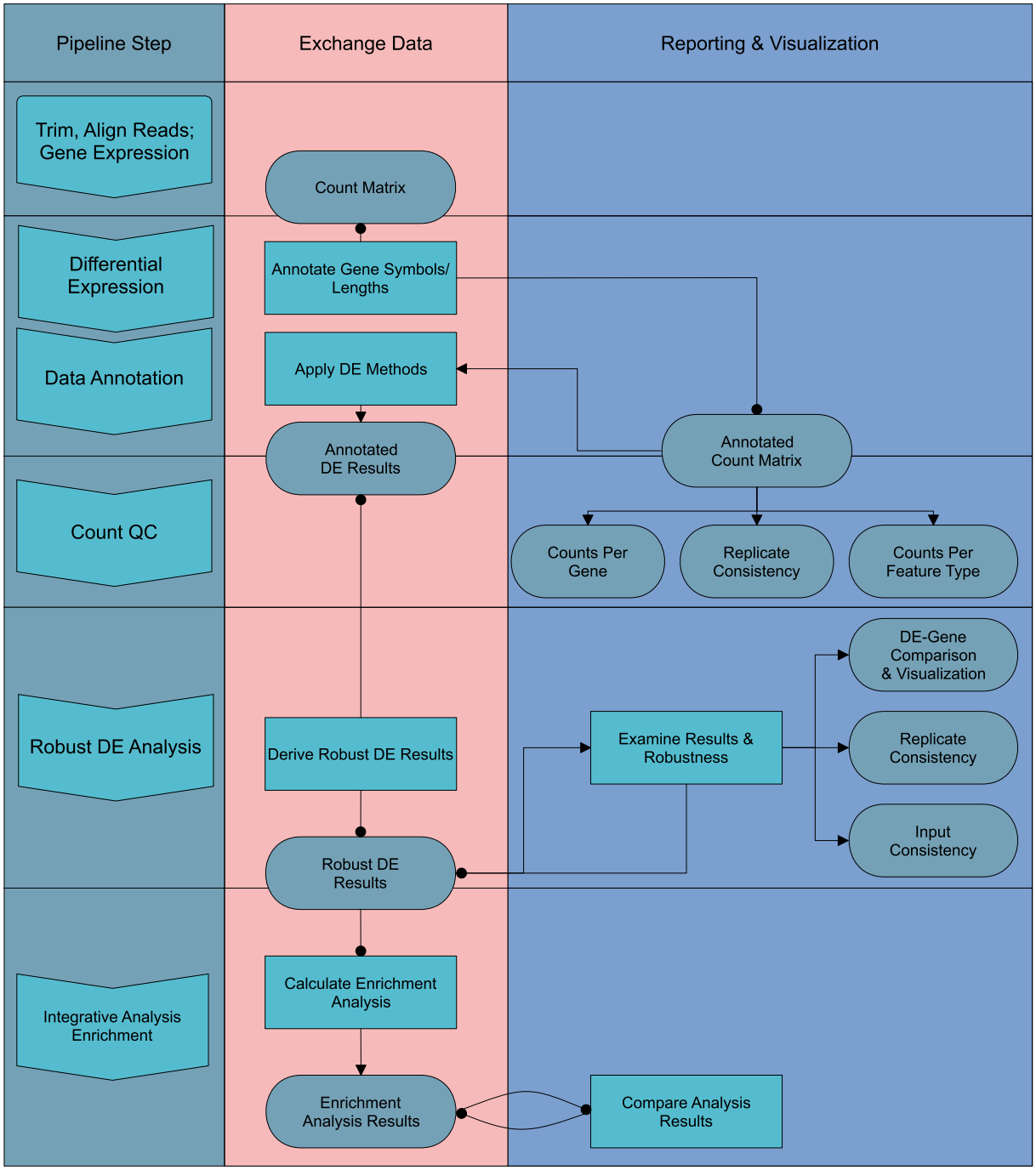


Figure 6.2: **The workflow of the robust DE pipeline RoDE**. The pipeline starts with the count matrices and first performs DE tests while annotating the count matrices with gene information and normalized counts. Following this, a count-level quality control is performed, before the DE results are examined for robustness. The final step is an enrichment analysis, which compares the given results from all inputs as well as the robust solutions. Dots at the edges' ends symbolize multiple in- or outputs.

Table 6.1: **Pipeline Tools for Robust Differential Analysis** The `RoDE` pipeline uses well established DE methods like DESeq2, edgeR or limma. New methods can easily be added. Likewise, the enrichment analysis bases upon well established packages like clusterProfiler, or miRNA-target gene sets from miRTarBase.

Pipeline Step	Method	Reference
DE Analysis Methods	DESeq	EnrichmentBrowser [99]
	DirectDESeq	DESeq2 [189]
	DirectDESeq2Paired	DESeq2 [189]
	msEmpiRe	MS-EmpiRe [7]
	nlEmpiRe	nlEmpiRe*
	limma	limma [255]
	edgeR	edgeR [208, 257]
Enrichment Analysis	KEGG, GO	clusterProfiler [340]
	Reactome	ReactomePA [339],
	miRTarBase	miRTarBase [128]
	Integrative miRNA Analysis	miRExplore (Chapter 2.3)

* Csaba, Gergely. Personal Communication. 2019.

annotation file (with scaffolds; without patches or haplotypes) using featureCounts (v1.6.3 [184]) with reads being assigned to all overlapping meta-features, but counting only read pairs with both ends aligned to the same chromosome on the same strand and delivering fragment counts (instead of read counts). Sample names in the output matrices were renamed into a human-interpretable format, naming the samples `stable_patient1` to `unstable_patient4`.

Workflow The analysis workflow can roughly be divided into five steps (Table 6.2). Each step builds upon the results from the previous steps, but can also be executed independently. The gene counts, which were quantified through the STAR mapping, will be referred to as STAR counts and those from the HISAT2 mapping as HISAT2 counts, respectively.

The first step in the workflow maps Ensembl gene IDs to gene symbols [336] and adds gene lengths to the input count matrices. Particularly the gene lengths are required for the calculation of FPKM [218] and TPM [318] values (Equation 6.1). For the FPKM and TPM calculation, the $\text{fragments\_mapped}_{\text{gene}}$ refer to the number of fragments used for the quantification of a specific gene (e.g. the count reported by gene quantification methods). The gene length of a gene $\text{length}_{\text{gene}}$ is assumed to be reported in base pairs (bp). The scaling factor 10^3 transforms this input into kilobases. In the case of paired-end sequencing, the fragment is a read-pair. For single-end sequencing, one can consider one fragment to be one read. In this case, the FPKM is identical to the often used RPKM (reads per kilobase million) value.

Table 6.2: **Pipeline Steps for Robust Differential Analysis** For each pipeline step the included analyses and visualization are listed.

Pipeline Step	Purpose
Foldchanges	Perform DE Analysis using specific methods
Counts Analysis	FeatureCount summary visualization
	Replicate comparison (un-/norm. counts)
	LogFC distribution between replicates (un-/norm. counts)
	LogFC distribution between all samples (un-/norm. counts)
	Read count histograms & Count heatmap
	Count frequency per replicate
	Counts per biotype per replicate
	Input Comparison per replicate (raw, TPM, RPKM)
DE Analysis (per DE method)	Overview (UpSet, Volcano)
	Count frequency per replicate (w/ gene symbols)
	Cluster expression values of replicates (UNTF*) heatmap, PCA, UMAP
	Heatmap of top 100 DE genes (UNTF*)
	TPM/FPKM count histograms & frequency
Combined DE Analysis	Overview (UpSet, Volcano)
	DE gene rank comparison
	Cluster expression values of replicates (UNTF*) heatmap, PCA, UMAP
	Heatmap of top 100 DE genes (UNTF*)
	DE gene overlap within DE method for top DE genes by adj. p-value, log2FC, log2FC (of sign. genes)
Robustness Check	Overlap (UpSet) of top DE genes by adj. p-value, log2FC, log2FC (of sign. genes)
Enrichment Analysis	ORA for KEGG, Reactome, GeneOntology (each domain) for each method, prefix, combined and robust.
	Comparison of results (sorted by q-value)
Enrichment Analysis Custom gene sets	ORA for miRTarBase, Epi-Genetic Gene-Sets for each method, prefix, combined and robust.
	Comparison of results (sorted by q-value)

* un-normalized (raw) counts, library-size normalized counts, TPM, FPKM (performed for each count type)

$$
\begin{aligned}
FPKM_{\text{gene}} &= \frac{\text{fragments\_mapped}_{\text{gene}} \cdot 10^3 \cdot 10^6}{\text{fragments\_mapped}_{\text{total}} \cdot \text{length}_{\text{gene}}} \\
TPM_{\text{gene}} &= A_{\text{gene}} \cdot \frac{10^6}{\sum_{i \in \text{genes}} A_i} \\
A_{\text{gene}} &= \frac{\text{total\_fragments\_mapped}_{\text{gene}} \cdot 10^3}{\text{length}_{\text{gene}}}
\end{aligned}
\tag{6.1}
$$

The annotated count matrices are used as input for the DE analysis using at least one of the DE methods listed in Table 6.1. Each input matrix yields one output matrix with annotated log-Fold-Changes (logFCs) and (adjusted) significance values (PVAL.ADJ) for the performed comparison. This output then is used as input for the count-level quality control step from the *Counts Analysis*. Within this analysis multiple count-level statistics are calculated and visualized. Among these are replicate comparisons (scatter plot of counts) as well as count frequencies per gene and biotype for each replicate.

The *DE analysis* step evaluates the results of the DE analysis performed in the *fold changes* step. At this stage, the results of the different DE methods are compared, and the specific results are visualized using a volcano and upset plot. The replicates are clustered regarding their gene expression results on the differential genes. These analyses are repeated for the combined results, where the robust result is derived from all input matrices. Additionally, overlapping DE genes are compared by method and mapper as well as the combined result. With this robustness check the overlap of the differentially expressed genes for all samples are evaluated.

The last stage is the enrichment analysis of common gene sets like KEGG, Reactome or GeneOntology, as well as custom gene sets, like epi-genetically relevant gene-sets or miRNA-target gene sets taken from miRTarBase and miRExplore (Chapter 2.3). The found significant sets (q-value < 0.1) are compared among all input variants as well as the combined and robust variant. The combined variant robustly combines all results processed by this pipeline (robustness), while the robust variant combines additional user-supplied DE results (e.g. additional micro-array experiments) with the results from all DE methods executed as part of this pipeline (generalizability).

All pipeline steps and outputs are summarized in Table 6.2.

User Interface The pipeline requires the user to specify the input for the differential analysis. This input consists of count matrices with associated sample (column) names. The user must specify a location where the output report with figures is stored, as well as a directory where data associated with the count-matrices is stored in. For down-stream analyses the user must name the organism of the samples as well as files which map the gene name from the count matrices to their respective length and gene symbol. This is required for calculating normalized expression values like RPKM and TPM (Equation 6.1), but also has practical relevance: the output should be interpretable by non-bioinformaticians,

which are used to gene symbols, instead of Entrez [195] or Ensembl [336] gene IDs. Finally, the input requires the user to define the conditions to compare, which is done from the column names of the input count matrices. If the user does not specify the DE methods to be used, results for all possible combinations of the available DE methods are calculated. For practical reasons, like run-time or amount of produced data, only required methods should be specified. Likewise, the user should specify for which results (of the DE method combinations) an enrichment analysis should be performed for.

Robust Combination The robust combination of multiple DE results is a key operation for a robust data analysis. This combination takes the robust DE result (which again might be a combination of results from one or more DE methods) of all input DE results and calculates a robust result. For genes with a common direction of the log-fold-change (that means all methods consistently show an up- or down-regulation), the robust result for a gene is defined as the lowest absolute log-fold-change and the maximal adjusted p-value. Genes, for which some methods predict a down- and other methods an up-regulation, or the other way around, are assigned a log-fold-change of 0 and an (adjusted) p-value of 1. Such genes would not be reliable.

Enrichment Analysis Custom gene-sets are processed using scripts written as part of this pipeline. For the over-representation analysis the hypergeometric test is used [101]. P-values are adjusted for multiple testing correction using the Benjamini-Hochberg procedure [24].

Availability The robust DE pipeline `RoDE` is part of the poreSTAT framework which is available through GitHub `https://github.com/mjoppich/poreSTAT/`. A `RoDE` Docker image (`https://hub.docker.com/repository/docker/mjoppich/porestat_de`) is available. Using this Docker image, all required python module and R library dependencies are already in place.

`RoDE` adheres to the FAIR principles. It is findable from GitHub and Docker. At both locations instructions for accessing the software are given. By using common genome annotation files for creating enhancement files, and relying on gene quantification output from common software like featureCounts [184], the pipeline is also interoperable. Finally, the pipeline is easily extendable and thus reusable.

Results and Discussion

Performing a DE analysis starts with the evaluation of the quantified gene expression values using a specified DE method. In this use-case a *robust* DE analysis should be conducted to see, whether the applied DE tools deviate from each other on both the DE gene level, and on down-stream analyses, like GO enrichment analysis. This robust DE analysis is discussed at the example of a biomedical experiment, where the differential gene expression between stable and unstable human plaque is studied. This is of high interest in order

to understand the final stage of atherosclerosis, where stable plaque becomes unstable, ruptures, and leads into the fabrication of a thrombus. It is thus important to understand the processes involved in this final stage of atheroprogression. It must be avoided that the results concluded from the bioinformatics analysis are artefacts of the bioinformatics pipeline. The results should be genuine observations in the biological samples. In order to ensure that observations are actually true, each DE method applies checks to adjust against the multiple-testing problematic (the chance that there is a false-positive due to the many genes tested), but most importantly each method makes distinct assumptions regarding how to decide whether a gene is differentially regulated. Because each method makes certain assumptions, e.g. on the distribution of gene counts, results from the various methods may differ. These assumptions must not reflect the actual truth, though. Hence, the idea behind *robust* DE analysis is to assume that at least those DE genes, which are found by multiple methods, are actually differentially regulated. At least these DE genes are not only significant due to a statistical assumption made by just one method. Obviously this may drastically influence how many and which genes are left over and will depict a rather conservative, yet robust, image of the data. Such differences between the single methods and the robust view on the data are assessed by the robust DE pipeline `RoDE`.

Pipeline Within the pipeline, each step is executed separately and independently. The created output filenames follow a specific pattern. Thus, by checking the existence of a file, it is possible to determine whether a specific step has already been done or still needs to be executed. This design decision has one disadvantage. Each sub-step operates independently, and therefore has to read in all needed inputs. With count matrices easily becoming very large (100MB range), this might induce a bottleneck for future operations. However, this ensures that every step is self-contained and introduces a certain resilience to the whole workflow: upon an error, the workflow can be continued.

Pipeline Invocation Given the input count matrices, the `DifferentialAnalysis.py` script can execute the analysis workflow fully, or in parts. For the use-case presented in this chapter, the command is shown in Listing 6.1. This command specifies the input count matrices `-counts`, the respective short names `-prefixes` (here: STAR or HISAT2) as well as the location to store count-related information `-diffreg`. The `-name` specifies the prefix of all DE analysis output files. With the organism information `-organism` as well as with the enhance- and lengths-files (`-enhance` and `-lengths`) the entities from the count matrices can be converted into human-interpretable gene symbols. These files are needed to annotate gene lengths required for absolute expression value calculation. The organism name is essential for the set enrichment methods, because some require a mapping to Entrez gene IDs. With the `-report` flag the location of the output HTML report can be set, while all other output files are saved in the `-save` folder. The `-de_methods` flag allows the user to set specific (combinations of) methods which will be used to derive fold-changes and significance values between conditions. The conditions themselves (the columns used from the count matrices) are specified with the `-cond1` and `-cond2` flags. The order must be

the same as for the count matrices. Here, the first set of condition names is used for the star counts, the second set for the HISAT2 counts, respectively. With the `-enrich-methods` flag the methods for which an enrichment analysis is performed can be restricted. By default, this is done for all specified `de_methods`. The `enrich_methods` must be a subset of the `de_methods`. Finally, the specific miRNA-related enrichment analyses are demanded with the `-mirnas` flag. Using the `-mir_disease "DOID:2349"` flag, the miRExplore query can be filtered. By specifying the disease context (`DOID:2349`), only miRNA-gene interactions, which are related to this ontology term or children thereof are queried from miRExplore. The term ID `DOID:2349` refers to the term arteriosclerosis, which has atherosclerosis (`DOID:1936`) as child.

Counts Analysis The initial counts analysis aims to investigate whether the acquired data are useful and of sufficient quality for further analysis. As such, the first evaluation is performed on the feautureCount [184] outputs, analysing the mapping and counting efficiency. This is evaluated per replicate and input (here: quantifications of the STAR and HISAT2 alignments). For this evaluation, the number of total alignments is visualized (Figure 6.3). Using the assigned categories (e.g. Assigned, Unassigned_MultiMapping) the user can easily spot what the problem of unassigned alignments was and why these were not counted. This categorization is supplied by featureCounts from the gene quantification step. On this data, a large fraction of reads (about 80%) remains unassigned, independent of the used mapper. Only about 20% of all alignments could be used succesfully for quantification. This might indicate a problem during library preparation, but does not seem to be a problem of the alignment. Most assigned reads align to protein coding genes, but also to several non-coding RNAs such as lncRNAs (Figure 6.4). This, however, is expected as total RNA was isolated for the sequencing. This analysis is provided for every replicate and input matrix with both absolute or relative values shown. Again, no major difference between the input matrices from both aligners can be observed.

After evaluating the single replicates, the input counts from all replicates (all stable and all unstable) are compared in pairwise scatter plots (Figure 6.5). From these it can be concluded that the replicates are similar. For genes with lower counts the variability is larger than for the highly expressed genes. Yet, for both inputs and all shown replicates, the data points are triangularly shaped, indicating good agreement between the replicates. Still, a more narrow distribution would be better, ideally showing a line if perfectly similar.

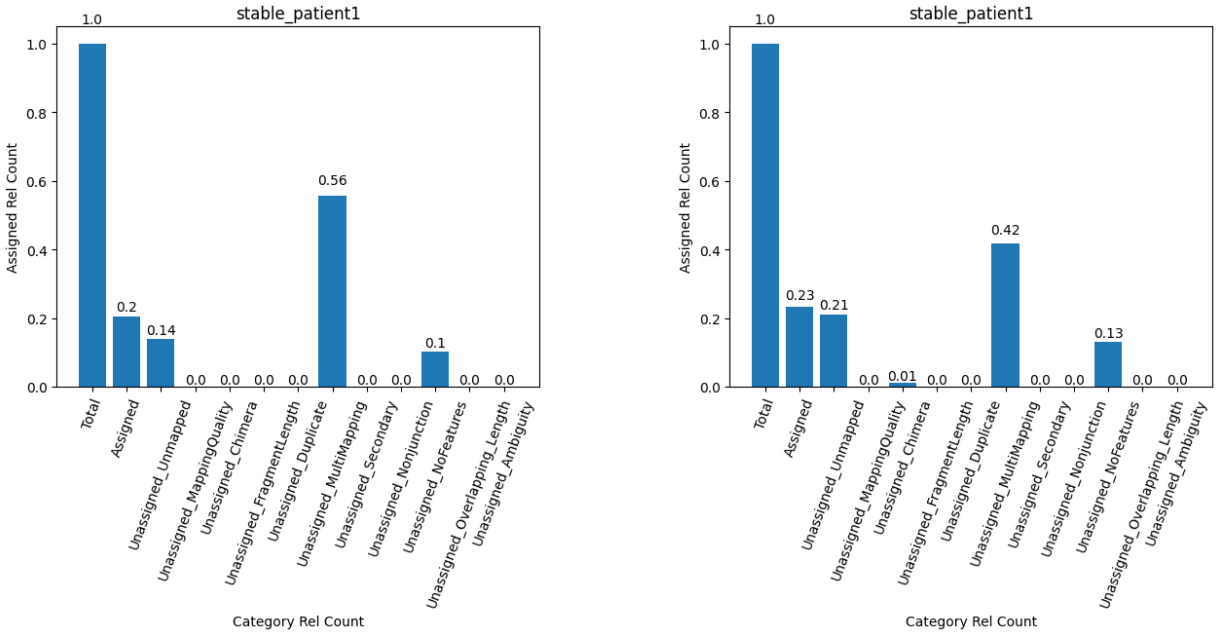
The input replicates are compared using scatter plots (Figure A.38b), where a high similarity between the STAR and HISAT2 alignments can be seen. This purely computational variability is much smaller than the biological variance observed earlier.

For each input, the expression values of the several replicates are compared in a heatmap (Figure A.38a, showing only genes with at least $t = 10$ counts). The `unstable_patient4` replicate behaves different compared to the other unstable sample, at least on a count level.

These results suggest that despite the low assignment-rate of alignments, mainly protein-coding genes were measured and that the replicates are mostly consistent and show similar patterns. The differences of the mapping strategy are quite small, but some outliers show a

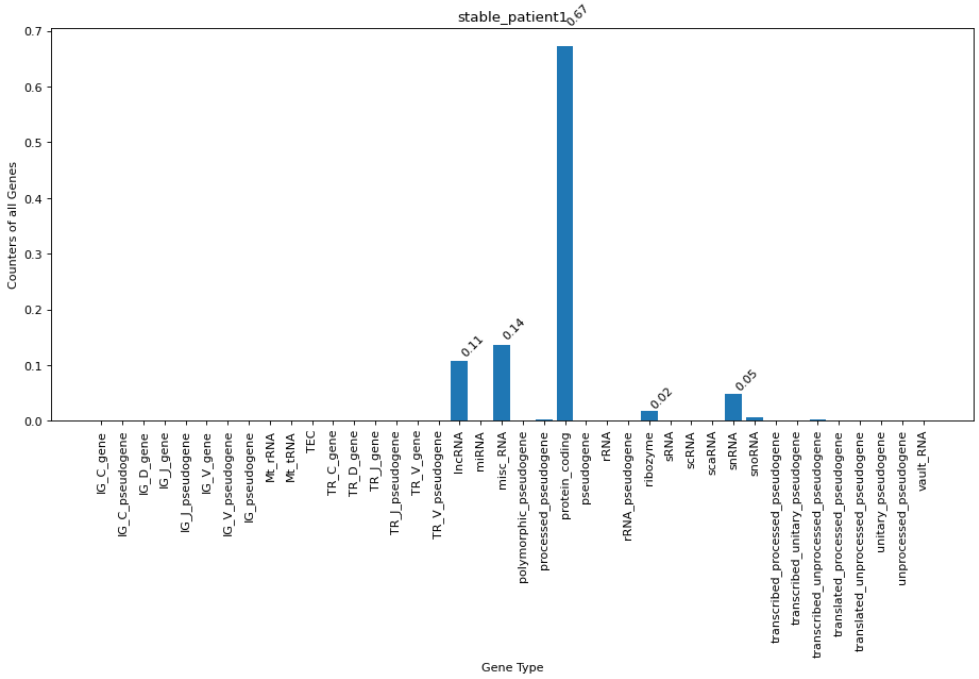
Listing 6.1: Command used to start the DifferentialAnalysis script for the presented use-case.

```
python3 porestat/DEtools/DifferentialAnalysis.py \
--counts \
counts/trim.star.prim.0.5.counts \
counts/trim.hisat2.prim.0.5.counts \
--diffreg \
diffregs/stable_vs_unstable.star.trimmed.diffreg/ \
diffregs/stable_vs_unstable.hisat.trimmed.diffreg/ \
--prefixes \
star \
hisat \
--name \
stable_vs_unstable \
--organism \
hsa \
--enhance \
ensembl.grch38.human.101.gtfout.list \
--lengths \
ensembl.grch38.human.101.gtfout.length.list \
--report \
./reports/stable_vs_unstable.html \
--save \
reports \
--fold_changes --stats --counts_analysis --enrichment \
--prefix-counts \
--de_methods \
"msEmpiRe" \
"DirectDESeq2" \
"msEmpiRe;DirectDESeq2" \
--enrich-methods "DirectDESeq2" "msEmpiRe;DirectDESeq2" \
--cond1 \
stable_patient1 stable_patient2 stable_patient3 stable_patient4 \
--cond1 \
stable_patient1 stable_patient2 stable_patient3 stable_patient4 \
--cond2 unstable_patient1 unstable_patient2 \
unstable_patient3 unstable_patient4 \
--cond2 unstable_patient1 unstable_patient2 \
unstable_patient3 unstable_patient4 \
--condition-no-path --synthetic-names --update \
--mirnas --mir_disease "DOID:2349"
```

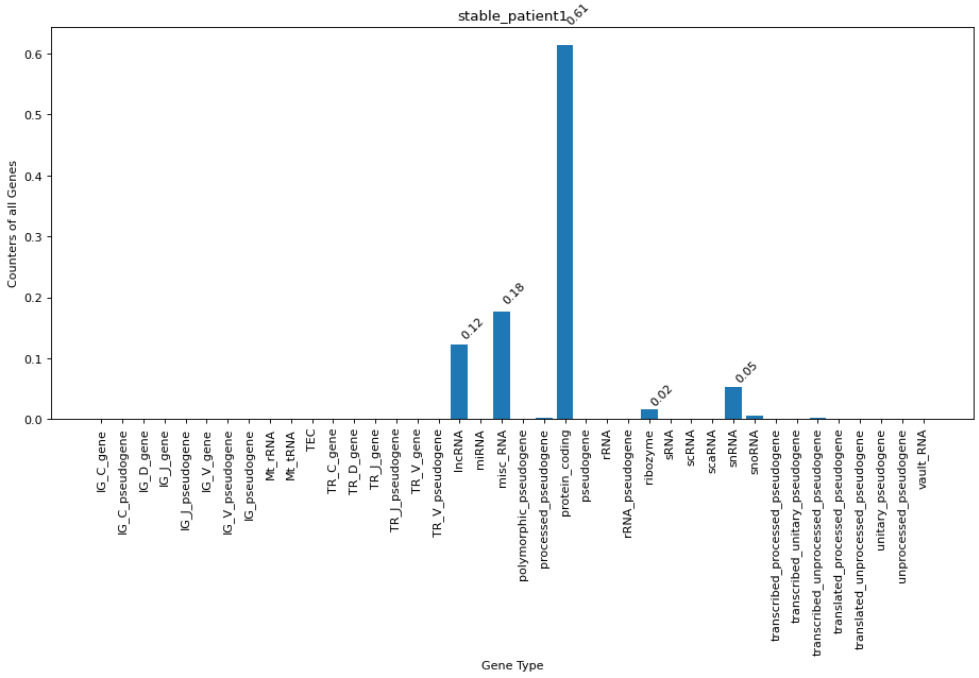


(a) STAR (b) HISAT2

Figure 6.3: **RoDE featureCounts summary** It is assessed how many *alignments* could be used for quantifying gene expression. Only about 20% of all alignments could be used, with a majority of alignments dropping out due to multi-mapping reads. This visualization is prepared per replicate and input.



(a) STAR



(b) HISAT2

Figure 6.4: **RoDE biotype assignment** The biotype assignment is important for checking whether the expected biological entities were extracted and sequenced. A majority of counts is assigned to protein coding genes. It can be noted that non-coding RNAs are counted, which is expected due to the total RNA-seq approach.

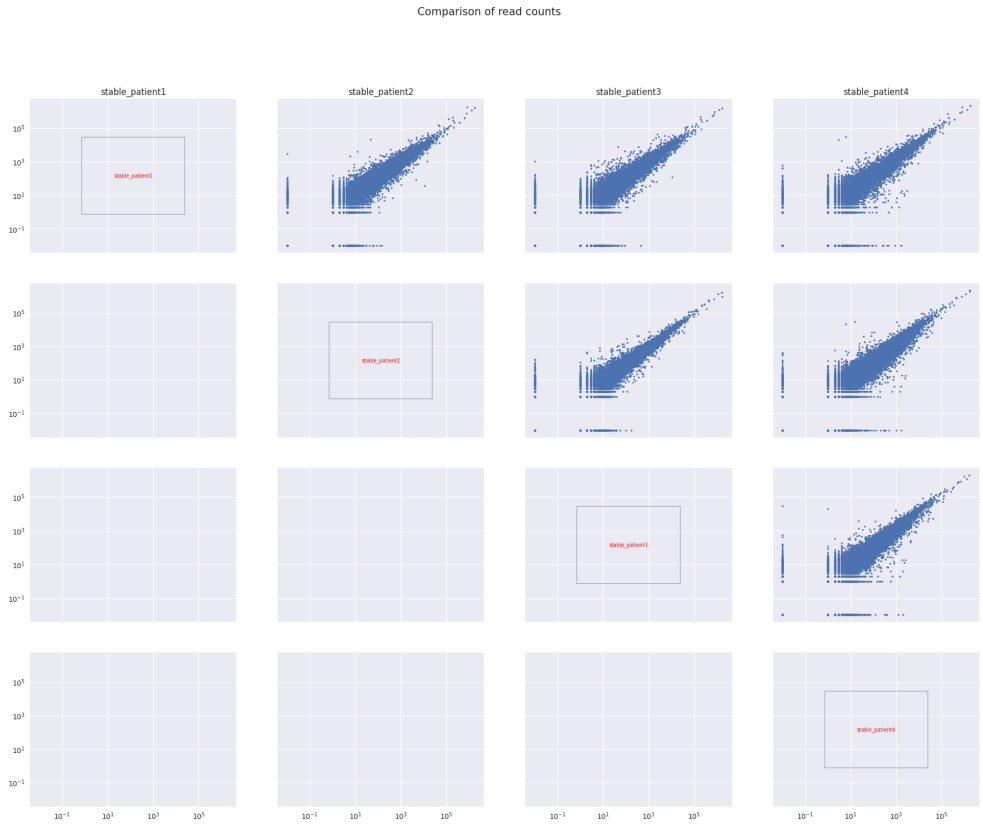
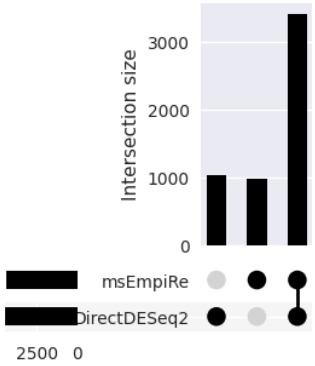
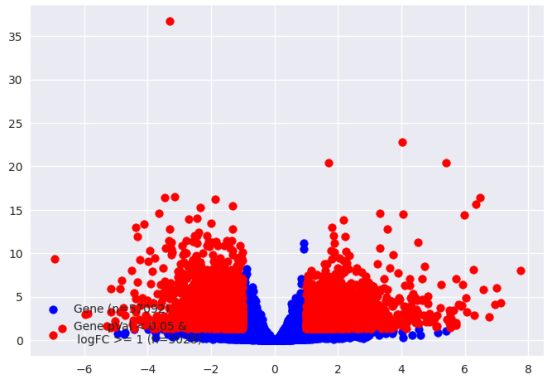


Figure 6.5: **RoDE replicate consistency** The consistency of replicates answers the question of how similar the replicates are. This can be checked by using pairwise scatter plots for all replicates within one condition (here: HISAT2 counts). In these data, the typical conical shape is observed indicating that the single replicates match. The wide scatter in the lower-count areas indicates a high variability in the low-expressed genes.



(a) UpSet plot of DE genes using the two DE methods (msEmpiRe and DirectDESeq2) for STAR input



(b) Volcano Plot of DE analysis using DirectDESeq2 on STAR input

Figure 6.6: **RoDE DE overview** Using the UpSet-plot (a) the two applied methods can easily be compared. The volcano plot (b) allows a fast assessment of the observed meaningful changes. A total of 3028 genes show a significant (adj. p-value < 0.05) absolute logFC ≥ 1.

higher deviation. Yet, on a raw count level, differences between stable and unstable can easily be spotted by eye in the corresponding heatmap.

DE Analysis The DE analysis is the core part of the robust pipeline. The initial overview is performed for each input matrix and DE method, showing the overlaps of the different DE methods' results (Figure 6.6a). Additionally, a volcano plot is presented to get a fast impression of the observed changes between the compared conditions (Figure 6.6b). These two results are plotted side by side for each input for an easy comparison. In this analysis, approximately 3000 DE-method-robust DE genes are found by both applied methods (msEmpiRe and DirectDESeq2).

The report continues with more detailed analyses of the replicate consistency per input. Using a heatmap, the distance between replicates is assessed for both inputs. Using all DE genes for this visualization (Figure 6.7), the stable replicates show a high similarity between each other (top left). For the unstable replicates, a higher heterogeneity in the samples can be noticed. This effect, however, vanishes after filtering for the top 500 up- and down-regulated genes (data not shown). The similarity of the stable replicates increases, and the unstable replicates show a higher similarity. This suggests that the top DE results divide the samples considerably well.

Another way to look into the similarity of the replicates is to consider the technique of dimensionality reduction. A traditional technique for this task is the use of Principal Component Analysis (PCA) on the input matrix (which might be counts, library-size normalized counts, RPKMs or TPMs). More recently, the use of UMAP [210] is recommended for this task, which is particularly suitable to visualize groups of data points while preserving relative proximities. UMAP was used to produce 2D embeddings of both the STAR and

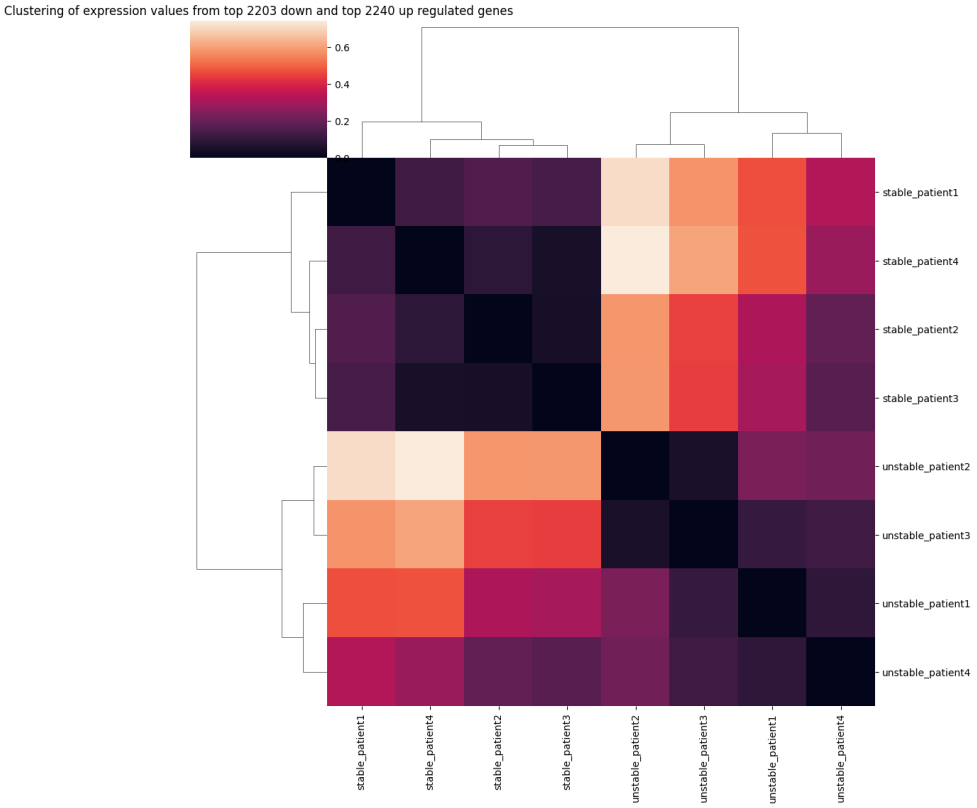


Figure 6.7: **RoDE DE replicate comparison** Using heatmaps, the distance (on DE gene counts) between replicates can be assessed (here: STAR counts; all DirectDESeq2 DE genes). Ideally, all samples of the same group show a small distance (e.g. dark). This expected pattern can be observed for the stable replicates, the unstable ones show a higher heterogeneity.

HISAT2 inputs (data not shown). For both inputs, the stable samples appear closer to each other than the unstable ones. This was already observed in the heatmaps. In both UMAP embeddings the unstable patient 4 sample seems to be more separated from the remaining unstable samples. Considering only the significantly regulated genes, all replicates show a high similarity and a high separation by condition (Figure A.39a). This matches the increased similarity observed in the heatmap.

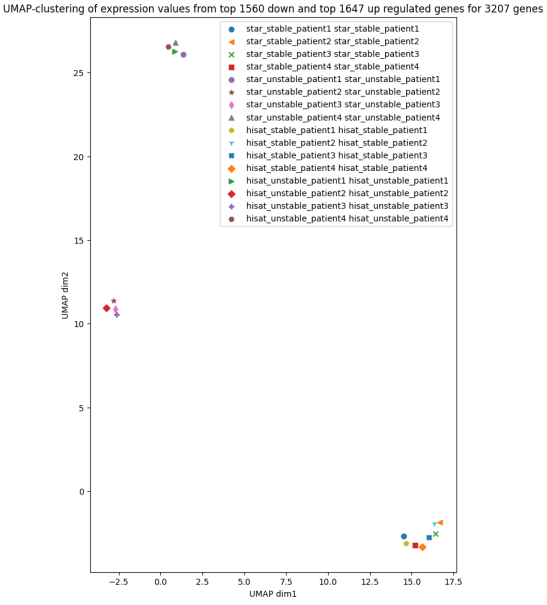
As part of the included visualizations, the pipeline plots a clustermap of the top 50 up- and down-regulated genes. Visualized are the column-wise z-scores of the library-size normalized, z-scaled and logarithmic raw expression values. This clustermap visualization is important for the experimentalists: it helps to get a first glance at the data, and thus is included in most publications. Furthermore, it helps to judge whether the replicates cluster together for the selected genes.

DE Robustness After preparing the results for all DE methods, the combined dataset from all inputs is created. The combined dataset contains the gene expression values from all inputs, and robustly combined DE results. Similar checks as for the regular DE results are conducted.

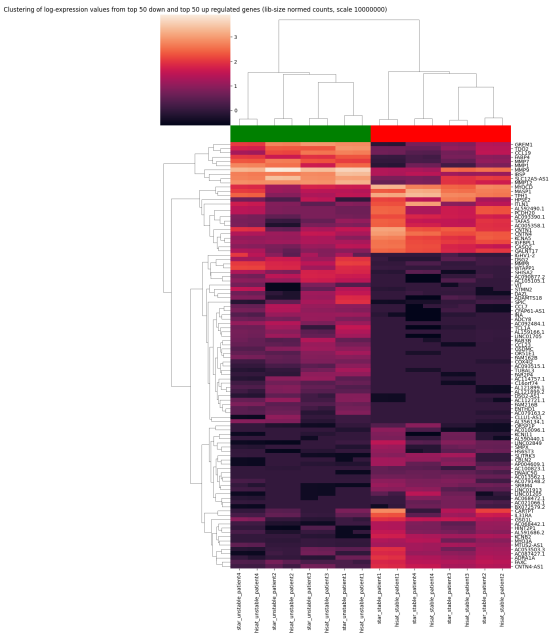
The most important checks are regarding the replicate consistency among all inputs. Hence, the UMAP embedding is calculated for all replicates from both inputs. Initially it can be observed that the UMAP embedding separates the samples by alignment method, underlining substantial differences between the alignment methods, because the gene quantification strategy for both aligners was the same. Considering only the DE genes (Figure 6.8a), the replicates are embedded at similar locations. This effect is stronger for the stable replicates than for the unstable ones. This suggests that there are differences between the aligners, but that the used DE genes divide the conditions well, while leaving aligner-specific differences particularly in the unstable samples. This observation is backed by the clustermap on the top 50 up- and down-regulated genes (Figure 6.8b), where the high similarity between the inputs can be seen.

The UpSet plot of the differentially regulated genes (Figure 6.9) shows that the majority of identified DE genes is robust. There are about 20% method-specific DE genes (844 for DESeq2 or 716 for msEmpiRe), which is by far more than the 57 HISAT2-specific genes, or the 23 STAR-specific ones. Overall this overview gives a good idea of the method- and aligner-specific DE genes, which can be identified by this robust approach.

The robust results can be compared in more detail. Checking the overlap between the top 250 DE genes (sorted by adjusted p-value, Figure 6.10a), compared with the overlap of the top 250 genes sorted by absolute logFC (Figure 6.10b) it must be noted that the overlap of the p-value sorted list is larger than the overlap of the fold-change sorted one, even if only significantly regulated genes are considerd (Figure 6.10c). This can also be observed on the gene rank-plot (Figure A.41), where the p-value ranked comparison shows only few differences. This suggests that there exist genes, which benefit from reads, which were unassignable (e.g. unmapped, mapped elsewhere) in the one aligner, but led to assignments in the other. However, the vast majority of DE genes does not show aligner



(a) Combined UMAP (all up/down-regulated DE genes)



(b) Combined clustermap top 100 DE genes

Figure 6.8: **RoDE combined dataset UMAP and clustermap evaluation** The dimensional reductions and clustering of expression values are repeated on the combined input samples. This allows a comparison on the different input methods, e.g. STAR and HISAT2. It can be noted that the stable replicates show no input-specific bias, while the unstable replicates do (a). This bias can not be observed in the clustermap of the top 50 up- and down-regulated genes (b).

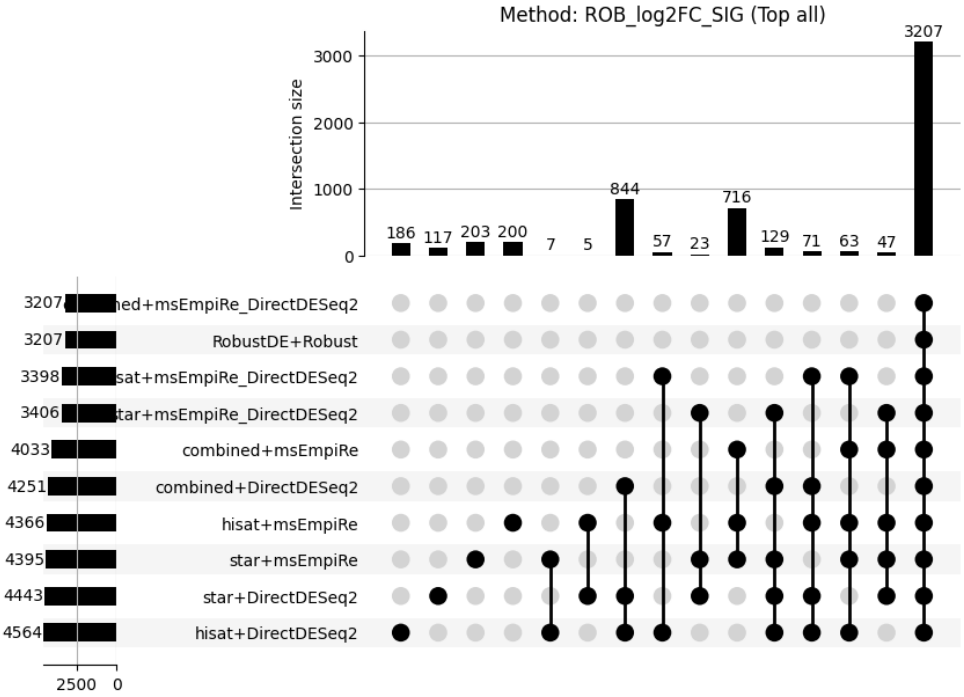
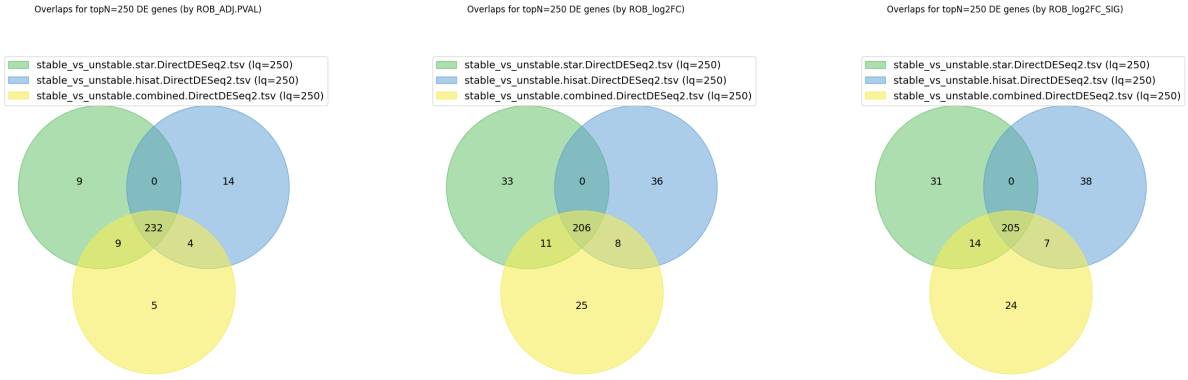


Figure 6.9: **RoDE replicate consistency and (robust) DE result comparison** The UpSet plot of all DE genes shows that there are differences between aligners and DE methods. However, the majority of the DE genes is common to all methods. There are some DE genes which are specific to the DE method. Few are specific to one aligner.



(a) Overlap for robust adj. p-value sorting

(b) Overlap for robust log2FC sorting

(c) Overlap for robust log2FC SIG sorting

Figure 6.10: **RoDE comparison of DE results from all DE methods and robust combination** As part of the robustness analysis poreSTAT calculates the overlap of the top differential genes, for three distinct ways of sorting the data. The overlaps for the top 250 DE genes (DESeq2) are shown, sorted by adjusted p-Value (a), absolute logFC (b) and absolute logFC of significant gene only (c). Green is the STAR input, blue the HISAT2 input and yellow is the robust combination of both. For all comparisons (which only differ in the order) it can be seen that a majority of reported DE genes is common. The significance estimation of the tools is more similar than the estimated logFCs ((a) has higher overlap between all results than (b) and (c)).

specific differences.

Enrichment Analysis The enrichment analysis is the final step of the pipeline. This step is subdivided into three parts: the enrichment analysis on general gene sets, such as KEGG, GO or Reactome, the enrichment analysis on custom gene sets and the miRExplore integration. For the former kind of analysis specific R-scripts are used, which perform the analysis using established R libraries (clusterProfiler [340] and ReactomePA [339]), while the custom gene sets are processed directly in python. The robustness evaluation on the results equals for both parts.

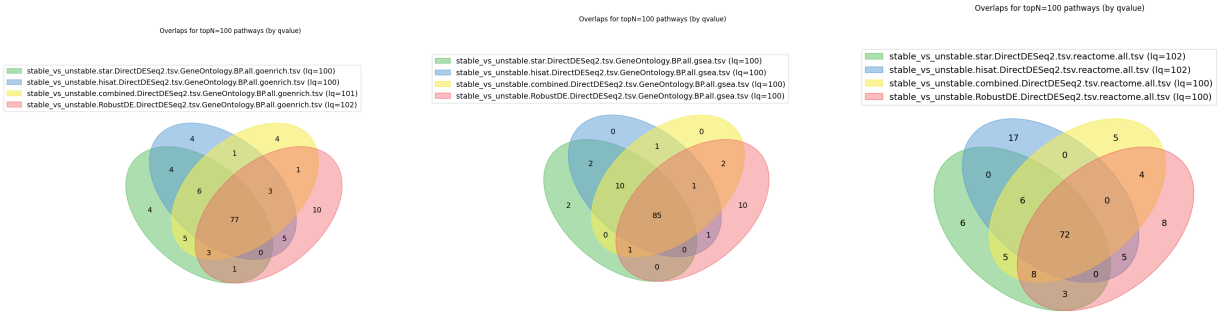
The significant gene sets (logFC $>$ 1.0 and adj. p-value $<$ 0.5) are annotated with matched genes by the respective R libraries. The output is saved in tabulator-separated files, which allows an easy computational processing and import into any spreadsheet office solution. The user can access these files via links from within the final report. For each gene set 3 distinct results are shown: one result considering all differential genes as input, and one result each for only considering up- or down-regulated genes. This feature was requested from multiple users, as this allows conclusions on whether a gene set is up- or down-regulated without performing the (often) more conservative significance testing of gene-set enrichment analysis (which would be the more trustful test, though) [295].

As part of the result presentation, the top enriched gene sets of all gene sets are visualized by bar plots (Figure A.40). This allows a fast and easy comparison between the different robust results. These plots show distinct bars for using all genes, and only the subset of all up- and down-regulated genes, respectively. The reported gene sets are sorted to maximize the difference between the UP and DOWN results, under the condition that both UP and DOWN results are significant. Using the miRTarBase [128] miRNA target sets for over-representation analysis, miR-297b, miR-146a and miR-410 are among the top ranking miRNAs and are thus potentially playing an important role in the transition from stable to unstable atherosclerotic plaque.

In order to ease the evaluation of the enrichment analysis results (distinct inputs, combined or robust analysis), Venn diagrams are used to display the gene-set overlaps. A majority of the results is robust (Figure 6.11). However, there is some fraction that is unique to the combined or robust analysis. Particularly these differences occur with the robust comparison input, as here significant genes are taken away, leading to missing results (in contrast to the other inputs) or a different ordering. Yet the vast majority of results (about 80%) are robust, which matches the minor differences observed in the DE genes. The difference in the used aligner (STAR or HISAT2) is marginal, but the highest with the Reactome Pathways.

For the GO (Biological Process domain) over-representation enrichment (Figure 6.11a) all input-unique enriched gene-sets are found in the other inputs, yet at higher ranks. This suggests that there is a difference in ordering, and also a difference in DE genes, but this does not yield highly different gene set enrichments. However, this does not seem to be generalizable: for the Reactome pathways (Figure 6.11c), there are pathways which are only found by using the HISAT2 input. Among the HISAT2-only pathways are methylation related ones (*DNA methylation*; *PRC2 methylates histones and DNA*; *RMTs methylate histone arginines*) as well as a acetylation related pathway (*HDACs deacetylate histones*). It is already known that epigenetic plays an important role in atheroprogression, but particularly histone and DNA methylation are known to be 'altered in atherosclerosis, suggesting a possible contribution of epigenetics in disease development' [108]. More recently, the contribution of histone deacetylases (HDACs) was brought into focus of regulating vascular cell homeostasis and thereby atherosclerosis [54]. There are further HISAT2-only pathways which support an epigenetic activity in atherosclerosis. These are missed when relying on STAR for alignments or the robust results. This shows, that in certain cases the choice of the aligner can make a difference, and that robust results may hide interesting and relevant observations. However, for a vast majority of results, the robust view on the data gives additional evidence and support. With the help of the robust DE pipeline it is possible to identify such differences, making it possible to rate the differences.

As part of the DE pipeline, an analysis of possible regulating miRNAs can be performed on interactions retrieved from miRExplore (Chapter 2.3). The results, which are discussed here, are from the method- and alignment-robust variant. Within the report, regulating miRNAs are made available in a table as well as in a network visualization (Figure 6.12). Strikingly, the performed over-representation analysis identified method-robust significantly regulating miRNAs (adj. p-value < 0.05) with three or more targets. These miRNAs



(a) Overlap GeneOntology Biological Process (ORA)

(b) Overlap GeneOntology Biological Process (GSEA)

(c) Overlap Reactome

Figure 6.11: **RoDE evaluation of enrichment robustness** Similar to the DE gene overlap comparison (Figure 6.10), these overlaps are also calculated for the q-value sorted enrichments (of both the general gene sets and custom one). Again, a majority of the top 100 reported gene sets does not differ between the inputs and the robust result (using different DE methods). The results determined by gene set enrichment analysis (b) seem to be more robust than those of the over-representation analysis (a).

are miR-155, miR-20a and miR-590, which are predicted to be less prevalent in unstable plaque. Other interesting and interacting miRNAs are miR-467b, which shares its targets with miR-590, and miR-181a, which shares targets with miR-20a. All these miRNAs (except for miR-467b) are also significantly enriched in the miRTarBase over-representation analysis (considering up-regulated DE genes). However, these miRNAs are not among the top enriched ones. Using the context-sensitive approach of the miRExplore integration, miRNAs, which are only significant because not context-relevant interactions are considered, are sorted out. With miRExplore, a focus on known and context-relevant miRNA regulators in arteriosclerosis (DOID:2349) is set.

It is already known that miR-155 participates in the atherogenesis. Therefore, it is an interesting target for clinical research regarding the resolution of atherosclerosis [39]. Particularly its role in macrophages, which are one of the many compounds of plaque, is of interest. The interaction with CSF1R has already been investigated further [323]. The miR-155 interactions with ADAM10 and FLT1 are verified by the miRTarBase resource [128] in accessions MIRT021045 and MIRT020776. The interaction of miR-155 with CD68 [344] is interesting, because CD68 is expressed by cells of the monocyte lineage (e.g. macrophages). CD68 is already known to be upregulated in unstable plaques [253]. The miR-155 interaction with CXCL8 is interesting due to controversy reports regarding the direction of regulation: there are reports that miR-155 overexpression reduces CXCL8 (also known as IL-8) production [298] in *Helicobacter pylori*-induced inflammation, or that silencing miR-155 in lipoprotein-stimulated macrophages promoted the release of CXCL8 [129]. Yet there are other reports which found (in chronic immune-mediated inflammatory dermatosis) that miR-155 overexpression increases CXCL8 prediction [319]. The presented data suggest that a lack of miR-155 increases CXCL8 production (in a

canonical way), which is a chemokine produced by macrophages or endothelial cells, thereby promoting angiogenesis [214], a key process in atherosclerosis. Because stable and unstable atherosclerotic plaques are compared, this found involvement of miR-155 is reassuring.

Another miRNA which was highlighted by the miRExplore analysis is miR-20a, which regulates PYCARD (ASC), but also TXNIP and NLRP3 [182]. While these interactions were explored in fibroblast-like synoviocytes, it is known that these cells play a crucial role in the pathogenesis of chronic inflammatory diseases, like rheuomatoid arthritis. The link between fibroblast-like synoviocytes, rheuomatoid arthritis and atherosclerosis was established by Bernhagen et al. [215]. The incorrect annotation of IL-18, due to text mining errors, does not affect the significance of miR-20a.

The presence of miR-590 in this comparison is not unsurprising: it was already found in the previous atheMir evaluation (Chapter 2.2), and regulates CCL2 expression in macrophages. Indeed, miR-590 targets CD68 and LPL directly [119] and thereby decreases plasma levels of pro-inflammatory cytokines. The interaction of miR-590 with IL-18 is also relevant for atheroprogression, indicating an involvement in angiogenesis [181].

Another interesting interaction is miR-467b targeting LPL, which was found to be related with an onset and development of cardiovascular disease [316]. Regarding the annotation of the miR-467b interaction with IL-18, this is due to the same text mining error as above. NLRP3 and IL18 are already discussed targets of miR-181a [285].

With the miRExplore extension, the robust DE results from earlier pipeline steps are used to predict regulating miRNAs. All discussed miRNAs are known to be relevant to the progression of atherosclerosis. The targeted genes were confirmed to play central roles in distinguishing stable plaques from unstable ones. The identified miRNAs are thus interesting targets for preventing atheroprogression in its late stage.

Conclusion

In this section the concept of a robust DE analysis has been demonstrated in a consequent way for an analysis in the context of atherosclerosis. RoDE makes the comparison of different input strategies (e.g. aligners or gene quantification), and differently processed DE results possible. Using the provided visualizations it can be analysed whether the replicates are of the required quality and sufficiently similar, and thus justify a further analysis. Particularly by diverse overlap comparisons, several DE methods can be compared and combined, such that differences in DE genes are highlighted. Furthermore, the effects of the different approaches on the enrichment analysis results are elaborated. This is important because differences on the DE gene-level could be meaningless for downstream analyses. However, if these difference on the gene-level induce changes in the significant functional pathways, these differences might still be of importance. In the presented analysis it could be seen that the choice of the methods (be it for alignment, quantification or DE) matters. The input-specific Reactome pathways were shown to be highly relevant for the use-case setting in atherosclerosis. This makes the point for RoDE: with this pipeline it is easy to spot such occurrences. If alignment or method-specific gene-sets are found, further research into these can be conducted. RoDE integrates with the miRNA-regulatory prediction introduced in

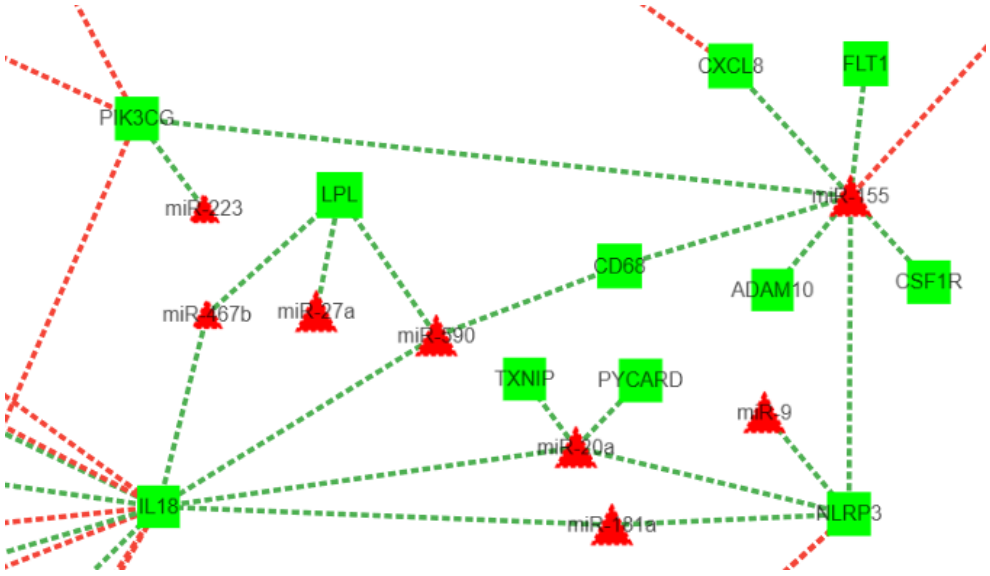


Figure 6.12: **RoDE miRNA-gene regulatory prediction using miRExplore** Excerpts from the miRExplore predicted miRNA regulatory network (robust, DirectDESeq2+msEmpiRe). Significant regulators ($p < 0.05$) in terms of overrepresented targets are miRNAs miR-155/20a/590, which are suspected to be down-regulated in unstable plaque.

this thesis with miRExplore (Chapter 2.3). The identified regulating miRNAs, identified from robust DE results, could be confirmed regarding their relevance in atherosclerosis.

RoDE is particularly suitable for preliminary experiments, which serve for hypothesis generation. By comparing and combining several analysis strategies the most can be retrieved from the data. This pipeline was applied in still ongoing projects. These projects are within the atherosclerosis context (yet unpublished) and within the SARS-CoV-2/COVID-19 context [238].

6.2 Building a Multi-modal Model of Atherosclerosis (Aorta3D)

Within research groups, or collaborative research centres, huge amounts of *omics*-data are acquired. While every group looks into a specific process of the research topics, there are similarities in the experiments, such as a common disease of interest. Moreover, the experimental outcomes could serve as comparison data for future experiments. Additionally, many non-*omics*-data and non-high-throughput data are generated, like microscopy images. While these are, in general, not comparable with high-throughput data, e.g. from sequencing or proteomics, recent advances in transcriptomics and proteomics make this possible. Using spatial transcriptomics and spatial proteomics, microscopy images with several staining can be brought in relation with such high-throughput data. Indeed, spatial transcriptomics has

been awarded *Method of the year 2020* by Nature Methods [84], and spatial transcriptomics has already been discussed in this thesis at the example of MALDI-TOF IMS (Chapter 4.4). But even without spatial transcriptomics, the availability of cell-type specific transcriptomics data, or IMS data, warrants an integrative analysis of these data types in a complex human disease context.

Introduction

In Chapter 4.4 methods for the analysis of IMS data have been presented. IMS enables the measurement of hundreds to thousands different masses and proteins within their biological topology. This ensures that little to no additional noise or contamination is introduced into the system, e.g. due to sample processing steps. Such measurements have been the domain of microscopy, combined with antibody staining. This, however, only allows the measurement of a few proteins (via antibodies) at a time, at high costs.

Any IMS method aims at providing an image of a sample while being able to resolve active analytes (proteins, peptides, lipids, etc.) at a higher rate than antibody staining. Typically, after performing the mass spectrometry, the sample is intact such that at least a stained microscopy image can (and should) still be acquired. There are even reports which claim that antibody staining is still possible [246].

Several possibilities exist to use IMS as base level analysis for an integrative setting. Prade et al. describe a multi-modal approach in which IMS data are integrated with additional microscopy data after antibody treatment [246]. Neumann et al. describe the use of multimodal IMS as the future for analysing medical and biological systems [226], specifically referring to the application of different IMS methods to the same sample, combined with traditional microscopy data. The rationale behind the usage of multiple of such methods is that each method has specific properties [226]: with matrix-assisted laser desorption/ionization (MALDI) IMS a high spatial resolution of typically around $5 - 20\mu m$ can be achieved, at a regular extraction rate. Using secondary ion mass spectrometry (SIMS) much higher primary ion doses and beam currents can be reached, allowing depth profiling and three-dimensional imaging (without the need of slicing the sample). Desorption electrospray ionization (DESI) is a minimally destructive technique, which can even be used in a clinical or surgical setting. Using the SPACiAL framework Prade et al. [246] are able to (manually) align multiple samples, e.g. from multiplexed immuno-stainings, and combine these with an IMS lipid measurement. This enables co-localization analyses, and cell-specific analyses.

With Aorta3D a similar idea is followed. Aorta3D is meant to suit as a 3D-data index for atherosclerosis related measurements integrating different measurement techniques and stages of the disease. Data is to be accessed either from a table, or via a graphical model of the disease. Upon selection of a disease element, e.g. a late-stage vessel with plaque, related experiments are shown, from which the user can select a relevant one. For this experiment, specific experimental results are displayed. Included experiments range from scRNA-seq experiments, over IMS data up to related imaging data. The latter two are aligned where possible, enabling an integrated analysis. Particularly the integration of cell type-specific

results, e.g. via scRNA-seq experiments or cell type predictions for IMS data (Chapters 4.1, 4.4), is a novel feature allowing integrative insights into the disease.

Methods

This section presents the methods applied in Aorta3D. The Aorta3D project is structured into three almost independent tasks. With a focus on IMS data, the interaction between the IMS data analysis framework `pIMZ` (Chapter 4.4) and Aorta3D is important, and a specific interface has been implemented. In order to align different images, such as microscopy images with the IMS results, or the IMS results within each other, specific alignment technology was implemented as part of the Bachelor thesis of Margaritha Olenchuk [232], which was supervised by this thesis' author. Finally, the web-platform for browsing all contained data was developed specifically for Aorta3D.

pIMZ integration The integration of Aorta3D with `pIMZ` (Chapter 4.4) is achieved via a common output format. All data is shared via json-formatted configuration files. The main configuration files contain information about the region-name, as well as links to files describing the identified segments (as image, pixel-coordinates and numpy-matrix), their marker masses, as well as to a hdf5 file containing m/z intensities required for data presentation.

Image Alignment The alignment of images and IMS data needs as input a json-file (Chapter A.1) containing links to the segmented image (as image and numpy-matrix [116]). Using the segmented image from the numpy-matrix and all pixels associated with non-background clusters, the boundaries of the measured object are calculated. These boundary images are then transferred into the registration pipeline, which extracts key features via the BRIEF algorithm implemented in scikit image [312]. With these features, the warp algorithm from scikit image can then calculate the transformation, which maps the measured area onto a reference (which is defined as the sample, which is most similar to all others).

From the aligned images, 3D representations of the sections are calculated. This is done using a python implementation of the SurfaceNets algorithm[3]. These 3D representations are referenced in the configuration file for easy access through Aorta3D. The workflow for aligning microscopy images has been implemented as part of Olenchuk's Bachelor thesis and therefore is not presented as part of this thesis [232].

Data presentation The Aorta3D web framework is made up from two components: the python-based flask[4] application and the TypeScript[5]/React[6]/MaterialUI[7]-based web-frontend. The server backend is responsible for delivering all required data. As such, it

[3]https://github.com/mjoppich/surfacenet_python/
[4]https://flask.palletsprojects.com/en/1.1.x/
[5]https://www.typescriptlang.org
[6]https://reactjs.org/
[7]https://material-ui.com/

is capable of reading and handling the json-configurations created from `pIMZ` and the image alignment stage. The server provides several routes from which the frontend can demand experimental data (e.g. expression data), images or descriptions. The frontend orchestrates the display of all relevant information and consists of multiple distinct components which were specifically created for Aorta3D. The `Aorta3DRenderer` can display and render objects in 3D, e.g. the aligned 3D models of IMS measured regions. The `Aorta3DClickableMap` allows user interactions on images and shows specific element information in the `Aorta3DElemInfo` component. The `Aorta3DRelatedExpsViewer` is designed to show related experiments, which might be pre-filtered on the selected structures (e.g. plaque) or even on the gene-level, when the selection is made in the `Aorta3DExpAnalyser` component, which displays expression data from either IMS or scRNA-seq experiments.

Results and Discussion

pIMZ and scRNA-seq integration The integration of Aorta3D with `pIMZ` (Chapter 4.4) and scRNA-seq analysis (Chapter 4.2) shows the high interoperability between these parts in an integrative setting. The integration with scRNA-seq is particularly easy, because the cell type prediction receives all relevant data as input: the marker genes per clusters. The cell type predictions are the regular output of the method. With `pIMZ`, the workflow is similar. In addition, `pIMZ` creates additional config files with both cell type information and information about the several clusters. This information is needed for the alignment of these regions. As part of the region alignment 3D models for each region are created, which can be visualized (Figure 6.13).

Data presentation The functionality of Aorta3D is described using the input data from the IMS project `pIMZ` presented in Chapter 4.4 (Slide D, regions 0, 1, 4, and 5, Figure 4.6). These data are accompanied by the single cell RNA-seq data of human and mouse atherosclerotic tissue, which were already described in Chapter 4.2.

After loading the Aorta3D website, the user is shown a summary statistics of all included experimental data. For each experiment type, the number of recorded experiments is listed. In addition, the annotated cell types or tissue regions are listed per experiment type. The main analysis page shows the 3D representation of an atherosclerotic blood vessel, including schematic elements like the vessel walls or some schematic sections (Figure 6.14). This view can contain a 3D representation of experimental IMS data. Using the 3D representation, data can easily be selected by disease stage or region. Schematic representations of specific cell types could be thought of as more general selectors.

On selecting a schematic representation of a section, this representation is loaded into the element info object (Figure 6.15a). The left part of this view gives general information about the element, and the right part features a ClickableMap. A ClickableMap object is a visualization, which can be used to select an element from an image. Here, the ClickableMap only serves as image display. Of higher interest is the list of related experiments which is shown below (Figure 6.15b). All experiments, which relate to the selected element in

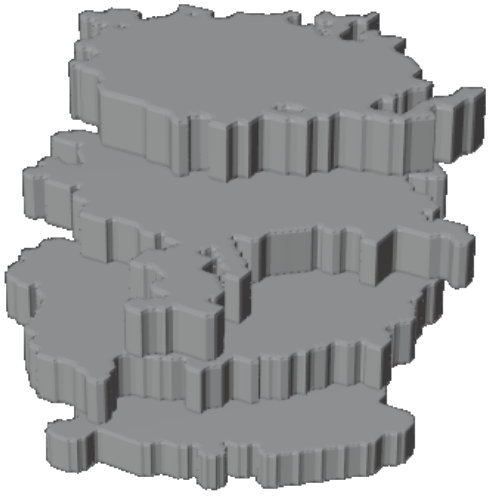


Figure 6.13: **Aorta3D region alignment** The 3D representation of the aligned regions 0, 1, 4 and 5 from slide D (Chapter 4.4, Figure 4.6).

the element info, are shown. This relation is defined by the experiment details which are annotated to both the selected element and the single experiments. The shown section features aorta (vessel) and plaque. Hence, all experiments which are related to these features are displayed in the *Related Experiments*. The related experiments list is searchable and filterable. Furthermore, detailed information for each experiment can be shown by clicking the *Details* button. Using the *Blend* slider, multiple images can be drawn over each other. By clicking on the detail button for the second experiment, a further experiment info is displayed (Figure 6.16a).

As already highlighted earlier, the image shown on an experiment info is a ClickableMap object (Figure 6.16a). A ClickableMap object allows the selection of a specific pixel on an image. More precisely, a click on a ClickableMap will select all pixels of that specific colour. For instance, clicking on any pixel of the central cluster on this image, sets this pixel as the selected pixel (Figure 6.16b) and shows the cluster selection in red for all pixels of the same cluster. The ClickableMap then queries the server for further information on this pixel, e.g. the pixel annotation, which here is *aorta*. In the case of spatial high-throughput data, like IMS data, *DE Analysis Results* for this region are displayed. For IMS and scRNA-seq the shown data are the marker masses and genes, respectively. With these data the user can easily spot genes or proteins which are relevant for the selected region. For the central cluster it can be seen that Ccl27a is highly up-regulated in this area. With only the cluster representation visible (like in Figure 6.16a), the user can hardly relate this finding. Thus, further imaging data can be blended into this image. This is done by

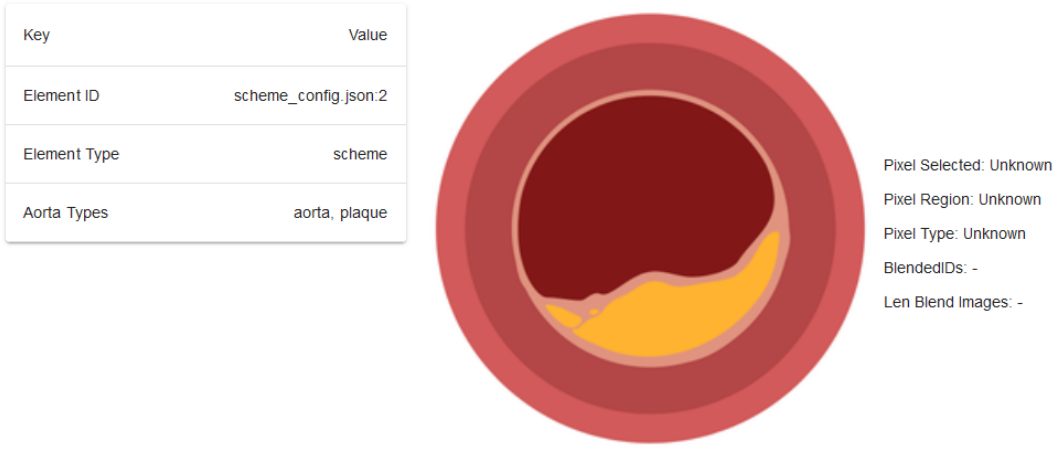
Figure 6.14: **Aorta3D visualization** The measured regions (Figure 6.13) are shown in the Aorta3D selector at the levels the user provided. Together with the actually measured samples, schematic objects show an atherosclerotic vessel.

selecting the slider in the *Blend* column of the related experiment viewer. The element info shows how many images are currently blended in. The interactive volcano plot is part of the *DE Analysis Results* view, which allows to explore the marker masses or genes by significance or fold-change. The `ClickableMap` is only shown for data with associated images, like IMS, but not scRNA-seq results.

From within the *DE Analysis Results* view the result list has a *Details* column with a button element. Upon clicking this button, the related experiments list will filter for experiments where the selected gene is among the marker genes.

Conclusion

With Aorta3D regular RNA-seq or scRNA-seq data can be combined with spatially resolved data, like IMS, or spatial scRNA-seq data. Using the 3D-interface, Aorta3D allows a fast browsing of all data along the progression of atherosclerosis. The 2D-ClickableMap-interface



(a) Selected schematic element in the element info object.

Related Experiments

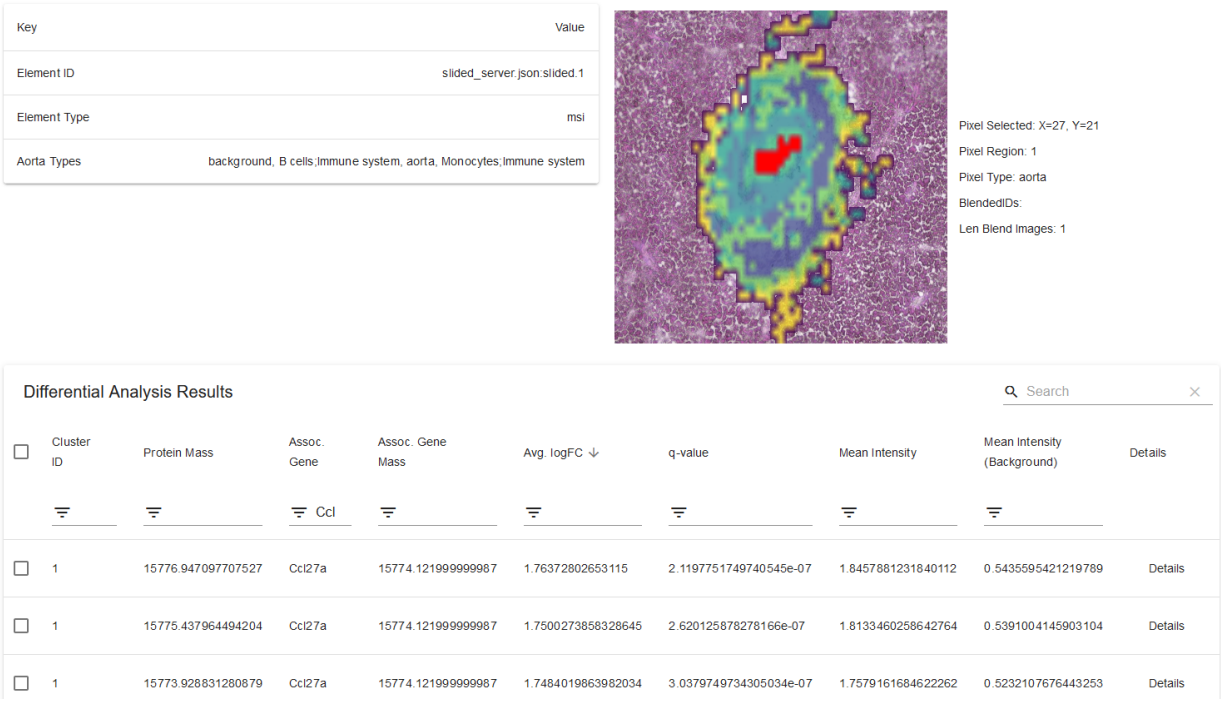
Search

	Experiment ID	Exp-Type	Detail Type	Sample Location	Plaque Level	Plaque Rate	Blend	Details
☐	slided_server.json:slided.0	msi	background; aorta; Basal cells;Epithelium		85			Details
☐	slided_server.json:slided.1	msi	background; B cells;Immune system; aorta; Monocytes;Immune system		75			Details
☐	slided_server.json:slided.2	msi	background; Mast cells;Immune system; aorta; Smooth muscle cells;Smooth muscle; Satellite cells;Skeletal muscle; Macrophages;Immune system		20			Details
☐	slided_server.json:slided.3	msi	background; B cells;Immune system; aorta; Satellite cells;Skeletal muscle; Smooth muscle cells;Smooth muscle; Macrophages;Immune system; B cells naive;Immune system; Basal cells;Epithelium		35			Details

(b) List of related experiments for the selected element (a).

Figure 6.15: **Aorta3D single element information and related experiments browser** Upon hovering over a specific element in the 3D browser (Figure 6.14), a detailed element information can be displayed. When selecting a schematic section, this section is shown as a clickable map (a). Relevant experiments are displayed in the related experiments view (b). Selecting an element therein shows additional information about this experiment.

(a) Experiment Info: Selected Slide D Region 0



(b) Selected Slide D Region 0 with blended image and cluster 1 selected

Figure 6.16: **Aorta3D blended image and cluster selection module via ClickableMap** Upon selecting Slide D region 0 from the related experiments the experiment info for this data is presented. Using the blend function, images from other experiments can be visualized in the current experiment info as well (a). Using the clickable map functionality, specific clusters on this image are shown. Clicking on the inner part of the shown artery selects all pixels of the same cluster and highlights these in red (b). Below the experiment info a *Differential Analysis Results* view is shown if such information is available. Here, this view contains all marker masses for the selected cluster identified by pIMZ.

then allows a quick selection of the actual region of interest. In the presented use-case, these were the clusters as determined by pIMZ, but images from microscopy are equally usable to identify cell types, for instance.

Last but not least, by integrating the many data types into one resource, Aorta3D suits as a 3D-index for multi-omics data. All datasets can be browsed simultaneously, which makes the identification of relevant data possible.

6.3 Conclusion

The robust DE pipeline RoDE takes count matrices, generated from the output of different read aligners or quantification methods, in order to produce a robust analysis of RNA-seq samples. Using RoDE assures not only reproducibility and replicability, but also creates a robust result. With the possibility to add further DE results into the analysis, a feature for more generalizability is implemented. During the pipeline's stages a focus is set on replicate consistency on different levels. First the robustness is evaluated on a gene level: for each (combination of) DE method(s) and all inputs. After the enrichment analysis, these robust comparisons are repeated on a gene-set level. Using such comparisons it is possible to spot irregularities and input-specific results which warrant further research: with the use-case data highly disease-relevant gene-sets were identified using just one specific combination. With a regular RNA-seq pipeline this would not have been identified as such, or even been missed, because the relevant combination of input and methods wouly maybe not have been performed. With the robust DE pipeline, not only robust results can be generated, but also method-specific results can be discovered. Finally, RoDE integrates the miRNA-regulatory prediction presented with miRExplore in Chapter 2.3.

With Aorta3D the results obtained from scRNA-seq analyses (Chapter 4.1) and pIMZ (Chapter 4.4) are combined. Aorta3D is a method to index atherosclerosis relevant data in 3D. By connecting both IMS and scRNA-seq, together with spatial information from IMS or microscopy, relevant data can be accessed by gene or protein name, spatial location, or, given annotated data, by cell type.

With the currently available data, Aorta3D can already serve as a frontend to pIMZ results and as an atherosclerotic experiment data index. However, given microscopy data which fits the IMS data, co-localization analyses could be performed using the already provided means of blended images, for instance. The accessibility of expression data from a 3D index, and the availability of DE analysis data from a spatial-2D view is a new way of accessing disease-related high-throughput data. It can be expected that such a framework makes the analysis of high-throughput data more intuitive and allows an easier discovery of interesting connections from both high-throughput IMS and scRNA-seq data, in combination with regular imaging experiments.

I think it's much more interesting to live not knowing than to have answers which might be wrong.

Richard Feynman

7 Perspectives for Future Research

The contributions of this thesis to the scientific community were lined out in the respective chapters. In general, the presented methods and frameworks were applied to a very specific context only. How these can be improved further in future research, e.g. by applying them to broader scientific contexts, is discussed in this chapter. Additionally, a transfer of the developed methods to other scientific questions could be rewarding. Finally, the contributions of this thesis are set into context with ongoing trends in research.

Text-mining miRNA-gene interactions (Chapter 2) The miRNA-gene interaction extraction relies on the NER of biomedical entities, including genes and miRNAs, in order to identify miRNA-gene co-occurrences. This process is followed by a two-pass interaction classification, first deciding whether a miRNA-gene interaction can be accepted, before the interaction direction is then predicted. Both classification tasks are rule-based classifications.

During the development of miRExplore it has been found that the NER software syngrep[1] does not find all gene symbols in certain circumstances, which could be explained with the hardcoded rules of syngrep. This problem particularly occurs in conditions where, for instance, nested gene symbols exist. Using the python-based re-implementation of syngrep, which performs the NER without such specific rules, the above behaviour can be circumvented. This, however, leads to a decrease in recall and precision of the NER, because some abbreviations are currently not correctly recognized (recall), and ambiguous abbreviations are not ruled out (precision). An improved version of syngrep for the gene- and miRNA-recognition, and with a correct abbreviation detection, might be beneficial. Another strategy for improving the detection rate of syngrep-identified miRNA-gene co-occurrences is the use of NLP to classify the entity type of found miRNA or gene entities. If

[1] Csaba, Gergely. Personal Communication. 2019.

a classification does not match the class gene or miRNA, the co-occurrence is rejected. This resembles a robust, NLP-assisted NER approach. Given the performance improvements of current hardware, NLP- and NER-methods, such a dual approach is computationally feasible.

In addition, it was noticed that for the rule-based miRNA-gene interaction classification missing context words were problematic. In some cases a specific rule was not triggered, because a fixed phrase introducing an interaction was not found. After adding the newly identified context word to the list of known context words, the interaction in question is identified correctly. This shows that the context rule in general is useful and working, but relies on additional lists of context words. These lists, however, are hard to create oneself, and the nature of language almost guarantees that the list will never be complete. Instead of manually curating this list, as well as other lists and rules, a supervised learning approach could learn these keywords. The current state-of-the-art software usually relies on a rule-based relation extraction. With the advent of novel deep-learning based models for text mining, and in particular relation extraction (like BERT [72]), also in the domain of biomedical text mining [171], the use of such technology for this task should be looked into. For regular relations such tasks have already been benchmarked with good results, even on document-level relation extraction [112], which is a harder problem than the sentence-level detection performed with miRExplore. Indeed, by manually curating the lists of context words for the rule-based approach, a manually supervised learning is performed. Adding words to the lists can be interpreted as adding more rules to the system. Hence, these rules and words could be learned in a supervised manner, leading to even better results. However, the down-side of this approach would be the requirement of adequately sized annotated training and test data. However, relying on pre-trained models like BioBERT, reduces the amount of required training data. In fact, the existing miRExplore database can be used as training data: even though the training data may not be 100% correct, it is sufficiently well to improve the pre-trained model. If the errors, which occur in the training data from miRExplore, are seldom enough, and thus appear not systematically, the chance is high that the fine-tuning will not learn these errors.

Finally, this thesis presents the interaction detection in the setting of miRNA-gene interactions. It could be shown that the presented approach achieves a reasonably well specificity and sensitivity, expressed by a F_1 score of about 0.95. But there are further biomolecular entities which can interact with each other, such as transcription-factor (TF)-gene interactions. There are no other regulation directions in this case, but the canonical regulation would be `TF-gene UP`. It would be interesting to apply the presented interaction classification to the TF-gene interaction problem. Having a database of both, miRNA-gene and TF-gene interactions would allow the construction of more complete gene regulatory networks, containing both repressors (miRNAs) and activators (TFs). This, hopefully, would allow for a better modelling and simulation of active regulators on DE analysis results.

bioGUI usability and tsxCount interoperability (Chapter 3) With the bioGUI framework, existing bioinformatics software is made available to users, even if they are not computer experts. Currently bioGUI does not rely on any repositories like conda[2], bioconda [66] or Chocolatey[3]. Providing install modules for any package available in bioconda, and automatically extracting the GUI template from the main script, or at least from python scripts, could massively boost the number of available software through bioGUI, and thereby contribute to a more user-friendly scientific domain. With the already existing tools this step would be possible with a considerable amount of work.

The performance of serialization techniques for bioinformatics problems, at the example of *k-mer* counting, has been benchmarked with tsxCount. Serialization is an important construct for any parallel application. However, many bioinformatics programs can not profit from serialization because the major scripting languages (R and python) only support the execution of one thread at a time. Providing an easy-to-use framework for parallelization could be a step towards supporting easy parallelization of R or python frameworks. Such a parallelization framework could profit from the presented benchmark, and may even choose the serialization technique dynamically during runtime, depending on the used hardware.

scRNA-seq and IMS data analysis (Chapter 4) In the discussion of the cell type prediction method cPred, it was observed that the prediction of the various differentiation stages of, for instance, monocytes is a challenge. While working with collaborators from the biomedical domain on still ongoing projects, it was noticed that several unique marker genes were missing from the databases. In fact, the databases had no *specifically unique* marker genes listed: marker genes, that if expressed and present, identify a cell type certainly. Providing a database of such cell type specific markers could resolve ambiguities, particularly for very similar cell types. Such a category of marker genes could be easily incorporated into the prediction model. It reflects a further category of cell type specific marker genes, with a specifically high weight. Additionally, such an incorporation of *specifically unique* marker genes adheres to the principles of the applied down-weighting scheme.

Regarding the analysis of IMS data, `pIMZ` was presented as an integrative framework. Unfortunately, only few public IMS datasets exist, even though large consortia like HuBMAP promise to release many IMS datasets in due time. However, such datasets are required in order to benchmark the multiple normalization techniques properly. Currently, `pIMZ` makes use of two differential testing methods taken from the diffxpy-package [330], and nlEmpiRe[4]. Particularly the ideas of the nlEmpiRe package, which were already applied to proteomic data [7], suit IMS data very well. The current implementation has runtime problems with large numbers of replicates, which can go into the hundreds with IMS data. The same reason prevents the application of the package to scRNA-seq data, even though the general ideas are applicable to such kinds of data, too. Taking the ideas of nlEmpiRe, transforming them to be applicable to high-replicate experiments, such as IMS and scRNA-seq, would be

[2]https://conda.io
[3]https://chocolatey.org/
[4]Csaba, Gergely. Personal Communication. 2020.

highly rewarding for more sensitive DE analyses. Particularly IMS data analyses would benefit from the prospected increase in sensitivity of the nlEmpiRe[5] approach, because the noise signal in the spectra can be quite intensive. Having a sensitive method, which can maintain the desired false discovery rate, could yield more genuinely differentially regulated masses or genes, even with these noisy measurement techniques. Such an implementation could be also well applicable to the DE analysis of scRNA-seq data, as the problems in this area are very similar. The current trend in analysis software goes into even more integrated analysis environments. As such, integrating set based enrichment methods into the `pIMZ` framework could be beneficial for the user experience. Hence, a combination of the robust DE pipeline with `pIMZ` could be rewarding, allowing the identification and elimination of computationally introduced results by performing the analysis for several normalization techniques and differential methods at the same time.

TGS data analysis (Chapter 5) Currently, the poreSTAT framework and *sequ-into* focus on the application in genomics projects. However, with the possibility of directly sequencing RNA using the Oxford Nanopore MinION device, the framework could be extended to provide distinct analyses for transcriptomics. Particularly with long reads, the identification of gene-fusions or alternative splicing events becomes easy. But also epigenetic regulations, like m5C (5-methylcytosine), or the m6A (N6-methyladenine) modification on the RNA-level, can easily be detected using this sequencing strategy [335]. Providing quality reports for such events would increase the applicability of poreSTAT.

Robust Differential Expression Analysis (Chapter 6.1) The analysis of DE genes is the most common analysis of RNA-seq data. It is the starting point for many, more focused, analyses of expression level data. Understanding the results of the DE analysis is important for understanding the implications of this analysis on the subsequent analyses. For the robust DE analysis pipeline several additions can be sought of. The design decision of independent analysis scripts should be overthought for performance reasons. Even if this design is kept, using a workflow management system for orchestrating all required analyses seems plausible, as the current analysis script becomes unhandy at more than 2000 lines of code. While the poreSTAT analysis framework can use interactive javascript-enabled figures within the reports, these have not yet been deployed to the robust pipeline. Particularly scatter plots (e.g. replicate consistency or volcano plots) would profit, as this would ease the identification of serious outliers. In general, the focus of the framework should include further analyses which aim at accessing problematic findings. For example, method-specific, or input-specific DE genes should be made directly available to the user, so that it can be decided how severe the differences are.

Aorta3D (Chapter 6.2) Providing a resource for accessing large amounts of data in a complicated setting, like complex human diseases, is challenging. Integrating multiple different experimental techniques, of both, low- and high-throughput nature, is even more

[5]Csaba, Gergely. Personal Communication. 2020.

challenging, because inherently heterogeneous data must be handled. With Aorta3D, this problem is tackled by using descriptions in an unstructured json-data format. Another feature of clinical research is, that many human diseases are explored at the example of animal models, like mouse models for atherosclerosis. This project would benefit from further integration of both mouse and human data. It is not fully understood what the differences in human and mouse atherosclerosis are, and where the mouse model reaches its limits. Using spatial information, together with proteomic and single cell evidence, the comparison of human and mouse atherosclerosis can be conducted. Having the Aorta3D framework available for this task is beneficial, as this framework allows the side-by-side comparison of multi-modal, cross-species data.

The question of whether machines can think, a question of which we now know that it is about as relevant as the question of whether submarines can swim.

Edsger W. Dijkstra

8 Conclusion

In the early 2000s, high-throughput experiments for measuring expression profiles were applied routinely in the form of microarray experiments. The amount of generated data, which needed to be pre-processed and analysed, required efficient analysis methods in order to cope with the amount of data and down-stream analyses. Nowadays, the microarray technique is displaced by sequencing experiments, as public repositories of experimental data show. But it is replaced with the even more computationally demanding analysis of sequencing data (Figure 1.2). The general workflow for analysing sequencing data differs from microarray measurement techniques. These differences are mainly located in the (pre-processing) steps required to quantify experimental outcomes, such as gene expression. The down-stream steps in the analysis of microarray and sequencing data are comparable. Such steps may, however, profit from the increasing amount of gained knowledge in specific contexts, e.g. processes involved in specific diseases. This prior knowledge can be used in the form of context knowledge about the system of interest, which allows for specifically tailored methodological approaches, particularly suitable for down-stream analyses. Advances in the development of computational resources and experimental techniques offer new opportunities for methodological improvements. These improvements in available knowledge, in available data sources and computational resources motivate a new focus on integrative methods in complex human disease contexts.

Efficient, context-sensitive, robust, user-friendly and easily interpretable analyses are required to cope with the massive amount of sequencing data available. The adequacy of a method for this purpose can be measured regarding its computational efficiency, for instance, its runtime, and by the ability to consider already existing knowledge (of the observed system). The latter requires specifically tailored-methods, e.g. methods deriving a context for data, or making use of context-sensitive data. Any performed analysis should be robust regarding the input data, but it should also be robust regarding the

applied methods, ensuring that genuine results are reported. With the observed increase in sequencing data, and sequencing methods, which become routine assays in laboratories, many non-bioinformaticians are performing bioinformatics analyses. By providing user-friendly methods and easily interpretable results, bioinformatics methods can be applied by a broader audience, which finally supports the interdisciplinary environment of bioinformatics.

In many cases, knowledge on specific contexts, such as disease-contexts, is available. Ignoring this knowledge leads to an immense loss in information. Being able to exploit this information for all reported findings makes it possible to trace the reasoning of results. Moreover, with this information, newly derived results can be set into context. In addition, context-sensitivity allows methods to focus on already known and thus relevant results, and better justified hypotheses. Context-sensitivity can be achieved by either performing a preselection, e.g. by using specific texts only, or by providing contextual meta-information for data, e.g. the context of the discussed miRNA-gene interactions.

By focusing on the user-friendliness of a method, its correct application can be supported. This also promotes that the user can make efficient use of the method and its results. Easily interpretable results make a method more valuable, because more users can apply the method, no relevant information is missed, while at the same time, the results are not over-interpreted by the users.

There exist many protocols to measure transcriptomic expression using sequencing technologies. With bulk RNA-seq the analysis pipeline differs between mRNA and miRNA expression: for miRNA expression specific genome annotations are used, but most importantly, the quantification strategy should follow more strict assumptions. Likewise, for long non-coding RNA sequencing, where not much is known about long non-coding RNA isoforms, the quantification strategy should reflect this. But even for regular mRNA transcriptomics, there is a major difference in the processing of bulk and scRNA-seq data. This highlights the requirement for new methods specialized for specific measurement techniques. But this also applies to computer scientific methods. For example, in the domain of text mining many building blocks, like the dependency graph prediction or entity classification, improved a lot, enabling new opportunities for consecutive methods.

As already pointed out, modern bioinformatics is driven by the many advances in gained knowledge, experimental techniques and computational methods. These advances motivate the need for new methods on a data extraction, (pre-)processing and down-stream analysis level, and, moreover, offer new opportunities for an integrated data analysis in the context of complex human diseases. This thesis addresses several of the above-mentioned opportunities. The contributions of this work focus on computational resources such as accessibility, usability and interoperability (Chapter 3), or on improving the analysis of (high-throughput) data on a methodological level, also by making new experimental techniques available (Chapters 2, 4, 5 and 6).

Chapter 3 describes methods for improving the accessibility and usability of bioinformatics software in general, and evaluates methods for interoperable and efficient parallel computation. bioGUI [145] (Chapter 3.1) enables users to perform bioinformatics analyses without any CL knowledge. Particularly for non-bioinformaticians this is a big step towards running bioinformatics analyses independently. It was found that bioGUI makes the use

of bioinformatics applications easier, compared to relying on a CLI. This finally increases the accessibility of bioinformatics software. With the tsxCount application (Chapter 3.2) several serialization methods were benchmarked at the example of *k-mer* counting. With only little performance differences between a highly specific serialization technique (TSX), and a broadly available one (OpenMP locks, OMP), interoperability considerations must be made when evaluating which method to choose. Such considerations, however, may have effects on the efficiency of the developed software. It is thus of high importance to have a clear understanding of the performance of the employed technology, but also of the targeted hardware and the problem at hand, in order to achieve an efficient implementation. The results from the tsxCount benchmark can be used for reference in real-life settings and thereby enable exploiting the underlying hardware in the best possible way.

With the results from bioGUI and tsxCount computational resources are made available and are efficiently exploited. The following chapters focus on a methodological level for providing new pre-processing methods for modern experimental techniques and data integrative methods for down-stream analysis.

In Chapter 2 text mining is employed to create a context-sensitive database of miRNA-gene interactions from public texts. For this purpose, several aspects regarding the use of text mining for biomedical knowledge discovery are addressed. With MORSED, methods for ontology-based research in structure extracted documents (Chapter 2.1) are implemented. Using these methods, ontologies are converted into synonym lists for the employed NER approach. Furthermore, structured text extraction from scientific literature in PDF format is achieved, together with the automatic assignment of therein contained paragraphs to their corresponding section. It was found that, depending on the topic of the synonyms and ontologies, different text sections, e.g. the methods section, contain more relevant information than other sections. For instance, a large fraction of experimental technique related named entities are only found in the methods section. In most cases, the abstract was not enough to cover all relevant named entities, which highlights the importance of using publicly available full texts, if possible. The applied PDF text extraction works well, and the developed section categorization works almost perfectly, if unrecoverable errors from the pdf text extraction are left out. Using the given NER approach, the programmatic extension of synonyms increases the amount of found named entities and hence helps to extract an as complete as possible context. Taken together, these methods build the foundation for (PDF-based) full text analysis using NER with synonyms derived from ontologies.

In a pilot study of a context-sensitive miRNA-gene interaction mining framework, atheMir [144] (Chapter 2.2), the rule-based extraction of miRNA-gene interactions is established. Going through all PubMed abstracts with such an approach to detect miRNA-gene interactions, creates a database of context-sensitive interactions complete and correct enough to suit as a base for writing a data-driven review on miRNA-gene interactions in atherosclerosis [144].

The atheMir framework was further improved to be robust and fast enough to cope with PMC full texts, while additionally improving over state-of-the-art miRNA-gene interaction detection methods. This is accomplished by the miRExplore framework (Chapter 2.3),

which includes methods for mining miRNA-gene interactions and storing these with context information in a database. The miRExplore framework provides additional means to assess when, and in which other contexts, specific miRNA-gene interactions were already observed, or whether a specific observation is genuinely new (Timeline feature). This sets new findings into the correct context. miRExplore connects with the presented DE gene expression pipeline `RoDE` (Chapter 6.1) with a greedy approach for identifying the most likely miRNA-regulators. This approach uses the identified DE genes, and predicts likely regulators by minimizing inconsistently regulated miRNA-gene pairs. Both these use-cases are not yet provided by existing frameworks. Due to the context-sensitive resource, already published interactions, e.g. from a broader context, can serve as an additional evidence for hypothesis generation. Taking DE results from RNA-seq experiments, and deriving a regulatory miRNA-gene network from the miRExplore database, enables an automatic, miRNA-centered interpretation of DE data. From this regulatory network, active miRNAs can be predicted. Moreover, applying this method to robust DE results, method-independent, and thus likely, miRNA-gene regulations are predicted. Finally, the integrative character of this framework and the presented analyses is emphasized.

In Chapters 4, 5 and 6, key challenges in modern bioinformatics analyses due to the increasing diversity of molecular possibilities and the resulting increase in experimental measurement methods are addressed. Methods for using data from three different sequencing methods (scRNA-seq, ONT MinION/TGS and bulk RNA-seq) are presented. All these methods require specific analysis steps, and have distinct properties, requiring distinct methods for data evaluation. This phenomenon is not restricted to sequencing techniques only, but holds true for proteomics and mass spectrometry techniques, like IMS, for spatially resolved measurements of the proteome.

In Chapter 4.2 the cPred method for cell type prediction is developed. Cell type prediction is an important task in any scRNA-seq analysis, because differences in the cell type composition of a sample can define different conditions. Hence, the identification of cell types in scRNA-seq datasets is of high importance. At the beginning of this thesis, no such methods were readily available, creating the need for a cell type prediction method. The cPred method is, in contrast to other state-of-the-art software, quite versatile: the input (expression) values can be derived from RNA-seq experiments and proteomic measurements alike. Only the presence and abundance of specific marker genes (or protein masses) is relevant. The cPred method relies on existing lists or databases of cell type-specific marker genes and thus is easily extendable. cPred could already be successfully applied within two COVID-19 related projects [228, 238]. In contrast to other methods, which rely on cell-level expression reference inputs, the use of a weighted-sum on averaged cluster-level expression values is more robust and less dependent on the measurement technique of the reference data. Contrary to other approaches, the weighted-sum approach only requires lists of marker genes, but no pre-processed scRNA-seq data with known cell type annotations as a reference. The cPred predictions are thus independent of the measurement technique.

In Chapter 4.4 the `pIMZ` framework for the analysis of MALDI-TOF IMS data is introduced. One of the main features of `pIMZ` is the focus on using python-based jupyter notebooks for data analysis, which is supported through respective examples and plotting

facilities. By using notebooks, analyses can be shared more easily, and reproducible analyses are promoted. Additionally, it enables a more informed and interactive analysis. With the `pIMZ` package the analysis of spatial mass spectrometry data becomes as usable and as streamlined as the analysis of scRNA-seq data. The analysis of single samples is possible, starting with typical pre-processing steps, like spectra extraction and normalization, and leading to down-stream analyses. `pIMZ` implements multiple clustering strategies and integrates several methods for marker mass identification. In addition, comparative analyses, integrative analyses of multiple samples at once, are supported. Moreover, data from large consortia, like the HuBMAP consortium, are made available through an integrated downloader. The `pIMZ` IMS analysis integrates cPred for cell type prediction, allowing the identification of cell types in specific regions of a tissue. This enables integrated visualizations of IMS data.

In Chapter 5 TGS techniques are made available by the poreSTAT framework and the *sequ-into* application. In the early 2010s sequencing was already applied routinely in biomedical research. One of the problems of the then used NGS methods is that only short, fixed-length fragments could be sequenced. With the advent of TGS techniques, whole DNA and RNA molecules can be sequenced, generating so-called long reads. These long reads offer new possibilities, particularly in genomics. However, they also require new pre-processing methods and also a different understanding in down-stream analyses. The methods presented in Chapters 5.1 and 5.2 are designed to work with these kinds of sequencing data. Both methods, poreSTAT and *sequ-into*, support the new input data format (FAST5), which may contain multiple reads at once, in contrast to other frameworks. The general workflow for analysing data starts with a quality control on the reads, before further down-stream tasks, like read-mapping or assembly, should be performed. At this stage, quality control checks can identify problems with the library preparation, which, if not detected, may endanger the sequencing and, thus, possibly render the acquired data unusable. If an alignment of the reads to a reference was performed, this alignment can subsequently be analysed in order to check whether the intended organism or correct molecules (e.g. mRNA vs. rRNA) are sequenced. All checks at the read- and alignment-level can be performed using the poreSTAT analysis framework presented in Chapter 5.1. An HTML-based report contains easily interpretable and shareable results, which can be explored in detail through interactive plots.

Chapter 5.2 introduces the *sequ-into* application [142]. *sequ-into* makes use of one of the advantages of the Oxford Nanopore MinION sequencer over regular NGS techniques: the analysis of reads directly during the sequencing process. Reads, which are processed live during sequencing, can be checked and analysed for certain properties during the sequencing run. This enables the fast identification of possible inconsistencies, contaminations or off-target sequences. With the *sequ-into* application this analysis is implemented using an incremental online analysis server. *sequ-into* combines the online analysis with an easy-to-use GUI. While *sequ-into* was originally designed for the analysis of prokaryotic samples, it works well for mammalian-sized genomes, even on regular laptop or desktop computers. In the discussed use-case, SarS-COV-2 sequences could be identified from a meta-genomic sample, including further bacterial genomes.

In Chapter 6.1, the robust DE pipeline RoDE for processing bulk RNA-seq data is presented. When several alignment methods, quantification strategies, differential testing methods and enrichment analyses create many combinations of applicable methods, it is almost impossible to keep track of the single results. However, in many cases researchers are only interested in genuine changes. But under any circumstances, the reporting of changes, which only originate from the use of a specific computational method, should be avoided. Using the presented robust DE pipeline RoDE, a DE analysis is performed with a specific focus on the question: which changes can be seen independently of the applied methods? Such *robust* changes, which most likely do not result from computational biases, are identified using the RoDE pipeline. It is also possible to investigate the opposite question: Are there changes which are only discovered by a specific way of processing the data? Regarding the discussed analysis using RoDE, it was possible to detect that certain disease-relevant results could only be discovered using one specific processing path. RoDE yields a method-robust result of the DE analysis. Its HTML report, which includes brief explanations of the performed analysis steps, supports the idea of the FAIR principles, with a focus on usability (Docker image) and interpretability (HTML report). This pipeline has been successfully applied in a complex human disease context [238].

In Chapter 6.2, the Aorta3D project provides an integrative 3D index of atherosclerosis-relevant data. Combining multi-omics data sources in one data model, and making this combined data accessible, is the key feature of Aorta3D. Given the many sequencing and analysis techniques, the need for a common index of relevant data is created. Through Aorta3D, users can access experimental data via several access vectors: via a 3D model of the disease progression, from spatial representations (e.g. images), or via reported DE genes. It supports filtering of experiments based on the analysed region, cell type and contained DE genes. In addition, the Aorta3D framework allows stacking of multiple experiments, such that these images are displayed on top of each other. This enables the selection of regions of interest, e.g. from IMS data, based on external imaging data, e.g. the antibody staining of specific cell types. Aorta3D serves as a 3D accessible index of atherosclerosis-relevant data. It provides a solid base for transferring knowledge obtained from model organisms to actual complex human diseases contexts.

This thesis contributes to current topics in bioinformatics research at each stage of a typical bioinformatics analysis workflow, during both pre-processing steps and down-stream analyses (Figure 1.1). The developed methods and frameworks are discussed at the example of complex human diseases, with an integrative view on biomedical data in mind. Using a rule-based text-mining approach, context-sensitive miRNA-gene interactions are extracted at high precision and recall. These can be used to check novel miRNA-gene interaction findings, or to predict likely miRNA regulators for gene expression analysis results. In the context of differential gene expression, the RoDE pipeline for robust DE analysis, also integrating the miRExplore miRNA-gene interaction database, has been developed and successfully applied. The cell type prediction method cPred, which has been applied to scRNA-seq data and IMS results, enables a multi-modal analysis of disease-related datasets. Using the cell type predictions researchers can evaluate which cells behave different in

a disease context, or how the cell composition changes in samples (of different disease stages). The `pIMZ` framework allows for the (differential) analysis of IMS data, which can be combined with scRNA-seq data using the 3D- and spatial index Aorta3D. Aorta3D organizes experiments not on a single entity (e.g. genes), but also inter-connects them through identified cell types, even on a spatial level. With the poreSTAT framework and *sequ-into* a further sequencing technology can be exploited: TGS, which is able to produce arbitrarily long-reads, thus requiring different analyses than NGS data. These topics are consolidated in the topics of usability and interoperability, which was a particular focus of the bioGUI and tsxCount projects.

All the discussed topics are applied to problems relating to complex human diseases, like the chronic inflammatory disease atherosclerosis, or the pandemic SARS-CoV-2 virus infection.

The nice thing about standards is that you have so many to choose from.

Andrew S. Tanenbaum

A Appendix

A.1 Common Bioinformatics Data Formats

Here commonly used data types are introduced. A full coverage of further common data types is given by Griffin et al. [109].

Sequencing Data: FASTA/FASTQ Both, the *FASTA* and the *FASTQ* file format are intended to store sequences. While the *FASTA* format is stores one sequence per entry, the *FASTQ* format allows to store an additional per-base annotation for each sequence.

The FASTA format (Figure A.1) starts an entry with the > sign immediately followed by the entry name until the first space. After the entry, a multitude of additional annotations may follow. However, there is no standardization of these additional annotations. The sequence itself is often formatted such that each line is at most 80 characters wide. However, the FASTA format also allows to store the whole sequence in one line.

The FASTQ format (Figure A.2) starts an entry with the @ sign immediately followed by the entry name until the first space. Any further information are annotations which might be ignored when reading the FASTQ file. One FASTQ entry always consists of 4 lines. The first line starts with the @ sign, followed by the sequence identifier or name. The second line contains the actual sequence. The third line only consists of the + as delimiter of the sequence and annotation line. The fourth line must have the same length as the second line. The x-th character of the fourth line is the quality score for the x-th base of the second line. The quality score is a machine estimate of how good or reliable the sequencing information at that position is.

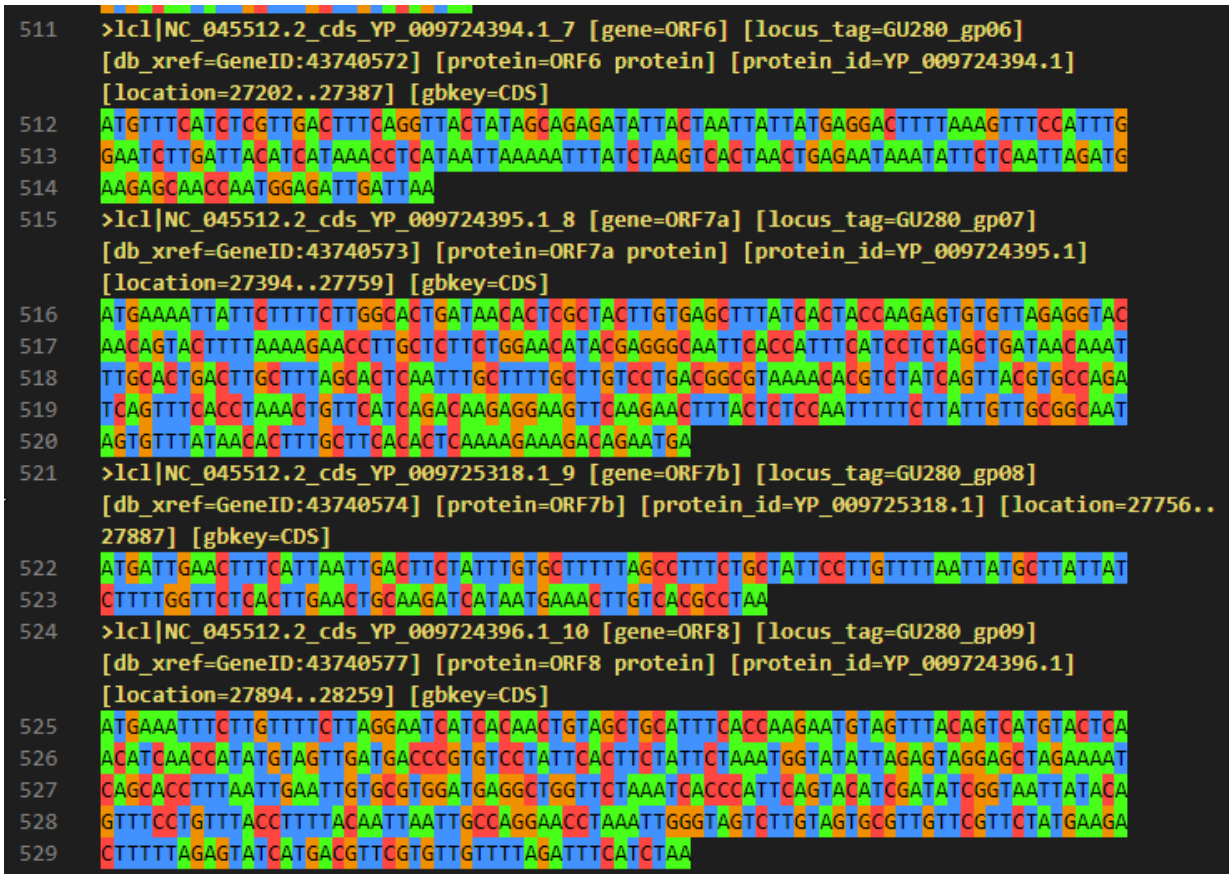


Figure A.1: **The FASTA format** is meant to store sequence information. For each entry it contains one sequence.

Oxford Nanopore MinION FAST5 format The FAST5 format (Figure A.3) stores sequences from Oxford Nanopore sequencing devices. One FAST5 file can contain one or more sequences, each with an own entry in the top-level hierarchy. All reads have a defined structure such that it can be derived what kind of read it is (1D or 2D read technology) and whether it is basecalled or not. Additionally, several meta information are available. In contrast to FASTQ files, FAST5 reads also contain the measured raw signals, which can be interpreted to re-basecall the read at a later time after further progress in the basecalling process was made. The advantage of the FAST5 format over FASTQ format is that more information can be retrieved for a specific read in a standardized way. FAST5 reads can not be accessed using a regular text editor, but on every operating system a free viewer is available. FAST5 files are specifically formatted HDF5 files, which is a standardized container format, for which every programming language also provides libraries for file access. While the Oxford Nanopore sequencing software MinKNOW used to write one read per file, nowadays the default behaviour is to collect a multitude of reads per FAST5 file. This has the advantage that no more several hundreds of thousands files are created, which stress any kind of (journaling) file system. A complete list of useful applications for ONT

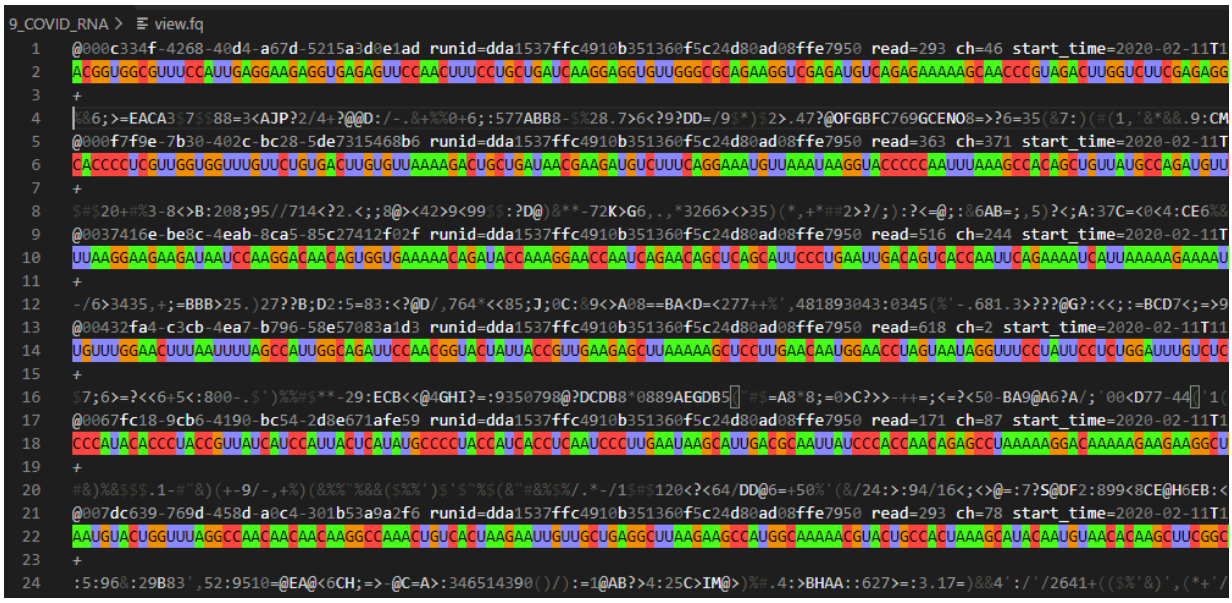


Figure A.2: **The FASTQ format** stores sequence information. For each entry it contains one sequence and an additional per-base annotation.

sequencing has recently been published [196].

Sequencing Data: genome feature format The *gff* or *gtf* file format (Figure A.4) is meant to store information about where genomic features are located on a genome. Apart from human-readable information at the beginning of the file (denoted with `#`), both gff and gtf file format are tab-based. The difference between gff and gtf basically is the formatting of the annotations, and how relations between features in different lines are represented. In general, both formats define one feature per line. For each feature, its source, location (sequence, type, start, stop, strand) and annotation (e.g. ID, notes, etc.) are given.

Sequencing Data: bam/sam The SAM and BAM file format are both alignment formats. These files describe the alignment of sequences (e.g. reads from sequencing) onto other sequences (the reference, e.g. genome). Therefore, for each entry the sequence it aligns to, the starting position and various other information are stored. While the SAM file format is uncompressed and can thus be read in any text viewer, the BAM format is compressed. Further information regarding this file format is already published [180].

For each alignment, a sequence of CIGAR (Concise Idiosyncratic Gapped Alignment Report) codes is available. A sequence `76H130M` tells that there are 76 bases with CIGAR code `H` (hard clipping), which are followed by 130 bases with CIGAR code `M`, which stands for *matches*. It should be noted, that these matching bases must not be exact matches (CIGAR code `=`), but mismatches can also be contained in a matching region. The full list of available CIGAR codes is available in [180].

IMS Data: imzML The imzML format (Figure A.5) is an XML-based format for storing spatial resolution mass spectrometry data. Within the imzML file there is one `spectrum` entry per stored spectrum. This entry stores the coordinates of the measurement, together with some meta information (like maximum, minimum intensity), as well as an offset in the binary storage file where all intensities of the spectrum are stored. This binary storage file accompanies every imzML file and usually has an `ibd` file ending.

Text mining: JATS-format The JATS-format[1] (Figure A.6) is used to store PMC full texts and PubMed abstracts in a machine-readable format. It is a specifically designed XML-based format. For each article, a `front`- and a `body`-group exist. The `front`-part contains all relevant meta information, such as PubMed/PMC-ID, journal, authors and affiliations. The `body`-part contains the respective text, partially structured in case of PMC full texts. Using specific XPaths and queries, as well as an XML-Tree-Parser, these JATS-formatted texts can be extracted and brought into a format which can be used for text mining.

Text mining: sentence, synonyms, ontologies The text mining applications in this work rely massively on the file formats already present in the working group of Prof. Zimmer. Since it was a prerequisite to use syngrep, the NER text mining tool previously developed in the group, all other work had to be brought in-line with data formats understood by syngrep. The synonym files and text input files (sentences) thus had to be formatted such that they are compatible.

For syngrep the sentence file (Figure A.7) is a tab-separated file format where the first column contains the sentence identifier, and the second column the text to search. Even though this information is not used in the way syngrep is called throughout this work, the sentence identifier also follows a specific semantic: separated by dots, the first entry is the document ID, the second entry defines the section and the last entry the sentence within this paragraph. The suffix-ID `.1.1` hence defines the title of a document. Further sections are the abstract (`.2`), the body (`.3`) as well as references and, where available, MeSH terms.

Frequently ontologies, like Gene Ontology, serve as input for the creation of synonyms. These ontologies and synonyms are the named entities that are searched in the literature-extracted sentences in order to derive the context of the found relations. An ontology file (Figure A.8) is a file which consists of multiple `TERM` entries. Each entry has an ID, which is used by other terms as reference. A human-readable name describes the term. A term may have a namespace, which provides information about which domain the term is part of. The definition defines the actual term, but often is quite lengthy. Hence, it can not be used as a synonym (or it should not). One of the key features of an ontology is that it forms a directed acyclic graph (DAG). Hence, each term defines also its parents using the `is_a` relation. Further keywords like the `intersection_of` or `relationship` define further relations, but are seldom relevant for NER. An ontology term may also propose

[1] https://www.niso.org/publications/z3996-2019-jats

multiple additional synonyms using the `synonym` keyword. Modifiers at the end suggest whether this is an exact synonym, or a broad one.

The synonym file format (Figure A.9) is a custom format used by the syngrep NER application [114]. One line represents one synonym. The primary synonym name is delimited by the `:` character, which must not be contained in the word. Any following synonyms are delimited by the pipe | character.

The syngrep index format (Figure A.10) contains all entities found by syngrep. For each found entity the following information are recorded: the sentence ID, which synonym was found (synonym file and line, and text), at which position and how long the match was. Additionally, the word which matched in the text is written out as well as an information whether this is a perfect hit or not.

Unstructured Data Format: JSON JSON is the JavaScript Object Notation format, which is published as part of the javascript programming language standard[2]. JSON itself is built on two structures: a collection of key-value-pairs and ordered lists of values. As such JSON is ideal for working with unstructured data, where some keys might be missing or where the value-data for a specific key may change at run-time. For a productive use it also becomes handy that the python programming language supports json directly by its *dictionary* object. Contrary to tables, json has the advantage that it can store arbitrary data, and that missing or additional keys pose no problem.

[2]https://www.ecma-international.org/ecma-262/9.0/

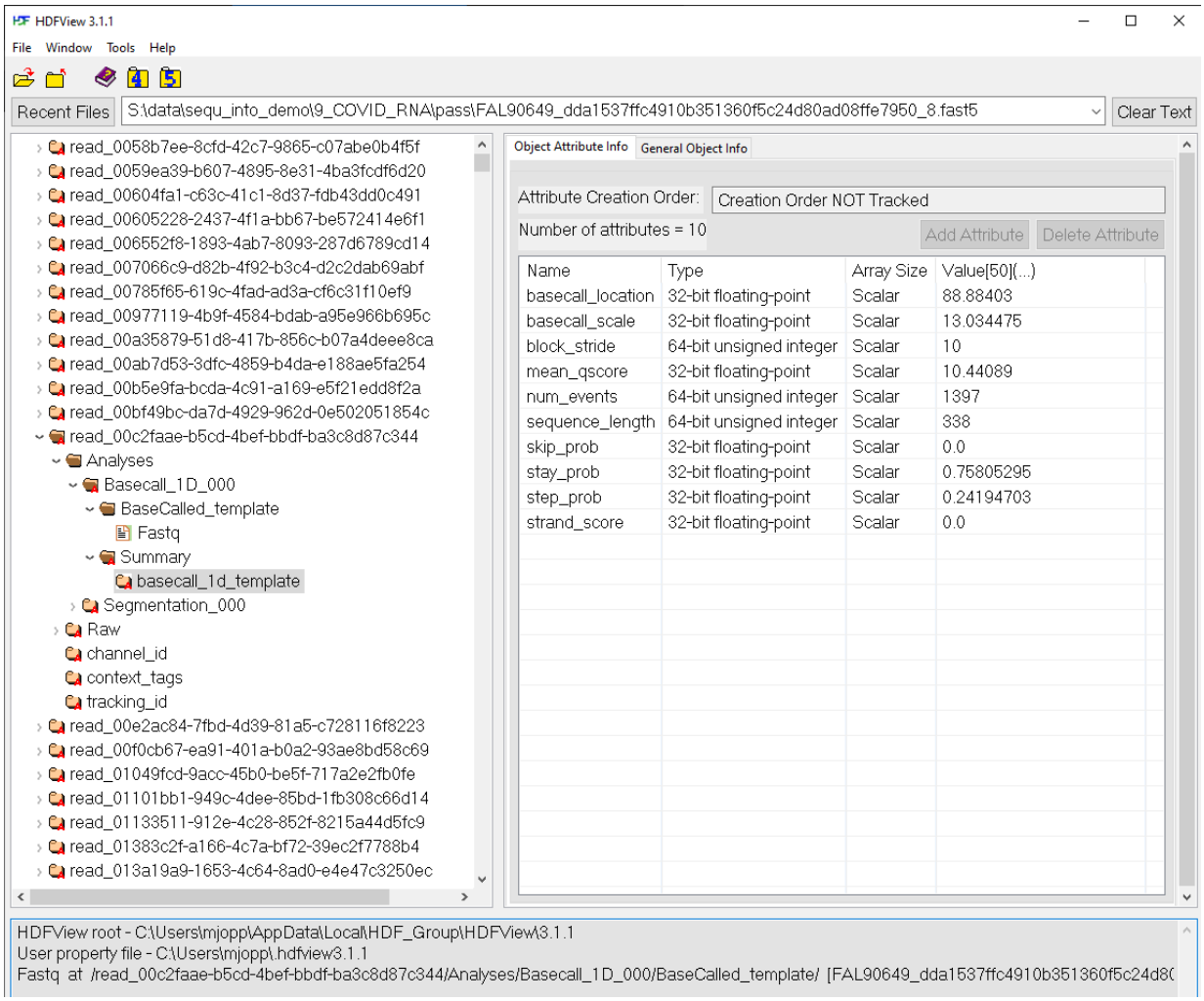


Figure A.3: **The FAST5 format** stores sequences from Oxford Nanopore sequencing devices. One FAST5 file can contain one or more sequences, each with an own entry in the top-level hierarchy. Each read has a defined structure such that it can be derived what kind of read it is (1D or 2D read technology) and whether it is basecalled or not. Additionally, several metadata are available. Only FAST5 reads also contain the measured raw signal, which can be interpreted to re-basecall the read at a later time after further progress in the basecalling process was made.

```
##gff-version 3
#!gff-spec-version 1.21
#!processor NCBI annotwriter
#!genome-build ASM985889v3
#!genome-build-accession NCBI_Assembly:GCF_009858895.2
##sequence-region NC_045512.2 1 29903
##species https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=2697049
NC_045512.2 RefSeq  region  1   29903   .   +   .   ID=NC_045512.2:1..29903;Dbxref=taxon:2697049;collection-date=Dec-2019;country=China;gb-acronym=SARS-CoV
NC_045512.2 RefSeq  five_prime_UTR  1   265 .   +   .   ID=id-NC_045512.2:1..265;gbkey=5'UTR
NC_045512.2 RefSeq  gene    266 21555   .   +   .   ID=gene-GU280_gp01;Dbxref=GeneID:43740578;Name=ORF1ab;gbkey=Gene;gene=ORF1ab;gene_biotype=protein_codin
NC_045512.2 RefSeq  CDS 266 13468   .   +   0   ID=cds-YP_009724389.1;Parent=gene-GU280_gp01;Dbxref=Genbank:YP_009724389.1,GeneID:43740578;Name=YP_00972438
NC_045512.2 RefSeq  CDS 13468   21555   .   +   0   ID=cds-YP_009724389.1;Parent=gene-GU280_gp01;Dbxref=Genbank:YP_009724389.1,GeneID:43740578;Name=YP_0097
NC_045512.2 RefSeq  mature_protein_region_of_CDS    266 805 .   +   .   ID=id-YP_009724389.1:1..180;Note=nsp1%3B produced by both pp1a and pp1ab;Parent=cds
NC_045512.2 RefSeq  mature_protein_region_of_CDS    806 2719    .   +   .   ID=id-YP_009724389.1:181..818;Note=produced by both pp1a and pp1ab;Parent=cds-Y
NC_045512.2 RefSeq  mature_protein_region_of_CDS    2720    8554    .   +   .   ID=id-YP_009724389.1:819..2763;Note=former nsp1%3B conserved domains are: N
NC_045512.2 RefSeq  mature_protein_region_of_CDS    8555    10054   .   +   .   ID=id-YP_009724389.1:2764..3263;Note=nsp4B_TM%3B contains transmembrane dom
NC_045512.2 RefSeq  mature_protein_region_of_CDS    10055   10972   .   +   .   ID=id-YP_009724389.1:3264..3569;Note=nsp5A_3CLpro and nsp5B_3CLpro%3B main
NC_045512.2 RefSeq  mature_protein_region_of_CDS    10973   11842   .   +   .   ID=id-YP_009724389.1:3570..3859;Note=nsp6_TM%3B putative transmembrane doma
NC_045512.2 RefSeq  mature_protein_region_of_CDS    11843   12091   .   +   .   ID=id-YP_009724389.1:3860..3942;Note=produced by both pp1a and pp1ab;Parent
NC_045512.2 RefSeq  mature_protein_region_of_CDS    12092   12685   .   +   .   ID=id-YP_009724389.1:3943..4140;Note=produced by both pp1a and pp1ab;Parent
NC_045512.2 RefSeq  mature_protein_region_of_CDS    12686   13024   .   +   .   ID=id-YP_009724389.1:4141..4253;Note=ssRNA-binding protein%3B produced by b
```

Figure A.4: **The GFF/GTF format** is meant to store genomic annotations. One feature is defined per line. For each feature it is annotated on which sequence it is located, where it was defined, what kind of feature it is and where it is located on the sequence. The last column defines the annotation of the feature as well as possible children or parents.

```
<spectrum defaultArrayLength="0" id="spectrum=11" index="11" spotID="11">
<referenceableParamGroupRef ref="spectrum"/>
<scanList count="1">
    <cvParam accession="MS:1000795" cvRef="MS" name="no combination"/>
    <scan instrumentConfigurationRef="instrumentConfiguration0">
    <userParam name="3DPositionX" value="72.829521"/>
    <userParam name="3DPositionY" value="16056.733398"/>
    <userParam name="3DPositionZ" value="0.000000"/>
    <cvParam accession="IMS:1000050" cvRef="IMS" name="position x" value="2"/>
    <cvParam accession="IMS:1000051" cvRef="IMS" name="position y" value="22"/>
    </scan>
</scanList>
<binaryDataArrayList count="2">
    <binaryDataArray encodedLength="0">
    <referenceableParamGroupRef ref="mzArray"/>
    <cvParam accession="IMS:1000103" cvRef="IMS" name="external array length" value="37100"/>
    <cvParam accession="IMS:1000104" cvRef="IMS" name="external encoded length" value="296800"/>
    <cvParam accession="IMS:1000102" cvRef="IMS" name="external offset" value="16"/>
    </binaryDataArray>
    <binaryDataArray encodedLength="0">
    <referenceableParamGroupRef ref="intensities"/>
    <cvParam accession="IMS:1000103" cvRef="IMS" name="external array length" value="37100"/>
    <cvParam accession="IMS:1000104" cvRef="IMS" name="external encoded length" value="148400"/>
    <cvParam accession="IMS:1000102" cvRef="IMS" name="external offset" value="1929216"/>
    </binaryDataArray>
</binaryDataArrayList>
<cvParam accession="MS:1000528" cvRef="MS" name="lowest observed m/z" unitAccession="MS:1000040" unitCvRef="MS" unitName="m/z" value="499.965973"/>
<cvParam accession="MS:1000527" cvRef="MS" name="highest observed m/z" unitAccession="MS:1000040" unitCvRef="MS" unitName="m/z" value="2000.070190"/>
<cvParam accession="MS:1000285" cvRef="MS" name="total ion current" unitAccession="MS:1000131" unitCvRef="MS" unitName="number of detector counts" value="11361.125977"/>
</spectrum>
```

Figure A.5: **The imzML format** is XML-based format for storing spatial resolution mass spectrometry data.

Figure A.6: **The JATS format** is XML-based format for storing full text information together with several meta information.

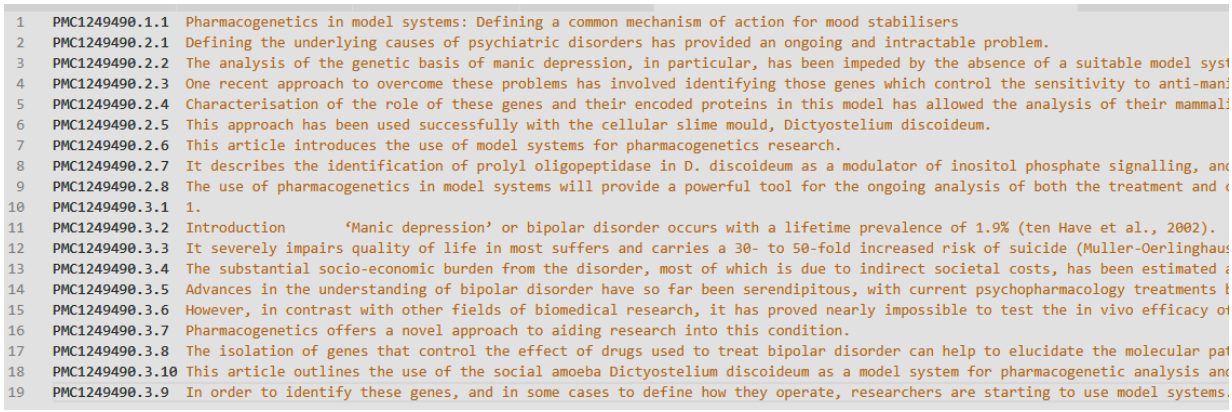


Figure A.7: **The syngrep sentence format** is a tab separated file. The first column contains the sentence ID, the second column the actual sentence.

```
[Term]
id: GO:0001796
name: regulation of type IIa hypersensitivity
namespace: biological_process
def: "Any process that modulates the frequency, rate, or extent of type IIa hypersensitivity, a type of
inflammatory response." [GOC:add, ISBN:0781735149]
is_a: GO:0002892 ! regulation of type II hypersensitivity
intersection_of: GO:0008150 ! biological_process
intersection_of: regulates GO:0001794 ! type IIa hypersensitivity
relationship: regulates GO:0001794 ! type IIa hypersensitivity

[Term]
id: GO:0001797
name: negative regulation of type IIa hypersensitivity
namespace: biological_process
def: "Any process that stops, prevents, or reduces the rate of type IIa hypersensitivity, a type of inflammatory
response." [GOC:add, ISBN:0781735149]
synonym: "down regulation of type IIa hypersensitivity" EXACT []
synonym: "down-regulation of type IIa hypersensitivity" EXACT []
synonym: "downregulation of type IIa hypersensitivity" EXACT []
synonym: "inhibition of type IIa hypersensitivity" NARROW []
is_a: GO:0001796 ! regulation of type IIa hypersensitivity
is_a: GO:0002893 ! negative regulation of type II hypersensitivity
intersection_of: GO:0008150 ! biological_process
intersection_of: negatively_regulates GO:0001794 ! type IIa hypersensitivity
relationship: negatively_regulates GO:0001794 ! type IIa hypersensitivity
```

Figure A.8: **The ontology obo format** stores an ontology in text format using defined entries.

```
4343  CCL16:CCL16|HGNC:10614|C-C motif chemokine ligand 16|SCYA16|small inducible cytokine subfamily A (Cys-Cys), member 16|
4344  CCL17:CCL17|HGNC:10615|C-C motif chemokine ligand 17|SCYA17|small inducible cytokine subfamily A (Cys-Cys), member 17|
4345  CCL18:CCL18|HGNC:10616|C-C motif chemokine ligand 18|SCYA18|small inducible cytokine subfamily A (Cys-Cys), member 18,
4346  CCL19:CCL19|HGNC:10617|C-C motif chemokine ligand 19|SCYA19|small inducible cytokine subfamily A (Cys-Cys), member 19|
4347  CCL20:CCL20|HGNC:10619|C-C motif chemokine ligand 20|SCYA20|small inducible cytokine subfamily A (Cys-Cys), member 20|
4348  CCL21:CCL21|HGNC:10620|C-C motif chemokine ligand 21|SCYA21|chemokine (C-C motif) ligand 21|small inducible cytokine s
4349  CCL22:CCL22|HGNC:10621|C-C motif chemokine ligand 22|SCYA22|small inducible cytokine subfamily A (Cys-Cys), member 22|
4350  CCL23:CCL23|HGNC:10622|C-C motif chemokine ligand 23|SCYA23|small inducible cytokine subfamily A (Cys-Cys), member 23|
4351  CCL24:CCL24|HGNC:10623|C-C motif chemokine ligand 24|SCYA24|small inducible cytokine subfamily A (Cys-Cys), member 24|
4352  CCL25:CCL25|HGNC:10624|C-C motif chemokine ligand 25|SCYA25|chemokine (C-C motif) ligand 25|small inducible cytokine s
4353  CCL26:CCL26|HGNC:10625|C-C motif chemokine ligand 26|SCYA26|small inducible cytokine subfamily A (Cys-Cys), member 26|
4354  CCL27:CCL27|HGNC:10626|C-C motif chemokine ligand 27|SCYA27|small inducible cytokine subfamily A (Cys-Cys), member 27|
```

Figure A.9: **The synonym file format** is a custom format used by the syngrep NER application [114]. One line represents one synonym. The primary synonym name is delimited by the **:** character (which must not be contained in any word). Any following synonyms are delimited by the pipe | character.

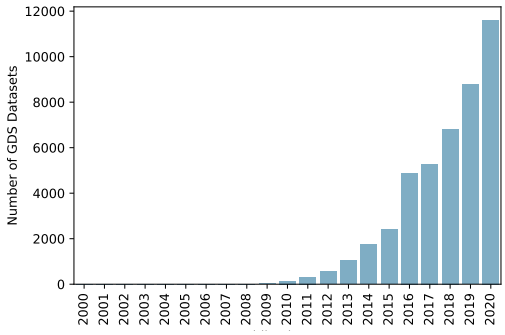
```
2   PMC2574629.1.1  0:9 miR-19  0   6   miR-19  true          0
3   PMC2574629.1.1  0:32    miR-101 8   7   miR-101 true          0
4   PMC2574629.1.1  0:44    miR-130 21  7   miR-130 true          0
5   PMC2574629.2.3  0:9 miR-19  19  6   miR-19  true          0
6   PMC2574629.2.3  0:32    miR-101 27  7   miR-101 true          0
7   PMC2574629.2.3  0:44    miR-130 40  7   miR-130 true          0
8   PMC2574629.3.18 0:9 miR-19a 0   7   miR-19a true          0
9   PMC2574629.3.18 0:32    miR-101 9   7   miR-101 true          0
10  PMC2574629.3.18 0:44    miR-130a    22  8   miR-130a    true          0
11  PMC2574629.3.21 0:9 miR-19a 172 7   miR-19a true          0
```

Figure A.10: **The syngrep index format** contains all entities found by syngrep. For each found entity the sentence ID is recorded, which synonym was found (by synonym file, line, and text), at which position and how long the match was. Additionally, the text word is written out as well as an information whether this is a perfect hit or not.

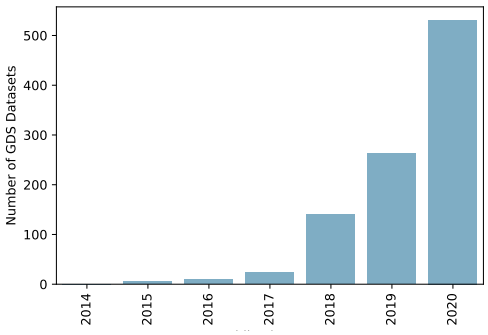
A.2 Chapter 1

For extracting the number of datasets in GEO the biopython [61] Entrez module was used with the database argument set to `gds`. The queries were performed per `year`:

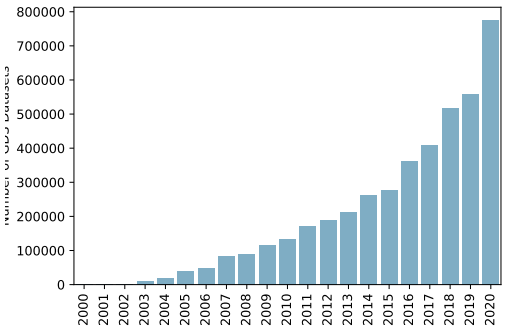
- Total number of experiments `{year}[Publication Date]`
- Total number of array experiments (expression profiling) `{year}[Publication Date] AND "expression profiling by array"[DataSet Type]`
- Total number of sequencing experiments (expression profiling) `{year}[Publication Date] AND "expression profiling by high throughput sequencing"[DataSet Type]`
- Total number of scRNA-seq experiments (expression profiling) `{year}[Publication Date] AND scRNA-seq AND "expression profiling by high throughput sequencing"[DataSet Type]`



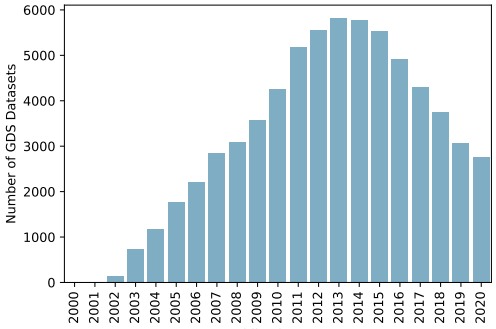
(a) Number of high throughput sequencing experiments in GEO per year



(b) Number of scRNA-seq experiments in GEO per year



(c) Number of experiments in GEO per year



(d) Number of array experiments in GEO per year

Figure A.11: **Number of expression profiling experiments (GEO)** (a) The number of sequencing experiments for expression profiling is continually increasing. (b) Each year rising numbers of scRNA-seq experiments are deposited in GEO. (c) Each year increasing numbers of experiments are deposited in the GEO. (d) Until 2013, rising numbers array experiments for expression profiling were deposited in GEO. Now, the number of yearly array experiments decreases.

A.3 Chapter 2

A.3.1 MORSED

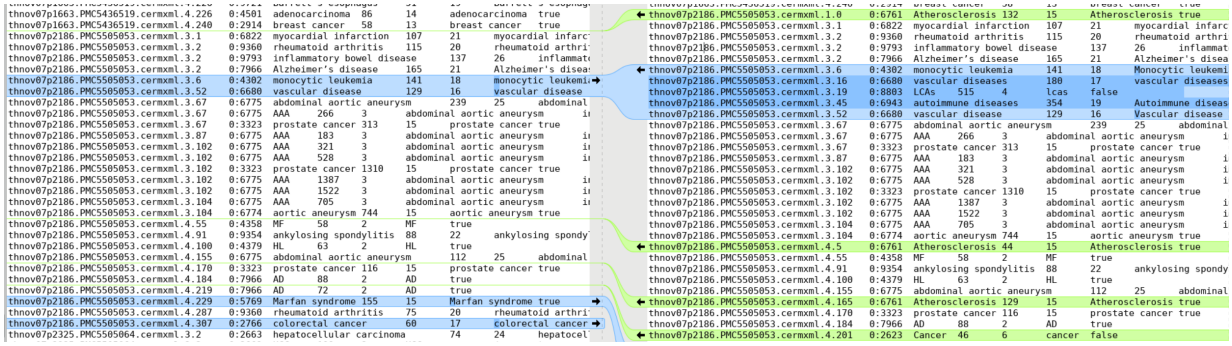


Figure A.12: **Diff on the text mining results between original (left) and inflated (right) synonyms**. It can be seen that the inflated version detects previously unknown synonyms, such as *vascular disease* or *atherosclerosis*.

MORSED provides an app and a website to access the presented methods. The MORSED methods use a PDF extraction tool which performs a structure-aware extraction of text from a pdf (Figure A.13). Thus, in addition to the actual text, also the section head-lines are available. Using this, occurrences of the targeted named entities can be filtered by section, allowing a more fine-grained search.

In order to provide the developed methods to a broad public, a cross-platform application was developed. The advantage of this application is that it works for any kind of organized keywords. The use-case focuses on terms on scientific evidence within the biological research domain. But the application could also be used on animal welfare research topics.

Using Electron as a starting point for the application allows a deployment on any Desktop computer (Windows, macOS, Linux). The application further relies on the Angular web technology.

Since the application itself does not perform the text-mining directly, a server framework is used to extract the text from the input PDF, NER is performed, and found synonyms and their location are returned to the application. This server also has a basic front-end to perform the above tasks on a specified set of contexts.

The ontology used in the text mining approach is created or provided by the user. The user can remove or add nodes in the ontology (Figure A.14). For each node, specific synonyms can be added. By default, the corresponding ontology term name is used as synonym. In contrast to a normal ontology, MORSED is restricted to only handle tree-like ontologies, which resemble a regular *mind map* — a concept most users would understand more easily than a complex ontology.

The user supplied ontology is sent to the web service for search of hits in specific pdf files. For this purpose the ontology is translated into a synonym file which contains for all ontology nodes the entities to look for.

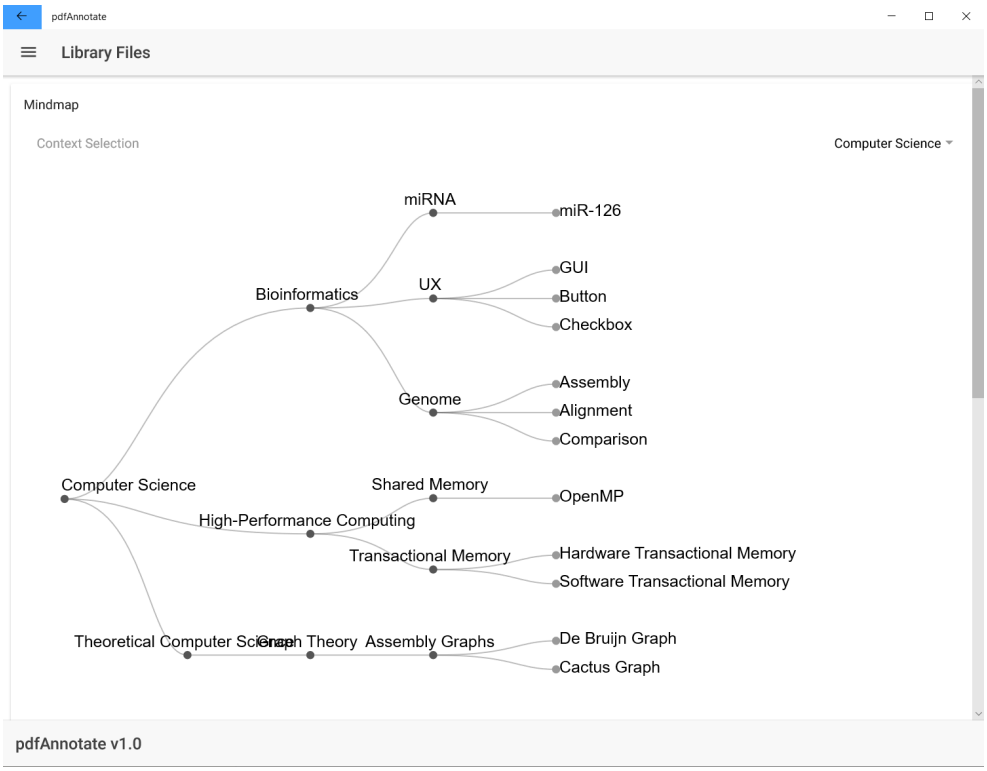


Figure A.13: **Screenshot of ontology selection in the MORSED app.** In the ontology view the user can select a root node which he/she wants to explore. This node is used for the initial filter and thus documents.

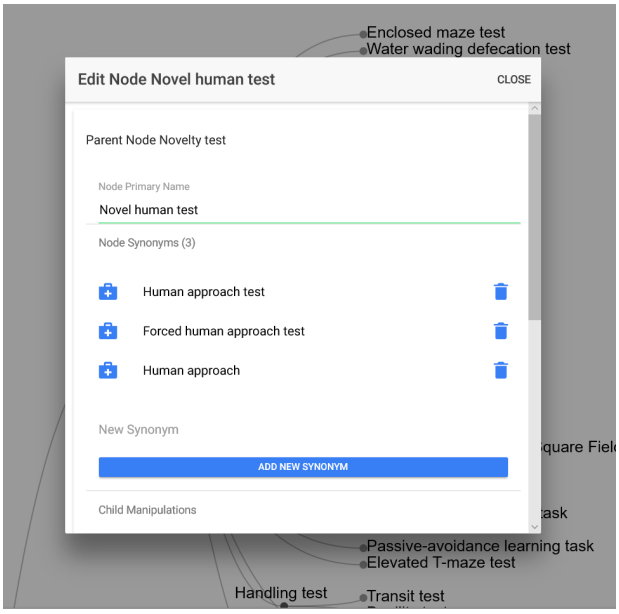


Figure A.14: **Screenshot of MORSED ontology creation.** Here the node name can be changed, as well as the registered synonyms. Further down in the menu, the (child) nodes can be manipulated (added or removed).

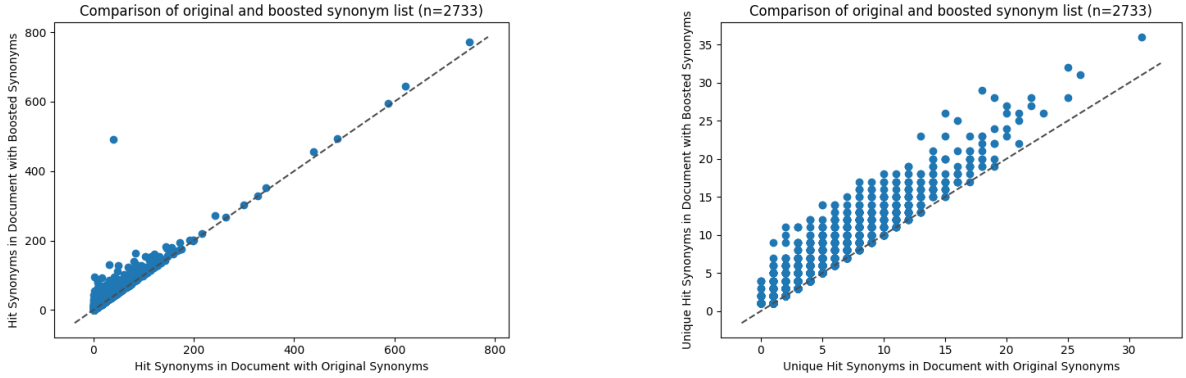
It is important to be able to filter hits efficiently. Especially in the context of the Evidence and Conclusion Ontology (ECO) or the Measurement Method Ontology (MMO) [280], the focus is on what to measure and how.

MORSED allows multiple filtering options. First, it can search for any synonym of a given node only, or also include hits of children. Additionally, not only single nodes may be searched for, but also any node containing a specific keyword as sub-string.

Several filters can be combined into a filter group. For each filter group, one filter has to evaluate true on a certain hit, in order to accept this hit (disjunction). If there exist multiple filter groups, each of the groups must evaluate true in order to accept the hit (conjunction).

For all filter groups, hits can be either searched in all identified sections, or, since the PDF is extracted aware of its structure, in specific sections.

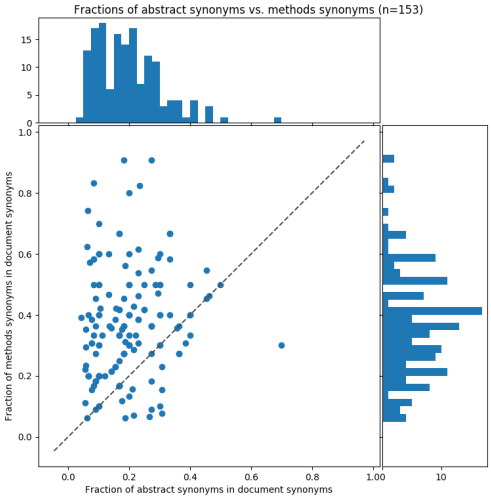
Additional Analyses



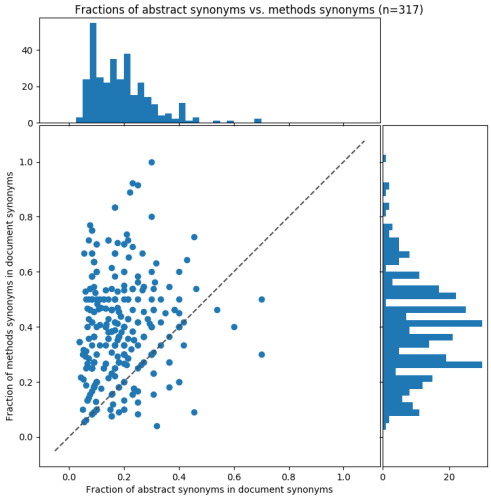
(a) Number of NER hits per document with the original ATOL synonyms and the inflated variant.

(b) Number of unique NER hits per document with the original ATOL synonyms and the inflated variant.

Figure A.15: **Comparison of original and inflated synonyms for ATOL** on the test corpus. While the amount of actual hits is increased by 150%, the amount of uniquely hit synonyms can be inflated by about 50%.

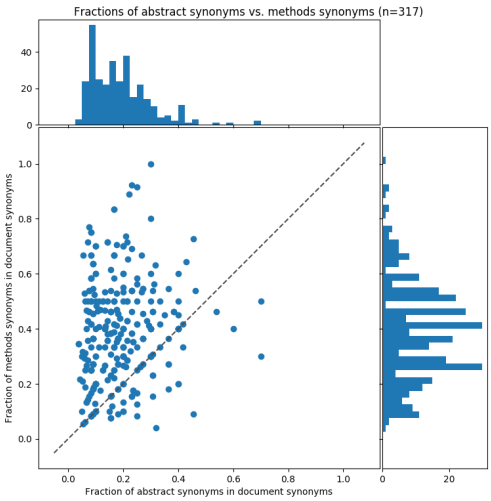


(a) Abstract versus methods synonyms for the ATOL context.

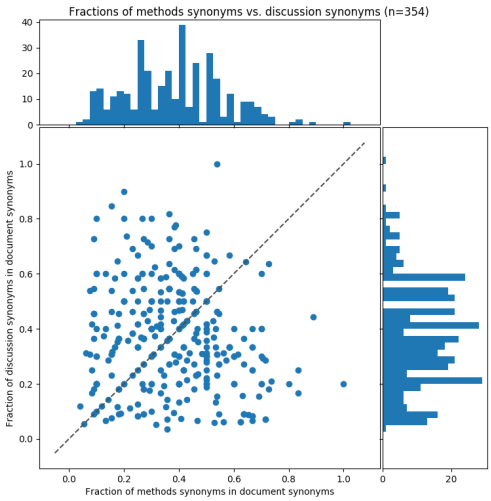


(b) Abstract versus methods synonyms for the inflated ATOL context.

Figure A.16: **Comparison of abstract and methods hits (ATOL)** In (a) the original ATOL context has been evaluated on the allxml dataset. In most documents, the methods synonyms make up 30% and more of all document synonyms. Using the inflated context (b), this effect can be seen even stronger. For most documents the fraction of abstract synonyms is less than 20%. Using the inflated context (b), about twice as many documents have more than 10 synonyms in both sections and hence are included in this overview.

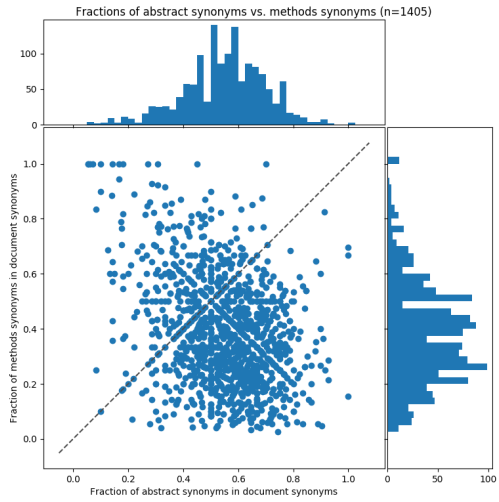


(a) Abstract versus methods synonyms for the inflated ATOL context. It can be seen that more synonyms are found in the methods part of a document.

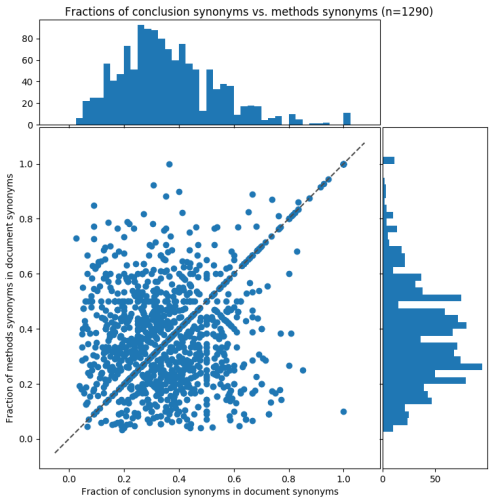


(b) Conclusion versus methods synonyms for the inflated ATOL context. The discussion rarely covers 30% or more of all document synonyms. More synonyms are found in the methods part of a document. The synonyms found in the conclusion rarely make up 50% of the document synonyms.

Figure A.17: **Comparison of abstract and conclusion hits (ATOL)** Comparison of the abstract (a) and conclusion (b) synonyms against the methods synonyms.

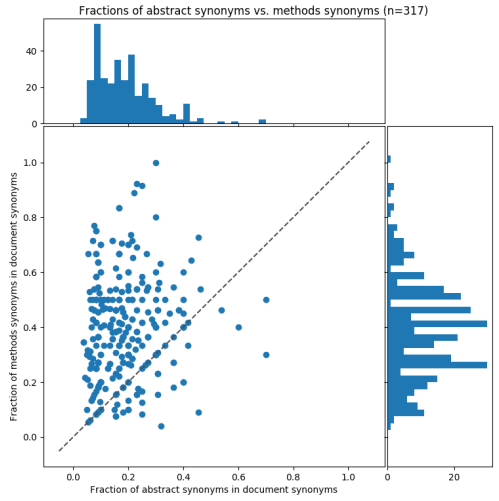


(a) Abstract versus methods synonyms for the inflated GO context. More synonyms are found in the methods part of a document.

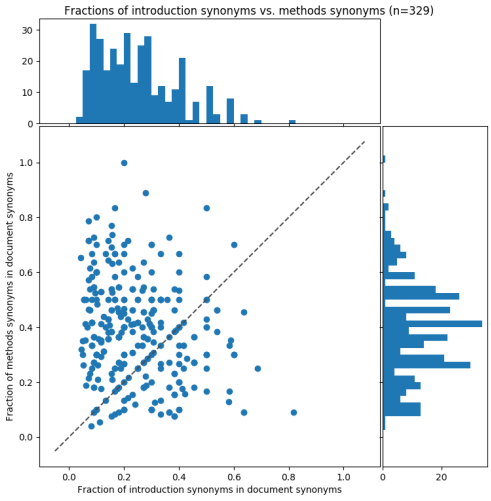


(b) Conclusion versus methods synonyms for the inflated GO context. The conclusion rarely covers 30% or more of all document synonyms. More synonyms are found in the methods part of a document. The synonyms found in the conclusion rarely make up 50% of the document synonyms.

Figure A.18: **Comparison of abstract and conclusion hits (GO)** Comparison of the synonyms found in the abstract (a) and conclusion (b) against the synonyms identified in the methods section of the atherosclerosis documents.

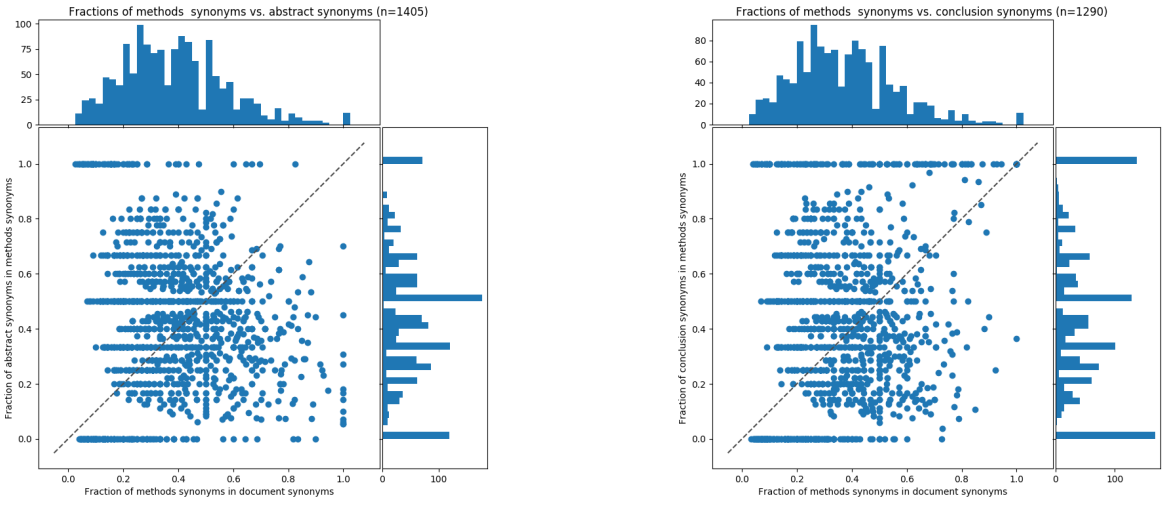


(a)



(b)

Figure A.19: **Comparison of abstract and introduction versus methods hits (ATOL)** The abstract synonyms overlap only by about 50% with the methods synonyms, while the methods synonyms make up 60% or more of all document synonyms (a). This holds also true for the introduction synonyms (b).



(a) (b)

Figure A.20: **Comparison of methods versus abstract and conclusion hits (GO)** The GO abstract synonyms overlap only by about 50% with the methods synonyms, while the methods synonyms make up 60% or more of all document synonyms (a). This holds also true for the conclusion synonyms (b).

A.3.2 atheMir

The atheMir software has been implemented by Markus Joppich and is available as release from `https://github.com/mjoppich/miRExplore/releases/tag/athemir`. The manuscript has been prepared by Markus Joppich. Ralf Zimmer contributed to the evaluation of the found interaction, the introduction of the manuscript and with general suggestions regarding text and figures. Christian Weber contributed with textual aspects as well as suggestions to improve figures. The accepted publication is available as open-access online article `https://doi.org/10.1055/s-0039-1693165`.

A.3.3 miRExplore

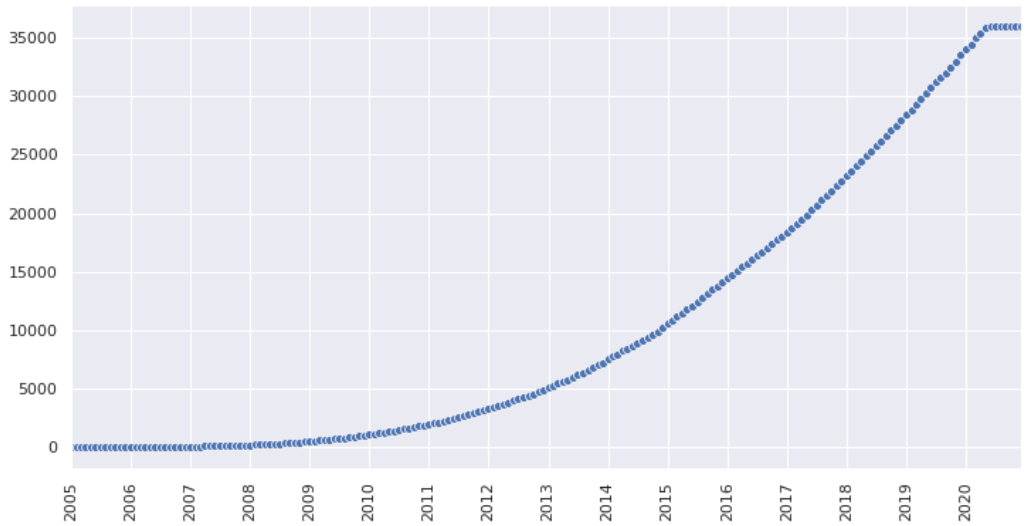


Figure A.21: **Number of PubMed articles relevant to miRNA-gene interactions** A cumulative histogram of published articles in PubMed with found miRNA-gene interactions. It can be seen that the number of miRNA-gene interactions per year still rises.

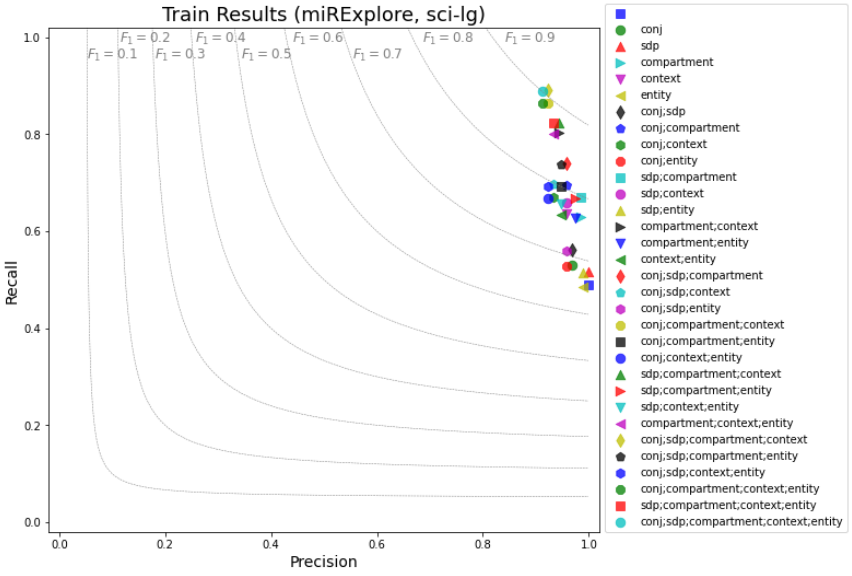


Figure A.22: **Detailed miRExplore performance (scispaCy sci-lg)** for miRNA-gene interaction prediction using the scispaCy model on the modified training data.

Table A.1: **Performance of miRExplore (interaction, scispaCy sci-lg model)** Interaction prediction is performed using the scispaCy sci-lg model on the modified test dataset.

Rules enabled	**Precision**	**Recall**	F_1
	1.000	0.595	0.746
conj	0.957	0.673	0.790
sdp	1.000	0.605	0.754
compartment	0.986	0.743	0.847
context	0.978	0.652	0.783
entity	1.000	0.595	0.746
conj;sdp	0.957	0.688	0.800
conj;compartment	0.949	0.873	0.910
conj;context	0.935	0.729	0.819
conj;entity	0.957	0.673	0.790
sdp;compartment	0.986	0.751	0.853
sdp;context	0.978	0.665	0.792
sdp;entity	1.000	0.605	0.754
compartment;context	0.964	0.821	0.887
compartment;entity	0.986	0.743	0.847
context;entity	0.978	0.652	0.783
conj;sdp;compartment	0.949	0.885	0.916
conj;sdp;context	0.935	0.746	0.830
conj;sdp;entity	0.957	0.688	0.800
conj;compartment;context	0.928	0.948	0.938
conj;compartment;entity	0.949	0.873	0.910
conj;context;entity	0.935	0.729	0.819
sdp;compartment;context	0.964	0.831	0.893
sdp;compartment;entity	0.986	0.751	0.853
sdp;context;entity	0.978	0.665	0.792
compartment;context;entity	0.964	0.821	0.887
conj;sdp;compartment;context	0.928	0.962	0.945
conj;sdp;compartment;entity	0.949	0.885	0.916
conj;sdp;context;entity	0.935	0.746	0.830
conj;compartment;context;entity	0.928	0.948	0.938
sdp;compartment;context;entity	0.964	0.831	0.893
conj;sdp;compartment;context;entity	0.928	0.962	0.945

Table A.2: **Performance of miRExplore (interaction, spaCy spacy-lg model)** Interaction prediction is performed using the spaCy spacy-lg model on the modified test dataset.

Rules enabled	**Precision**	**Recall**	F_1
	1.000	0.595	0.746
conj	0.986	0.624	0.764
sdp	0.891	0.586	0.707
compartment	0.855	0.698	0.769
context	0.986	0.642	0.777
entity	1.000	0.595	0.746
conj;sdp	0.877	0.617	0.725
conj;compartment	0.841	0.748	0.792
conj;context	0.971	0.673	0.795
conj;entity	0.986	0.624	0.764
sdp;compartment	0.768	0.679	0.721
sdp;context	0.877	0.634	0.736
sdp;entity	0.891	0.586	0.707
compartment;context	0.848	0.770	0.807
compartment;entity	0.855	0.698	0.769
context;entity	0.986	0.642	0.777
conj;sdp;compartment	0.754	0.732	0.743
conj;sdp;context	0.862	0.669	0.753
conj;sdp;entity	0.877	0.617	0.725
conj;compartment;context	0.833	0.827	0.830
conj;compartment;entity	0.841	0.748	0.792
conj;context;entity	0.971	0.673	0.795
sdp;compartment;context	0.761	0.750	0.755
sdp;compartment;entity	0.768	0.679	0.721
sdp;context;entity	0.877	0.634	0.736
compartment;context;entity	0.848	0.770	0.807
conj;sdp;compartment;context	0.746	0.811	0.777
conj;sdp;compartment;entity	0.754	0.732	0.743
conj;sdp;context;entity	0.862	0.669	0.753
conj;compartment;context;entity	0.833	0.827	0.830
sdp;compartment;context;entity	0.761	0.750	0.755
conj;sdp;compartment;context;entity	0.746	0.811	0.777

Table A.3: **Performance of miRExplore (interaction, spaCy BIONLP13CG model)** Interaction prediction is performed using the spaCy BIONLP13CG model on the modified test dataset.

Rules enabled	**Precision**	**Recall**	F_1
	1.000	0.595	0.746
conj	0.913	0.624	0.741
sdp	1.000	0.608	0.756
compartment	0.906	0.706	0.794
context	0.978	0.649	0.780
entity	1.000	0.595	0.746
conj;sdp	0.913	0.640	0.752
conj;compartment	0.826	0.765	0.794
conj;context	0.891	0.672	0.766
conj;entity	0.913	0.624	0.741
sdp;compartment	0.906	0.714	0.799
sdp;context	0.978	0.662	0.789
sdp;entity	1.000	0.608	0.756
compartment;context	0.884	0.772	0.824
compartment;entity	0.906	0.706	0.794
context;entity	0.978	0.649	0.780
conj;sdp;compartment	0.826	0.776	0.800
conj;sdp;context	0.891	0.687	0.776
conj;sdp;entity	0.913	0.640	0.752
conj;compartment;context	0.804	0.822	0.813
conj;compartment;entity	0.826	0.765	0.794
conj;context;entity	0.891	0.672	0.766
sdp;compartment;context	0.884	0.782	0.830
sdp;compartment;entity	0.906	0.714	0.799
sdp;context;entity	0.978	0.662	0.789
compartment;context;entity	0.884	0.772	0.824
conj;sdp;compartment;context	0.804	0.835	0.819
conj;sdp;compartment;entity	0.826	0.776	0.800
conj;sdp;context;entity	0.891	0.687	0.776
conj;compartment;context;entity	0.804	0.822	0.813
sdp;compartment;context;entity	0.884	0.782	0.830
conj;sdp;compartment;context;entity	0.804	0.835	0.819

Table A.4: **Comparison of miRExplore with other miRNA-gene mining approaches** Results are obtained on the original Bagewadi [19] benchmark or on the modified version (annotated with *mod.*).

Rules enabled	**Precision**	**Recall**	F_1
miRExplore/atheMir	0.720	0.770	0.744
miRExplore/BioNLP (mod.)	0.804	0.835	0.819
miRExplore/sci-lg	0.920	0.850	0.884
miRExplore/sci-lg (mod.)	0.928	0.962	0.945
miRExplore/spacy-lg (mod.)	0.746	0.811	0.777
miRSel	0.550	1.000	0.710
miRTex	0.920	0.820	0.867
ProMiner	0.410	0.450	0.429
ReLeX	0.480	0.790	0.597

Table A.5: **Performance of miRExplore (regulation, scispacy sci-lg model)** Interaction and regulation prediction is performed using the scispacy sci-lg model on the test dataset.

Rules enabled	**Precision**	**Recall**	F_1
	0.740	0.449	0.559
compartment	0.774	0.616	0.686
between	0.720	0.507	0.595
counts	0.846	0.797	0.821
return	0.369	0.594	0.455
compartment;between	0.766	0.645	0.700
compartment;counts	0.896	0.848	0.871
compartment;return	0.681	0.623	0.651
between;counts	0.889	0.855	0.871
between;return	0.767	0.652	0.705
counts;return	0.892	0.848	0.870
compartment;between;counts	0.894	0.870	0.882
compartment;between;return	0.767	0.652	0.705
compartment;counts;return	0.938	0.899	0.918
between;counts;return	0.933	0.906	0.919
compartment;between;counts;return	0.938	0.920	0.929

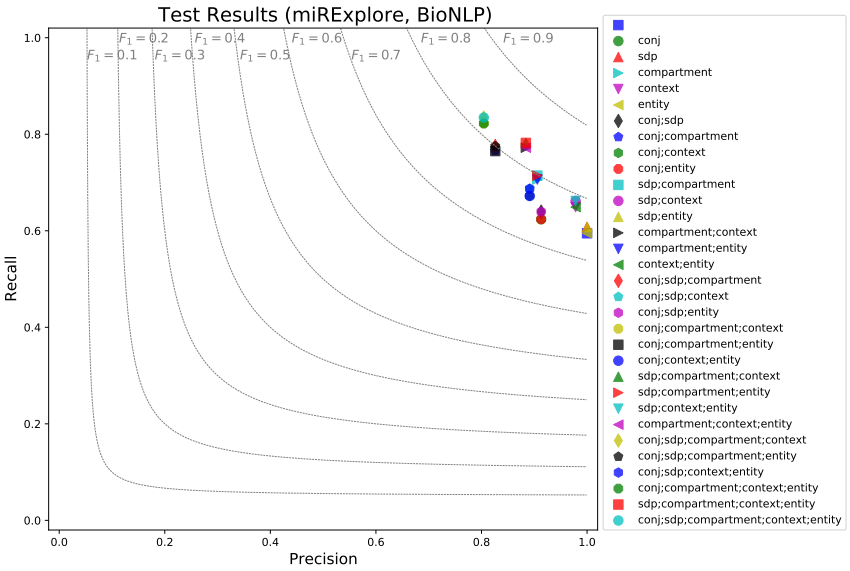
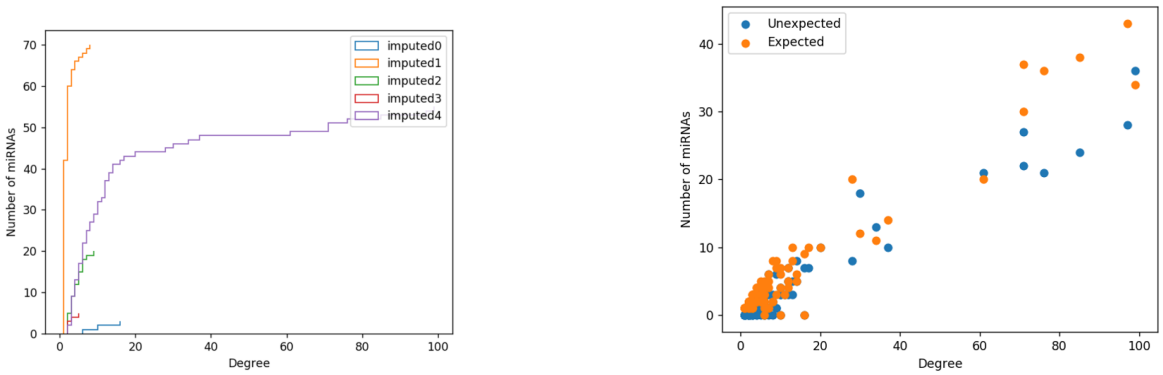
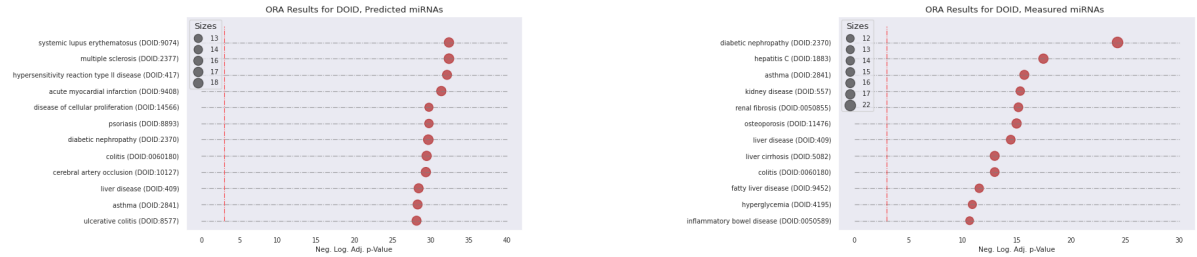


Figure A.23: **Detailed miRExplore performance (scispaCy BioNLP)** for miRNA-gene interaction prediction using the scispaCy (BioNLP) model on the modified test data



(a) (b)

Figure A.24: **miRExplore: Detailed view of active miRNA imputation** (a) Most miRNAs are imputed by the last step (imputed4), particularly also those miRNAs with a high degree. (b) Only rarely a miRNA has more imputed inconsistencies (unexpected) than consistently regulated edges (expected). The inconsistencies occur because the algorithm tries to assign yet missing miRNAs. The induced inconsistencies do not induce more unexplained miRNAs.



(a) Enriched disease ontology terms for the predicted miRNAs

(b) Enriched disease ontology terms for the measured miRNAs

Figure A.25: **miRExplore: miRNA over-representation in DOID terms** Both the predicted and imputed miRNA targets are enriched for genes associated with asthma. Strikingly, for the predicted miRNAs mostly auto-immune diseases are listed.

Benchmark Changes

In general the published benchmark from Bagewadi et al. [19] was used. However, it was noticed that for few interactions the benchmark contains arguable annotations.

These have been changed for the final benchmark used in this thesis.

Document 19941032

```
  <sentence id="miRNA-corp.d1.s1" origId="19941032.s4" text="Addition of exogenous miRNA-128 to
       CRL-1690 and CRL-2610 GBM cell lines (a) restored 'homeostatic' ARP5 (ANGPTL6), Bmi-1
      and E2F-3a expression, and (b) significantly decreased the proliferation of CRL-1690 and
      CRL-2610 cell lines. ">
  <entity charOffset="22-30" id="miRNA-corp.d1.s1.e0" text="miRNA-128" type="Specific_miRNAs"/>
  <entity charOffset="99-102" id="miRNA-corp.d1.s1.e1" text="ARP5" type="Genes/Proteins"/>
  <entity charOffset="105-111" id="miRNA-corp.d1.s1.e2" text="ANGPTL6" type="Genes/Proteins"/>
  <entity charOffset="115-119" id="miRNA-corp.d1.s1.e3" text="Bmi-1" type="Genes/Proteins"/>
  <entity charOffset="152-174" id="miRNA-corp.d1.s1.e4" text="significantly decreased" type="
      Relation_Trigger"/>
  <pair e1="miRNA-corp.d1.s1.e0" e2="miRNA-corp.d1.s1.e1" id="miRNA-corp.d1.s1.p0" interaction
      ="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d1.s1.e0" e2="miRNA-corp.d1.s1.e2" id="miRNA-corp.d1.s1.p1" interaction
      ="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d1.s1.e0" e2="miRNA-corp.d1.s1.e3" id="miRNA-corp.d1.s1.p2" interaction
      ="True" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 18262516

```
<sentence id="miRNA-corp.d11.s1" origId="18262516.s9" text="Moreover, the levels of the apl-1
    transcription are modulated by the activity of let-7 family microRNAs. ">
  <entity charOffset="28-32" id="miRNA-corp.d11.s1.e0" text="apl-1" type="Genes/Proteins"/>
  <entity charOffset="52-60" id="miRNA-corp.d11.s1.e1" text="modulated" type="Relation_Trigger
      "/>
  <entity charOffset="81-85" id="miRNA-corp.d11.s1.e2" text="let-7" type="Specific_miRNAs"/>
  <entity charOffset="94-102" id="miRNA-corp.d11.s1.e3" text="microRNAs" type="Non-
      Specific_miRNAs"/>
  <pair e1="miRNA-corp.d11.s1.e2" e2="miRNA-corp.d11.s1.e0" id="miRNA-corp.d11.s1.p0"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
```

```
  <pair e1="miRNA-corp.d11.s1.e3" e2="miRNA-corp.d11.s1.e0" id="miRNA-corp.d11.s1.p1"
      interaction="True" type="Non-Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 19424584

```
<sentence id="miRNA-corp.d48.s1" origId="19424584.s5" text="Consistently, miR-221/222 knocked-
    down through antisense 2'-OME-oligonucleotides increased p27Kip1 in U251 glioma
    subcutaneous mice and strongly reduced tumor growth in vivo through up regulation of
    p27Kip1. ">
  <entity charOffset="14-24" id="miRNA-corp.d48.s1.e0" text="miR-221/222" type="Specific_miRNAs
      "/>
  <entity charOffset="91-97" id="miRNA-corp.d48.s1.e1" text="p27Kip1" type="Genes/Proteins"/>
  <entity charOffset="107-112" id="miRNA-corp.d48.s1.e2" text="glioma" type="Diseases"/>
  <entity charOffset="127-130" id="miRNA-corp.d48.s1.e3" text="mice" type="Species"/>
  <entity charOffset="153-157" id="miRNA-corp.d48.s1.e4" text="tumor" type="Diseases"/>
  <entity charOffset="182-194" id="miRNA-corp.d48.s1.e5" text="up regulation" type="
      Relation_Trigger"/>
  <entity charOffset="199-205" id="miRNA-corp.d48.s1.e6" text="p27Kip1" type="Genes/Proteins"/>
  <pair e1="miRNA-corp.d48.s1.e0" e2="miRNA-corp.d48.s1.e1" id="miRNA-corp.d48.s1.p0"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d48.s1.e0" e2="miRNA-corp.d48.s1.e2" id="miRNA-corp.d48.s1.p1"
      interaction="False" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d48.s1.e0" e2="miRNA-corp.d48.s1.e4" id="miRNA-corp.d48.s1.p2"
      interaction="False" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d48.s1.e0" e2="miRNA-corp.d48.s1.e6" id="miRNA-corp.d48.s1.p3"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 20479936

```
<sentence id="miRNA-corp.d53.s6" origId="20479936.s4" text="miR-29b downregulates the
    expression of luciferase through hPGRN or mouse PGRN (mPGRN) 3'UTRs, and the regulation was
     abolished by mutations in the miR-29b binding site. ">
  <entity charOffset="156-167" id="miRNA-corp.d53.s6.e0" text="binding site" type="
      Relation_Trigger"/>
  <entity charOffset="0-6" id="miRNA-corp.d53.s6.e1" text="miR-29b" type="Specific_miRNAs"/>
  <entity charOffset="8-35" id="miRNA-corp.d53.s6.e2" text="downregulates the expression" type
      ="Relation_Trigger"/>
  <entity charOffset="68-72" id="miRNA-corp.d53.s6.e3" text="mouse" type="Species"/>
  <entity charOffset="74-77" id="miRNA-corp.d53.s6.e4" text="PGRN" type="Genes/Proteins"/>
  <entity charOffset="103-112" id="miRNA-corp.d53.s6.e5" text="regulation" type="
      Relation_Trigger"/>
  <entity charOffset="148-154" id="miRNA-corp.d53.s6.e6" text="miR-29b" type="Specific_miRNAs
      "/>
  <pair e1="miRNA-corp.d53.s6.e1" e2="miRNA-corp.d53.s6.e4" id="miRNA-corp.d53.s6.p0"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d53.s6.e6" e2="miRNA-corp.d53.s6.e4" id="miRNA-corp.d53.s6.p1"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 20479936

```
<sentence id="miRNA-corp.d54.s9" origId="20840605.s9" text="Our data indicate that
    overexpression of miR-21 protects against ischemic neuronal death, and that downregulation
    of FASLG, a tumor necrosis
    factor-a family member and an important cell death-inducing ligand whose gene is targeted
    by miR-21, probably mediates the neuroprotective effect. ">
```

```
  <entity charOffset="126-149" id="miRNA-corp.d54.s9.e0" text="tumor necrosis factor-a" type="
      Genes/Proteins"/>
  <entity charOffset="23-36" id="miRNA-corp.d54.s9.e1" text="overexpression" type="
      Relation_Trigger"/>
  <entity charOffset="41-46" id="miRNA-corp.d54.s9.e2" text="miR-21" type="Specific_miRNAs"/>
  <entity charOffset="99-112" id="miRNA-corp.d54.s9.e3" text="downregulation" type="
      Relation_Trigger"/>
  <entity charOffset="117-121" id="miRNA-corp.d54.s9.e4" text="FASLG" type="Genes/Proteins"/>
  <entity charOffset="222-229" id="miRNA-corp.d54.s9.e5" text="targeted" type="Relation_Trigger
      "/>
  <entity charOffset="234-239" id="miRNA-corp.d54.s9.e6" text="miR-21" type="Specific_miRNAs"/>
  <pair e1="miRNA-corp.d54.s9.e2" e2="miRNA-corp.d54.s9.e4" id="miRNA-corp.d54.s9.p0"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d54.s9.e2" e2="miRNA-corp.d54.s9.e0" id="miRNA-corp.d54.s9.p1"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d54.s9.e6" e2="miRNA-corp.d54.s9.e4" id="miRNA-corp.d54.s9.p2"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d54.s9.e6" e2="miRNA-corp.d54.s9.e0" id="miRNA-corp.d54.s9.p3"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
<sentence id="miRNA-corp.d54.s10" origId="20840605.s8" text="Moreover, overexpression of miR-21
     in neurons significantly reduced FASLG levels, and introduction of an miR-21 mimic into
    293-HEK cells substantially reduced luciferase activity in a reporter system containing the
     3'-UTR of Faslg. ">
  <entity charOffset="225-229" id="miRNA-corp.d54.s10.e0" text="Faslg" type="Genes/Proteins"/>
  <entity charOffset="46-66" id="miRNA-corp.d54.s10.e1" text="significantly reduced" type="
      Relation_Trigger"/>
  <entity charOffset="10-23" id="miRNA-corp.d54.s10.e2" text="overexpression" type="
      Relation_Trigger"/>
  <entity charOffset="28-33" id="miRNA-corp.d54.s10.e3" text="miR-21" type="Specific_miRNAs"/>
  <entity charOffset="60-79" id="miRNA-corp.d54.s10.e4" text="reduced FASLG levels" type="
      Relation_Trigger"/>
  <entity charOffset="68-72" id="miRNA-corp.d54.s10.e5" text="FASLG" type="Genes/Proteins"/>
  <entity charOffset="105-110" id="miRNA-corp.d54.s10.e6" text="miR-21" type="Specific_miRNAs
      "/>
  <pair e1="miRNA-corp.d54.s10.e3" e2="miRNA-corp.d54.s10.e0" id="miRNA-corp.d54.s10.p0"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d54.s10.e3" e2="miRNA-corp.d54.s10.e5" id="miRNA-corp.d54.s10.p1"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d54.s10.e6" e2="miRNA-corp.d54.s10.e0" id="miRNA-corp.d54.s10.p2"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d54.s10.e6" e2="miRNA-corp.d54.s10.e5" id="miRNA-corp.d54.s10.p3"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 20378606

```
<sentence id="miRNA-corp.d66.s4" origId="20378606.s3" text="To test our hypothesis, a predicted
     microRNA-binding site was found in the 3'-UTR of angiopoietin-1 using bioinformatics;
    variant rs2507800 was identified to be located in the miR-211-binding site of angiopoietin-
    1. ">
  <entity charOffset="36-43" id="miRNA-corp.d66.s4.e0" text="microRNA" type="Non-
      Specific_miRNAs"/>
  <entity charOffset="85-98" id="miRNA-corp.d66.s4.e1" text="angiopoietin-1" type="Genes/
      Proteins"/>
  <entity charOffset="176-182" id="miRNA-corp.d66.s4.e2" text="miR-211" type="Specific_miRNAs
      "/>
  <entity charOffset="200-213" id="miRNA-corp.d66.s4.e3" text="angiopoietin-1" type="Genes/
      Proteins"/>
  <pair e1="miRNA-corp.d66.s4.e2" e2="miRNA-corp.d66.s4.e3" id="miRNA-corp.d66.s4.p0"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d66.s4.e2" e2="miRNA-corp.d66.s4.e1" id="miRNA-corp.d66.s4.p1"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
```

```
  <pair e1="miRNA-corp.d66.s4.e0" e2="miRNA-corp.d66.s4.e3" id="miRNA-corp.d66.s4.p2"
      interaction="False" type="Non-Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d66.s4.e0" e2="miRNA-corp.d66.s4.e1" id="miRNA-corp.d66.s4.p3"
      interaction="False" type="Non-Specific_miRNAs-Genes/Proteins"/>
</sentence>
<sentence id="miRNA-corp.d66.s8" origId="20378606.s0" text="A functional variant in the 3'-UTR
    of angiopoietin-1 might reduce stroke risk by interfering with the binding efficiency of
    microRNA
    211.  ">
<entity charOffset="38-51" id="miRNA-corp.d66.s8.e0" text="angiopoietin-1" type="Genes/Proteins
    "/>
<entity charOffset="66-71" id="miRNA-corp.d66.s8.e1" text="stroke" type="Diseases"/>
<entity charOffset="124-135" id="miRNA-corp.d66.s8.e2" text="microRNA 211" type="
    Specific_miRNAs"/>
<entity charOffset="102-108" id="miRNA-corp.d66.s8.e3" text="binding" type="Relation_Trigger"/>
<pair e1="miRNA-corp.d66.s8.e2" e2="miRNA-corp.d66.s8.e0" id="miRNA-corp.d66.s8.p0" interaction
    ="True" type="Specific_miRNAs-Genes/Proteins"/>
<pair e1="miRNA-corp.d66.s8.e2" e2="miRNA-corp.d66.s8.e1" id="miRNA-corp.d66.s8.p1" interaction
    ="False" type="Specific_miRNAs-Diseases"/>
</sentence>
```

Document 20489155

```
  <sentence id="miRNA-corp.d69.s5" origId="20489155.s5" text="miR-107 has been implicated in
      Alzheimer's disease pathogenesis, and sequence elements in the open reading frame-rather
      than the 3' untranslated region-of GRN mRNA are recognized by miR-107 and are highly
      conserved among vertebrate species. ">
  <entity charOffset="31-49" id="miRNA-corp.d69.s5.e0" text="Alzheimer's disease" type="
      Diseases"/>
  <entity charOffset="0-6" id="miRNA-corp.d69.s5.e1" text="miR-107" type="Specific_miRNAs"/>
  <entity charOffset="155-157" id="miRNA-corp.d69.s5.e2" text="GRN" type="Genes/Proteins"/>
  <entity charOffset="182-188" id="miRNA-corp.d69.s5.e3" text="miR-107" type="Specific_miRNAs
      "/>
  <pair e1="miRNA-corp.d69.s5.e1" e2="miRNA-corp.d69.s5.e0" id="miRNA-corp.d69.s5.p0"
      interaction="False" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d69.s5.e1" e2="miRNA-corp.d69.s5.e2" id="miRNA-corp.d69.s5.p1"
      interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d69.s5.e3" e2="miRNA-corp.d69.s5.e0" id="miRNA-corp.d69.s5.p2"
      interaction="False" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d69.s5.e3" e2="miRNA-corp.d69.s5.e2" id="miRNA-corp.d69.s5.p3"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 20633539 The Bax/Bcl-2 ratio can be decreased by either decreasing only Bax (interaction) or both, or by increasing Bcl-2. In any case, an interaction between miR-21 and both Bax and Bcl-2 is recorded (the latter being arguable).

```
  <sentence id="miRNA-corp.d70.s1" origId="20633539.s0" text="MiR-21 protected human
      glioblastoma U87MG cells from chemotherapeutic drug temozolomide induced apoptosis by
      decreasing Bax/Bcl-2 ratio and caspase-3 activity.  ">
  <entity charOffset="0-5" id="miRNA-corp.d70.s1.e0" text="MiR-21" type="Specific_miRNAs"/>
  <entity charOffset="17-34" id="miRNA-corp.d70.s1.e1" text="human glioblastoma" type="Diseases
      "/>
  <entity charOffset="17-21" id="miRNA-corp.d70.s1.e2" text="human" type="Species"/>
  <entity charOffset="120-122" id="miRNA-corp.d70.s1.e3" text="Bax" type="Genes/Proteins"/>
  <entity charOffset="124-128" id="miRNA-corp.d70.s1.e4" text="Bcl-2" type="Genes/Proteins"/>
  <entity charOffset="140-148" id="miRNA-corp.d70.s1.e5" text="caspase-3" type="Genes/Proteins
      "/>
```

```
  <pair e1="miRNA-corp.d70.s1.e0" e2="miRNA-corp.d70.s1.e5" id="miRNA-corp.d70.s1.p0"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d70.s1.e0" e2="miRNA-corp.d70.s1.e1" id="miRNA-corp.d70.s1.p1"
      interaction="False" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d70.s1.e0" e2="miRNA-corp.d70.s1.e3" id="miRNA-corp.d70.s1.p2"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d70.s1.e0" e2="miRNA-corp.d70.s1.e4" id="miRNA-corp.d70.s1.p3"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>

<sentence id="miRNA-corp.d70.s12" origId="20633539.s14" text="However, such effect was partly
    prevented by treatment of cells with miR-21 overexpression before, which appeared to
    downregulate the Bax expression, upregulate the Bcl-2 expression and decrease caspase-3
    activity. ">
<entity charOffset="76-89" id="miRNA-corp.d70.s12.e0" text="overexpression" type="
    Relation_Trigger"/>
<entity charOffset="69-74" id="miRNA-corp.d70.s12.e1" text="miR-21" type="Specific_miRNAs"/>
<entity charOffset="117-147" id="miRNA-corp.d70.s12.e2" text="downregulate the Bax expression"
    type="Relation_Trigger"/>
<entity charOffset="134-136" id="miRNA-corp.d70.s12.e3" text="Bax" type="Genes/Proteins"/>
<entity charOffset="150-180" id="miRNA-corp.d70.s12.e4" text="upregulate the Bcl-2 expression"
    type="Relation_Trigger"/>
<entity charOffset="165-169" id="miRNA-corp.d70.s12.e5" text="Bcl-2" type="Genes/Proteins"/>
<entity charOffset="195-203" id="miRNA-corp.d70.s12.e6" text="caspase-3" type="Genes/Proteins
    "/>
<pair e1="miRNA-corp.d70.s12.e1" e2="miRNA-corp.d70.s12.e3" id="miRNA-corp.d70.s12.p0"
    interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
<pair e1="miRNA-corp.d70.s12.e1" e2="miRNA-corp.d70.s12.e5" id="miRNA-corp.d70.s12.p1"
    interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
<pair e1="miRNA-corp.d70.s12.e1" e2="miRNA-corp.d70.s12.e6" id="miRNA-corp.d70.s12.p2"
    interaction="False" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 18607543

```
  <sentence id="miRNA-corp.d80.s5" origId="18607543.s5" text="We identified putative miR sites
      in the CDK6 including microRNA
      124a, a brain enriched microRNA. ">
  <entity charOffset="23-25" id="miRNA-corp.d80.s5.e0" text="miR" type="Non-Specific_miRNAs"/>
  <entity charOffset="40-43" id="miRNA-corp.d80.s5.e1" text="CDK6" type="Genes/Proteins"/>
  <entity charOffset="55-67" id="miRNA-corp.d80.s5.e2" text="microRNA 124a" type="
      Specific_miRNAs"/>
  <entity charOffset="87-94" id="miRNA-corp.d80.s5.e3" text="microRNA" type="Non-
      Specific_miRNAs"/>
  <pair e1="miRNA-corp.d80.s5.e0" e2="miRNA-corp.d80.s5.e1" id="miRNA-corp.d80.s5.p0"
      interaction="False" type="Non-Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d80.s5.e2" e2="miRNA-corp.d80.s5.e1" id="miRNA-corp.d80.s5.p1"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d80.s5.e3" e2="miRNA-corp.d80.s5.e1" id="miRNA-corp.d80.s5.p2"
      interaction="False" type="Non-Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 19228967

```
  <sentence id="miRNA-corp.d87.s9" origId="19228967.s8" text="BAG2 levels in cells are under
      the physiological control of the microRNA miR-128a, which can tune paired helical
      filament Tau levels in neurons. ">
  <entity charOffset="64-71" id="miRNA-corp.d87.s9.e0" text="microRNA" type="Non-
      Specific_miRNAs"/>
  <entity charOffset="73-80" id="miRNA-corp.d87.s9.e1" text="miR-128a" type="Specific_miRNAs"/>
```

```
  <entity charOffset="122-124" id="miRNA-corp.d87.s9.e2" text="Tau" type="Genes/Proteins"/>
  <entity charOffset="0-3" id="miRNA-corp.d87.s9.e3" text="BAG2" type="Genes/Proteins"/>
  <pair e1="miRNA-corp.d87.s9.e0" e2="miRNA-corp.d87.s9.e2" id="miRNA-corp.d87.s9.p0"
      interaction="False" type="Non-Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d87.s9.e0" e2="miRNA-corp.d87.s9.e3" id="miRNA-corp.d87.s9.p1"
      interaction="False" type="Non-Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d87.s9.e1" e2="miRNA-corp.d87.s9.e2" id="miRNA-corp.d87.s9.p2"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d87.s9.e1" e2="miRNA-corp.d87.s9.e3" id="miRNA-corp.d87.s9.p3"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

Document 15648093

```
  <sentence id="miRNA-corp.d88.s8" origId="15648093.s3" text="An inverse correlation has been
      shown in B cell chronic lymphocytic leukemia (B-CLL) between miR-15a and miR-16-1
      expression and the expression levels of arginyl-tRNA synthetase
      (RARS), an enzyme which associates with the cofactor p43 in the aminoacyl-tRNA synthetase
       complex. ">
  <entity charOffset="3-21" id="miRNA-corp.d88.s8.e0" text="inverse correlation" type="
      Relation_Trigger"/>
  <entity charOffset="41-75" id="miRNA-corp.d88.s8.e1" text="B cell chronic lymphocytic
      leukemia" type="Diseases"/>
  <entity charOffset="78-82" id="miRNA-corp.d88.s8.e2" text="B-CLL" type="Diseases"/>
  <entity charOffset="93-99" id="miRNA-corp.d88.s8.e3" text="miR-15a" type="Specific_miRNAs"/>
  <entity charOffset="105-112" id="miRNA-corp.d88.s8.e4" text="miR-16-1" type="Specific_miRNAs
      "/>
  <entity charOffset="202-211" id="miRNA-corp.d88.s8.e5" text="associates" type="
      Relation_Trigger"/>
  <entity charOffset="154-176" id="miRNA-corp.d88.s8.e6" text="arginyl-tRNA synthetase" type="
      Genes/Proteins"/>
  <entity charOffset="179-182" id="miRNA-corp.d88.s8.e7" text="RARS" type="Genes/Proteins"/>
  <entity charOffset="133-149" id="miRNA-corp.d88.s8.e8" text="expression levels" type="
      Relation_Trigger"/>
  <pair e1="miRNA-corp.d88.s8.e3" e2="miRNA-corp.d88.s8.e1" id="miRNA-corp.d88.s8.p0"
      interaction="True" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d88.s8.e3" e2="miRNA-corp.d88.s8.e2" id="miRNA-corp.d88.s8.p1"
      interaction="True" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d88.s8.e3" e2="miRNA-corp.d88.s8.e6" id="miRNA-corp.d88.s8.p2"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d88.s8.e3" e2="miRNA-corp.d88.s8.e7" id="miRNA-corp.d88.s8.p3"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d88.s8.e4" e2="miRNA-corp.d88.s8.e1" id="miRNA-corp.d88.s8.p4"
      interaction="True" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d88.s8.e4" e2="miRNA-corp.d88.s8.e2" id="miRNA-corp.d88.s8.p5"
      interaction="True" type="Specific_miRNAs-Diseases"/>
  <pair e1="miRNA-corp.d88.s8.e4" e2="miRNA-corp.d88.s8.e6" id="miRNA-corp.d88.s8.p6"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
  <pair e1="miRNA-corp.d88.s8.e4" e2="miRNA-corp.d88.s8.e7" id="miRNA-corp.d88.s8.p7"
      interaction="True" type="Specific_miRNAs-Genes/Proteins"/>
</sentence>
```

A.4 Chapter 3

A.4.1 bioGUI

The publication was created by Markus Joppich and supervised by Ralf Zimmer. The software was implemented by Markus Joppich, and most of the manuscript was written by

Markus Joppich. Ralf Zimmer contributed with several improvements to the manuscript. The accepted publication is available as open-access online article `https://doi.org/10.7717/peerj.8111`.

A.4.2 tsxCount

The software has been implemented by Markus Joppich. The manuscript has been prepared by Markus Joppich, with several suggestions by Wolfgang Joppich. Ralf Zimmer contributed with general suggestions regarding text and figures. This manuscript is currently under revision at PeerJ Computer Science.

Hardware Transactional Memory for multi-threaded k-mer counting

A prominent example of genomic entity counting is the counting of k-long substrings (k-*mer*s). It is of particular importance in the process of genome assembly, error correction (of sequenced reads) as well as for further sequence feature related tasks. Novel sequencing methods, like the Oxford Nanopore MinION, generate noisy reads, increasing the number of almost unique k-*mer*s. Hence, using regular integers to count such k-*mer*s is a waste of memory, particularly for large datasets. A sparse counting implementation can leverage this problem and deliver memory-efficient implementations.

When implementing such a sparse counting data structure in a multi-threaded environment, it must be ensured that no two threads try to write to the same location at the same time, as then results will be incorrect. Hence, the update-access to a specific field must be serialized. Four different serialization methods in the context of *k-mer* counting using a sparse representation have been implemented. The application framework can either use regular mutex-based locks (PTHREAD), OpenMP locks (OMP), an atomic compare-and-swap (CAS) and hardware transactional memory (TSX) serialization. In our example application it can be seen that the CAS approach performs worse than the lock-based approaches, but shows a high efficiency with more threads. Our results show that the TSX approach is the fastest explored technique for a larger number of threads, directly followed by OMP. The gap between TSX and OMP is quite narrow, but OMP is more convenient to implement.

Introduction

Counting genomic entities is a frequent task in many bioinformatics workflows. For instance, counting k-long substrings (k-*mers*) is important in the process of genome assembly [68, 160] and read error correction [32, 143], but also is an important ingredient to genome indexing [155]. Further use-cases of *k-mer*-based approaches can be found in the area of metagenomics, using the Kraken tools, like KrakenUniq [37]. Several genome assemblers, like Celera [223] or Canu [160], rely on building a graph-based structure upon k-*mer*s.

It is thus not surprising that researchers are also interested in determining the *k-mer* frequency using stand-alone *k-mer* counters like KMC2/3 [71, 158] or KCMBT [198]. Even computational frameworks for *k-mer*-based analysis are developed [202].

Still, *k-mer* counting is set into focus, as continuing benchmarks of *k-mer* counters show. While Manekar et al. focus on different published tools [199] in their benchmark, Li et al. prepared a micro-benchmark to compare multiple *k-mer* counting strategies[3]. Hence, *k-mer* counting is a prominent task in genomics. But particularly with new sequencing methods, like the Oxford Nanopore MinION, noisy reads are generated, increasing the amount of almost unique *k-mer*s. Simple, non-memory optimized *k-mer* counters which use the same number of storage bits for each *k-mer*, e.g. 32-bit unsigned integer, then waste a lot of memory for low-count *k-mer*s, which most likely originate from sequencing errors and occur only few times. A sparse counting implementation can leverage this problem and deliver memory-efficient implementations.

[203] introduced such a sparse implementation with Jellyfish, but highlighted an important problem when counting in parallel: only a single thread may write to an array entry at a time. Therefore, in any multi-threaded environment it must be ensured that no two threads try to write to the same memory location at the same time, as then results will be incorrect. Hence, the update-access to a specific field must be serialized.

As already mentioned, several benchmarks compare different *k-mer* counters or counting strategies. However, none of these benchmarks focuses on the serialization method. While there are benchmarks which compare or systematically evaluate such serialization methods and TSX in particular [277, 337, 192], the task solved within the protected areas are not too complex.

In Jellyfish [203] employ a specific method to avoid locking of the fields writing into the fields by using an atomic compare-and-swap operation. In general, mutex- or lock-based approaches are used for serialization. Transactional memory is an alternative serialization approach which shows promising results [159, 247]. Here we want to take advantage of the algorithmic idea from [203], in order to explore and benchmark serialization methods: regular serialization via a global mutex (PTHREAD) and transactional mutexes (OMP), C++ atomic compare and swap (CAS) and hardware transactional memory (TSX).

Understanding the advantages or short-comings of these methods in the context of *k-mer* counting delivers insights on whether, or not, these techniques are useful for bioinformatics applications. Moreover, our hash map implementation allows storing arbitrary (bit-encoded) data, such that it could not only be used for *k-mer*s, but any other data which has similar characteristics as well.

The following sections explain the algorithm, the implemented modifications with respect to [203] and also introduce the different serialization methods. Finally, the results of the benchmarking are discussed.

[3] https://github.com/lh3/kmer-cnt/

Methods

Code Availability

The tsxCount source code with analysis scripts is available from GitHub `https://github.com/mjoppich/tsxCount`.

Correctness

The used datasets are regular MinION direct RNA-seq samples. Hence, exact results could be counted using any other *k-mer* counter or even simple python scripts. The aggregated results are shown in Table A.6. It has been asserted that the counts reported by these scripts and the presented counter implementation are identical. For the time measurements, this check has been disabled, however.

Used Hardware

All experiments have been performed on three machines. One (laptop) computer is equipped with 32 GB RAM and an Intel(R) Core(TM) i7-7820HQ CPU with 4 cores and 8 logical processors. GCC 7.4 was used to compile the application. The execution here runs within the Windows Subsystem for Linux environment. The processor provides 256 KB L1-cache, and 1 MB and 8 MB L2/3 cache, respectively. Another (xeon server) computer is equipped with 128 GB RAM and an Intel(R) Xeon(R) W-2145 CPU with 8 cores and 16 logical processors. GCC 7.5 was used to compile the application. The processor provides 512 KB L1-cache, and 8 MB and 11 MB L2/3 cache, respectively. A further (silver server) computer with two sockets (each equipped with an Intel(R) Xeon(R) Silver 4214 CPU with 12 cores and 24 logical processors) and 96 GB of RAM was used. Each CPU has 768kb L1-cache, 12 MB L2-cache and 16.5 MB L3-cache. The application here was compiled using clang++ 5.0.1.

The input was read from local SSDs on the laptop and the xeon server. The silver server is connected to a NFS system. For performance measurements, the project was built with `cmake` [205] build type `RELWITHDEBINFO` (`-O2 -g`). Unless otherwise noted, no OMP proc bind was used. Wall-clock time evaluation and memory consumption analysis has been performed using `/usr/bin/time --verbose`.

In addition, a performance analysis using Intel vTune Amplifier[4] has been conducted.

For the profiling, the different implementations have to read in FASTQ files with (long) reads and count the occurring *k-mer*s. All implementations use the same I/O and counting framework. The reported run times are averaged over three distinct runs.

Test Dataset

In order to test the implementations on a real dataset, a direct RNA-seq dataset from SRA with accession ID `SRR5989373` was used. The dataset has been sub-divided into

[4]https://software.intel.com/en-us/vtune

Table A.6: **tsxCount overview of test datasets** Overview of the SCER dataset and its subsets. The full dataset contains 241 446 transcriptomic reads obtained through MinION long-read sequencing. The original dataset is available from NCBI SRA under accession `SRR5989373`.

Dataset name	Number of reads	Number of distinct 14-mers
SCER.6_25	15.090	9.473.626
SCER.12_5	30.181	16.203.616
SCER.25	60.362	27.186.456
SCER.50	120.723	42.447.595
SCER.100	241.446	66.266.341

multiple parts using a fraction of reads. Using transcriptomic reads from direct RNA-seq is of particular interest, as two important features are given: the MinION sequencing typically has a sequence identity/correctness of roughly 15%, generating many almost singleton *k-mer*s. Additionally, poly-A tails may yield very high counts if these are sequenced as well. Thus, this test set is expected to create a very bi-modal distribution of *k-mer* counts.

For this benchmark, the task has been set to count 14-mers. The parameter l for the hash map has been set to 26 creating 2^{26} storage fields of size 32 bits. The field size, meaning the data type the big integer class uses internally for storage is 8 bit (`uint8_t`). Hence, each big integer uses an array of 4 unsigned integers to store data in. The characteristics of all datasets is given in Table A.6. Using only 14-mers has the advantage that exact counting could be performed using a python script to generate comparison data and ensure that all implementations deliver correct results.

Evaluation

The purpose of this paper is not to have a new, fast *k-mer* counting tool. Here we want to compare multiple ways to implement a sparse memory entity counter, which essentially differs only in the serialization technique used. It is thus not intended to compare with specific *k-mer* counting tools, instead, the different serialization techniques are benchmarked against each other.

Algorithmic Framework

Hash Map implementation

The hash map implementation used here follows the ideas presented by [203]. While Marcais et al. store *key* and *value* in two different arrays, here $(key, value)$ pairs are stored in a contiguous array. The *key* part is used to store the *k-mer* information, and the *value* part stores the number of observations for the specific *k-mer*. Using a specific bijective hashing function (like Marcais et al.) for the *k-mer*, the upper bits of the *k-mer* are encoded in the

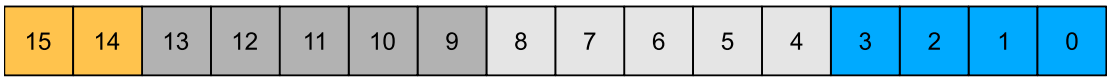


Figure A.26: **tsxCount bit representation of a key-value pair** The upper 12 bits (grey+orange) represent the *k-mer* information and also encode for the actual *k-mer*. The lower 4 bits are the value bits (blue). The upper 2 bits (orange) are not needed in case of a secondary entry for a *k-mer* and can be used as additional storage then (func-part). The grey bits contain the hash value from the bijective hashing function and determine, together with the *k-mer*'s position, the *k-mer* itself. In the case of a secondary entry for a *k-mer* (overflow), the dark grey fields store the number of reprobes needed to reach this element, and the light grey fields store the number of reprobes from the last element.

position the *k-mer* is stored in. These bits can then be used for further purposes. The structure of such a key-value-pair is shown in Figure A.26.

The difference to the original implementation [203] is when storing an additional field for a specific *k-mer* after an overflow of the initial storage part has occurred. Given 4 storage bits, this is the case if the *k-mer* is counted $2^4 = 16$ times or more. Similar to Marcais et al., a secondary field is created for this *k-mer* and the number of reprobes needed to reach this field is stored in the *key*-part. This requires fewer bits than the *key* is long — for a hash map with 2^l entries, at most l fields. The remaining bits $(2 * k - l)$ may serve as additional value fields and will be referred to as the *func*-part of the key. During evaluation of our implementation it has been noticed that this is not working, at least not with the reprobe-function from the original paper $reprobe(i) = i * (i + 1)/2$ for the i-th retry. It can be shown that two different *k-mer*s can be positioned at the same location within the hash map, with the same number of reprobes back to the previous element (Table A.10). This leads to incorrect counts. In order to avoid this, the bits reserved for storing the reprobe count are divided. The lower bits store the number of reprobes from the last element, while the upper bits store the number of reprobes needed to reach this element. This combination will only match for exactly the same element, but obviously limits the number of allowed reprobes to $2^{2*k-l-1}$, which did not pose a problem in our tests.

UBigInt Implementation

Our approach is designed to count arbitrarily large *k-mer*, in contrast to many other implementations which require storing the *k-mer* in a single integer. This however has the drawback, that particularly for large values of k, e.g. $k = 128$ requiring $256bit$ (2bit per base), no suitable data type is available. In order to accomplish this, multiple base fields of a specific bit-size (e.g. 8bit) are concatenated to derive the actually desired value. Using this, however, implicates that atomic operations can hardly be used, and an overhead for copying over multiple fields is introduced.

General Implementation

The general idea for the counter implementation was taken from [203]. In order to compare the different serialization methods, the same function body has been used for all serialization techniques, with exception to the CAS implementation, for reasons explained in section 15. When adding a *k-mer* to the counter (Code Example 1), first possible positions are calculated. Then it is checked whether a matching *k-mer* entry is found at these positions. If not, this *k-mer* was not entered into the counter before and the first empty field is used as initial storage. If the field was already occupied, it is checked whether the key of this field matches the *k-mer* and if so, the contained value is incremented.

Code Example 1: add *k-mer* in hash-map

```
   Input  : A k-mer as TSX::kmer
   Output : true if add was successful, false otherwise
 1 function addKmer (kmer);
 2 i=0;
 3 for i ≤ allowedRetries do
 4 |   pos = getPosition(kmer, i);
 5 |   lock(pos);                  // lock current position for increment
 6 |   if empty(pos) then
 7 |   |   map[pos] = makeKeyVal(kmer, i, 1) ;       // empty field; insert
   |   |    count 1
 8 |   |   unlock(pos);     // unlock current position after increment
 9 |   |   break;
10 |   else
11 |   |   if kmer(map[pos]) == kmer then ;          // field matches kmer
12 |   |
13 |   |   |   hasOverflow = increment(kmer, pos, i) ;      // increment count
14 |   |   |   unlock(pos); // unlock current position after increment
15 |   |   |   if hasOverflow then
16 |   |   |   |   handleOverflow(kmer, pos, i) ;   // handle overflow - will
   |   |   |   |    lock next positions
17 |   |   |   end
18 |   |   |   break;
19 |   |   else
20 |   |   |   ++i ;   // position does not match, try next position
21 |   |   |   unlock(pos);        // unlock current position for retry
22 |   |   end
23 |   end
24 |   return added
25 end
```

The `increment` function (Code Example 2) has to handle two cases. The first case is the most common case: the value in the storage bits of an entry has to be incremented. If this was successful, again two cases must be considered. First, if no overflow occurred, the operation completed successfully and may return. If an overflow occurred, and if this was not a secondary position, then the overflow must be handled. Hence, the increment function returns 2, signalling that the calling function must care about the overflow. Otherwise, if this was a secondary position, it is tried to increment the func-part of the key, hoping to prevent an overflow. Only if there happens an overflow again, the calling function must take care for handling the overflow.

The `handleOverflow` function (Supplemental Material, Code Example 4) is similar to the `addkmer` function (Code Example 1). A matching position must be found. If the algorithm finds an empty field, a new overflow location is created (value 1, with reprobes set back to the original element). Otherwise, it checks whether the number of reprobes to the initial position and the number of reprobes to the last position match. If so, this field is incremented using the function described in Code Example 2.

Running the application

All implemented serialization techniques share a common data processing function (Code Example 3). This function controls the streaming of reads from a FASTQ file and feeds the reads into the counter. This procedure is implemented using OpenMP [53, 65] tasks for easy parallelization later on. The I/O is always handled by the OpenMP master thread. The number of reads per task of course also influences the performance. Here, the number of reads per task was set to 40 for all comparisons. After creating the chunk of 40 reads, first all *k-mer*s are computed within the task, before each *k-mer* is added to the final counter.

Locking implementation

Two different serialization techniques using locks have been implemented: first a version using regular mutexes (`PTHREAD`) and second a version using OpenMP `omp_lock_hint_speculative` with `omp_init_lock_with_hint` (OMP). The speculative hint refers to the circumstance that the programmer suggests that the operation should be implemented using speculative techniques, such as transactional memory, according to the OpenMP 5.0 API specification[5]. These two variants work similarly, but differ in their implementation. The OpenMP lock variant makes use of hardware transactional memory (if possible), while the PTHREAD version uses regular mechanisms.

In general, both versions make a hash map position exclusive for a specific thread. If a position is already assigned to a different thread, the lock can not be acquired. Otherwise, the lock is assigned to the specific thread. Such a procedure can, for instance, be implemented using a vector of assigned positions per thread. All operations to this vector then must be protected by a specific lock, allowing only one thread to pass at a time.

[5]https://www.openmp.org/spec-html/5.0/openmp.html

Code Example 2: increment *k-mer* in hash-map

```
Input  : A k-mer as TSX::kmer, the position to increment, and the retry count i
Output : true if add was successful, false otherwise
1  function increment (kmer, pos, i);
2  handleFuncOverflow = false;
3  while not incremented do
4  |  state = incrementKeyValue(kmer, position, rets)
5  |  if error(state) then
6  |  |  continue ;                                  // retry on error
7  |  else
8  |  |  if overflow(state) then ;                  // Overflow occurred
9  |  |
10 |  |  |  if funcIsValue then // Try to increment func-part first
11 |  |  |  |  handleFuncOverflow = true;
12 |  |  |  else
13 |  |  |  |  return 2 ;        // Overflow not contained, must be
   |  |  |  |   handled
14 |  |  |  end
15 |  |  else
16 |  |  |  return 1 ;                        // added with no overflow
17 |  |  end
18 |  end
19 end
20 if handleFuncOverflow then
21 |  while not incremented do
22 |  |  state = incrementFunc(kmer, position, rets) ; // increment func part
23 |  |  if error(state) then
24 |  |  |  continue ;                                  // retry on error
25 |  |  else
26 |  |  |  if overflow(state) then // func-part overflow
27 |  |  |  |  handleOverflow(kmer, position, rets) ;    // handles overflow
28 |  |  |  end
29 |  |  |  return 1;
30 |  |  end
31 |  end
32 end
```

Unlocking a position requires to lock the list of locked positions per thread, remove a specific position from the thread list, and unlock the list of locked positions.

In the SERIAL implementation, the calls to acquire, lock and unlock a position are empty function bodies.

Code Example 3: Process of reading FASTQ-file and adding *k-mer* to counter

```
   Input  : A FASTQ file sFilename
   Output : A Counter with all k-mers counted.
 1 function processFASTQ (sFilename);
 2 tsxMap = TSXHashMap(options)
 3 fastqReader = FASTQReader(sFilename)
   // initialize OpenMP
 4 #pragma omp parallel num_ threads(threads)
       // only executed by one/master thread
 5     #pragma omp master
 6         while pReader.hasNext() do
               // new task for each chunk of reads
 7             #pragma omp task firstprivate(pEntries) shared(iPosOfInterest)
 8                 std::vector<FASTQEntry> oEntries = pReader.getEntries(40);
 9                 for entry in oEntries do
10                     allKmers = getKmersForEntry(entry);    // calculate all
                        k-mers
11                     for kmer in allKmers do
12                         tsxMap.addKmer(kmer);       // increment for each
                            k-mer
13                     end
14                 end
15         end
```

CAS implementation

The CAS operation is an atomic compare and swap operation, which atomically compares the content of two memory locations and inserts a new value, if the previous comparison was positive.

Instead of directly copying the incremented key-value-pair into the hash map, this approach first fetches the current key-value-pair, stores a copy of it, then calculates the incremented version, recognizes overflows, and then calls a specific version of the copy-to-memory function, which performs this operation using the CAS operation. In contrast to the other serialization techniques, a further copy of the array position before the increment must be maintained.

TSX implementation

TSX refers to a hardware transactional memory implementation built into many Intel CPUs. It allows starting transactions, which are only committed if no other transaction has interfered with the initial transaction. If a conflict occurs, all changes are reverted and a user-defined resolution strategy must be executed. Here we follow the following strategy:

in case of a regular abort, e.g. due to conflicts, the transaction is repeated.

The locking implementation can be used for this serialization method, in general, but before fetching the original value from the hash map, the transaction must be started using the `_xbegin ()` directive. The transaction is to be finished if the incremented key-value pair was stored back into the hash map. This is done by calling the `_xend()` directive. If the value could not be stored (e.g. field not empty), the transaction is aborted, and the resolution strategy is applied.

For TSX is it important to have all required values already in cache, as the transaction will otherwise fail/abort. Hence, before starting the transaction, the required data is prefetched. The destination memory location also has to be prefetched, however in write-mode, which here is achieved by calling the `__atomic_store(pPos+i, pPos+i, __ATOMIC_RELAXED)` directive. This directive reads the memory at position `pPos+i` and writes the same value back at that position. Being an atomic operation, no serialization is required.

Results

In order to identify the advantages or disadvantages of the different serialization techniques, performance evaluations using several fractions of a publicly available transcriptomic sequencing dataset have been conducted (Table A.6). The measurements show, that in general, the different serialization techniques perform well (Figures A.27-A.29) and all show a considerable speed-up with more threads (Tables A.7-A.9).

Summary

Laptop computer (Table A.7) SERIAL is the fastest implementation on a single thread. OMP and CAS show super-linear speed-up on real cores. PTHREAD and OMP are less sensitive for hyper-threading. TSX is the fastest implementation on the maximum number of threads.

Xeon Server (Table A.8) Super-linear speed-up can be seen on real cores for all parallel implementations. Using 16 hyper-threads TSX still reaches an efficiency of more than 70%, but has not yet reached the times of OMP. The SERIAL method is the fastest implementation on a single thread.

Silver Server (Tables A.9, A.11, A.12) On a single thread PTHREAD is the fastest of all the parallel implementations, but TSX and OMP are within a range of less than 8%. CAS is almost four times slower.

At 12 threads (real cores of a single socket) this changes considerably. Due to excellent efficiency TSX is the fastest implementation now. Because of super-linear speed-up CAS is already faster than PTHREAD and only about 6% slower than OMP. The lock-based implementations have efficiencies of only 50% and less (OMP, PTHREAD).

Exploiting the twelve additional real cores of the second socket does not yield advantages. TSX with 24 threads is about 20 seconds or 9% faster. CAS improves by about 45%, whereas

PTHREAD and OMP slow down. Using hyper-threading on every core (48 threads) shows again that there is only a small gain for TSX (less than 10%) and CAS (about 16%). The slow-down for PTHREAD and OMP is dramatically.

Detailed Results

On the laptop computer some difference between the OMP and PTHREAD implementation can be seen for an increasing number of threads, although both use a mutex lock implementation. However, the OMP implementation internally may use transactional memory for the locking. It is also interesting to note that the compare-and-swap (CAS) method is significantly slower than the other serialization techniques. This might have multiple reasons: unlike the jellyfish implementation [203], our CAS implementation has not been specifically optimized and does not use SSE/AVX optimized CAS operations (simple GNU version). Surprisingly, the more threads are used, the better the CAS version performs. Particularly for the CAS version, the achieved speed-up is (super-)linear up to 48 threads on the silver server (Table A.9). The speed-up of the lock-based implementations (OMP, PTHREAD) stagnates with more than 12 threads on the silver server, both CAS and TSX show a higher speed-up. While the TSX implementation is considerably faster, it can be seen that only the CAS implementation scales perfectly up to 48 threads, which was also reported by [203].

The main finding can be seen most clearly with the large dataset on the silver server with 48 threads (Figure A.29). Here it is observed that for a low number of threads the lock-based approaches have an advantage over TSX. With more threads this advantage becomes smaller, and TSX overtakes the lock based versions already at 6 threads on the silver server (Figure A.29, Table A.9). On the xeon server this can not be seen this early, and also not this clear. At 16 threads the PTHREAD implementation slows down and TSX catches up with the OMP implementation – which is also observed on the laptop. The observed behaviour on the silver server does not depend on the possible options for binding the threads to the cores (`OMP_PROC_BIND`, Figures A.29, A.32, A.33). It should be noted that without binding threads to a specific core (Figure A.33, Table A.12) the CAS implementation even overtakes TSX. This is not surprising, since TSX will require many cache transfers, even across socket boundaries, which the CAS version does not require. However, in absolute values the bound versions are faster than the unbound setting (Tables A.9, A.11 and A.12).

The CAS implementation is the most time-consuming one. It is, however, noteworthy, that the speed-up achieved by CAS is comparable to the other serialization techniques, and remains linear with more threads. The SERIAL method is the fastest implementation on a single thread. OMP, PTHREAD and TSX vary in time within a range of less than 10%. Nevertheless, as long as the number of threads is smaller or equal to the number of physical cores super-linear speed-up for all parallel implementations can be observed on the xeon server. On the silver server this is similar for the threads on the first socket. This is likely due to the better cache behaviour when using more physical cores, which have their own local cache.

Table A.7: **tsxCount runtimes (SCER.6_25, laptop)** With more threads, the TSX implementation improves, overtaking the OMP implementation at 8 threads.

Method/Threads	1	2	4	8
CAS [s]	411.53	172.31	100.70	81.86
Speed-up	1.00	2.39	4.09	5.03
Efficiency	1.00	1.19	1.02	0.63
SERIAL [s]	168.76	-	-	-
Speed-up	1.00	-	-	-
Efficiency	1.00	-	-	-
OMP [s]	218.70	80.40	47.79	35.39
Speed-up	1.00	2.72	4.58	6.18
Efficiency	1.00	1.36	1.14	0.77
TSX [s]	174.42	81.70	55.25	31.42
Speed-up	1.00	2.13	3.16	5.55
Efficiency	1.00	1.07	0.79	0.69
PTHREAD [s]	203.97	102.07	56.24	49.09
Speed-up	1.00	2.00	3.63	4.16
Efficiency	1.00	1.00	0.91	0.52

Table A.8: **tsxCount runtimes (SCER.100, xeon server)** The OMP and PTHREAD implementations perform similarly well on few threads. On 16 threads the TSX implementation catches up with the OMP one. The SERIAL implementation does not protect memory accesses and thus may yield incorrect results, but shows the overhead involved for serialization.

Method/Threads	1	2	4	8	16
CAS [s]	10689.00	4732.67	1756.80	838.79	650.02
Speed-up	1.00	2.26	6.08	12.74	16.44
Efficiency	1.00	1.13	1.52	1.59	1.03
OMP [s]	3091.63	1239.50	610.97	368.68	281.65
Speed-up	1.00	2.49	5.06	8.39	10.98
Efficiency	1.00	1.25	1.27	1.05	0.69
PTHREAD [s]	3014.45	1225.92	616.46	381.87	319.00
Speed-up	1.00	2.46	4.89	7.89	9.45
Efficiency	1.00	1.23	1.22	0.99	0.59
TSX [s]	3208.22	1289.70	793.97	408.40	284.63
Speed-up	1.00	2.49	4.04	7.86	11.27
Efficiency	1.00	1.24	1.01	0.98	0.70
SERIAL/NOLOCK[1] [s]	3000.77	1169.83	690.77	278.01	247.33
Speed-up	1.00	2.57	4.34	10.79	12.13
Efficiency	1.00	1.28	1.09	1.35	0.76

[1]: The parallelized SERIAL method has been measured on more than one thread only once.

Table A.9: **tsxCount runtimes (SCER.100, silver server, `OMP_PROC_BIND=spread`)** With few threads the performance of OMP and TSX is similar. With a larger number of threads the performance of TSX becomes best. Strikingly the CAS implementation scales well and outperforms the lock-based variants at 16 threads.

Method/Threads	1	2	6	9	12	16	20	24	36	42	48
CAS [s]	12324.67	3776.67	1478.81	698.14	551.22	401.22	370.13	303.10	315.59	274.65	253.07
Speed-up	1.00	3.26	8.33	17.65	22.36	30.72	33.30	40.66	39.05	44.87	48.70
Efficiency	1.00	1.63	1.39	1.96	1.86	1.92	1.66	1.69	1.08	1.07	1.01
OMP [s]	3348.43	1282.85	482.37	457.17	523.57	599.19	687.00	778.17	820.46	785.25	795.39
Speed-up	1.00	2.61	6.94	7.32	6.40	5.59	4.87	4.30	4.08	4.26	4.21
Efficiency	1.00	1.31	1.16	0.81	0.53	0.35	0.24	0.18	0.11	0.10	0.09
TSX [s]	3558.31	1354.32	429.32	450.15	264.84	221.18	231.16	241.86	296.04	215.56	218.89
Speed-up	1.00	2.63	8.29	7.90	13.44	16.09	15.39	14.71	12.02	16.51	16.26
Efficiency	1.00	1.31	1.38	0.88	1.12	1.01	0.77	0.61	0.33	0.39	0.34
PTHREAD [s]	3320.98	1280.21	578.57	556.70	577.84	591.13	614.56	635.46	722.86	756.53	789.31
Speed-up	1.00	2.59	5.74	5.97	5.75	5.62	5.40	5.23	4.59	4.39	4.21
Efficiency	1.00	1.30	0.96	0.66	0.48	0.35	0.27	0.22	0.13	0.10	0.09

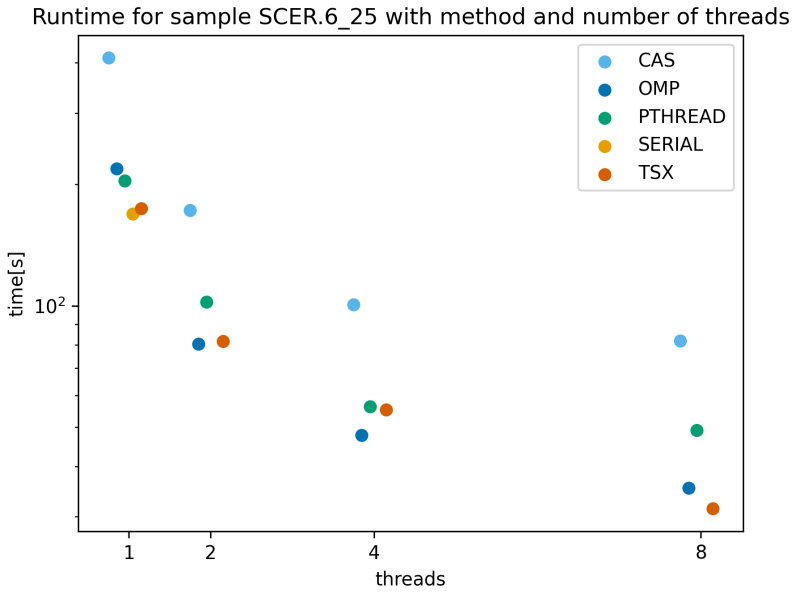


Figure A.27: **tsxCount runtimes (SCER.6_25, laptop)** With more threads, the TSX implementation improves, overtaking the OMP implementation at 8 threads.

On the laptop OMP and PTHREAD with four threads and four cores show no super-linear speed-up, but the efficiency is still above 90%. This is likely caused by the smaller cache on the laptop CPU in comparison to the server CPUs (see Tables A.7 and A.8).

The speed-up decreases as soon as the number of threads exceeds the number of cores. Nevertheless, still efficiencies of more than 70% with 8 threads on 4 cores can be seen on the laptop. On the xeon server (16 threads on 8 cores) a higher drop in efficiency for PTHREAD can be noticed, whereas CAS, OMP and TSX still show satisfying speed-up and efficiencies of more than 70% (Table A.8).

On the xeon server, the SERIAL implementation was also executed in parallel (SERIAL/NOLOCK). It must be noted that the parallel execution of SERIAL/NOLOCK does not yield correct results, because no serialization is employed. However, it can be regarded as a baseline for an ideal serialization, which allows a rough estimation of the overhead induced by the serialization techniques. Not using any serialization, most executions with multiple threads do not terminate properly or abort. Hence, the parallel SERIAL implementation was only executed once.

Performance Evaluation

In order to compare the behaviour of TSX (Figure A.30) and OMP (Figure A.31) implementations the Intel vTune Amplifier was used. In the TSX results it can clearly be seen that two threads (cpu_0 and cpu_1) report higher abort cycles than all others. These are the threads on the core with the OpenMP master thread which is responsible for IO.

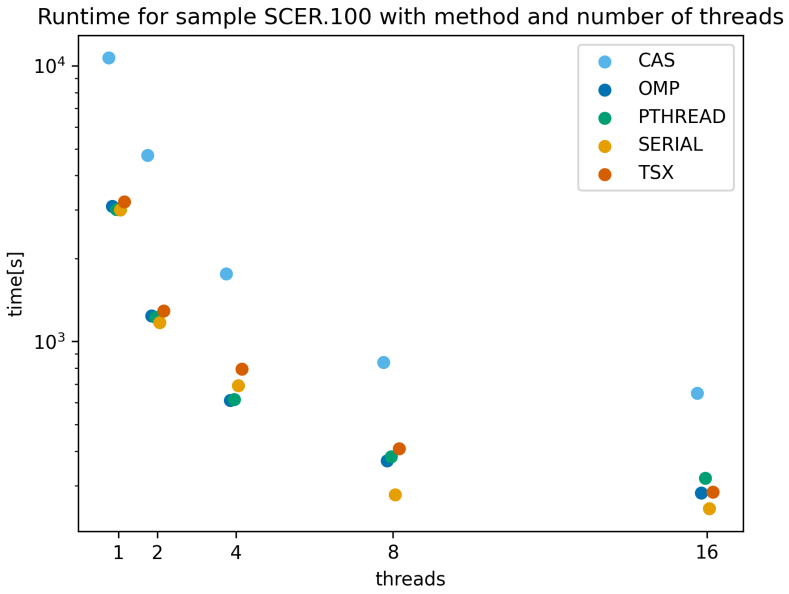


Figure A.28: **tsxCount runtimes (SCER.100, xeon server)** The OMP and PTHREAD implementations perform similarly well on few threads. On 16 threads the TSX implementation catches up with the OMP one. The SERIAL implementation does not protect memory accesses and thus may yield incorrect results, but shows the overhead involved for serialization.

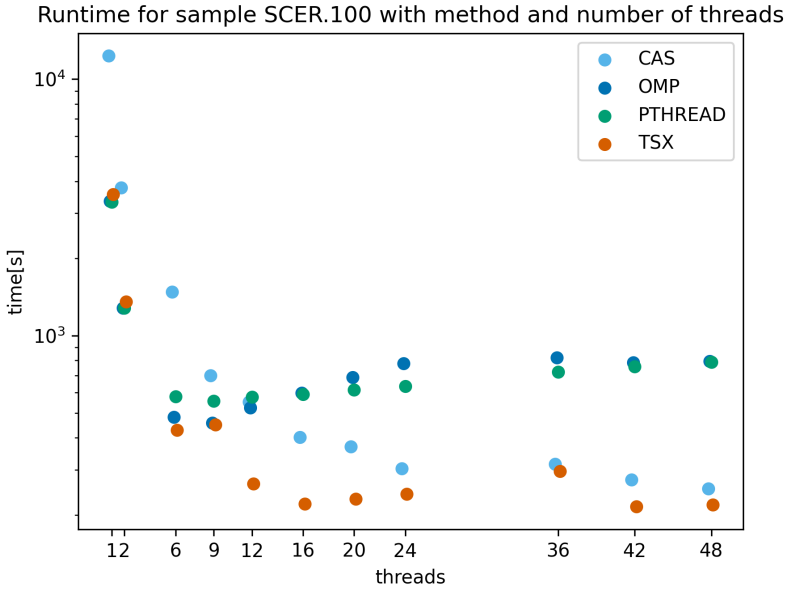


Figure A.29: **tsxCount runtimes (SCER.100, silver server, `OMP_PROC_BIND=spread`)** With few threads the performance of OMP and TSX is similar. With a larger number of threads the performance of TSX becomes best. Strikingly the CAS implementation scales well and outperforms the lock-based variants at 16 threads.

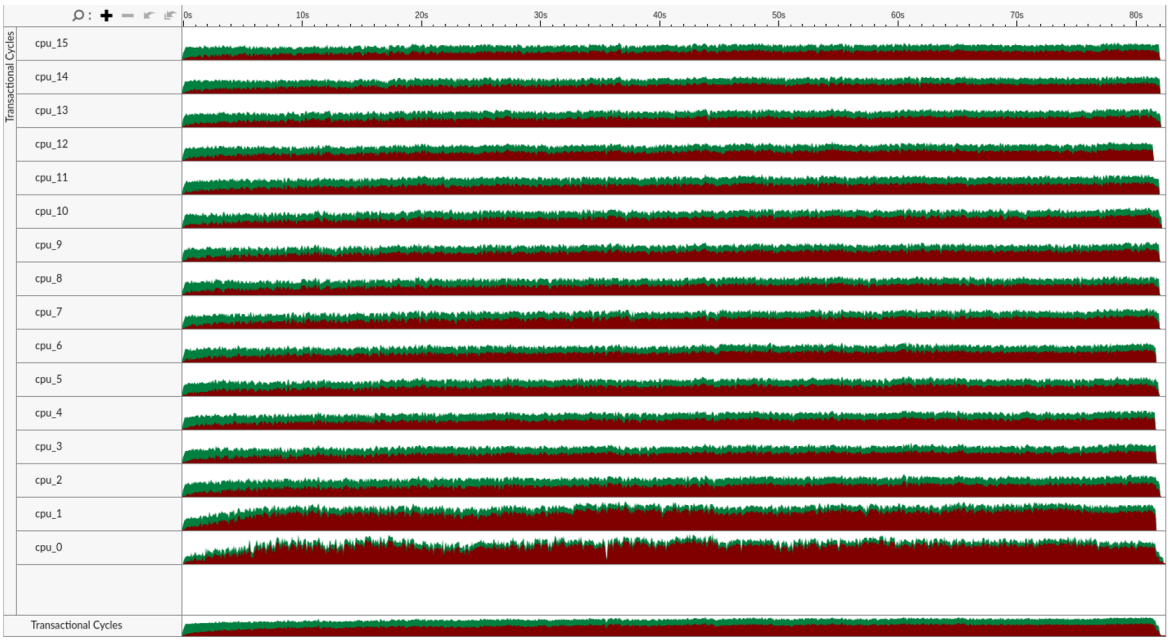


Figure A.30: **tsxCount TSX profiling with Intel vTune Amplifier** Transactional profiling of SCER.25 sample with Intel vTune Amplifier using tsx-exploration. Higher abort rates (brown area) can be noticed towards the end of the run.

Furthermore, an increase of abort cycles over time can be observed, not only in the master threads. This is plausible: more fields in the hash map are filled, hence more retries are needed. Also overflows occur more frequently towards the end, requiring more count operations which again create more transactions which could abort.

The OMP analysis looks as expected. Most of the time all threads are very busy. In contrast to TSX, the spin and overhead time is constant over the time. This is also as expected, since each increment will only acquire one lock at a time.

Discussion

The results demonstrate that hardware transactional memory is a useful and fast serialization technique, however, lacking interoperability. While the computational and software engineering overhead is quite large compared to lock-based serialization techniques, this pays out at a larger number of threads. This is, most likely, due to a significant increase in locking-activities for multiple threads: more threads need to lock hash map positions, and the query for locked positions takes more time. On the other hand, if more threads have more distinct elements to count, only few transactions will be in conflict. Hence, a slightly larger overhead for hardware transactions is taking less work than the locking overhead.

Using the bioinformatics problem of counting *k-mer*s using a hash map, which uses a memory-efficient representation of *k-mer* counts, requiring large protected operations, is a well-suited benchmark for several serialization techniques. It allows the comparison of

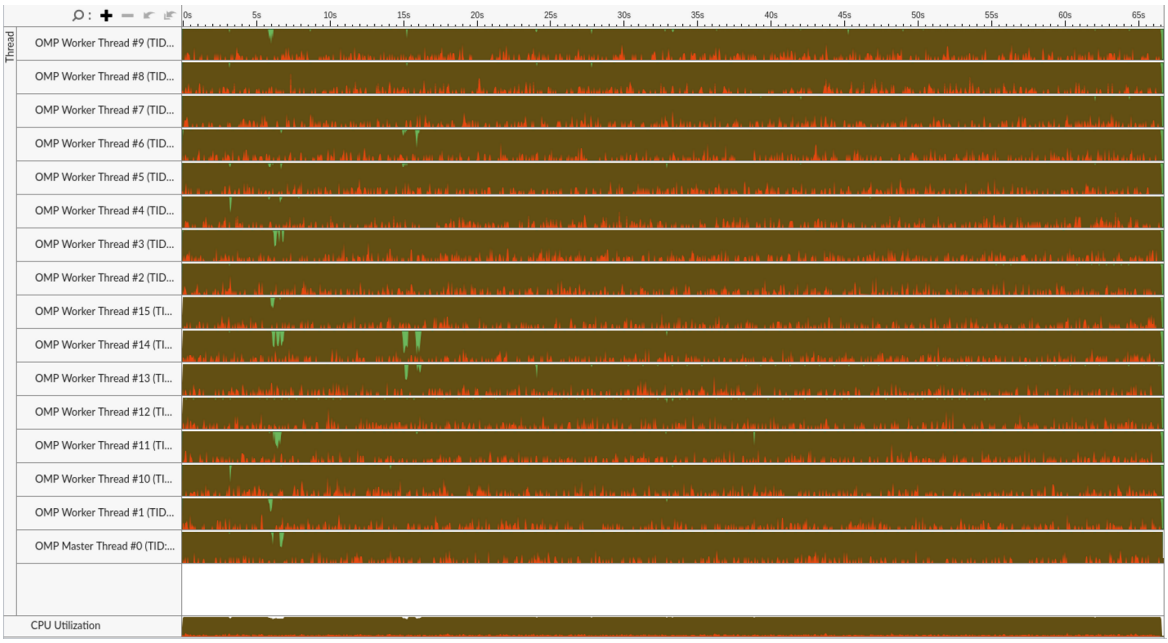


Figure A.31: **tsxCount OMP profiling with Intel vTune Amplifier** Transactional profiling of SCER.25 sample with Intel vTune Amplifier using Hotspot analysis. The waiting time for locks (red area) can be noticed throughout the run.

several techniques far away from pure numerics or simple increments. Moreover, the *k-mer* counting problem can be applied to arbitrarily large datasets, hence allowing the use of as many threads as possible.

In contrast to existing studies by [277, 247, 121], here we perform a significant workload on the data within the (lock/TSX-)protected areas. The benchmark is conducted on a real world problem and compares the different serialization techniques on the same problem. For the *k-mer* increment task, first a hash map position has to be retrieved, parsed, compared, incremented, and re-assembled. In particular, the retrieved values represent a bit-encoded data-structure on their own, which needs to be maintained. This requires more operations within the transactions than only the increment of an array position, for instance. Here, five different implementations can be compared: Lock-based methods (SERIAL, PTHREAD, OMP), TSX and CAS.

Very surprising have been the results obtained for the CAS implementation. CAS is the most different and special technique to implement, because existing code from the lock-based approaches needs to make sure that a *before-changes* copy is created, maintained and used to compare when writing the manipulated data. In this study it performed slowest, which is not surprising, requiring at least twice as much memory read/write operations. However, with many threads, the CAS implementation overtakes the lock-based approaches and performs almost as good as the TSX implementation. Only using the default CAS operation, without making use of SSE or AVX, this is possibly not the limit of the CAS approach (as shown by [203]). However, since all other implementations also do not make

use of such additional features, this comparison is as fair as possible. Nonetheless, the recorded linear speed-up up to 48 threads is astonishing enough compared to the other methods.

Of special interest is the comparison of the lock-based and TSX versions. For any lock-based method, the same building blocks and interface could be used. OMP and PTHREAD methods only differ in the locking mechanism, hence in the underlying implementation. The SERIAL version always acquires a virtual lock automatically. It thus uses the same implementation as versions with locks. It is interesting to see that there is a performance difference in using the PTHREAD-lock approach in comparison to the guided-lock version from OMP, which internally might use hardware transactional memory. Particularly with many threads, the OMP variant is more efficient than the PTHREAD version (Figure A.28, 8 vs. 16 threads), even though this depends on the platform (Figure A.29). Since the OMP implementation may use transactional locks internally (user-independent runtime decision), this could be a result from different hardware, compilers and libraries or computers. Even though the TSX implementation only performs better than the lock-based methods for 9 or more threads on the silver server, its performance with fewer threads is not significantly slower than OMP, for instance. Compared to the lock-based methods, TSX shows a linear speed-up for more threads used, but eventually stagnates after 16 threads. This stagnation can have multiple reasons, e.g. a saturation of memory IO or an increase in the transaction abort rate. In all experiments it can, however, be seen that with more threads, TSX becomes faster and achieves a better speed-up than the lock-based approaches.

Using hardware transactional memory as serialization techniques is quite new. There are no common patterns available, neither are distinct examples nor best practices. While a lock can protect an arbitrarily large code area, hardware transactions are limited in their size. A transaction is aborted if it exceeds its capacity or if not all required memory locations are cached. Both of these limitations likewise depend on the used hardware, compiler and implementation. In addition, transactions might be aborted due to a debugger operating within transactional regions, or if transactions are nested. During the implementation phase we found cache misses being the main reason for aborted transactions. Prefetching the necessary data (both for read and write) resolved this problem. The need for having data within the cache limits the applicability of transactions in general. This forces the user to keep the code within a transaction as small as possible. We have seen that on devices with smaller caches (like the laptop computer here), more prefetching is required to avoid cache misses and aborted transactions. With more prefetching of data, the implementation however gets less performant. Apart from the platform specificity of TSX (Intel), this makes the TSX implementation less portable and interoperable, because for each platform it has to be evaluated which level of prefetching is required.

Besides the pure computational efficiency, also the time required to implement the serialization techniques must be considered. The usage of locks has been very straight forward and easy to implement and to debug. The different locking mechanisms only demand a different lock initialization, and maybe a slightly different syntax for acquiring or releasing a lock. Finally, the overall time and resources needed to make a code threadsafe using locks is small, which is the opposite experience to using TSX.

Conclusion

In this paper we investigated how hardware transactional memory can be used as serialization technique for counting genomic entities, *k-mer*s. We compared the TSX strategy against no locks (SERIAL), pthread-mutex (PTHREAD), omp-locks (OMP) and a CAS implementation. The real-live benchmark is large enough for a significant workload and includes both biases introduced by the sequencing technology (error rate), and the transcriptomic sample origin (poly-A ends at reads, leading to high counts for view *k-mer*s). The obtained results can be used to make an informed decision on which serialization technique suits the respective environment better.

Given the differences between the PTHREAD and OMP implementations, it can be noted that the hinted-lock implementation (OMP) is more performant, possibly due to using hardware transactions internally. However, the difference, in general, is neglectable.

For our application, OMP (using speculative locks) and TSX serialization have been the most performant implementations on few threads. With an increasing number of threads TSX becomes advantageous compared to OpenMP locks, and CAS shows increasing efficiency, which can be seen by its linear speed-up, even at a high number of threads. Using only few threads, an OMP lock-based serialization technique can be preferred. Not only because OMP is at least as fast as TSX, but the small advantage of TSX in time efficiency is considerably offset by the experienced difficulties during implementation, debugging and its lower platform-robustness. If more threads are intended to be used, or if in general a linear speed-up is required, TSX and CAS are useful choices. In our benchmark CAS has the most constant speed-up, but having high initial costs, this only pays out with more than 24 threads, in our example, and can not overtake TSX. However, TSX is not available on all CPU platforms. The TSX implementation is also not platform robust regarding cache misses by heavily depending on the available caches of the processor. Hence, given that the OMP-approach is much easier to implement and faster or as fast for fewer threads, the OMP lock-based serialization wins also for reasons of interoperability and availability on all platforms.

In summary, the main question one has to answer before choosing a serialization technique, is on which platform the given software will be run. For a general purpose software, the usage of TSX is hardly possible, because only some Intel CPUs support TSX. For software which is intended to be run with many threads, the CAS approach can be useful, as it delivers an excellent speed-up with very high initial costs. If the target is a regular workstation, with an average CPU and thread count, lock-based approaches remain the favoured serialization technique, also being the most interoperable choice.

Supplemental Material

Reprobe Problem

An example for the reprobe problem mentioned in Section A.4.2 is given in Table A.10. In this example, positions 823, 828 and 830 are already blocked by other *k-mer*s. Further, assume that first *k-mer* X is incremented, requiring an overflow. Since position 830 is

Supplemental Material, Table A.10: Double-match scenario for the same position (same number of reprobes).

Position *k-mer X*	Reprobes	Reprobes from last insert	Description
829	0		initial
830	1	1	blocked
832	2	2	overflow

Position *k-mer Y*	Reprobes	Reprobes from last insert	
822	0		Initial
823	1	1	blocked
825	2	2	overflow
828	3	1	blocked

832	4	2	overflow; **Incorrect Increment!**

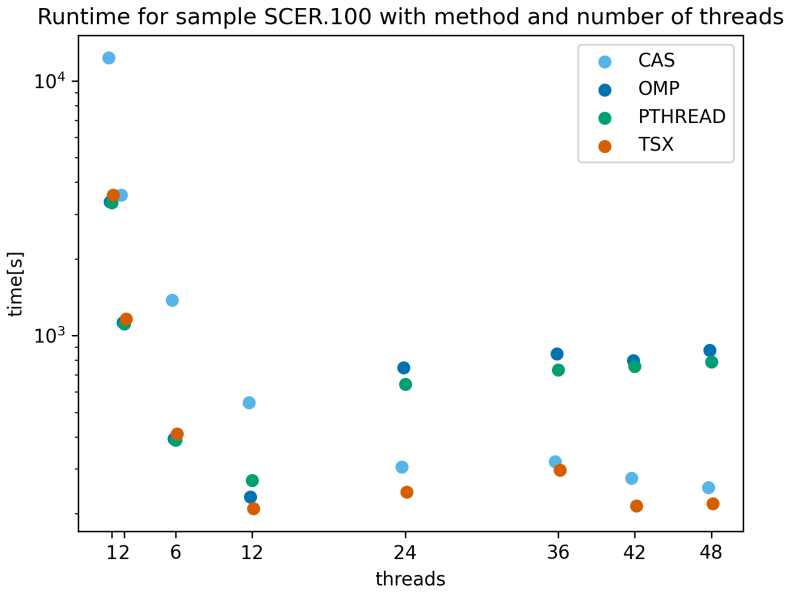
blocked, the second retry finds an empty field at position 832 (determined by the hashing function). The number of reprobes required from the last position is 2, which is encoded in the (*key*, *value*)-entry. A second *k-mer Y* has an initial entry at position 822. This *k-mer* already had an overflow, which is stored in field 825. Given that field 828 is already occupied, the next overflow will be stored in field 832. This is 2 reprobes away from the last entry for this $k - mer$. Hence, the encoded reprobes of 2, which were stored by *k-mer X*, match, and the overflow of *k-mer* Y increments a field of *k-mer* X. This can be circumvented by storing both the total number of reprobes as well as the number of reprobes from the last field.

Code Examples

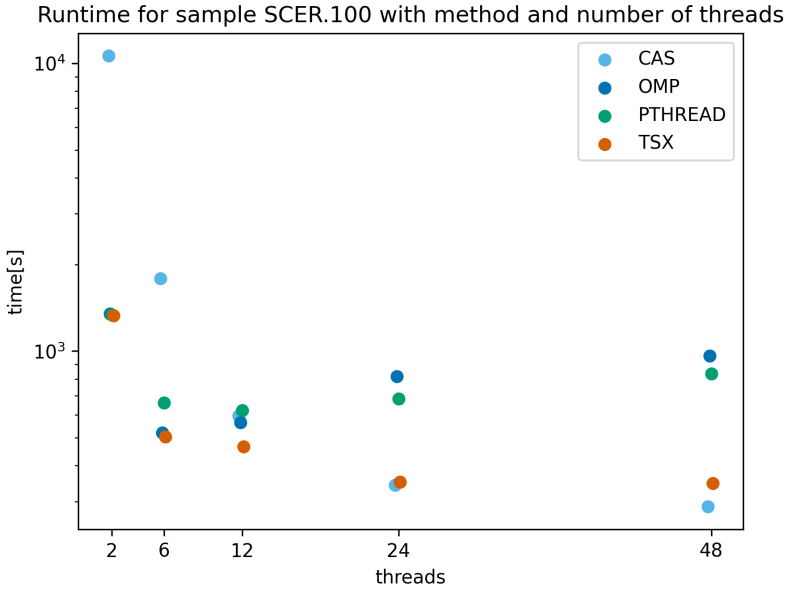
The pseudo-code for the handling of overflows during the increment operation is shown in Supplemental Material, Code Example 4.

Additional Experimental Results

Additional experimental results for the full dataset on the silver server using `OMP_PROC_BIND=close` (Figure A.32, Table A.11) and `OMP_PROC_BIND=false` (Figure A.33, Table A.12).



Supplemental Material, Figure A.32: **tsxCount runtimes (SCER.100, silver server, `OMP_PROC_BIND=close`)** With few threads the performance of OMP and TSX is similar. With a larger number of threads the performance of TSX becomes best. Strikingly the CAS implementation scales well and outperforms the lock-based variants at 24 threads. The performance of all implementations is similar to using `OMP_PROC_BIND=spread`.



Supplemental Material, Figure A.33: **tsxCount runtimes (SCER.100, silver server, `OMP_PROC_BIND=FALSE`)** With few threads the performance of OMP and TSX is similar. With a larger number of threads the performance of TSX becomes best. Strikingly, the CAS implementation scales well and outperforms the TSX and lock-based variants at 24 threads. The absolute run-times in this mode are higher than using `OMP_PROC_BIND=spread`.

Supplemental Material, Code Example 4: Handle overflow during *k-mer* increment

```
Input  : A k-mer as TSX::kmer, the position to increment, and the retry count i
Output : true if add was successful, false otherwise
1  function handleOverflow (kmer, pos, i);
2  rets=0;
3  while rets ≤ allowedRetries do
4      pos = getPosition(kmer, rets)
5      lock(pos);                  // lock current position for increment
6      if empty(pos) then
7          map[pos] = makeKeyVal(kmer, rets, 1);     // new secondary field
8          unlock(pos);     // unlock current position after increment
9          break
10     else
11         if kmer matches position then
12             hasOverflow = increment(kmer, pos, rets); // increment
                secondary field
13             unlock(pos); // unlock current position after increment
14             if hasOverflow then // further overflow
15                 handleOverflow(kmer, pos, rets);
16             end
17             break
18         else
19             ++rets;
20             unlock(pos);  // unlock current position, retry at next
                position
21         end
22     end
23 end
```

Supplemental Material, Table A.11: **tsxCount runtimes (SCER.100, silver server, `OMP_PROC_BIND=close`)** With few threads the performance of OMP and TSX is similar. With a larger number of threads the performance of TSX becomes best. Strikingly the CAS implementation scales well and outperforms the lock-based variants at 24 threads. The performance of all implementations is similar to using `OMP_PROC_BIND=spread`.

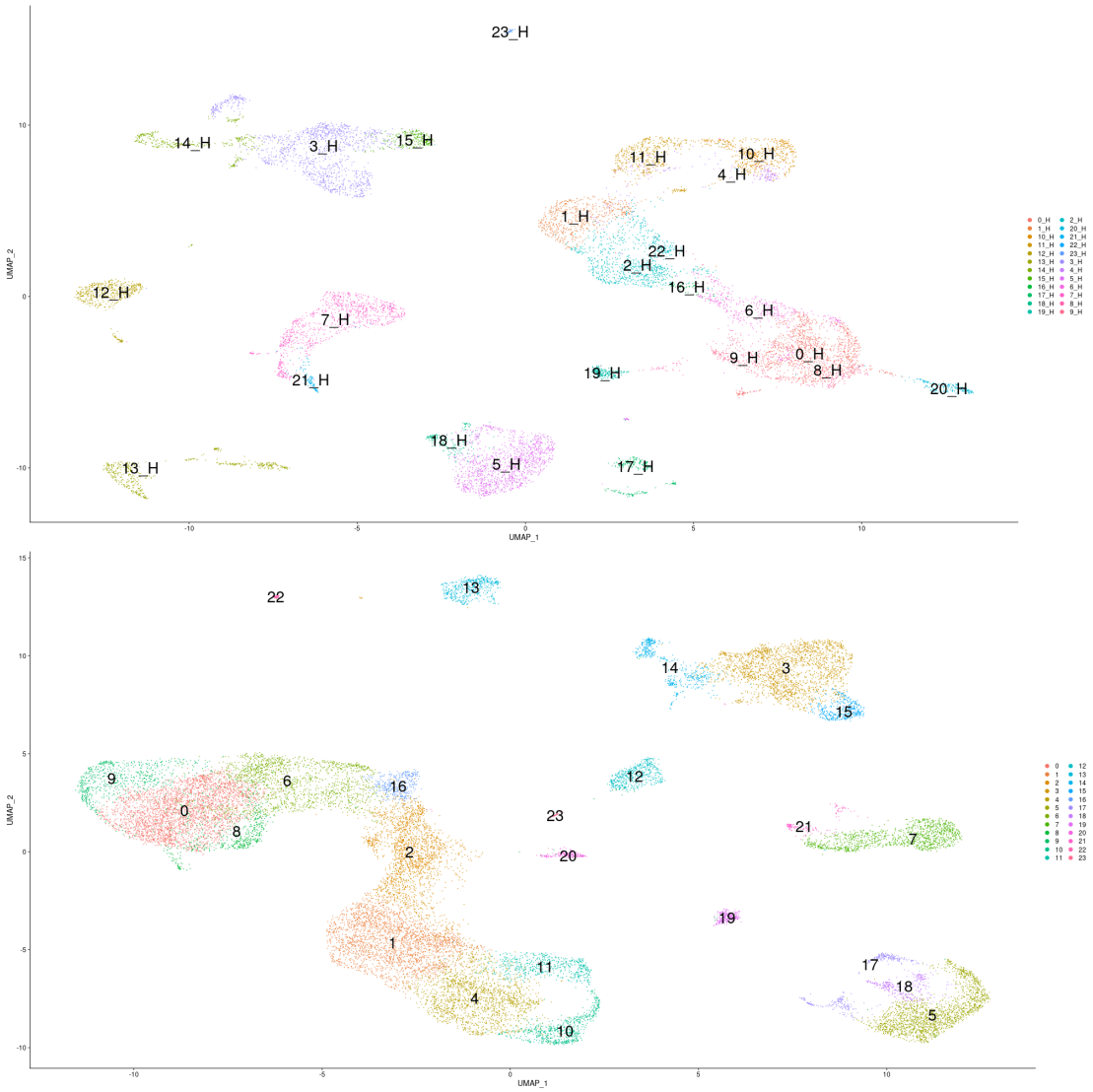
Method/ Threads	1	2	6	12	24	36	42	48
CAS [s]	12324.67	3568.14	1379.34	545.64	304.69	319.21	275.78	253.24
Speed-up	1.00	3.45	8.94	22.59	40.45	38.61	44.69	48.67
Efficiency	1.00	1.73	1.49	1.88	1.69	1.07	1.06	1.01
OMP [s]	3348.43	1125.84	393.04	232.26	748.84	846.91	797.11	877.19
Speed-up	1.00	2.97	8.52	14.42	4.47	3.95	4.20	3.82
Efficiency	1.00	1.49	1.42	1.20	0.19	0.11	0.10	0.08
TSX [s]	3558.31	1163.92	410.36	208.96	243.34	295.86	214.25	218.42
Speed-up	1.00	3.06	8.67	17.03	14.62	12.03	16.61	16.29
Efficiency	1.00	1.53	1.45	1.42	0.61	0.33	0.40	0.34
PTHREAD [s]	3320.98	1110.12	387.69	269.29	643.60	732.45	757.09	789.51
Speed-up	1.00	2.99	8.57	12.33	5.16	4.53	4.39	4.21
Efficiency	1.00	1.50	1.43	1.03	0.22	0.13	0.10	0.09

Supplemental Material, Table A.12: **tsxCount runtimes (SCER.100, silver server, `OMP_PROC_BIND=FALSE`)** With few threads the performance of OMP and TSX is similar. With a larger number of threads the performance of TSX becomes best. Strikingly, the CAS implementation scales well and outperforms the TSX and lock-based variants at 24 threads. The absolute run-times in this mode are higher than using `OMP_PROC_BIND=spread`.

Method/Threads	2	6	12	24	48
CAS [s]	10624.67	1788.53	596.18	341.68	287.76
Speed-up	1.00	5.94	17.82	31.10	36.92
Efficiency	1.00	0.99	1.49	1.30	0.77
OMP [s]	1347.23	519.19	564.77	815.36	963.44
Speed-up	1.00	2.59	2.39	1.65	1.40
Efficiency	1.00	0.43	0.20	0.07	0.03
PTHREAD [s]	1338.51	659.69	621.23	682.34	834.29
Speed-up	1.00	2.03	2.15	1.96	1.60
Efficiency	1.00	0.34	0.18	0.08	0.03
TSX [s]	1326.87	504.17	465.45	350.14	347.36
Speed-up	1.00	2.63	2.85	3.79	3.82
Efficiency	1.00	0.44	0.24	0.16	0.08

A.5 Chapter 4

A.5.2 cPred



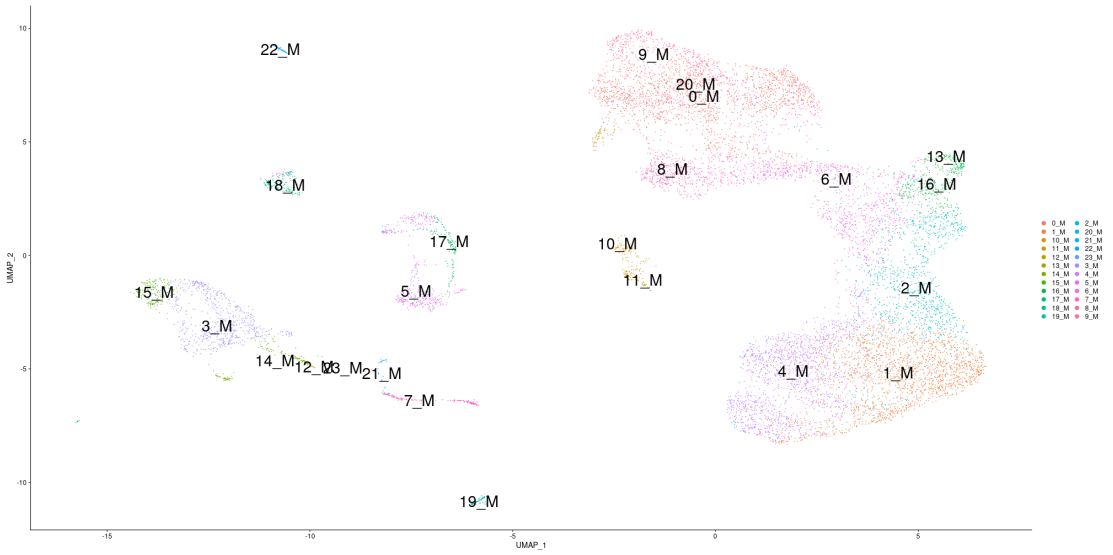
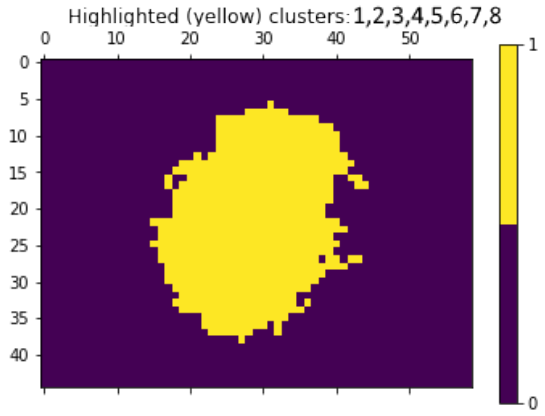
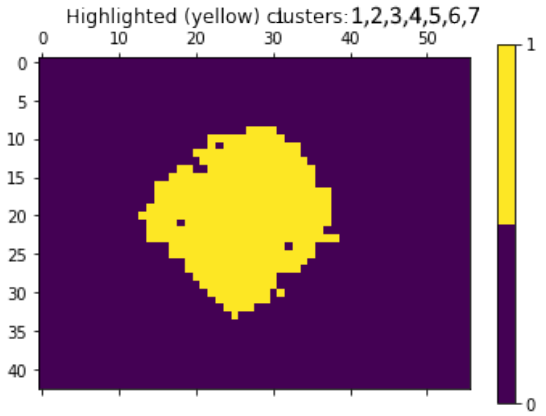


Figure A.34: **UMAP representation of GSE131780 dataset (scRNA-seq analysis of human/mouse plaque)**. The top UMAP shows the clustering of the human dataset only, the bottom one the mouse dataset, respectively. In the middle the integrated, human and mouse dataset is shown.

A.5.4 `pIMZ`



(a) Slide D, Region 0: Selected pixels (yellow).



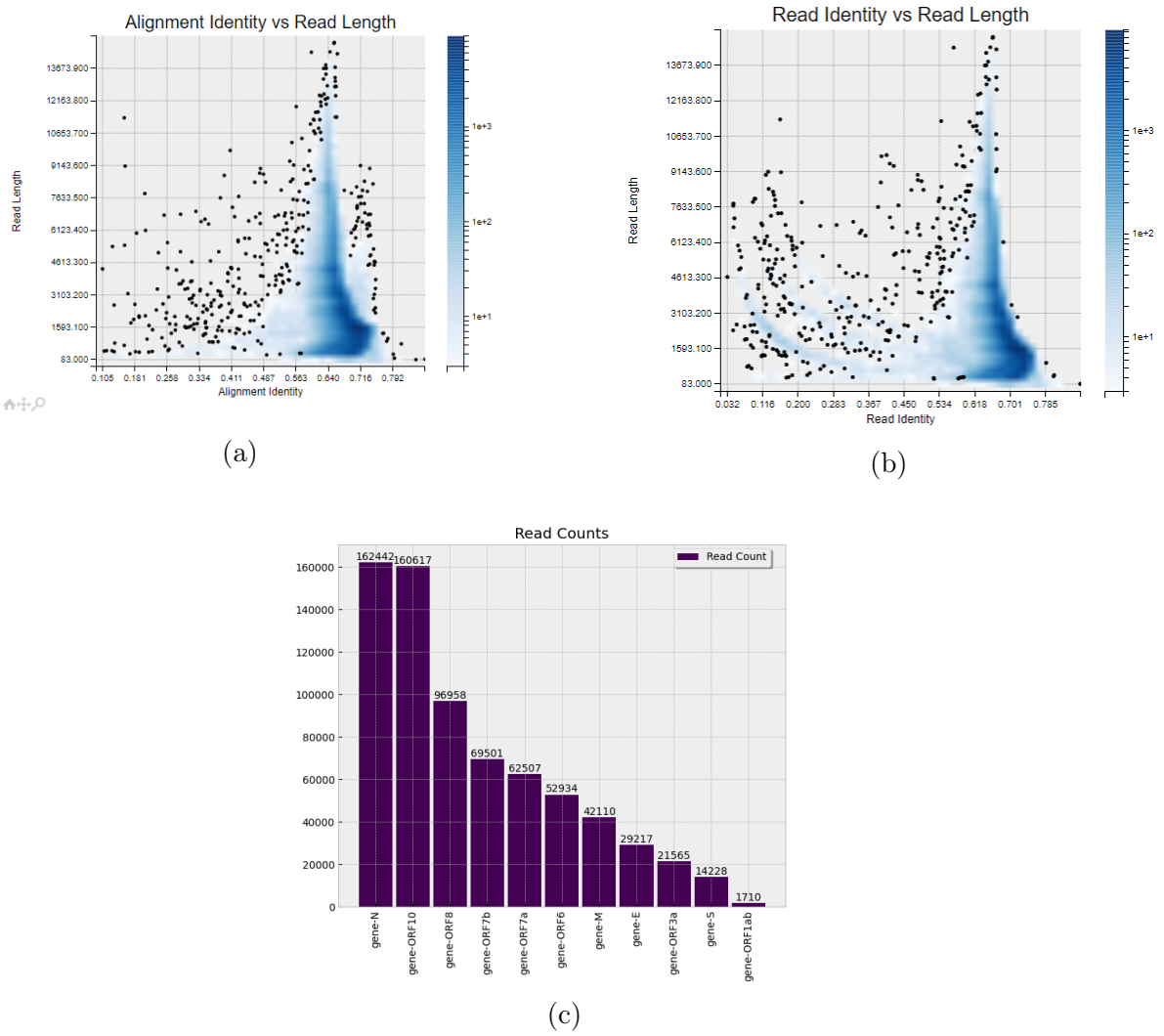
(b) Slide D, Region 1: Selected pixels (yellow).

Figure A.35: **`pIMZ` comparing all clusters (Region 0 vs Region 1)** An analysis over multiple SpectraRegion objects is possible using the CombinedSpectra. All non-background pixels (shown in yellow) of Region 0 (a) are compared with Region 1 (b).

A.6 Chapter 5

A.6.1 poreSTAT

Using the correct SARS-CoV-2 genome with accession number `MT007544.1`, the poreSTAT read alignment analysis yields better results compared to the initial analysis discussed in Chapter 5.1 (Figure A.36). Both, the alignment identity and the read identity improve, suggesting that the reads fit the reference genome better. A read identity of 75% is in scope for the used version 9.4.1 read chemistry. The counts per feature are better distributed, with the nucleocapsid phosphoprotein (gene-N) being the most abundant transcript.



(a) (b)

(c)

Figure A.36: **poreSTAT improved alignment overview** (a) reporting read length versus alignment identity, (b) reporting read length vs. read identity and (c) feature counts for the Australian virus genome (accession `MT007544.1`).

A.6.2 sequ-into

sequ-into was initially developed in the course of the iGEM 2018 competition by the iGEM Team Munich 2018. Active development of the base application was performed by Margaritha Olenchuk and Julia Mayer under the supervision of Markus Joppich. After the competition, *sequ-into* received several important changes implemented by Markus Joppich: the analysis and reporting works in an incremental, online fashion such that only new data must be processed. In contrast to the original application, this required a redesign of the whole backend. Instead of relying on the result from an asynchronous system call, as in the initial version, an asynchronous HTTP request to a locally spawned server has to take place. The manuscript has been prepared by Markus Joppich, with several suggestions by Margaritha Olenchuk and Julia Mayer. Ralf Zimmer, Quirin Emslander and Luisa Jimenez-Soto contributed with general suggestions regarding text and figures. The accepted publication is available as open-access online article `https://doi.org/10.1016/j.csbj.2020.05.014`.

A.7 Chapter 6

A.7.1 Robust Differential Expression

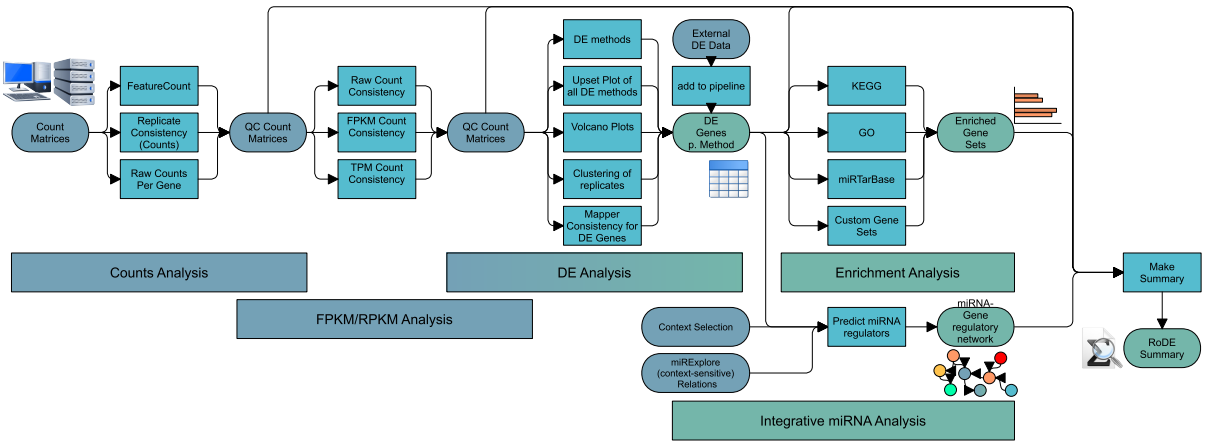
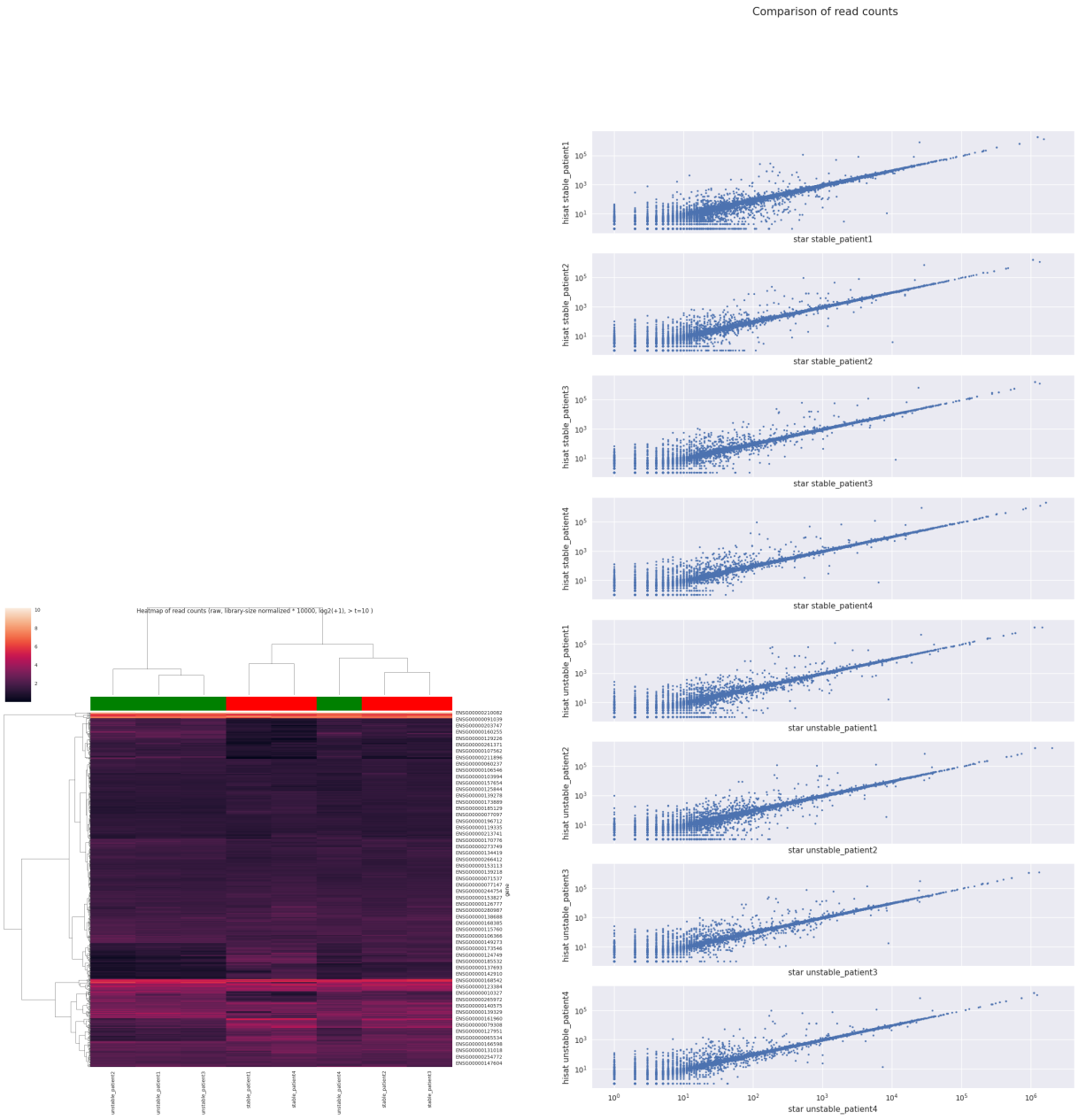


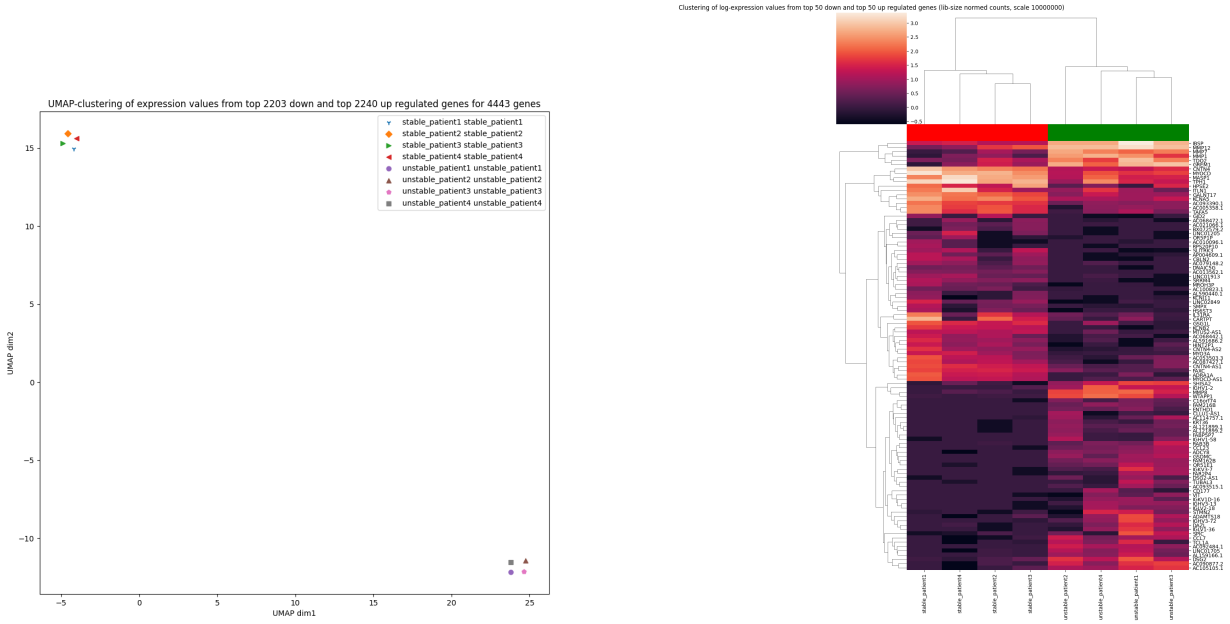
Figure A.37: **RoDE workflow in detail** The workflow of the RoDE pipeline in more detail. In each of the five stages (yellow), several analyses (rectangles) are performed of specified inputs (ellipses). Using the output of the previous stage as input for the next stage nicely demonstrates the pipeline character of RoDE .



(a) Count Heatmap (STAR)

(b) Input Read Count Comparison

Figure A.38: **RoDE robust pipeline (count comparisons)** (a) The count heatmap (library-size normalized $\times 10000$, filtered for genes with > 10 expression, log_2 with pseudo-count 1) can be used to verify how similar the single samples are. It must be noted, that one unstable sample 4 differs from the other unstable samples. (b) The Comparison of the input counts reveals that these are very similar, with some differences in the low-expressed genes.



(a) STAR UMAP all DE

(b) STAR clustermap top 100 DE genes

Figure A.39: **RoDE UMAP and clustermap evaluation on DE genes** After performing DE, the dimensional reduction can also be done only on the differential genes (a). The two groups form distinct clusters of high similarity, which can be expected if the DE analysis was performed correctly. Using a clustermap (b), the top 50 up- and down-regulated genes can be displayed. This again shows that these genes divide the two groups properly.

Figure A.40: **RoDE miRTarBase target gene enrichments** The top enriched gene sets of the custom gene sets are visualized by bar plots. These plots differentiate between considering all genes, all up- or all down-regulated genes. Reported gene sets are sorted such that either UP or DOWN sets are significant.

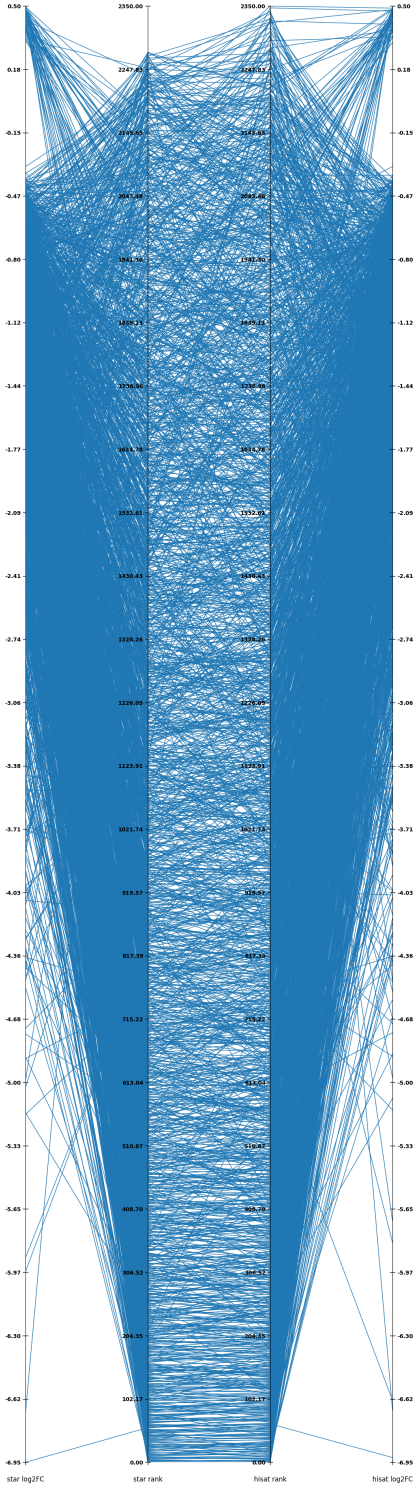


Figure A.41: **RoDE robust pipeline rank plot** The rank plot is a parallel coordinate plot with the logFCs on the outer axes and the rank of the p-value on the inner ones. This plot gives a fast overview of how different the results from to DE results are.

Bibliography

Articles, Proceedings, Books

[1] Afgan, Enis, Baker, Dannon, Beek, Marius van den et al. 'The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update.' In: *Nucleic acids research* 44.W1 (May 2016), W3–W10.

[2] Agarwal, Vikram, Bell, George W., Nam, Jin-Wu et al. 'Predicting effective microRNA target sites in mammalian mRNAs'. In: *eLife* 4.e05005 (Aug. 2015), pp. 1–38.

[3] Agarwala, Richa, Barrett, Tanya, Beck, Jeff et al. 'Database resources of the National Center for Biotechnology Information'. In: *Nucleic Acids Research* 44.D1 (Jan. 2016), pp. D7–D19.

[4] Ahmed, Nauman, Lévy, Jonathan, Ren, Shanshan et al. 'GASAL2: A GPU accelerated sequence alignment library for high-throughput NGS data'. In: *BMC Bioinformatics* 20.1 (Oct. 2019), p. 520.

[5] Alberts, B. *Molecular Biology of the Cell*. W.W. Norton, 2017. ISBN: 9781317563754.

[6] Ameres, Stefan L. and Zamore, Phillip D. 'Diversifying microRNA sequence and function'. In: *Nature Reviews Molecular Cell Biology* 14.8 (Aug. 2013), pp. 475–488.

[7] Ammar, Constantin, Gruber, Markus, Csaba, Gergely et al. 'MS-EmpiRe utilizes peptide-level noise distributions for ultra-sensitive detection of differentially expressed proteins'. In: *Molecular and Cellular Proteomics* 18.9 (Sept. 2019), pp. 1880–1892.

[8] An, Omer, Tan, Kar Tong, Li, Ying et al. 'CSI NGS portal: An online platform for automated NGS data analysis and sharing'. In: *International Journal of Molecular Sciences* 21.11 (May 2020), p. 3828.

[9] Anand, Praveen, Puranik, Arjun, Aravamudan, Murali et al. 'SARS-CoV-2 strategically mimics proteolytic activation of human ENaC'. In: *eLife* 9.e58603 (May 2020).

[10] Anders, Simon and Huber, Wolfgang. 'Differential expression analysis for sequence count data'. In: *Genome Biology* 11.10 (Oct. 2010), R106.

[11] Anders, Simon, Pyl, Paul Theodor and Huber, Wolfgang. 'HTSeq-A Python framework to work with high-throughput sequencing data'. In: *Bioinformatics* 31.2 (Jan. 2015), pp. 166–169.

[12] Angerer, Philipp, Simon, Lukas, Tritschler, Sophie et al. 'Single cells make big data: New challenges and opportunities in transcriptomics'. In: *Current Opinion in Systems Biology* 4 (Aug. 2017), pp. 85–91.

[13] Aran, Dvir, Looney, Agnieszka P., Liu, Leqian et al. 'Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage'. In: *Nature Immunology* 20.2 (Feb. 2019), pp. 163–172.

[14] Argelaguet, Ricard, Velten, Britta, Arnol, Damien et al. 'Multi-Omics Factor Analysis - a framework for unsupervised integration of multi-omics data sets'. In: *Molecular Systems Biology* 14.6 (June 2018).

[15] Ashburner, Michael, Ball, Catherine A., Blake, Judith A. et al. 'Gene Ontology: tool for the unification of biology'. In: *Nat Genet* 25.1 (May 2000), pp. 25–29.

[16] Athar, Awais, Füllgrabe, Anja, George, Nancy et al. 'ArrayExpress update - From bulk to single-cell expression data'. In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D711–D715.

[17] Attwood, Teresa K., Blackford, Sarah, Brazas, Michelle D. et al. 'A global perspective on evolving bioinformatics and data science training needs'. In: *Briefings in Bioinformatics* 20.2 (Mar. 2019), pp. 398–404.

[18] Baccin, Chiara, Al-Sabah, Jude, Velten, Lars et al. 'Combined single-cell and spatial transcriptomics reveal the molecular, cellular and spatial bone marrow niche organization'. In: *Nature Cell Biology* 22.1 (Jan. 2020), pp. 38–48.

[19] Bagewadi, Shweta, Bobić, Tamara, Hofmann-Apitius, Martin et al. 'Detecting miRNA Mentions and Relations in Biomedical Literature'. In: *F1000Research* 3 (Oct. 2015), p. 205.

[20] Baker, Monya and Penny, Dan. 'Is there a reproducibility crisis?' In: *Nature* 533.7604 (May 2016), pp. 452–454.

[21] Bankevich, Anton, Nurk, Sergey, Antipov, Dmitry et al. 'SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing.' In: *Journal of computational biology : a journal of computational molecular cell biology* 19.5 (May 2012), pp. 455–77.

[22] Barrett, Tanya, Wilhite, Stephen E., Ledoux, Pierre et al. 'NCBI GEO: Archive for functional genomics data sets - Update'. In: *Nucleic Acids Research* 41.D1 (Jan. 2013), pp. D991–D995.

[23] Bemis, Kyle D., Harry, April, Eberlin, Livia S. et al. 'Cardinal: An R package for statistical analysis of mass spectrometry-based imaging experiments'. In: *Bioinformatics* 31.14 (July 2015), pp. 2418–2420.

[24] Benjamini, Yoav and Hochberg, Yosef. 'Controlling the False Discovery Rate : A Practical and Powerful Approach to Multiple Testing'. In: *Journal of the Royal Statistical Society. Series B (Methodological) Journal of the Royal Statistical Society* 57.1 (Dec. 1995), pp. 289–300.

[25] Bergenstråhle, Joseph, Larsson, Ludvig and Lundeberg, Joakim. 'Seamless integration of image and molecular analysis for spatial transcriptomics workflows'. In: *BMC Genomics* 21.1 (July 2020), p. 482.

[26] Bernardo, Bianca C, Ooi, Jenny YY, Lin, Ruby CY et al. 'miRNA therapeutics: A new class of drugs with potential therapeutic applications in the heart'. In: *Future Medicinal Chemistry* 7.13 (Sept. 2015), pp. 1771–1792.

[27] Betel, Doron, Wilson, Manda, Gabow, Aaron et al. 'The microRNA.org resource: targets and expression'. In: *Nucleic Acids Research* 36.suppl_1 (Jan. 2008), pp. D149–D153.

[28] Bobryshev, Yuri V. 'Transdifferentiation of smooth muscle cells into chondrocytes in atherosclerotic arteries in situ: Implications for diffuse intimal calcification'. In: *Journal of Pathology* 205.5 (Apr. 2005), pp. 641–650.

[29] Bokhart, Mark T., Nazari, Milad, Garrard, Kenneth P. et al. 'MSiReader v1.0: Evolving Open-Source Mass Spectrometry Imaging Software for Targeted and Untargeted Analyses'. In: *Journal of the American Society for Mass Spectrometry* 29.1 (2018), pp. 8–16.

[30] Bolchini, Davide, Finkelstein, Anthony, Perrone, Vito et al. 'Better bioinformatics through usability analysis'. In: *Bioinformatics* 25.3 (Feb. 2009), pp. 406–412.

[31] Boldogkői, Zsolt, Moldován, Norbert, Balázs, Zsolt et al. 'Long-Read Sequencing - A Powerful Tool in Viral Transcriptome Research'. In: *Trends in Microbiology* 27.7 (July 2019), pp. 578–592.

[32] Bolger, Anthony, Denton, Alisandra, Bolger, Marie et al. 'LOGAN : A framework for LOssless Graph-based ANalysis of high throughput sequence data'. In: *bioRxiv* (Aug. 2017), pp. 1–13.

[33] Bolger, Anthony M, Lohse, Marc and Usadel, Bjoern. 'Trimmomatic: a flexible trimmer for Illumina sequence data'. In: *Bioinformatics* 30.15 (Apr. 2014), pp. 2114–2120.

[34] Bost, Pierre, Giladi, Amir, Liu, Yang et al. 'Host-Viral Infection Maps Reveal Signatures of Severe COVID-19 Patients'. In: *Cell* 181.7 (June 2020), 1475–1488.e12.

[35] Boué, Stéphanie, Talikka, Marja, Westra, Jurjen Willem et al. 'Causal biological network database: a comprehensive platform of causal biological network models focused on the pulmonary and vascular systems'. In: *Database (Oxford)* 2015 (Jan. 2015).

[36] Braschi, Bryony, Denny, Paul, Gray, Kristian et al. 'Genenames.org: The HGNC and VGNC resources in 2019'. In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D786–D792.

[37] Breitwieser, F. P., Baker, D. N. and Salzberg, S. L. 'KrakenUniq: Confident and fast metagenomics classification using unique k-mer counts'. In: *Genome Biology* 19.1 (Nov. 2018), p. 198.

[38] Brouwer, R. W.W., Van den hout, M. C.G.N., Grosveld, F. G. et al. 'NARWHAL, a primary analysis pipeline for NGS data'. In: *Bioinformatics* 28.2 (Jan. 2012), pp. 284–285.

[39] Bruen, Robyn, Fitzsimons, Stephen and Belton, Orina. 'MiR-155 in the resolution of atherosclerosis'. In: *Frontiers in Pharmacology* 10.MAY (May 2019), p. 463.

[40] Bruford, Elspeth A., Braschi, Bryony, Denny, Paul et al. 'Guidelines for human gene nomenclature'. In: *Nature Genetics* 52.8 (Aug. 2020), pp. 754–758.

[41] Bult, Carol J., Blake, Judith A., Smith, Cynthia L. et al. 'Mouse Genome Database (MGD) 2019'. In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D801–D806.

[42] Butcher, Matthew J. and Galkina, Elena V. 'Phenotypic and functional heterogeneity of macrophages and dendritic cell subsets in the healthy and atherosclerosis-prone aorta'. In: *Frontiers in Physiology* 3 MAR (Mar. 2012), p. 44.

[43] Butler, Andrew, Hoffman, Paul, Smibert, Peter et al. 'Integrating single-cell transcriptomic data across different conditions, technologies, and species'. In: *Nature Biotechnology* 36.5 (May 2018), pp. 411–420.

[44] Caminati, Marco, Pham, Duy Le, Bagnasco, Diego et al. 'Type 2 immunity in asthma'. In: *World Allergy Organization Journal* 11.1 (June 2018).

[45] Campello, Ricardo J.G.B., Moulavi, Davoud and Sander, Joerg. 'Density-based clustering based on hierarchical density estimates'. In: *Lecture Notes in Computer Science*. Vol. 7819 LNAI. PART 2. Springer, Berlin, Heidelberg, 2013, pp. 160–172.

[46] Cao, Jing and Zhang, Song. 'A Bayesian extension of the hypergeometric test for functional enrichment analysis'. In: *Biometrics* 70.1 (Mar. 2014), pp. 84–94.

[47] Carbon, S., Dietze, H., Lewis, S. E. et al. 'Expansion of the gene ontology knowledgebase and resources: The gene ontology consortium'. In: *Nucleic Acids Research* 45.D1 (Jan. 2017), pp. D331–D338.

[48] Carbon, Seth, Ireland, Amelia, Mungall, Christopher J. et al. 'AmiGO: Online access to ontology and annotation data'. In: *Bioinformatics* 25.2 (Jan. 2009), pp. 288–289.

[49] Cardoso, Fatima, Veer, Laura J. van't, Bogaerts, Jan et al. '70-Gene Signature as an Aid to Treatment Decisions in Early-Stage Breast Cancer'. In: *New England Journal of Medicine* 375.8 (Aug. 2016), pp. 717–729.

[50] Carriço, J. A., Rossi, M., Moran-Gilad, J. et al. 'A primer on microbial bioinformatics for nonbioinformaticians'. In: *Clinical Microbiology and Infection* 24.4 (Apr. 2018), pp. 342–349.

[51] Castro-Wallace, Sarah L., Chiu, Charles Y., John, Kristen K. et al. 'Nanopore DNA Sequencing and Genome Assembly on the International Space Station'. In: *Scientific Reports* 7.1 (Dec. 2017), pp. 1–12.

[52] Chang, Ting Ting, Yang, Hsin Ying, Chen, Ching et al. 'CCL4 inhibition in atherosclerosis: Effects on plaque stability, endothelial cell adhesiveness, and macrophages activation'. In: *International Journal of Molecular Sciences* 21.18 (Sept. 2020), pp. 1–19.

[53] Barbara Chapman, Weiming Zheng, Guang R. Gao et al., eds. *A Practical Programming Model for the Multi-Core Era*. Vol. 4935. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-69302-4.

[54] Chen, Xiaona, He, Yanhong, Fu, Wenjun et al. 'Histone Deacetylases (HDACs) and Atherosclerosis: A Mechanistic and Pharmacological Review'. In: *Frontiers in Cell and Developmental Biology* 8 (Nov. 2020), p. 581015.

[55] Chen, Xing, Xie, Di, Wang, Lei et al. 'BNPMDA: Bipartite network projection for MiRNA–Disease association prediction'. In: *Bioinformatics* 34.18 (Sept. 2018). Ed. by Alfonso Valencia, pp. 3178–3186.

[56] Chibucos, Marcus C., Siegele, Deborah A., Hu, James C. et al. 'The evidence and conclusion ontology (ECO): Supporting GO annotations'. In: *Methods in Molecular Biology*. Vol. 1446. Humana Press Inc., 2017, pp. 245–259.

[57] Cho, Sooyoung, Jang, Insu, Jun, Yukyung et al. 'miRGator v3.0: a microRNA portal for deep sequencing, expression profiling and mRNA targeting'. In: *Nucleic Acids Research* 41.D1 (2012), pp. D252–D257.

[58] Chou, Chih Hung, Shrestha, Sirjana, Yang, Chi Dung et al. 'MiRTarBase update 2018: A resource for experimentally validated microRNA-target interactions'. In: *Nucleic Acids Research* 46.D1 (Jan. 2018), pp. D296–D302.

[59] Choudhury, Sagnik Ray, Wang, Shuting and Giles, C. Lee. 'Scalable algorithms for scholarly figure mining and semantics'. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. New York, New York, USA: Association for Computing Machinery, June 2016, pp. 1–6.

[60] Chow, Yit-Lai, Teh, Lai Kuan, Chyi, Loh Huey et al. 'Lipid Metabolism Genes in Stroke Pathogenesis: The Atherosclerosis'. In: *Current Pharmaceutical Design* 26.34 (2020), pp. 4261–4271.

[61] Cock, Peter J.A., Antao, Tiago, Chang, Jeffrey T. et al. 'Biopython: Freely available Python tools for computational molecular biology and bioinformatics'. In: *Bioinformatics* 25.11 (June 2009), pp. 1422–1423.

[62] Coelho, Ana L., Schaller, Matthew A., Benjamim, Claudia F. et al. 'The Chemokine CCL6 Promotes Innate Immunity via Immune Cell Activation and Recruitment'. In: *The Journal of Immunology* 179.8 (Oct. 2007), pp. 5474–5482.

[63] Conesa, Ana, Madrigal, Pedro, Tarazona, Sonia et al. 'A survey of best practices for RNA-seq data analysis'. In: *Genome Biology* 17.1 (Jan. 2016), pp. 1–19.

[64] D'Antonio, Mattia, De Meo, Paolo D.Onorio, Pallocca, Matteo et al. 'RAP: RNA-Seq Analysis Pipeline, a new cloud-based NGS web application'. In: *BMC Genomics* 16.6 (June 2015), pp. 1–11.

[65] Dagum, L. and Menon, R. 'OpenMP: an industry standard API for shared-memory programming'. In: *IEEE Computational Science and Engineering* 5.1 (Jan. 1998), pp. 46–55.

[66] Dale, Ryan, Grüning, Björn, Sjödin, Andreas et al. 'Bioconda: Sustainable and comprehensive software distribution for the life sciences'. In: *Nature Methods* 15.7 (July 2018), pp. 475–476.

[67] Delcher, Arthur L, Bratke, Kirsten A, Powers, Edwin C et al. 'Identifying bacterial genes and endosymbiont DNA with Glimmer.' In: *Bioinformatics* 23.6 (Mar. 2007), pp. 673–9.

[68] Denisov, Gennady, Walenz, Brian, Halpern, Aaron L et al. 'Consensus generation and variant detection by Celera Assembler.' In: *Bioinformatics* 24.8 (Apr. 2008), pp. 1035–40.

[69] Dennis, Glynn, Sherman, Brad T., Hosack, Douglas A. et al. 'Genome-wide distribution of 5-formylcytosine in embryonic stem cells is associated with transcription and depends on thymine DNA glycosylase'. In: *Genome Biology* 4.5 (May 2003), P3.

[70] Deonier, Richard C., Tavaré, Simon and Waterman, Michael S. *Computational Genome Analysis*. New York, NY: Springer New York, 2005.

[71] Deorowicz, Sebastian, Kokot, Marek, Grabowski, Szymon et al. 'KMC 2: Fast and resource-frugal k-mer counting'. In: *Bioinformatics* 31.10 (2015), pp. 1569–1576.

[72] Devlin, Jacob, Chang, Ming Wei, Lee, Kenton et al. 'BERT: Pre-training of deep bidirectional transformers for language understanding'. In: *2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. June 2019, pp. 4171–4186.

[73] Di Bartolo, Belinda A., Psaltis, Peter J., Bursill, Christina A. et al. 'Translating evidence of HDL and plaque regression'. In: *Arteriosclerosis, Thrombosis, and Vascular Biology* 38.9 (July 2018), pp. 1961–1968.

[74] DI Tommaso, Paolo, Chatzou, Maria, Floden, Evan W. et al. 'Nextflow enables reproducible computational workflows'. In: *Nature Biotechnology* 35.4 (Apr. 2017), pp. 316–319.

[75] Dobin, Alexander, Davis, Carrie A, Schlesinger, Felix et al. 'STAR: ultrafast universal RNA-seq aligner.' In: *Bioinformatics* 29.1 (Jan. 2013), pp. 15–21.

[76] Doerr, Allison. 'Mass spectrometry imaging takes off'. In: *Nature Methods* 15.1 (Jan. 2018), p. 32.

[77] Donaldson, Callum J., Lao, Ka Hou and Zeng, Lingfang. 'The salient role of microRNAs in atherogenesis'. In: *J Mol Cell Cardiol* 122 (Sept. 2018), pp. 98–113.

[78] Dugas, Martin, Weninger, F., Merk, S. et al. 'A generic concept for large-scale microarray analysis dedicated to medical diagnostics'. In: *Methods of Information in Medicine* 45.2 (2006), pp. 146–152.

[79] Duhan, Naveen and Kaundal, Rakesh. 'pySeqRNA: An automated Python package for RNA sequencing data analysis'. In: *Poster 28th International Conference on Intelligent Systems for Molecular Biology (ISMB) 2020*. 2020.

[80] Dweep, Harsh and Gretz, Norbert. 'miRWalk2.0: a comprehensive atlas of microRNA-target interactions'. In: *Nature Methods* 12.8 (Aug. 2015), p. 697.

[81] Ebbert, Mark T.W., Jensen, Tanner D., Jansen-West, Karen et al. 'Systematic analysis of dark and camouflaged genes reveals disease-relevant genes hiding in plain sight'. In: *Genome Biology* 20.1 (May 2019), pp. 1–23.

[82] Edgar, Ron, Domrachev, Michael and Lash, Alex E. 'Gene Expression Omnibus: NCBI gene expression and hybridization array data repository'. In: *Nucleic Acids Research* 30.1 (Jan. 2002), pp. 207–210.

[83] Editorial Nature Methods. 'Method of the Year 2019: Single-cell multimodal omics'. In: *Nature methods* 17.1 (Jan. 2020), p. 1.

[84] Editorial Nature Methods. 'Method of the Year 2020: spatially resolved transcriptomics'. In: *Nature Methods* 18.1 (Jan. 2021), pp. 1–1.

[85] Egholm, Cecilie, Heeb, Lukas E.M., Impellizzieri, Daniela et al. 'The regulatory effects of interleukin-4 receptor signaling on neutrophils in type 2 immune responses'. In: *Frontiers in Immunology* 10.OCT (Oct. 2019), p. 2507.

[86] Eliasson, Lena, Poy, Matthew N., MacDonald, Patrick E. et al. 'A pancreatic islet-specific microRNA regulates insulin secretion'. In: *Nature* 432.7014 (Nov. 2004), pp. 226–230.

[87] Elliott, Brendan, Kirac, Mustafa, Cakmak, Ali et al. 'PathCase: Pathways database system'. In: *Bioinformatics* 24.21 (Nov. 2008), pp. 2526–2533.

[88] Erhard, Florian, Haas, Jürgen, Lieber, Diana et al. 'Widespread context dependency of microRNA-mediated regulation'. In: *Genome Res* 24.6 (June 2014), pp. 906–19.

[89] Fernandez, Dawn M., Rahman, Adeeb H., Fernandez, Nicolas F. et al. 'Single-cell immune landscape of human atherosclerotic plaques'. In: *Nature Medicine* 25.10 (Oct. 2019), pp. 1576–1588.

[90] Fernandez-Cuesta, Lynnette, Sun, Ruping, Menon, Roopika et al. 'Identification of novel fusion genes in lung cancer using breakpoint assembly of transcriptome sequencing data'. In: *Genome Biology* 16.1 (Jan. 2015), p. 7.

[91] Fernández-Ruiz, Irene. 'Atherosclerosis: A new role for lncRNAs in atherosclerosis'. In: *Nature Reviews Cardiology* 15.4 (Mar. 2018), p. 195.

[92] Föll, Melanie Christine, Moritz, Lennart, Wollmann, Thomas et al. 'Accessible and reproducible mass spectrometry imaging data analysis in Galaxy'. In: *GigaScience* 8.12 (Dec. 2019).

[93] Frankish, Adam, Diekhans, Mark, Ferreira, Anne Maud et al. 'GENCODE reference annotation for the human and mouse genomes'. In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D766–D773.

[94] Franzén, Oscar, Gan, Li Ming and Björkegren, Johan L.M. 'PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data'. In: *Database : the journal of biological databases and curation* 2019 (Jan. 2019).

[95] Fuior, Elena V. and Gafencu, Anca V. 'Apolipoprotein c1: Its pleiotropic effects in lipid metabolism and beyond'. In: *International Journal of Molecular Sciences* 20.23 (Nov. 2019), pp. 1–25.

[96] Fundel, Katrin, Küffner, Robert and Zimmer, Ralf. 'RelEx - Relation extraction using dependency parse trees'. In: *Bioinformatics* 23.3 (2006), pp. 365–371.

[97] Furge, Laura Lowe, Stevens-Truss, Regina, Moore, D. Blaine et al. 'Vertical and horizontal integration of bioinformatics education: A modular, interdisciplinary approach'. In: *Biochemistry and Molecular Biology Education* 37.1 (2009), pp. 26–36.

[98] Fury, Wen, Batliwalla, Franak, Gregersen, Peter K. et al. 'Overlapping probabilities of top ranking gene lists, hypergeometric distribution, and stringency of gene selection criterion'. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*. 2006, pp. 5531–5534.

[99] Geistlinger, Ludwig, Csaba, Gergely and Zimmer, Ralf. 'Bioconductor's EnrichmentBrowser: Seamless navigation through combined results of set- & network-based enrichment analysis'. In: *BMC Bioinformatics* 17.1 (Jan. 2016), p. 45.

[100] Gencer, Selin, Evans, Bryce R., Vorst, Emiel P.C. van der et al. 'Inflammatory Chemokines in Atherosclerosis'. In: *Cells* 10.2 (Jan. 2021), p. 226.

[101] Gentleman, Robert and Falcon, Seth. 'Hypergeometric Testing Used for Gene Set Enrichment Analysis'. In: *Bioconductor Case Studies* (2008).

[102] Giardine, Belinda, Riemer, Cathy, Hardison, Ross C et al. 'Galaxy: a platform for interactive large-scale genome analysis.' In: *Genome research* 15.10 (Oct. 2005), pp. 1451–5.

[103] Gibb, Sebastian and Strimmer, Korbinian. 'Maldiquant: A versatile R package for the analysis of mass spectrometry data'. In: *Bioinformatics* 28.17 (Sept. 2012), pp. 2270–2271.

[104] Gieseck, Richard L., Wilson, Mark S. and Wynn, Thomas A. 'Type 2 immunity in tissue repair and fibrosis'. In: *Nature Reviews Immunology* 18.1 (Jan. 2018), pp. 62–76.

[105] Glass, Kimberly and Girvan, Michelle. 'Finding New Order in Biological Functions from the Network Structure of Gene Annotations'. In: *PLOS Computational Biology* 11.11 (Nov. 2015). Ed. by Lilia M. Iakoucheva, e1004565.

[106] Goicuria, Haize. 'Smooth muscle cell characterization and transcriptomic analysis in human carotid atherosclerotic plaques'. PhD thesis. Universidad del País Vasco - Euskal Herriko Unibertsitatea, 2018.

[107] Golik, Wiktoria, Dameron, Olivier, Bugeon, Jérôme et al. 'ATOL: The multi-species livestock trait ontology'. In: *Communications in Computer and Information Science*. Vol. 343 CCIS. Springer, Berlin, Heidelberg, Nov. 2012, pp. 289–300.

[108] Greißel, Anna, Culmes, Mihaela, Napieralski, Rudolf et al. 'Alternation of histone and DNA methylation in human atherosclerotic carotid plaques'. In: *Thrombosis and Haemostasis* 114.2 (Nov. 2015), pp. 390–402.

[109] Griffin, Philippa C., Khadake, Jyoti, LeMay, Kate S. et al. 'Best practice data life cycle approaches for the life sciences'. In: *F1000Research* 6 (Aug. 2017), p. 1618.

[110] Griffiths-Jones, Sam, Saini, Harpreet Kaur, Van Dongen, Stijn et al. 'miRBase: Tools for microRNA genomics'. In: *Nucleic Acids Research* 36.SUPPL. 1 (Jan. 2008), pp. D154–D158.

[111] Gruber, Thomas R. 'Toward principles for the design of ontologies used for knowledge sharing'. In: *International Journal of Human - Computer Studies* 43.5-6 (Nov. 1995), pp. 907–928.

[112] Han, Xiaoyu and Wang, Lei. 'A novel document-level relation extraction method based on BERT and entity information'. In: *IEEE Access* 8 (2020), pp. 96912–96919.

[113] Han, Yixing, Gao, Shouguo, Muegge, Kathrin et al. 'Advanced Applications of RNA Sequencing and Challenges.' en. In: *Bioinformatics and biology insights* 9.Suppl 1 (Jan. 2015), pp. 29–46.

[114] Hanisch, Daniel, Fundel, Katrin, Mevissen, Heinz Theodor et al. 'ProMiner: Rule-based protein and gene entity recognition'. In: *BMC Bioinformatics* 6.SUPPL.1 (2005), pp. 1–9.

[115] Hannay, Mike. 'Pragmatic function assignment and word order variation in a functional grammar of English'. In: *Journal of Pragmatics* 16.2 (1991), pp. 131–155.

[116] Harris, Charles R., Millman, K. Jarrod, Walt, Stéfan J. van der et al. 'Array programming with NumPy'. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362.

[117] Hartel, Frank W., De Coronado, Sherri, Dionne, Robert et al. 'Modeling a description logic vocabulary for cancer research'. In: *J Biomed Inform* 38.2 (Apr. 2005), pp. 114–129.

[118] Hartmann, Petra, Schober, Andreas and Weber, Christian. 'Chemokines and microRNAs in atherosclerosis'. In: *Cellular and Molecular Life Sciences* 72.17 (May 2015), pp. 3253–3266.

[119] He, Ping Ping, OuYang, Xin Ping, Li, Yuan et al. 'MicroRNA-590 inhibits lipoprotein lipase expression and prevents atherosclerosis in apoE knockout mice'. In: *PLoS ONE* 10.9 (Sept. 2015). Ed. by Kottarappat N Dileepan, e0138788.

[120] Heng, Tracy S.P., Painter, Michio W., Elpek, Kutlu et al. 'The immunological genome project: Networks of gene expression in immune cells'. In: *Nature Immunology* 9.10 (2008), pp. 1091–1094.

[121] Heuveline, Vincent, Janko, Sven, Karl, Wolfgang et al. 'Software transactional memory, OpenMP and Pthread implementations of the conjugate gradients method - A preliminary evaluation'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 7851 LNCS. Springer, Berlin, Heidelberg, 2013, pp. 300–313.

[122] Hicks, Stephanie C., Townes, F. William, Teng, Mingxiang et al. 'Missing data and technical variability in single-cell RNA-sequencing experiments'. In: *Biostatistics* 19.4 (Oct. 2018), pp. 562–578.

[123] Ho, Joses, Tumkaya, Tayfun, Aryal, Sameer et al. 'Moving beyond P values: data analysis with estimation graphics'. In: *Nature Methods* 16.7 (July 2019), pp. 565–566.

[124] Hortells, Luis, Sur, Swastika and St. Hilaire, Cynthia. 'Cell Phenotype Transitions in Cardiovascular Calcification'. In: *Frontiers in Cardiovascular Medicine* 5 (Mar. 2018), p. 27.

[125] Hozza, Michal, Vinař, Tomáš and Brejová, Broňa. 'How big is that genome? Estimating genome size and coverage from k-mer abundance spectra'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9309 (Sept. 2015), pp. 199–209.

[126] Hsu, Sheng-Da, Chu, Chia-Huei, Tsou, Ann-Ping et al. 'miRNAMap 2.0: genomic maps of microRNAs in metazoan genomes'. In: *Nucleic Acids Research* 36.suppl_1 (Nov. 2007), pp. D165–D169.

[127] Hua, Lei and Quan, Chanqin. 'A Shortest Dependency Path Based Convolutional Neural Network for Protein-Protein Relation Extraction'. In: *BioMed Research International* 2016 (June 2016).

[128] Huang, Hsi-Yuan, Lin, Yang-Chi-Dung, Li, Jing et al. 'miRTarBase 2020: updates to the experimentally validated microRNA-target interaction database'. In: *Nucleic Acids Research* 48.D1 (Oct. 2019), pp. D148–D154.

[129] Huang, Ri Sheng, Hu, Guan Qiong, Lin, Bin et al. 'Microrna-155 silencing enhances inflammatory response and lipid uptake in oxidized low-density lipoprotein-stimulated human THP-1 macrophages'. In: *Journal of Investigative Medicine* 58.8 (Dec. 2010), pp. 961–967.

[130] Hubmap Consortium. 'The human body at cellular resolution: the NIH Human Biomolecular Atlas Program'. In: *Nature* 574.7777 (Oct. 2019), pp. 187–192.

[131] Hunt, Martin, Silva, Nishadi De, Otto, Thomas D. et al. 'Circlator: Automated circularization of genome assemblies using long sequencing reads'. In: *Genome Biology* 16.1 (Dec. 2015), p. 294.

[132] Hunter, John D. 'Matplotlib: A 2D graphics environment'. In: *Computing in Science and Engineering* 9.3 (May 2007), pp. 90–95.

[133] Imai, Kazuo, Tarumoto, Norihito, Misawa, Kazuhisa et al. 'A novel diagnostic method for malaria using loop-mediated isothermal amplification (LAMP) and MinION™ nanopore sequencer'. In: *BMC Infectious Diseases* 17.1 (Sept. 2017), p. 621.

[134] Insull, William. 'The Pathology of Atherosclerosis: Plaque Development and Plaque Responses to Medical Treatment'. In: *American Journal of Medicine* 122.1 SUPPL. (Jan. 2009), S3–S14.

[135] Jackson, Ampadu Okyere, Regine, Mugwaneza Annick, Subrata, Chakrabarti et al. 'Molecular mechanisms and genetic regulation in atherosclerosis'. In: *Int J Cardiol Heart Vasc* 21 (Dec. 2018), pp. 36–44.

[136] Jassal, Bijay, Matthews, Lisa, Viteri, Guilherme et al. 'The reactome pathway knowledgebase'. In: *Nucleic Acids Research* 48.D1 (Jan. 2020), pp. D498–D503.

[137] Ji, Bo Ya, You, Zhu Hong, Cheng, Li et al. 'Predicting miRNA-disease association from heterogeneous information network with GraRep embedding model'. In: *Scientific Reports* 10.1 (Dec. 2020), pp. 1–12.

[138] Jiang, Qinghua, Wang, Yadong, Hao, Yangyang et al. 'miR2Disease: a manually curated database for microRNA deregulation in human disease'. In: *Nucleic Acids Research* 37.suppl_1 (Oct. 2008), pp. D98–D104.

[139] Jiang, W., Zhang, Y., Meng, F. et al. 'Identification of active transcription factor and miRNA regulatory pathways in Alzheimer's disease'. In: *Bioinformatics* 29.20 (Oct. 2013), pp. 2596–2602.

[140] Johnson, Jason L. 'Elucidating the contributory role of microRNA to cardiovascular diseases (a review)'. In: *Vascular Pharmacology* (Oct. 2018).

[141] Jones, Jessica E.C., Rabhi, Nabil, Orofino, Joseph et al. 'The Adipocyte Acquires a Fibroblast-Like Transcriptional Signature in Response to a High Fat Diet'. In: *Scientific Reports* 10.1 (Dec. 2020), pp. 1–15.

[142] Joppich, Markus, Olenchuk, Margaryta, Mayer, Julia et al. 'SEQU-INTO: Early detection of impurities, contamination and off-targets (ICOs) in long read/MinION sequencing'. In: *Computational and Structural Biotechnology Journal* 18 (May 2020), pp. 1342–1351.

[143] Joppich, Markus, Schmidl, Dirk, Bolger, Anthony M et al. 'PAGANtec: OpenMP Parallel Error Correction for Next-Generation Sequencing Data'. In: *OpenMP: Heterogenous Execution and Data Movements - 11th International Workshop on OpenMP, IWOMP 2015, Aachen, Germany, October 1-2, 2015, Proceedings*. Ed. by Christian Terboven, Bronis R de Supinski, Pablo Reble et al. Vol. 9342. Lecture Notes in Computer Science. Springer, 2015, pp. 3–17.

[144] Joppich, Markus, Weber, Christian and Zimmer, Ralf. 'Using Context-Sensitive Text Mining to Identify miRNAs in Different Stages of Atherosclerosis'. In: *Thrombosis and Haemostasis* 119.08 (Aug. 2019), pp. 1247–1264.

[145] Joppich, Markus and Zimmer, Ralf. 'From command-line bioinformatics to bioGUI'. In: *PeerJ* 2019.11 (Nov. 2019), e8111.

[146] Joy, Tisha and Hegele, Robert A. 'Is raising HDL a futile strategy for atheroprotection?' In: *Nature Reviews Drug Discovery* 7.2 (Feb. 2008), pp. 143–155.

[147] Joyce, Andrew R. and Palsson, Bernhard. 'The model organism as a system: Integrating 'omics' data sets'. In: *Nature Reviews Molecular Cell Biology* 7.3 (Mar. 2006), pp. 198–210.

[148] Kanehisa, Minoru, Furumichi, Miho, Sato, Yoko et al. 'KEGG: Integrating viruses and cellular organisms'. In: *Nucleic Acids Research* 49.D1 (Jan. 2021), pp. D545–D551.

[149] Karagkouni, Dimitra, Paraskevopoulou, Maria D., Chatzopoulos, Serafeim et al. 'DIANA-TarBase v8: A decade-long collection of experimentally supported miRNA-gene interactions'. In: *Nucleic Acids Research* 46.D1 (Jan. 2018), pp. D239–D245.

[150] Katz, Daniel S., Gruenpeter, Morane and Honeyman, Tom. 'Taking a fresh look at FAIR for research software'. In: *Patterns* 2.3 (Mar. 2021), p. 100222.

[151] Kaya, Koray D, Karakülah, Gökhan, Yakicier, Cengiz M et al. 'mESAdb: microRNA expression and sequence analysis database.' In: *Nucleic acids research* 39.Database issue (Jan. 2011), pp. D170–80.

[152] Kern, Fabian, Fehlmann, Tobias and Keller, Andreas. 'On the lifetime of bioinformatics web services'. In: *Nucleic Acids Research* 48.22 (Dec. 2020), pp. 12523–12533.

[153] Kim, Daehwan, Paggi, Joseph M., Park, Chanhee et al. 'Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype'. In: *Nature Biotechnology* 37.8 (Aug. 2019), pp. 907–915.

[154] Kinjo, Sonoko, Monma, Norikazu, Misu, Sadahiko et al. 'Maser: One-stop platform for NGS big data from analysis to visualization'. In: *Database* 2018.2018 (Jan. 2018), pp. 1–12.

[155] Kirsche, Melanie, Das, Arun and Schatz, Michael. 'Sapling: Accelerating Suffix Array Queries with Learned Data Models'. In: *bioRxiv* (Jan. 2020), p. 2020.01.29.925768.

[156] Kılıç, Ayşe, Santolini, Marc, Nakano, Taiji et al. 'A systems immunology approach identifies the collective impact of 5 miRs in Th2 inflammation'. In: *JCI insight* 3.11 (June 2018).

[157] Koh, Hiromi W.L., Fermin, Damian, Vogel, Christine et al. 'iOmicsPASS: network-based integration of multiomics data for predictive subnetwork discovery'. In: *npj Systems Biology and Applications* 5.1 (Dec. 2019), pp. 1–10.

[158] Kokot, Marek, Dlugosz, Maciej and Deorowicz, Sebastian. 'KMC 3: counting and manipulating k-mer statistics'. In: *Bioinformatics (Oxford, England)* 33.17 (Sept. 2017), pp. 2759–2761.

[159] Kordic, Branislav, Popovic, Marko, Popovic, Miroslav et al. 'A protein structure prediction program architecture based on a software transactional memory'. In: *ACM International Conference Proceeding Series*. New York, New York, USA: Association for Computing Machinery, Sept. 2019, pp. 1–9.

[160] Koren, Sergey, Walenz, Brian P, Berlin, Konstantin et al. 'Canu: Scalable and accurate long-read assembly via adaptive κ-mer weighting and repeat separation'. In: *Genome Research* 27.5 (May 2017), pp. 722–736.

[161] Kozomara, Ana, Birgaoanu, Maria and Griffiths-Jones, Sam. 'MiRBase: From microRNA sequences to function'. In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D155–D162.

[162] Kozomara, Ana and Griffiths-Jones, Sam. 'MiRBase: Integrating microRNA annotation and deep-sequencing data'. In: *Nucleic Acids Research* 39.SUPPL. 1 (Jan. 2011), pp. D152–D157.

[163] Kozomara, Ana and Griffiths-Jones, Sam. 'miRBase: annotating high confidence microRNAs using deep sequencing data'. In: *Nucleic Acids Res* 42.D1 (Jan. 2014), pp. D68–D73.

[164] Krzywinski, Martin and Altman, Naomi. 'Comparing samples-part i'. In: *Nature Methods* 11.3 (Mar. 2014), pp. 215–216.

[165] Kumar, Kishore R., Cowley, Mark J. and Davis, Ryan L. 'Next-Generation Sequencing and Emerging Technologies'. In: *Seminars in Thrombosis and Hemostasis* 45.7 (Oct. 2019), pp. 661–673.

[166] Lähnemann, David, Köster, Johannes, Szczurek, Ewa et al. 'Eleven grand challenges in single-cell data science'. In: *Genome Biology* 21.1 (Feb. 2020), p. 53.

[167] Langmead, Ben and Salzberg, Steven L. 'Fast gapped-read alignment with Bowtie 2.' In: *Nature methods* 9.4 (Apr. 2012), pp. 357–9.

[168] Langmead, Ben, Trapnell, Cole, Pop, Mihai et al. 'Ultrafast and memory-efficient alignment of short DNA sequences to the human genome'. In: *Genome Biology* 10.3 (Mar. 2009), R25.

[169] Lauzon, N., Dufresne, M., Beaudoin, A. et al. 'Forensic analysis of latent fingermarks by silver-assisted LDI imaging MS on nonconductive surfaces'. In: *Journal of Mass Spectrometry* 52.6 (June 2017), pp. 397–404.

[170] Lauzon, Nidia and Chaurand, Pierre. 'Detection of exogenous substances in latent fingermarks by silver-assisted LDI imaging MS: Perspectives in forensic sciences'. In: *Analyst* 143.15 (Aug. 2018), pp. 3586–3594.

[171] Lee, Jinhyuk, Yoon, Wonjin, Kim, Sungdong et al. 'BioBERT: A pre-trained biomedical language representation model for biomedical text mining'. In: *Bioinformatics* 36.4 (Sept. 2020), pp. 1234–1240.

[172] Leggett, Richard M., Heavens, Darren, Caccamo, Mario et al. 'NanoOK: Multi-reference alignment analysis of nanopore sequencing data, quality and error profiles'. In: *Bioinformatics* 32.1 (Sept. 2015), pp. 142–144.

[173] Leinonen, Rasko, Sugawara, Hideaki and Shumway, Martin. 'The sequence read archive'. In: *Nucleic Acids Research* 39.SUPPL. 1 (Jan. 2011), pp. D19–D21.

[174] Lenzerini, Maurizio. 'Data integration'. In: *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '02*. New York, New York, USA: Association for Computing Machinery (ACM), 2002, p. 233.

[175] Lex, Alexander, Gehlenborg, Nils, Strobelt, Hendrik et al. 'UpSet: Visualization of intersecting sets'. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 1983–1992.

[176] Li, Bo and Dewey, Colin N. 'RSEM: Accurate transcript quantification from RNA-seq data with or without a reference genome'. In: *Bioinformatics: The Impact of Accurate Quantification on Proteomic and Genetic Analysis and Research* 12.1 (Dec. 2014), pp. 41–74.

[177] Li, Gang, Ross, Karen E., Arighi, Cecilia N. et al. 'miRTex: A Text Mining System for miRNA-Gene Relation Extraction'. In: *PLoS Computational Biology* 11.9 (Sept. 2015), pp. 1–24.

[178] Li, H. and Durbin, R. 'Fast and accurate short read alignment with Burrows-Wheeler transform'. In: *Bioinformatics* 25.14 (July 2009), pp. 1754–1760.

[179] Li, Heng. 'Minimap2: pairwise alignment for nucleotide sequences'. In: *Bioinformatics* 34.18 (Sept. 2018), pp. 3094–3100.

[180] Li, Heng, Handsaker, Bob, Wysoker, Alec et al. 'The Sequence Alignment/Map format and SAMtools.' In: *Bioinformatics* 25.16 (Aug. 2009), pp. 2078–9.

[181] Li, Te Mao, Liu, Shan Chi, Huang, Ya Hsin et al. 'YKL-40-induced inhibition of miR-590-3p promotes interleukin-18 expression and angiogenesis of endothelial progenitor cells'. In: *International Journal of Molecular Sciences* 18.5 (May 2017), p. 920.

[182] Li, Xiao Feng, Shen, Wen Wen, Sun, Ying Yin et al. 'MicroRNA-20a negatively regulates expression of NLRP3-inflammasome by targeting TXNIP in adjuvant-induced arthritis fibroblast-like synoviocytes'. In: *Joint Bone Spine* 83.6 (Dec. 2016), pp. 695–700.

[183] Liao, Mingfeng, Liu, Yang, Yuan, Jing et al. 'Single-cell landscape of bronchoalveolar immune cells in patients with COVID-19'. In: *Nature Medicine* 26.6 (June 2020), pp. 842–844.

[184] Liao, Yang, Smyth, Gordon K. and Shi, Wei. 'FeatureCounts: An efficient general purpose program for assigning sequence reads to genomic features'. In: *Bioinformatics* 30.7 (Apr. 2014), pp. 923–930.

[185] Liberzon, Arthur, Birger, Chet, Thorvaldsdóttir, Helga et al. 'The Molecular Signatures Database Hallmark Gene Set Collection'. In: *Cell Systems* 1.6 (Dec. 2015), pp. 417–425.

[186] Litviňuková, Monika, Talavera-López, Carlos, Maatz, Henrike et al. 'Cells of the adult human heart'. In: *Nature* 588.7838 (Dec. 2020), pp. 466–472.

[187] Liu, Yuexin. 'A global immune gene expression signature for human cancers'. In: *Oncotarget* 10.20 (Mar. 2019), pp. 1993–2005.

[188] Loman, Nicholas J. and Quinlan, Aaron R. 'Poretools: A toolkit for analyzing nanopore sequence data'. In: *Bioinformatics* 30.23 (Dec. 2014), pp. 3399–3401.

[189] Love, Michael I., Huber, Wolfgang and Anders, Simon. 'Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2'. In: *Genome Biology* 15.12 (Dec. 2014), p. 550.

[190] Ma, Changlin, Zhang, Yong and Zhang, Maoyuan. 'Tree kernel-based protein-protein interaction extraction considering both modal verb phrases and appositive dependency features'. In: *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*. New York, New York, USA: Association for Computing Machinery, Inc, May 2015, pp. 655–660.

[191] Mabbott, Neil A., Baillie, J. K., Brown, Helen et al. 'An expression atlas of human primary cells: Inference of gene function from coexpression networks'. In: *BMC Genomics* 14.1 (Sept. 2013), p. 632.

[192] Machado Pereira, Marcio, Gaudet, Matthew, Nelson Amaral, J. et al. 'Study of hardware transactional memory characteristics and serialization policies on Haswell'. In: *Parallel Computing* 54 (May 2016), pp. 46–58.

[193] Mack, Christopher. 'Fibroblasts'. In: *Atherosclerosis: Risks, Mechanisms, and Therapies*. Hoboken, NJ: John Wiley & Sons, Inc, Mar. 2015, pp. 129–140.

[194] Magi, Alberto, Semeraro, Roberto, Mingrino, Alessandra et al. 'Nanopore sequencing data analysis: state of the art, applications and challenges'. In: *Briefings in Bioinformatics* (June 2017).

[195] Maglott, Donna, Ostell, Jim, Pruitt, Kim D. et al. 'Entrez Gene: Gene-centered information at NCBI'. In: *Nucleic Acids Research* 33.suppl_1 (Jan. 2005), pp. D54–D58.

[196] Makałowski, Wojciech and Shabardina, Victoria. 'Bioinformatics of nanopore sequencing'. In: *Journal of Human Genetics* 65.1 (Jan. 2020), pp. 61–67.

[197] Malmberg, M. M., Spangenberg, G. C., Daetwyler, H. D. et al. 'Assessment of low-coverage nanopore long read sequencing for SNP genotyping in doubled haploid canola (Brassica napus L.)' In: *Scientific Reports* 9.1 (Dec. 2019), pp. 1–12.

[198] Mamun, Abdullah Al, Pal, Soumitra and Rajasekaran, Sanguthevar. 'KCMBT: A k-mer Counter based on Multiple Burst Trees'. In: *Bioinformatics* 32.18 (Sept. 2016), pp. 2783–2790.

[199] Manekar, Swati C. and Sathe, Shailesh R. 'A benchmark study of k-mer counting methods for high-throughput sequencing'. In: *GigaScience* 7.12 (Dec. 2018), pp. 1–13.

[200] Mangul, Serghei, Martin, Lana S., Eskin, Eleazar et al. 'Improving the usability and archival stability of bioinformatics software'. In: *Genome Biology* 20.1 (Feb. 2019), p. 47.

[201] Mangul, Serghei, Mosqueiro, Thiago, Abdill, Richard J. et al. 'Challenges and recommendations to improve the installability and archival stability of omics computational tools'. In: *PLoS Biology* 17.6 (June 2019), e3000333.

[202] Mapleson, Daniel, Accinelli, Gonzalo Garcia, Kettleborough, George et al. 'KAT: A K-mer analysis toolkit to quality control NGS datasets and genome assemblies'. In: *Bioinformatics* 33.4 (Feb. 2017), pp. 574–576.

[203] Marçais, Guillaume and Kingsford, Carl. 'A fast, lock-free approach for efficient parallel counting of occurrences of k-mers.' In: *Bioinformatics* 27.6 (Mar. 2011), pp. 764–70.

[204] Marneffe, Marie-Catherine de and Manning, Christopher D. 'The Stanford typed dependencies representation'. In: *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*. August. 2008, pp. 1–8.

[205] Martin, Ken, Hoffman, Bill, Cedilnik, Andy et al. *Mastering CMake Fifth Edition*. Fifth Edit. Kitware Inc., 2010.

[206] Mayer, Gerhard, Müller, Wolfgang, Schork, Karin et al. 'Implementing FAIR data management within the German Network for Bioinformatics Infrastructure (de.NBI) exemplified by selected use cases'. In: *Briefings in Bioinformatics* bbab010 (Feb. 2021), pp. 1–14.

[207] McCabe, Sean D., Lin, Dan Yu and Love, Michael I. 'Consistency and overfitting of multi-omics methods on experimental data'. In: *Briefings in Bioinformatics* 21.4 (July 2020), pp. 1277–1284.

[208] McCarthy, Davis J., Chen, Yunshun and Smyth, Gordon K. 'Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation'. In: *Nucleic Acids Research* 40.10 (May 2012), pp. 4288–4297.

[209] McInnes, Leland and Healy, John. 'Accelerated Hierarchical Density Based Clustering'. In: *IEEE International Conference on Data Mining Workshops, ICDMW*. Vol. 2017-Novem. IEEE Computer Society, Dec. 2017, pp. 33–42.

[210] McInnes, Leland, Healy, John and Melville, James. 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'. In: *arxiv.org* (Feb. 2018).

[211] McManus, Michael T. and Sharp, Phillip A. 'Gene silencing in mammals by small interfering RNAs'. In: *Nat Rev Genet* 3.10 (Oct. 2002), pp. 737–747.

[212] Mering, Christian von, Jensen, Lars J., Kuhn, Michael et al. 'STRING 7 - Recent developments in the integration and prediction of protein interactions'. In: *Nucleic Acids Research* 35.SUPPL. 1 (Jan. 2007), pp. D358–D362.

[213] Mirza, Bilal, Wang, Wei, Wang, Jie et al. 'Machine learning and integrative analysis of biomedical big data'. In: *Genes* 10.2 (Jan. 2019), p. 87.

[214] Mohr, Thomas, Haudek-Prinz, Verena, Slany, Astrid et al. 'Proteome profiling in IL-1β and VEGF-activated human umbilical vein endothelial cells delineates the interlink between inflammation and angiogenesis'. In: *PLoS ONE* 12.6 (June 2017).

[215] Morand, Eric F., Leech, Michelle and Bernhagen, Jürgen. 'MIF: A new cytokine link between rheumatoid arthritis and atherosclerosis'. In: *Nature Reviews Drug Discovery* 5.5 (Apr. 2006), pp. 399–411.

[216] Morganti, Stefania, Tarantino, Paolo, Ferraro, Emanuela et al. 'Next generation sequencing (NGS): A revolutionary technology in pharmacogenomics and personalized medicine in cancer'. In: *Advances in Experimental Medicine and Biology*. Vol. 1168. Springer, Nov. 2019, pp. 9–30.

[217] Morse, Christina, Tabib, Tracy, Sembrat, John et al. 'Proliferating SPP1/MERTK-expressing macrophages in idiopathic pulmonary fibrosis'. In: *European Respiratory Journal* 54.2 (Aug. 2019).

[218] Mortazavi, Ali, Williams, Brian a, McCue, Kenneth et al. 'Mapping and quantifying mammalian transcriptomes by RNA-Seq.' In: *Nature methods* 5.7 (July 2008), pp. 621–8.

[219] Moss, Eric G. 'MicroRNAs: Hidden in the genome'. In: *Current Biology* 12.4 (Feb. 2002), R138–40.

[220] Muhl, Lars, Genové, Guillem, Leptidis, Stefanos et al. 'Single-cell analysis uncovers fibroblast heterogeneity and criteria for fibroblast and mural cell identification and discrimination'. In: *Nature Communications* 11.1 (Dec. 2020), pp. 1–18.

[221] Müller, H. M., Van Auken, K M, Li, Y et al. 'Textpresso Central: A customizable platform for searching, text mining, viewing, and curating biomedical literature'. In: *BMC Bioinformatics* 19.1 (Mar. 2018), p. 94.

[222] Murphy, K M and Weaver, C. *Janeway's Immunobiology: Ninth International Student Edition*. Garland Science, Taylor & Francis Group, LLC, 2016. ISBN: 9780815345510.

[223] Myers, Eugene W., Sutton, Granger G., Delcher, Art L. et al. 'A whole-genome assembly of Drosophila'. In: *Science* 287.5461 (Mar. 2000), pp. 2196–2204.

[224] Naeem, Haroon, Küffner, Robert, Csaba, Gergely et al. 'miRSel: automated extraction of associations between microRNAs and genes from the biomedical literature.' In: *BMC Bioinformatics* 11 (Mar. 2010), p. 135.

[225] Nam, Seungyoon, Li, Meng, Choi, Kwangmin et al. 'MicroRNA and mRNA integrated analysis (MMIA): a web tool for examining biological functions of microRNA expression'. In: *Nucleic Acids Research* 37.suppl_2 (May 2009), W356–W362.

[226] Neumann, Elizabeth K., Djambazova, Katerina V., Caprioli, Richard M. et al. 'Multimodal Imaging Mass Spectrometry: Next Generation Molecular Mapping in Biology and Medicine'. In: *Journal of the American Society for Mass Spectrometry* (Sept. 2020).

[227] Neumann, Mark, King, Daniel, Beltagy, Iz et al. 'ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing'. In: *Proceedings of the 18th BioNLP Workshop and Shared Task*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 319–327.

[228] Nicolai, Leo, Leunig, Alexander, Brambs, Sophia et al. 'Vascular neutrophilic inflammation and immunothrombosis distinguish severe COVID-19 from influenza pneumonia'. In: *Journal of Thrombosis and Haemostasis* (Nov. 2020), jth.15179.

[229] Niessner, Alexander and Weyand, Cornelia M. 'Dendritic cells in atherosclerotic disease'. In: *Clinical Immunology* 134.1 (Jan. 2010), pp. 25–32.

[230] Noor, Zainab, Ahn, Seong Beom, Baker, Mark S et al. 'Mass spectrometry-based protein identification in proteomics – a review'. In: *Briefings in Bioinformatics* 22.2 (Feb. 2020), pp. 1620–1638.

[231] Nosek, Brian A. and Errington, Timothy M. 'What is replication?' In: *PLoS Biology* 18.3 (Mar. 2020), e3000691.

[232] Olenchuk, Margaryta. '3D-Representation of High-Dimensional Heterogeneous Experimental Data'. Bachelor thesis. LMU Munich, 2019.

[233] Papageorgiou, Louis, Eleni, Picasi, Raftopoulou, Sofia et al. 'Genomic big data hitting the storage bottleneck'. In: *EMBnet.journal* 24 (Apr. 2018), e910.

[234] Paschke, C., Leisner, A., Hester, A. et al. 'Mirion - A software package for automatic processing of mass spectrometric images'. In: *Journal of the American Society for Mass Spectrometry* 24.8 (June 2013), pp. 1296–1306.

[235] Patel, Arpan, Belykh, Evgenii, Miller, EricJ et al. 'MinION rapid sequencing: Review of potential applications in neurosurgery'. In: *Surgical Neurology International* 9.1 (Aug. 2018), p. 157.

[236] Peck, David, Crawford, Emily D., Ross, Kenneth N. et al. 'A method for high-throughput gene expression signature analysis'. In: *Genome Biology* 7.7 (July 2006), pp. 1–6.

[237] Peels, Rik. 'Replicability and replication in the humanities'. In: *Research Integrity and Peer Review* 4.1 (Dec. 2019), p. 2.

[238] Pekayvaz, Kami, Leunig, Alexander, Kaiser, Rainer et al. 'Protective immune trajectories in early viral containment of non-pneumonic SARS-CoV-2 infection'. In: *bioRxiv* (Feb. 2021), p. 2021.02.03.429351.

[239] Peng, Yong and Croce, Carlo M. 'The role of microRNAs in human cancer'. In: *Signal Transduction and Targeted Therapy* 1.4 (Jan. 2016).

[240] Pertea, Mihaela, Kim, Daehwan, Pertea, Geo M et al. 'Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown.' In: *Nature Protocols* 11.9 (Sept. 2016), pp. 1650–67.

[241] Pessotti, Rita de Cassia, Hansen, Bridget L, Zacharia, Vineetha M. et al. 'High Spatial Resolution Imaging Mass Spectrometry Reveals Chemical Heterogeneity Across Bacterial Microcolonies'. In: *Analytical Chemistry* (Nov. 2019).

[242] Plaisier, Seema B., Taschereau, Richard, Wong, Justin A. et al. 'Rank-rank hypergeometric overlap: Identification of statistically significant overlap between gene-expression signatures'. In: *Nucleic Acids Research* 38.17 (July 2010), e169.

[243] Pont, Frédéric, Tosolini, Marie and Fournié, Jean J. 'Single-Cell Signature Explorer for comprehensive visualization of single cell signatures across scRNA-seq datasets'. In: *Nucleic Acids Research* 47.21 (Dec. 2019), e133.

[244] Pop, Mihai and Salzberg, Steven L. 'Bioinformatics challenges of new sequencing technology'. In: *Trends in Genetics* 24.3 (2008), pp. 142–149.

[245] Porta Siegel, Tiffany, Hamm, Gregory, Bunch, Josephine et al. 'Mass Spectrometry Imaging and Integration with Other Imaging Modalities for Greater Molecular Understanding of Biological Tissues'. In: *Molecular Imaging and Biology* 20.6 (Dec. 2018), pp. 888–901.

[246] Prade, Verena M., Kunzke, Thomas, Feuchtinger, Annette et al. 'De novo discovery of metabolic heterogeneity with immunophenotype-guided imaging mass spectrometry'. In: *Molecular Metabolism* 36 (June 2020), p. 100953.

[247] Pyankov, S. A. and Babichev, S. L. 'Transactional memory as an approach to building a lock-free data structure'. In: *Procedia Computer Science*. Vol. 162. Elsevier B.V., Jan. 2019, pp. 76–81.

[248] Qi, Furong, Qian, Shen, Zhang, Shuye et al. 'Single cell RNA sequencing of 13 human tissues identify cell types and receptors of human coronaviruses'. In: *Biochemical and Biophysical Research Communications* 526.1 (May 2020), pp. 135–140.

[249] Quan, Changqin, Wang, Meng and Ren, Fuji. 'An unsupervised text mining method for relation extraction from biomedical literature'. In: *PLoS ONE* 9.7 (2014), pp. 1–8.

[250] Ràfols, Pere, Heijs, Bram, Del Castillo, Esteban et al. 'RMSIproc: An R package for mass spectrometry imaging data processing'. In: *Bioinformatics* 36.11 (June 2020), pp. 3618–3619.

[251] Rehmsmeier, Marc, Steffen, Peter, Höchsmann, Matthias et al. 'Fast and effective prediction of microRNA/target duplexes'. In: *RNA* 10.10 (Oct. 2004), pp. 1507–1517.

[252] Reinert, Knut, Dadi, Temesgen Hailemariam, Ehrhardt, Marcel et al. 'The SeqAn C++ template library for efficient sequence analysis: A resource for programmers'. In: *Journal of Biotechnology* 261 (Nov. 2017), pp. 157–168.

[253] Ren, Shiyan, Fan, Xueqiang, Peng, Liang et al. 'Expression of NF-κB, CD68 and CD105 in carotid atherosclerotic plaque'. In: *Journal of Thoracic Disease* 5.6 (Dec. 2013), pp. 771–776.

[254] Richard, Hugues, Schulz, Marcel H, Sultan, Marc et al. 'Prediction of alternative isoforms from exon expression levels in RNA-Seq experiments'. In: *Nucleic Acids Research* 38.10 (June 2010), e112.

[255] Ritchie, Matthew E., Phipson, Belinda, Wu, Di et al. 'Limma powers differential expression analyses for RNA-sequencing and microarray studies'. In: *Nucleic Acids Research* 43.7 (Jan. 2015), e47.

[256] Rizzo, Stefania, Coen, Matteo, Sakic, Antonija et al. 'Sudden coronary death in the young: Evidence of contractile phenotype of smooth muscle cells in the culprit atherosclerotic plaque'. In: *International Journal of Cardiology* 264 (Aug. 2018), pp. 1–6.

[257] Robinson, Mark D., McCarthy, Davis J. and Smyth, Gordon K. 'edgeR: A Bioconductor package for differential expression analysis of digital gene expression data'. In: *Bioinformatics* 26.1 (Nov. 2009), pp. 139–140.

[258] Rodriguez-Esteban, Raul and Iossifov, Ivan. 'Figure mining for biomedical research'. In: *Bioinformatics* 25.16 (Aug. 2009), pp. 2082–2084.

[259] Rojas-Rengifo, Diana F., Ulloa-Guerrero, Cindy P., Joppich, Markus et al. 'Tryptophan usage by Helicobacter pylori differs among strains'. In: *Scientific Reports* 9.1 (Dec. 2019), p. 873.

[260] Ruan, Jue and Li, Heng. 'Fast and accurate long-read assembly with wtdbg2'. In: *Nature Methods* 17.2 (Feb. 2020), pp. 155–158.

[261] Rupaimoole, Rajesha and Slack, Frank J. 'MicroRNA therapeutics: Towards a new era for the management of cancer and other diseases'. In: *Nature Reviews Drug Discovery* 16.3 (Feb. 2017), pp. 203–221.

[262] Ryan, Daniel J., Spraggins, Jeffrey M. and Caprioli, Richard M. 'Protein identification strategies in MALDI imaging mass spectrometry: a brief review'. In: *Current Opinion in Chemical Biology* 48 (Feb. 2019), pp. 64–72.

[263] Sanderson, Nicholas D, Street, Teresa L, Foster, Dona et al. 'Real-time analysis of nanopore-based metagenomic sequencing from infected orthopaedic devices'. In: *BMC Genomics* 19.1 (Dec. 2018), p. 714.

[264] Santovito, Donato, Egea, Virginia and Weber, Christian. 'Small but smart: MicroRNAs orchestrate atherosclerosis development and progression'. In: *Biochimica et Biophysica Acta - Molecular and Cell Biology of Lipids* 1861.12 (Dec. 2016), pp. 2075–2086.

[265] Sato, Hiroyasu, Kato, Rina, Isogai, Yuki et al. 'Analyses of group III secreted phospholipase A2 transgenic mice reveal potential participation of this enzyme in plasma lipoprotein modification, macrophage foam cell formation, and atherosclerosis'. In: *Journal of Biological Chemistry* 283.48 (Nov. 2008), pp. 33483–33497.

[266] Schleyer, Guy, Shahaf, Nir, Ziv, Carmit et al. 'In plaque-mass spectrometry imaging of a bloom-forming alga during viral infection reveals a metabolic shift towards odd-chain fatty acid lipids'. In: *Nature Microbiology* 4.3 (Mar. 2019), pp. 527–538.

[267] Schneider, M. V. and Jungck, J. R. 'Editorial: International, interdisciplinary, multilevel bioinformatics training and education'. In: *Briefings in Bioinformatics* 14.5 (Sept. 2013), pp. 527–527.

[268] Scholz, Matthew B., Lo, Chien Chi and Chain, Patrick S.G. 'Next generation sequencing and bioinformatic bottlenecks: The current state of metagenomic data analysis'. In: *Current Opinion in Biotechnology* 23.1 (Feb. 2012), pp. 9–15.

[269] Schramm, Thorsten, Hester, Alfons, Klinkert, Ivo et al. 'ImzML - A common data format for the flexible exchange and processing of mass spectrometry imaging data'. In: *Journal of Proteomics* 75.16 (Aug. 2012), pp. 5106–5110.

[270] Schriml, Lynn M, Mitraka, Elvira, Munro, James et al. 'Human Disease Ontology 2018 update: classification, content and workflow expansion'. In: *Nucleic Acids Research* 47.D1 (Jan. 2018), pp. D955–D962.

[271] Schuler, Robert, Bugacov, Alejandro, Blow, Matthew et al. 'Toward FAIR Knowledge Turns in Bioinformatics'. In: *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019*. Institute of Electrical and Electronics Engineers Inc., Nov. 2019, pp. 1240–1242.

[272] Schülke, Stefan. 'Induction of interleukin-10 producing dendritic cells as a tool to suppress allergen-specific T helper 2 responses'. In: *Frontiers in Immunology* 9.MAR (Mar. 2018), p. 1.

[273] Schulz, Sandra, Becker, Michael, Groseclose, M. Reid et al. 'Advanced MALDI mass spectrometry imaging in pharmaceutical research and drug development'. In: *Current Opinion in Biotechnology* 55 (Feb. 2019), pp. 51–59.

[274] Sedlazeck, Fritz J., Lee, Hayan, Darby, Charlotte A. et al. 'Piercing the dark matter: Bioinformatics of long-range sequencing and mapping'. In: *Nature Reviews Genetics* 19.6 (June 2018), pp. 329–346.

[275] Shaath, Hibah, Vishnubalaji, Radhakrishnan, Elkord, Eyad et al. 'Single-Cell Transcriptome Analysis Highlights a Role for Neutrophils and Inflammatory Macrophages in the Pathogenesis of Severe COVID-19'. In: *Cells* 9.11 (Oct. 2020).

[276] Shahi, Priyanka, Loukianiouk, Serguei, Bohne-Lang, Andreas et al. 'Argonaute - a database for gene regulation by mammalian microRNAs'. In: *Nucleic Acids Research* 34.suppl_1 (Jan. 2006), pp. D115–D118.

[277] Shanbhag, Anil, Pirk, Holger and Madden, Sam. 'Locality-Adaptive Parallel Hash Joins Using Hardware Transactional Memory'. In: 10195 (2017), pp. 118–133.

[278] Sharifi-Noghabi, Hossein, Zolotareva, Olga, Collins, Colin C. et al. 'MOLI: Multi-omics late integration with deep neural networks for drug response prediction'. In: *Bioinformatics*. Vol. 35. 14. Oxford University Press, July 2019, pp. i501–i509.

[279] Shen, Shihao, Park, Juw Won, Lu, Zhi-xiang et al. 'rMATS: Robust and flexible detection of differential alternative splicing from replicate RNA-Seq data'. In: *Proceedings of the National Academy of Sciences* 111.51 (Dec. 2014), E5593–E5601.

[280] Shimoyama, Mary, Nigam, Rajni, Mclntosh, Leslie Sanders et al. 'Three ontologies to define phenotype measurement data'. In: *Frontiers in Genetics* 3.MAY (May 2012), p. 87.

[281] Signoretto, Marco, Van De Plas, Raf, De Moor, Bart et al. 'Tensor versus matrix completion: A comparison with application to spectral data'. In: *IEEE Signal Processing Letters* 18.7 (July 2011), pp. 403–406.

[282] Simpson, Jared T, Workman, Rachael E, Zuzarte, P C et al. 'Detecting DNA cytosine methylation using nanopore sequencing'. In: *Nature Methods* 14.4 (Apr. 2017), pp. 407–410.

[283] Singh, Amrit, Shannon, Casey P., Gautier, Benoît et al. 'DIABLO: An integrative approach for identifying key molecular drivers from multi-omics assays'. In: *Bioinformatics* 35.17 (Sept. 2019), pp. 3055–3062.

[284] Smith, Jennifer R., Park, Carissa A., Nigam, Rajni et al. 'The clinical measurement, measurement method and experimental condition ontologies: Expansion, improvements and new applications'. In: *Journal of Biomedical Semantics* 4.1 (Oct. 2013), p. 26.

[285] Song, Jinyang, Yang, Shaonan, Yin, Ruihua et al. 'MicroRNA-181a regulates the activation of the NLRP3 inflammatory pathway by targeting MEK1 in THP-1 macrophages stimulated by ox-LDL'. In: *Journal of Cellular Biochemistry* 120.8 (Aug. 2019), pp. 13640–13650.

[286] Sović, Ivan, Šikić, Mile, Wilm, Andreas et al. 'Fast and sensitive mapping of nanopore sequencing reads with GraphMap'. In: *Nature Communications* 7 (Apr. 2016), p. 11307.

[287] Spraker, Joseph E., Luu, Gordon T. and Sanchez, Laura M. 'Imaging mass spectrometry for natural products discovery: a review of ionization methods'. In: *Natural Product Reports* 37.2 (July 2019), pp. 150–162.

[288] Spraker, Joseph E., Sanchez, Laura M., Lowe, Tiffany M. et al. 'Ralstonia solanacearum lipopeptide induces chlamydospore development in fungi and facilitates bacterial entry into fungal tissues'. In: *ISME Journal* 10.9 (Sept. 2016), pp. 2317–2330.

[289] Spraker, Joseph E., Wiemann, Philipp, Baccile, Joshua A. et al. 'Conserved responses in a war of small molecules between a plant-pathogenic bacterium and fungi'. In: *mBio* 9.3 (May 2018).

[290] Steffensen, Lasse B., Conover, Cheryl A. and Oxvig, Claus. 'PAPP-A and the IGF system in atherosclerosis: what's up, what's down?' In: *American journal of physiology. Heart and circulatory physiology* 317.5 (Nov. 2019), H1039–H1049.

[291] Stickels, Robert R., Murray, Evan, Kumar, Pawan et al. 'Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2'. In: *Nature Biotechnology* (Dec. 2020), pp. 1–7.

[292] Stodden, Victoria, Seiler, Jennifer and Ma, Zhaokun. 'An empirical analysis of journal policy effectiveness for computational reproducibility'. In: *Proceedings of the National Academy of Sciences of the United States of America* 115.11 (Mar. 2018), pp. 2584–2589.

[293] Stuart, Tim, Butler, Andrew, Hoffman, Paul et al. 'Comprehensive Integration of Single-Cell Data'. In: *Cell* 177.7 (June 2019), 1888–1902.e21.

[294] Stuart, Tim and Satija, Rahul. 'Integrative single-cell analysis'. In: *Nature Reviews Genetics* 20.5 (May 2019), pp. 257–272.

[295] Subramanian, Aravind, Tamayo, Pablo, Mootha, Vamsi K et al. 'Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles.' In: *Proceedings of the National Academy of Sciences of the United States of America* 102.43 (Oct. 2005), pp. 15545–50.

[296] Surjit, Milan and Lal, Sunil K. 'The nucleocapsid protein of the SARS coronavirus: Structure, function and therapeutic potential'. In: *Molecular Biology of the SARS-Coronavirus*. Springer Berlin Heidelberg, 2010, pp. 129–151.

[297] Swales, John G., Hamm, Gregory, Clench, Malcolm R. et al. 'Mass spectrometry imaging and its application in pharmaceutical research and development: A concise review'. In: *International Journal of Mass Spectrometry* 437 (Mar. 2019), pp. 99–112.

[298] Tang, Bin, Xiao, Bin, Liu, Zhen et al. 'Identification of MyD88 as a novel target of miR-155, involved in negative regulation of Helicobacter pylori-induced inflammation'. In: *FEBS Letters* 584.8 (Apr. 2010), pp. 1481–1486.

[299] Tang, Lin. 'FAIR your data'. In: *Nature Methods* 17.2 (Feb. 2020), p. 127.

[300] Tang, Xiaoning, Huang, Yongmei, Lei, Jinli et al. 'The single-cell sequencing: New developments and medical applications'. In: *Cell and Bioscience* 9.1 (June 2019), pp. 1–9.

[301] Tarca, Adi Laurentiu, Draghici, Sorin, Bhatti, Gaurav et al. 'Down-weighting overlapping genes improves gene set analysis.' In: *BMC Bioinformatics* 13.1 (Dec. 2012), p. 136.

[302] Tarraga, Joaquin, Gallego, Asunción, Arnau, Vicente et al. 'HPG pore: An efficient and scalable framework for nanopore sequencing data'. In: *BMC Bioinformatics* 17.1 (Dec. 2016), p. 107.

[303] Tárraga, Joaquín, Arnau, Vicente, Martínez, Héctor et al. 'Acceleration of short and long DNA read mapping without loss of accuracy using suffix array'. In: *Bioinformatics* 30.23 (Dec. 2014), pp. 3396–3398.

[304] Tenenhaus, Arthur, Philippe, Cathy, Guillemot, Vincent et al. 'Variable selection for generalized canonical correlation analysis'. In: *Biostatistics* 15.3 (July 2014), pp. 569–583.

[305] Teng, Haotian, Cao, Minh Duc, Hall, Michael B. et al. 'Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning'. In: *GigaScience* 7.5 (May 2018).

[306] The COVID-19 Genomics UK Consortium. 'An integrated national scale SARS-CoV-2 genomic surveillance network'. In: *The Lancet Microbe* 1.3 (June 2020), e99–e100.

[307] Tkaczyk, Dominika, Szostek, Paweł, Fedoryszak, Mateusz et al. 'CERMINE: Automatic extraction of structured metadata from scientific literature'. In: *International Journal on Document Analysis and Recognition* 18.4 (July 2015), pp. 317–335.

[308] Torang, Arezo, Gupta, Paraag and Klinke, David J. 'An elastic-net logistic regression approach to generate classifiers and gene signatures for types of immune cells and T helper cell subsets'. In: *BMC Bioinformatics* 20.1 (Aug. 2019), p. 433.

[309] Tsukamoto, Kosuke, Mani, D. R., Shi, Jianru et al. 'Identification of apolipoprotein D as a cardioprotective gene using a mouse model of lethal atherosclerotic coronary artery disease'. In: *Proceedings of the National Academy of Sciences of the United States of America* 110.42 (Oct. 2013), pp. 17023–17028.

[310] Urbich, Carmen, Kuehbacher, Angelika and Dimmeler, Stefanie. 'Role of microRNAs in vascular diseases, inflammation, and angiogenesis'. In: *Cardiovasc Res* 79.4 (July 2008), pp. 581–588.

[311] Van Der Maaten, Laurens and Hinton, Geoffrey. 'Visualizing data using t-SNE'. In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2625.

[312] Van Der Walt, Stéfan, Schönberger, Johannes L., Nunez-Iglesias, Juan et al. 'Scikit-image: Image processing in python'. In: *PeerJ* 2014.1 (June 2014), e453.

[313] Venet, David, Dumont, Jacques E. and Detours, Vincent. 'Most random gene expression signatures are significantly associated with breast cancer outcome'. In: *PLoS Computational Biology* 7.10 (Oct. 2011). Ed. by Isidore Rigoutsos, e1002240.

[314] Verbeeck, Nico, Caprioli, Richard M. and Van de Plas, Raf. 'Unsupervised machine learning for exploratory data analysis in imaging mass spectrometry'. In: *Mass Spectrometry Reviews* (2019).

[315] Veselkov, Kirill, Sleeman, Jonathan, Claude, Emmanuelle et al. 'BASIS: High-performance bioinformatics platform for processing of large-scale mass spectrometry imaging data in chemically augmented histology'. In: *Scientific Reports* 8.1 (Dec. 2018).

[316] Vickers, Kasey C., Rye, Kerry Anne and Tabet, Fatiha. 'MicroRNAs in the onset and development of cardiovascular disease'. In: *Clinical Science* 126.3 (Feb. 2014), pp. 183–194.

[317] Virtanen, Pauli, Gommers, Ralf, Oliphant, Travis E. et al. 'SciPy 1.0: fundamental algorithms for scientific computing in Python'. In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272.

[318] Wagner, Günter P., Kin, Koryu and Lynch, Vincent J. 'Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples'. In: *Theory in Biosciences* 131.4 (Dec. 2012), pp. 281–285.

[319] Wang, Haizhen, Zhang, Yujin, Luomei, Junzi et al. 'The miR-155/GATA3/IL37 axis modulates the production of proinflammatory cytokines upon TNF-α stimulation to affect psoriasis development'. In: *Experimental Dermatology* 29.7 (July 2020), pp. 647–658.

[320] Wang, Qing, Ni, Chong-ming, Li, Zhen et al. 'Efficient and accurate prediction of transmembrane topology from amino acid sequence only'. In: *bioRxiv* (May 2019), p. 627307.

[321] Wang, Zhong, Gerstein, Mark and Snyder, Michael. 'RNA-Seq: a revolutionary tool for transcriptomics.' In: *Nature Reviews Genetics* 10.1 (Jan. 2009), pp. 57–63.

[322] Watson, Mick, Thomson, Marian, Risse, Judith et al. 'PoRe: An R package for the visualization and analysis of nanopore sequencing data'. In: *Bioinformatics* 31.1 (Jan. 2015), pp. 114–115.

[323] Wei, Yuanyuan, Zhu, Mengyu, Corbalán-Campos, Judit et al. 'Regulation of Csf1r and Bcl6 in macrophages mediates the stage-specific effects of MicroRNA-155 on atherosclerosis'. In: *Arteriosclerosis, Thrombosis, and Vascular Biology* 35.4 (Apr. 2015), pp. 796–803.

[324] Weinberger, Tobias, Esfandyari, Dena, Messerer, Denise et al. 'Ontogeny of arterial macrophages defines their functions in homeostasis and inflammation'. In: *Nature Communications* 11.1 (Dec. 2020), p. 4549.

[325] Westergaard, David, Stærfeldt, Hans Henrik, Tønsberg, Christian et al. 'A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts'. In: *PLoS Computational Biology* 14.2 (Feb. 2018), e1005962.

[326] Wheeler, Travis J and Eddy, Sean R. 'nhmmer: DNA homology search with profile HMMs.' In: *Bioinformatics* 29.19 (Oct. 2013), pp. 2487–9.

[327] Wick, Ryan R, Judd, Louise M and Holt, Kathryn E. 'Performance of neural network basecalling tools for Oxford Nanopore sequencing'. In: *bioRxiv* (Feb. 2019), p. 543439.

[328] Wilkinson, Mark D., Dumontier, Michel, Aalbersberg, IJsbrand Jan et al. 'Comment: The FAIR Guiding Principles for scientific data management and stewardship'. In: *Scientific Data* 3.1 (Mar. 2016), pp. 1–9.

[329] Wirka, Robert C., Wagh, Dhananjay, Paik, David T. et al. 'Atheroprotective roles of smooth muscle cell phenotypic modulation and the TCF21 disease gene as revealed by single-cell analysis'. In: *Nature Medicine* (July 2019), p. 1.

[330] Wolf, F. Alexander, Angerer, Philipp and Theis, Fabian J. 'SCANPY: Large-scale single-cell gene expression data analysis'. In: *Genome Biology* 19.1 (Feb. 2018), p. 15.

[331] Wolf, Marc P. and Hunziker, Patrick. 'Atherosclerosis: Insights into Vascular Pathobiology and Outlook to Novel Treatments'. In: *Journal of Cardiovascular Translational Research* 13.5 (Oct. 2020), pp. 744–757.

[332] Wong, Nathan and Wang, Xiaowei. 'miRDB: an online resource for microRNA target prediction and functional annotations'. In: *Nucleic Acids Research* 43.D1 (Nov. 2014), pp. D146–D152.

[333] Xiao, Feifei, Zuo, Zhixiang, Cai, Guoshuai et al. 'miRecords: an integrated resource for microRNA-target interactions'. In: *Nucleic Acids Research* 37.suppl_1 (Nov. 2008), pp. D105–D110.

[334] Xu, Hao, Zhong, Liang, Deng, Jiaxin et al. 'High expression of ACE2 receptor of 2019-nCoV on the epithelial cells of oral mucosa'. In: *International Journal of Oral Science* 12.1 (Feb. 2020).

[335] Xu, Liu and Seki, Masahide. 'Recent advances in the detection of base modifications using the Nanopore sequencer'. In: *Journal of Human Genetics* 65.1 (Oct. 2020), pp. 25–33.

[336] Yates, Andrew D., Achuthan, Premanand, Akanni, Wasiu et al. 'Ensembl 2020'. In: *Nucleic Acids Research* 48.D1 (Jan. 2020), pp. D682–D688.

[337] Yoo, Richard M., Hughes, Christopher J., Lai, Konrad et al. 'Performance evaluation of Intel® Transactional Synchronization Extensions for high-performance computing'. In: *International Conference for High Performance Computing, Networking, Storage and Analysis, SC* (2013).

[338] You, Zhu Hong, Huang, Zhi An, Zhu, Zexuan et al. 'PBMDA: A novel and effective path-based computational model for miRNA-disease association prediction'. In: *PLoS Computational Biology* 13.3 (Mar. 2017). Ed. by Edwin Wang, e1005455.

[339] Yu, Guangchuang and He, Qing Yu. 'ReactomePA: An R/Bioconductor package for reactome pathway analysis and visualization'. In: *Molecular BioSystems* 12.2 (Jan. 2016), pp. 477–479.

[340] Yu, Guangchuang, Wang, Li Gen, Han, Yanyan et al. 'ClusterProfiler: An R package for comparing biological themes among gene clusters'. In: *OMICS A Journal of Integrative Biology* 16.5 (May 2012), pp. 284–287.

[341] Yu, Guangchuang, Wang, Li Gen, Yan, Guang Rong et al. 'DOSE: An R/Bioconductor package for disease ontology semantic and enrichment analysis'. In: *Bioinformatics* 31.4 (Feb. 2015), pp. 608–609.

[342] Yukselen, Onur, Turkyilmaz, Osman, Ozturk, Ahmet Rasit et al. 'DolphinNext: A distributed data processing platform for high throughput genomics'. In: *BMC Genomics* 21.1 (Apr. 2020).

[343] Zhang, Xinxin, Lan, Yujia, Xu, Jinyuan et al. 'CellMarker: A manually curated resource of cell markers in human and mouse'. In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D721–D728.

[344] Zhang, Zhidong, Liang, Kai, Zou, Gangqiang et al. 'Inhibition of miR-155 attenuates abdominal aortic aneurysm in mice by regulating macrophage-mediated inflammation'. In: *Bioscience Reports* 38.3 (May 2018).

[345] Zhu, Jinfang. 'T helper 2 (Th2) cell differentiation, type 2 innate lymphoid cell (ILC2) development and regulation of interleukin-4 (IL-4) and IL-13 production'. In: *Cytokine* 75.1 (Sept. 2015), pp. 14–24.

[346] Zuo, Shuguang, Dai, Gongpeng and Ren, Xuequn. 'Identification of a 6-gene signature predicting prognosis for colorectal cancer'. In: *Cancer Cell International* 19.1 (Jan. 2019), p. 6.

Abbreviations

CL command-line. 64, 178

CLI command-line interface. 64, 66, 179

DE Differential Expression. viii, x, 6, 7, 9, 12, 14, 15, 18, 62, 90, 103–105, 108, 111, 112, 118, 119, 129, 131, 136–138, 140–146, 152–161, 165, 166, 169, 172, 174, 180, 182, 241

DNA deoxyribonucleic acid. 7, 125, 131, 132, 181

DO Disease Ontology. 30, 31

ECO Evidence & Conclusion Ontology. vii, 22, 24–26, 28, 29

FAIR Findable, Accessible, Interoperable and Reusable. 2, 3, 6, 9, 11, 34, 37, 61, 63, 68, 69, 79, 92, 111, 145, 182

GIL global interpreter lock. 121, 122

GO Gene Ontology. vii, 15, 22, 24, 29–31, 37, 58, 59, 145, 158

GUI graphical user interface. 64, 66, 87, 136, 181

IMS imaging mass-spectrometry. viii, 3, 6, 7, 11, 12, 15, 71, 77, 87–90, 92–94, 101, 104, 111–113, 162–166, 169, 173, 174, 180–183

IPC inter-process communication. 121, 122

miRNA micro-RNA. vii, ix, xi, 3, 5–7, 11–13, 15–18, 31–49, 51–62, 76, 137, 147, 158–161, 169, 171, 172, 178–180, 182, 205, 207

ML machine learning. 34, 50

MMO Measurement Method Ontology. 22, 26, 29

mRNA messenger RNA. 8, 12, 14, 15, 33, 45, 57, 61, 73, 74, 123, 125, 178

NER named-entity recognition. 18–21, 23–25, 29–31, 34, 37, 46–49, 61, 171, 172, 179, 195, 197

NGS next-generation sequencing. viii, 5, 7, 8, 13, 65, 115, 116, 125, 129, 132, 137, 139, 181, 183

NLP natural language processing. 23, 32, 37, 47, 171, 172

RNA ribonucleic acid. 9, 16, 72, 73, 92, 115, 125, 127, 131, 132, 140, 174, 181

RNA-seq RNA-sequencing. 5, 7–9, 12, 56, 64, 72, 84, 111, 129, 137–140, 150, 166, 169, 174, 178, 180, 182

scRNA-seq single-cell RNA-seq. vii, 3, 5, 7, 9–12, 71–80, 82, 84, 89, 94, 101, 104, 111, 112, 162–166, 169, 173, 174, 178, 180–183, 194, 237

SDP shortest dependency path. vii, 38, 39, 42, 49–51

TGS third-generation sequencing. 6, 7, 12, 14, 65, 115, 116, 129, 131, 135–137, 174, 180, 181, 183

Mine, yours, these are all bourgeois categories!
The Kangaroo

Acknowledgements

I would like to say thanks to all who supported me in the last years, to all, who endured my mood in the last years, and to all, who repaired the lights in the journey's tunnel when these seemed to be broken. Thank you!

I would like to use this opportunity to thank my advisor Ralf Zimmer for giving me the opportunity to not only write my dissertation at his chair, but especially for providing the setting to explore many interesting topics within the greater setting of atherosclerosis within the CRC 1123!

Special thanks also go to all, current and past, members of the research unit bioinformatics, with whom a lot of topics were discussed, sometimes at lunch, sometimes in the offices, and sometimes while getting destroyed at the table soccer. Particular thanks go to Constantin Ammar who shared his office with me, and was regularly appointed 'R specialist' against his will. Many thanks to Gergely Csaba for helping me out in (emotional) discussions on text mining and sequencing projects. Being the initiator of the Bioinformatics Honesty Shop I absolutely thank all the customers of this soul food providing endeavour.

Without collaborations this work would not have been possible. I thank my Bachelor thesis student Rita Olenchuk for her amazing work during her Bachelor's thesis, but also before, on the work within the iGEM project and *sequ-into*, and thereafter as HiWi. I thank Quirin Emslander, Julia Mayer, and Luisa Jimenez-Soto for their contribution with *sequ-into*. Warm thanks also go to Christian Weber for his support and work with atheMir. Finally, I would like to thank Kami Pekayvaz, Tobias Weinberger, Konstantin Stark and Christian Schulz for the collaborations on (sc)RNA-seq projects so far, and hopefully, also beyond this thesis.

I am grateful to Martin Hofmann-Apitius for reviewing this thesis, and to Andreas Butz and Heinrich Hußmann for being part of my dissertation committee.

Special thanks go to my wife and my parents for always being there for me, for supporting me, and encouraging me to stay motivated. Thanks to my friends, and in particular Christian Plewnia, for keeping in touch with me, for being a brother in arms, on the virtual battlefields, car-soccer-arenas, and also in real-life.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Eberhardzell, 04.02.2022
Ort, Datum

Markus Joppich
Unterschrift