

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

**Advancing Single-Molecule Imaging Analysis via
Deep Learning**

Simon Martin Wanninger

aus

München, Deutschland

2024

Erklärung

Diese Dissertation wurde im Sinne von § 7 der Promotionsordnung vom 28. November 2011 von Herrn Dr. Prof. Don C. Lamb betreut.

Eidesstattliche Versicherung

Diese Dissertation wurde eigenständig und ohne unerlaubte Hilfe erarbeitet.

München, 12.02.2024

Simon Wanninger

Dissertation eingereicht am 12.02.2024

1. Gutachter: Dr. Prof. Don C. Lamb

2. Gutachter: Dr. Prof. Philip Tinnefeld

Mündliche Prüfung am 30.04.2024

“Machines take me by surprise with great frequency.”

Alan Turing

Abstract

Advancing Single-Molecule Imaging Analysis via Deep Learning

by Simon Martin Wanninger

Single-molecule experiments have revolutionized our understanding of the physical world, offering unparalleled insights into dynamic processes. However, a bottleneck persists in the time-consuming and potentially biased nature of data analysis. The main goal of this thesis is to address these issues through the development of deep learning techniques tailored for the analysis of fluorescence data specifically focusing on surface-based single-molecule measurements of Förster resonance energy transfer (FRET). The culmination of this effort is *Deep-LASI* (Deep-Learning Assisted Single-molecule Imaging analysis), a software suite leveraging the predictive capabilities of deep neural networks (DNNs). Designed for rapid analysis of single-color, two-color FRET, and three-color FRET data, *Deep-LASI* fully automates trajectory sorting and FRET correction factor determination, followed by the automated prediction of observed states and state dwell times for each molecule. The pre-trained ensemble of DNNs are able to analyze previously unseen data sets in approximately 20–100 ms per trajectory. In extensive benchmarking, the DNNs demonstrated their efficacy through ground truth simulations and comparisons with manually analyzed experimental data, validated by expert users. Beyond the development of these deep learning techniques, *Deep-LASI* has evolved into a comprehensive software suite that provides robust methods for extracting raw intensity data from single-molecule movies across multiple channels. A key feature was the integration of alternative approaches for user intervention, applicable to every step that the DNNs undertake automatically. This user-centric framework of *Deep-LASI* encompasses human evaluation of single-molecule trajectories, offering flexibility to override DNN classifications, and the option to employ hidden Markov models (HMM) for the analysis of kinetic rates, along with various downstream analysis methods post trajectory sorting. Additionally, the evaluation of various software tools for extracting kinetic rate constants from single-molecule FRET trajectories is presented in this thesis. By analyzing specific data sets with different levels of complexity, the comparison of all employed methods shed light on their limitations and revealed important aspects that need to be considered for consistent analysis results. Lastly, established computational methods were coupled with experimental data to elucidate the conformational dynamics of bacterial adhesin SdrG, which can form an extremely mechanostable complex with its target peptide human fibrinogen β (Fg β). The SdrG:Fg β complex can withstand forces greater than 2 nN, representing the strongest non-covalent bond of known to date. Combining molecular dynamics simulations with single-molecule FRET measurements provided new insights into the behavior of the locking strand and ligand-induced structural changes of the SdrG protein.

Contents

Abstract	iv
List of Figures	vii
1 Introduction	1
2 Deep Learning	4
2.1 Definition of deep learning	4
2.2 Deep Neural Networks	5
2.2.1 Feedforward Neural Networks	6
2.2.2 Convolutional Neural Network (CNN)	9
2.2.3 Long short-term memory (LSTM) networks	12
2.3 Learning Algorithms	14
2.3.1 Initialization	14
2.3.2 Backpropagation	15
2.3.3 Adaptive Moment Estimation (ADAM)	16
2.4 Regularization Techniques	17
2.4.1 L1 and L2 Regularization	18
2.4.2 Batch Normalization	18
2.4.3 Dropout	20
2.4.4 Data augmentation	20
2.4.5 Label smoothing	20
2.4.6 Pruning	21
2.5 Training and Monitoring	21
2.5.1 Hyperparameters	21
2.5.2 Model inspection	22
3 Fluorescence	27
3.1 Principles of fluorescence	27
3.1.1 Energy diagram	27
3.1.2 Franck-Condon principle and Stokes shift	28
3.2 Förster resonance energy transfer	29
3.2.1 Fundamentals of FRET	29
3.2.2 Two-color FRET	31
3.2.3 Three-color FRET	32
3.2.4 Accurate FRET efficiencies	33
3.2.5 Determination of correction factors	34
3.3 Data acquisition	36
3.3.1 Total internal reflection fluorescence (TIRF)	36
3.3.2 Confocal microscopy	38

4 Publications	40
4.1 Paper 1: Deep-LASI:deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures	40
4.1.1 Motivation and main results	40
4.1.2 Brief description of the method	41
4.1.3 Outlook	42
4.2 Paper 2: Deep-LASI, Single-molecule Data Analysis Software	44
4.2.1 Motivation and main results	44
4.2.2 Outlook	44
4.3 Paper 3: A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories	47
4.3.1 Motivation and main results	47
4.3.2 Outlook	50
4.4 Paper 4: Single-molecule FRET reveals conformational changes of bacterial adhesin SdrG upon ligand binding	51
4.4.1 Motivation and main results	51
4.5 Brief description of the method	51
4.5.1 Outlook	52
5 Summary and Conclusion	54
Bibliography	56
Acknowledgements	62
A Appended Papers	63
A.1 Paper 1: Deep-LASI: deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures	64
A.2 Paper 2: Deep-LASI, Single-Molecule Data Analysis Software	129
A.3 Paper 3: A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories	156
A.4 Paper 4: Elucidating the conformational changes of bacterial adhesin SdrG via single-molecule FRET and all-atom MD simulations	227

List of Figures

2.1	Definition of deep learning.	5
2.2	Multilayer perceptron concept.	6
2.3	Overview of activation functions.	8
2.4	CNN architecture of VGG16.	10
2.5	Feature hierarchy of a deep neural network.	11
2.6	Long short-term memory concept.	12
2.7	The variance-bias tradeoff.	19
2.8	Loss and accuracy during training.	23
2.9	CNN weight distributions.	24
2.10	BiLSTM parameter distributions.	25
2.11	Dense layer parameter distributions.	26
3.1	Jablonski Diagram of various excitation and emission processes.	27
3.2	The Franck-Condon principle.	28
3.3	Theory of FRET	30
3.4	FRET spectra	31
3.5	Transition pathways in three-color FRET	32
3.6	TIRF microscopy.	37
3.7	Confocal microscopy.	39
4.1	Deep-LASI workflow.	42
4.2	Deep-LASI software overview.	45
4.3	Kinetic software challenge.	49
4.4	SdrG:Fg β complex overview.	52

Chapter 1

Introduction

Artificial intelligence in single molecule research

In recent years, the field of artificial intelligence, especially deep learning, has emerged as a powerful tool for various fields, enabling machines to learn complex patterns in data and make accurate predictions. While deep learning gained prominence as a modern technology, its origins can be traced back several decades, with roots in the development of neural networks. The concept of neural networks came from the field of neuroscience, where researchers sought to understand the workings of the brain. Early models of neural networks were based on the idea of a perceptron, a mathematical model of a single neuron. The perceptron model was first introduced by Frank Rosenblatt in 1957 and it was designed to learn from examples by adjusting its weights to minimize the error between its predicted output and the correct output.¹ The limitations of the perceptron model, especially its inability to learn nonlinear patterns from the data, prevented it from becoming popular. These shortcomings were addressed later by Ivakhnenko and Lapa with the development of neural networks with multiple layers, known as deep neural networks.² However, training deep neural networks was a challenging task, as it required significant computational resources and a large amount of labeled data. In the 2000s, the emergence of graphics processing units (GPUs) and the availability of large datasets led to a resurgence of interest in deep learning. Researchers developed new algorithms, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), that were capable of learning complex patterns in data.³ In addition, new training techniques, such as dropout⁴ and batch normalization⁵, were developed to improve the training of deep neural networks. The substantial progress in deep learning led to a wide range of applications across all data-driven research fields and industries. In the natural sciences, particularly in the field of single-molecule studies, deep learning techniques have enabled new possibilities for understanding complex biological processes at the molecular level. Notable example applications include:

- Single-molecule Förster Resonance Energy Transfer (smFRET): Deep neural networks have been developed to rapidly categorize a large number of fluorescence intensity traces in smFRET data.^{6,7}
- Localization Microscopy: Deep learning models have been employed to precisely localize individual point emitters and analyze single molecule tracking data. These models can predict the trajectory of individual molecules with high accuracy, enabling the study of dynamic processes within cells.^{8,9}
- Protein Folding Prediction: Predicting the 3D structure of proteins is a critical task in bioinformatics. Deep learning approaches, including AlphaFold developed by DeepMind, have shown remarkable success in predicting protein structures from amino acid sequences, contributing to our understanding of protein function.¹⁰
- Drug Discovery: Deep learning models have been used to predict the binding affinities of molecules to specific protein targets. This is invaluable in drug discovery, as it accelerates the identification of potential drug candidates and reduces the need for expensive and time-consuming laboratory experiments.¹¹

- **Genomic Sequence Analysis:** Deep learning techniques, such as convolutional neural networks (CNNs), have been applied to analyze genomic sequences. They can identify important motifs, regulatory elements, and potential disease-associated mutations, aiding genetic research.¹²
- **Molecular Dynamics Simulations:** Deep learning has been integrated into molecular dynamics simulations to enhance their efficiency and accuracy. Neural networks can provide approximations of complex energy landscapes, improving our understanding of molecular behavior under different conditions.¹³
- **Cryo-Electron Microscopy (Cryo-EM):** Deep learning has had a significant impact on Cryo-EM, allowing for faster and more accurate reconstruction of 3D structures from noisy and limited data. Cryo-EM combined with deep learning has advanced structural biology research.¹⁴
- **Single Molecule Sensing:** Deep learning models have been applied to the analysis of signals from single molecule sensors. They can enhance the detection sensitivity and specificity in fields like nanopore sequencing, facilitating DNA and RNA sequencing at the single molecule level.¹⁵
- **Phylogenetics:** Deep learning has been used for phylogenetic tree construction and species identification based on DNA and RNA sequences. This aids in understanding evolutionary relationships and biodiversity.¹⁶

These examples represent only a fraction of the application scenarios realized since deep learning methods are essentially employed in every field that provides sufficient data for training and benefits from large-scale prediction or analysis. However, it is of utmost importance to approach these technologies with caution and critical scrutiny, primarily due to the lack of transparency on how deep neural networks arrive at their conclusions. Furthermore, hidden biases and unwanted behaviors of all kinds can arise when training such models, rendering them useless or even detrimental. This thesis introduces the theoretical fundamentals of deep learning, outlines the challenges encountered during the training of deep neural networks and provides guidelines on how to address these challenges.

Fluorescence microscopy

Within the scope of this thesis, deep neural networks (DNNs) are mainly developed for the application in fluorescence microscopy. Therefore, it is crucial to emphasize the fundamentals of fluorescence microscopy as they provide the theoretical foundation of generating and understanding the datasets used for training DNNs. Fluorescence microscopy is a powerful research tool with a rich history and a wide range of applications. Especially in single-molecule research, it is an invaluable technique due to its high sensitivity and precision. Unlike bulk measurements that provide averaged information across a multitude of molecules, single-molecule approaches offer distinct advantages by enabling the observation and analysis of individual systems. Isolating single molecules in a controlled environment enables the confirmation or reevaluation of hypotheses generated by molecular dynamics simulations or other computational predictions. Combined with fluorescence microscopy, single-molecule studies provide insights into the heterogeneity inherent in biological systems, including the impact of environmental factors and local conditions. At its core, fluorescence microscopy harnesses the unique property of certain molecules to absorb photons at one wavelength and emit them at another, longer wavelength. This phenomenon, known as fluorescence, serves as the basis for visualizing specific molecules and extracting information over time. Fluorescence microscopy operates by selectively exciting fluorescent molecules, called fluorophores, and capturing the emitted fluorescence to either create high-resolution images or to probe the behavior of single molecular systems one at a time. In many cases, covalent labeling is employed, where the fluorescent dye is chemically attached to the molecule of interest. This labeling method is particularly useful for investigating the conformational dynamics of proteins or nanosystems such as DNA origami structures. To obtain information about different molecular states, quenching techniques are often used to deliberately introduce alterations in the emitted fluorescence intensity. One of the most important quenching mechanisms is called Förster resonance energy transfer (FRET),

which can be used to precisely determine the proximity between two or more fluorophores on a nanometer scale.^{17–23} Acting as a molecular ruler, FRET reveals changes in distances within or between molecules over a time. However, when using two fluorophores (2-color FRET), only a single distance is obtained, which is often insufficient for a comprehensive interpretation of the data. To overcome this limitation, a common strategy involves labeling the molecule of interest at different positions and measuring them separately. This approach allows for the acquisition of various distances, leading to a more thorough understanding of the molecular structure of the biomolecule and its kinetics. However, this strategy entails measuring multiple samples at similar conditions with each measurement containing thousands of molecules that need to be analyzed. Another approach is labeling the molecules with three fluorophores (3-color FRET), yielding three distances simultaneously and providing insights into intricate kinetic behaviors.^{24–29} The drawback of 3-color FRET is the increased complexity of data analysis due to the interdependence of the three FRET efficiencies and the higher number of available intensity channels that need to be considered. Regardless of the number of employed fluorophores, research projects often involve probing molecules in different environments or binding kinetics that depend on the concentration of a specific substrate. The extensive data collection and statistical analysis significantly increase the amount of effort, required expertise and time consumption needed to obtain accurate results. Therefore, high-throughput analysis tools are indispensable for processing large volumes of data in a reliable way. This thesis provides a detailed discussion of multi-color FRET and its practical considerations, including a theoretical basis for fluorescence microscopy in single-molecule research.

Outline

Chapter 2 begins with a concise definition of artificial intelligence, describes the fundamentals of deep learning and continues with detailed aspects regarding the training process of deep neural networks. Various deep neural network (DNN) architectures, ranging from traditional feedforward networks to specialized structures like convolutional neural networks (CNN) and long short-term memory (LSTM) networks, are introduced and explored. This chapter also provides detailed insights into learning algorithms employed in the actual training procedure, including initialization, backpropagation and adaptive moment estimation (ADAM). Additionally, a comprehensive summary of state of the art regularization techniques is presented, which play a crucial role in forcing DNNs to generalize a task and correctly handle new data. The final sections of chapter 2 describe the nuances of training and monitoring models, discussing critical aspects like hyperparameters and model inspection. Chapter 3 introduces the principles of fluorescence with a focus on Förster resonance energy transfer (FRET), a key phenomenon used for all publications presented in this thesis. In detail, chapter 3 introduces the fundamentals of FRET and its applications in single-molecule research employing two and three fluorophores. The section concludes with a description of common data acquisition techniques including total internal reflection fluorescence (TIRF) and confocal microscopy. Chapter 4 provides an overview of the work published in the context of this thesis, where the integration of deep learning techniques with fluorescence studies is explored and exploited. The developed software suite *Deep-LASI* for the analysis of single molecule time trajectories and the integrated deep neural networks are presented in two distinct publications. Two additional publications involve a blind benchmark of various software tools used for analyzing kinetic rates from single-molecule FRET trajectories and the study of conformational changes of bacterial adhesin SdrG upon ligand binding via single-molecule FRET and molecular dynamics (MD) simulations. Chapter 5 summarizes the work presented in this thesis and offers final conclusions for each project.

Chapter 2

Deep Learning

2.1 Definition of deep learning

Artificial intelligence (AI) is a growing field with many practical uses, ongoing research areas and encompasses different disciplines (Figure 2.1). Historically, the real challenge for AI emerged in solving problems that humans find effortless to do but struggle to describe formally, e.g. via mathematical equations or complex rules. Most of these tasks are solved by humans intuitively, such as identifying written words or recognizing distinct patterns in images. Numerous AI projects have attempted to encode information about the world into formal languages. Subsequently, computers are capable of autonomously processing these statements using logical inference rules. This approach is commonly referred to as the knowledge base approach in the field of AI. However, knowledge bases require a high level of human supervision to implement formal rules and still struggle to tackle complex problems.³⁰ The general issue with hard-coded knowledge is that AI models do not have the capability to learn on their own. The automatic extraction of patterns from raw data without human interference is termed machine learning. Popular machine learning algorithms are the logistic regression, widely used for binary classification tasks, and naive Bayes, which employs Bayes' theorem for probabilistic classification. While these methods are very efficient, they rely on distinct features that have to be designed and clearly defined beforehand. As the complexity of these features increases, the learning process becomes more difficult and susceptible to errors. This challenge can be tackled by using machine learning to simultaneously identify how the input representation relates to the desired output and what the input representation should be. This is referred to as "representation learning" and circumvents manual feature engineering. It also allows AI systems to quickly adapt to new tasks without needing much human intervention. With representation learning, algorithms can find suitable features in a fraction of time compared to the manual design process and often achieve higher accuracy.³¹ A classic example of representation learning is the autoencoder. It consists of two parts: an encoder that changes the input data into a different form and a decoder that changes this new form back to the original. Autoencoders are trained to keep as much information as possible during this transformation while also making sure the new form has certain useful properties. However, real-world applications need to consider the extremely high diversity in data that may represent only one desired class. Hence, a major contributor to the performance of an autoencoder is the size of the training data. Ideally, the training data should represent features with high variance, such that the autoencoder can learn abstract features in an unbiased way and ignore redundant information. This shifts the workload from designing features by hand to obtaining a training dataset with feature representations of high quality. Moreover, the requirement for a substantial amount of training data can be an obstacle for those who lack the resources to gather such data. Deep learning provides an effective solution to this problem by introducing representations that are formulated using simpler, underlying representations. This can be achieved by training a deep neural network (DNN). Essentially, a DNN is a mathematical function that transforms a set of input values into output values. This function is constructed by combining multiple, simpler functions. Each application of these distinct mathematical functions can be viewed as generating a new representation of the input. While the demands on the training dataset in deep learning remain high, they can be more readily met since the training data is labeled with the desired categories, i.e. the clustering of information is supervised by a human, in contrast to the training data for an autoencoder. As a result, a DNN can efficiently learn the most accurate method to compress the information within the training data

into the specified categories. In the scope of this thesis the focus lies on the supervised learning method using deep neural networks. The following section discuss the elementary operations of DNNs, their architectural aspects, and the training algorithms.

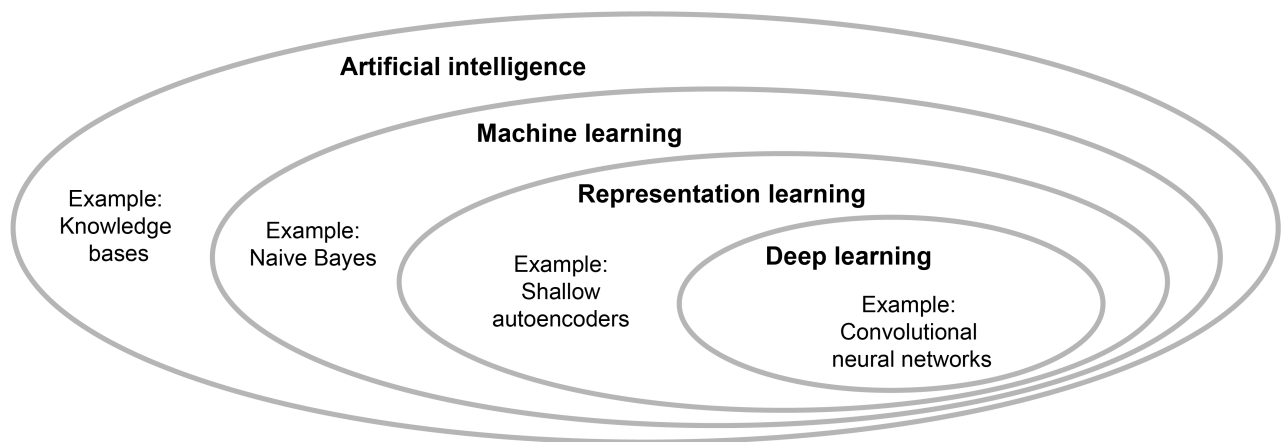


Figure 2.1 | Venn diagram of AI. Deep learning is a form of representation learning, which is in turn a subset of machine learning—a technique employed in various but not all AI methodologies.

2.2 Deep Neural Networks

Deep neural network (DNN) is a term that encompasses a broad category of neural network models that are characterized by having multiple layers, known as hidden layers, between the input and output layers. These hidden layers enable DNNs to learn hierarchical and complex representations of data, making them particularly effective in capturing intricate patterns and features in various domains. Within the field of deep neural networks, several different model types exist, each designed for specific tasks or data types. For various classification tasks, commonly used model types are:

- **Feedforward Neural Networks (FNNs):** FNNs (section 2.2.1) are a fundamental type of neural network where information travels forward in one direction from the input layer through one or more hidden layers to the output layer. In these networks, the neurons in a given layer connect to the neurons in the subsequent layer. This property differentiates FNNs from recurrent neural networks (listed below), that include connections between neurons in the same layer.
- **Convolutional Neural Networks (CNNs):** In principle, CNNs (section 2.2.2) are a specialized form of a FNN, since the information flows forward from the input to the output. However, CNNs are specialized for processing grid-like data, such as images and videos. They perform convolutions to automatically learn features from input data and are widely used in computer vision tasks like image classification and object detection.
- **Recurrent Neural Networks (RNNs):** Recurrent Neural Networks (RNNs) are a type of neural network designed for sequential data processing. Unlike FNNs, RNNs have connections that form directed cycles, allowing them to maintain a hidden state that captures information about previous inputs in the sequence. This inherent memory capability makes RNNs well-suited for tasks involving temporal dependencies, such as natural language processing, time series analysis, and speech recognition. However, traditional RNNs face challenges in capturing long-term dependencies due to issues like exponentially diminishing gradients during backpropagation 2.3.2. Variants like Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks have been developed to address these challenges and improve the effectiveness of RNNs in handling sequential data.

- Long Short-Term Memory (LSTM): LSTMs (section 2.2.3) distinguish themselves from traditional RNNs by incorporating a more sophisticated memory mechanism that mitigates the vanishing gradient problem. LSTMs address this issue through a carefully designed memory cell and gating mechanisms. The memory cell, equipped with input, forget, and output gates, enables precise control over the information flow. The intricate architecture of LSTMs allow them to selectively store and access information over extended sequences, making them more adept at capturing long-term dependencies in data.

It is important to note that these different model types can also be a single layers within an overarching architecture, i.e. a DNN does not have to be exclusively designed in one particular way. Such hybrid DNNs can contain elements of multiple techniques and use their strengths at different layers of depth. FNNs, CNNs and LSTMs are explained in detail below.

2.2.1 Feedforward Neural Networks

Feedforward neural networks (FNNs) are classic models in the field of deep learning. The primary objective of a FNN is to approximate a given function f^* . For instance, in the case of a classifier, the function $y = f^*(x)$ maps an input x to a corresponding class y . Within a FNN, a mapping $y = f(x; \theta)$ is defined, and the network learns to optimize the parameters θ to achieve the most accurate function approximation. The term 'feedforward' comes from the sequential flow of information during the evaluation of the function, originating from x , traversing through the intermediate computations and ultimately arriving at the output y . Notably, there are no feedback connections within the model where outputs are fed back into the network itself. When FNNs are expanded to incorporate feedback connections, they are referred to as recurrent neural networks (section 2.2.3).

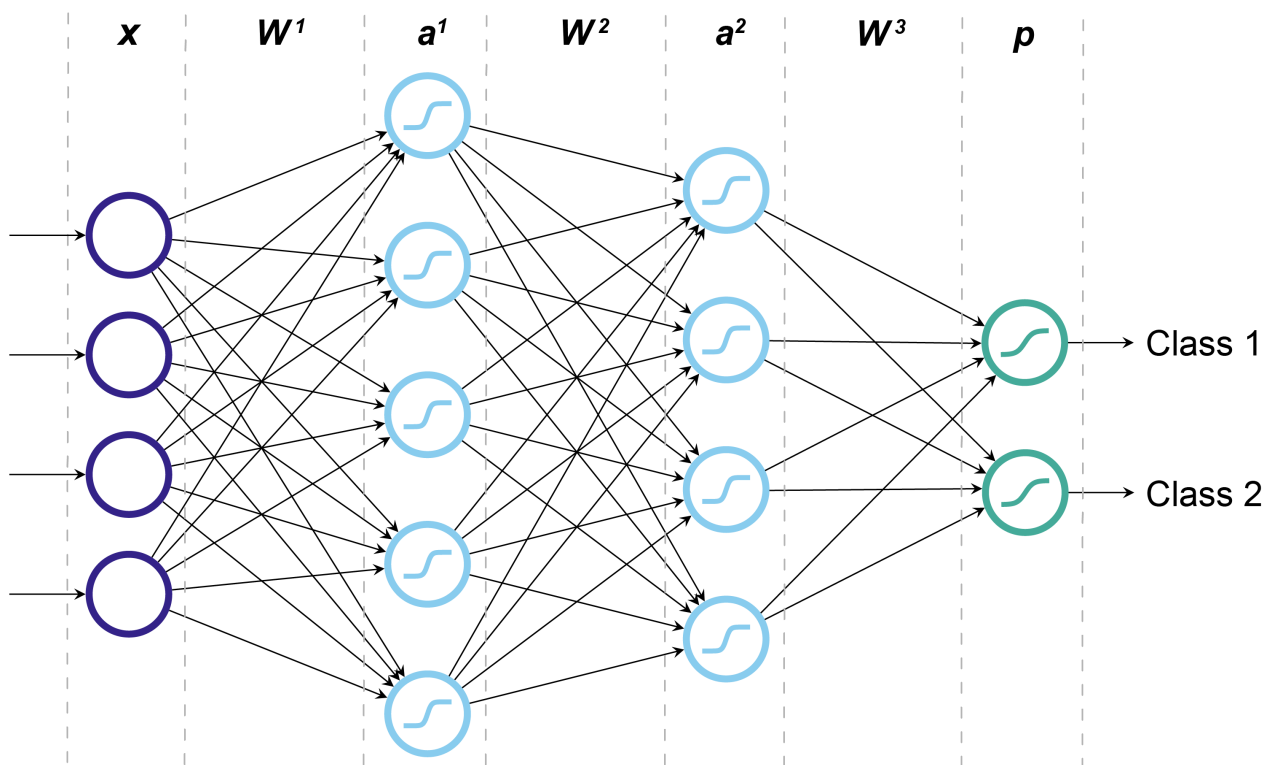


Figure 2.2 | Multilayer perceptron (MLP). The input data x , embedded in the first layer, is guided through the network and undergoes various mathematical operations, represented as neurons. In the final layer, all inputs are compressed and transformed into a probability value for each class.

FNN architecture and operations

The first conception of a feedforward neural networks (FNN) was a multilayer perceptron (MLP), also known as a dense neural network.¹ In a MLP, illustrated in Figure 2.2, all neurons are connected between two consecutive layers, which are commonly referred to as fully connected or dense layers. Therefore, output values from neurons in one layer serve as inputs to neurons in the next layer. While MLPs are less commonly used in modern deep learning models, they continue to serve as a foundational concept that leads to recurrent neural networks. The main structure of MLPs, consisting of an input layer, hidden layers, and an output layer, is also retained in modern convolutional neural networks (section 2.2.2). Additionally, their essential component, the fully connected layer, is still frequently used in the output layer of advanced classifiers. In the context of a MLP, several equations govern the flow of information and transformations within the neural network. Figure 2.2 shows the concept of a MLP architecture and introduces the following parameters:

- \mathbf{x} is the input data, which can be one- or multidimensional. Each element of \mathbf{x} typically corresponds to a specific input feature, i.e. a recorded intensity value of a camera.
- $\mathbf{W}^{(l)}$ represents the weights associated with the connections between neurons in the $(l - 1)^{\text{th}}$ layer and the l^{th} layer of a neural network. Each element $w_{ij}^{(l)}$ in this matrix corresponds to the weight of the connection between the i^{th} neuron in the $(l - 1)^{\text{th}}$ layer and the j^{th} neuron in the l^{th} layer.
- $\mathbf{a}^{(l)}$ contains the activation values of all neurons in the l^{th} layer of a neural network. These activations are calculated using a specific activation function (section 2.2.1) applied to the weighted sum of inputs to each neuron. The values in $\mathbf{a}^{(l)}$ represent the output of the neurons in that layer after processing the input.
- \mathbf{p} represents the probabilities of different classes in the final output layer of a classification model. Each element p_i in \mathbf{p} corresponds to the probability of the input belonging to the i^{th} class. These probabilities are often generated using a softmax activation function, ensuring that they sum to 1 and can be interpreted as probabilities. The class with the highest probability is typically predicted as the final output.

For the first layer of neurons, the weighted sum of inputs is calculated for each neuron as:

$$z_j = \sum_{i=1}^n (w_{i,j} \cdot x_i) + b_j \quad (2.1)$$

where z_j represents the weighted sum of the j^{th} neuron, $w_{i,j}$ is the weight of the connection between the i^{th} input and the j^{th} neuron, x_i is the value of the i^{th} input and b_j is the bias term for the j^{th} neuron. The bias term is a learnable constant that can be intuitively understood as the activation potential observed in biological neurons. Note that the input x_i changes to a_i for subsequent layers, as they take the output of previous neurons as input:

$$z_j = \sum_{i=1}^n (w_{i,j} \cdot a_i) + b_j \quad (2.2)$$

The weighted sum is then passed through an activation function to introduce non-linearity, as discussed in the next subsection 2.2.1. This was not the case in the original MLP, which employed simple step functions that did not allow the network to solve non-linear problems.

Activation functions

Activation functions play a crucial role in deep learning, as they allow for solving non-linear problems and have a significant influence on the learning process. They are continuously being developed to improve the training and performance of deep neural networks. Figure 2.3 summarizes modern activation functions that

have shown to be effective for specific types of data and address common challenges during the training procedure. Assuming sigmoidal activation functions, the output of a single neuron, a_j , can be expressed as:

$$a_j = \frac{1}{1 + e^{-z_j}} \tag{2.3}$$

In the final layer of a classifier model, the sigmoid activation function can be only used for binary classification and has to be extended for multi-classification. This is called the softmax function, which converts the weighted sums into probabilities of multiple classes. The softmax function for the k^{th} class is defined as:

$$p_k = \frac{e^{z_k}}{\sum_{n=1}^N e^{z_n}} \tag{2.4}$$

where p_k represents a probability that the output belongs to the k^{th} class, z_k is the unnormalized score of for class k and $\sum_{n=1}^N e^{z_n}$ is the sum of the exponential unnormalized scores over all classes. The probability-like value p is also commonly referred to as the confidence of a model.

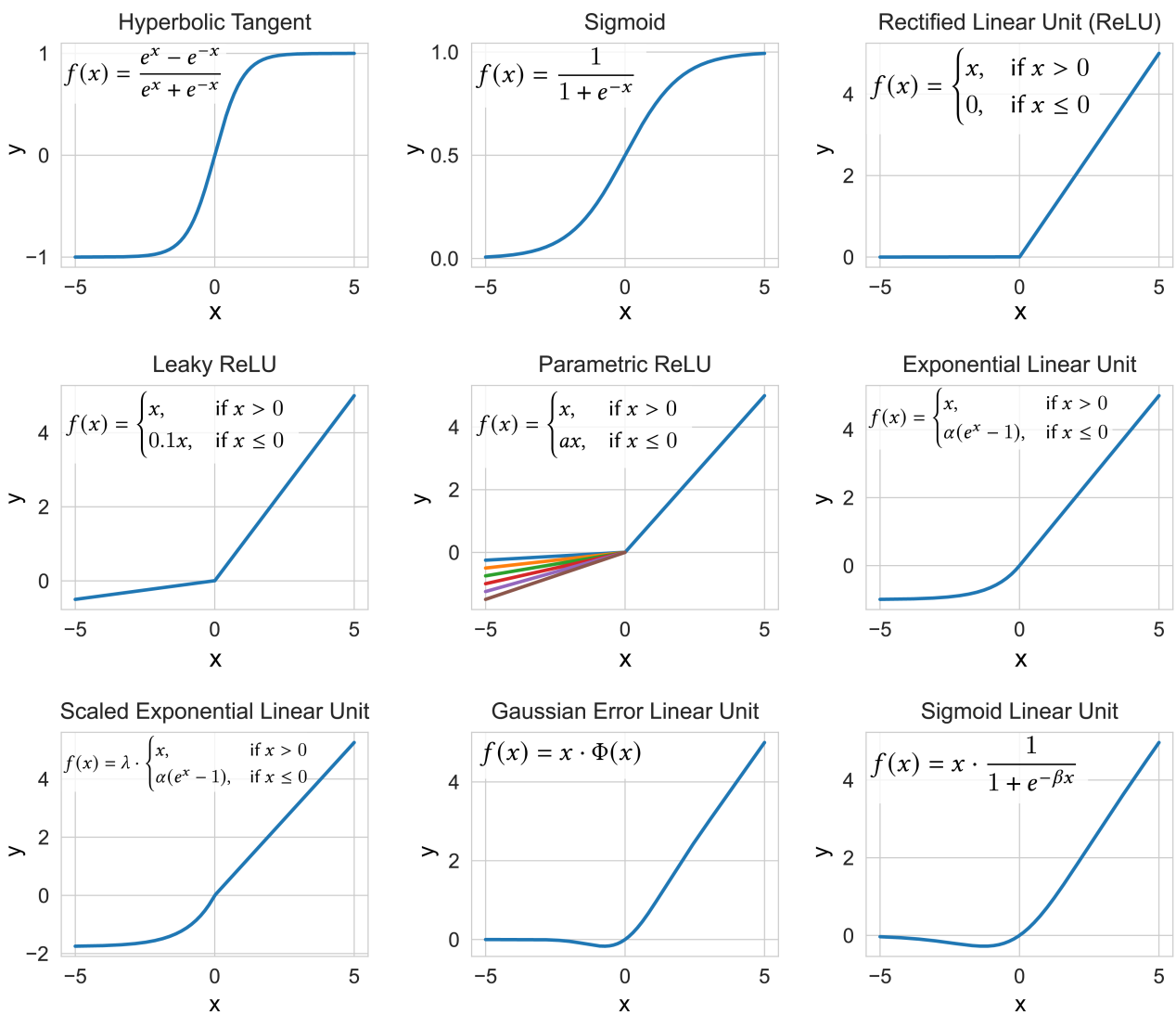


Figure 2.3 | Activation functions overview. Commonly used activation functions used in deep neural networks. Each function applies a non-linear transformation to the input data, allowing the model to learn complex relationships and enabling gradient-based learning.

2.2.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) represent a pivotal advancement in deep learning, particularly suited for tasks involving grid-like data such as images and videos.³² CNNs are inspired by the human visual system and have demonstrated remarkable performance improvements in tasks like image classification, object detection, and even natural language processing. They also offer a distinct advantage in terms of model efficiency, i.e. a reduced number of trainable parameters while retaining the performance. CNNs follow the same principles as feedforward networks (FNNs, subsection 2.2.1), being designed to approximate complex functions by undergoing training to optimize all network parameters and provide accurate classifications. The term 'convolutional' derives from the specialized layers and operations employed in the network's evaluation. In particular, neurons in a CNN consist of a small weight matrices, referred to as kernels, which scan the input data in a raster-like manner for feature extraction. The outputs of neurons are frequently collected in pooling layers (section 2.2.2) to pass on the most important features, a process that is often repeated until the final classification layer.

CNN architecture

Unlike traditional FNNs, CNNs incorporate convolutional and pooling layers, introducing a hierarchy to feature extraction and abstraction. This hierarchy allows CNNs to automatically learn and detect local patterns and features within the input data. The sequential flow of information within CNNs mirrors that of feedforward networks, originating from the input data and proceeding through these layers of computation to yield the final output. One of the key innovations brought by CNNs is the fundamental neuron operation, referred to as convolution, which allows the network to capture local patterns while sharing parameters across different regions. This operation not only enhances performance but also significantly reduces the number of parameters that need to be trained. Hence, the key distinction is the absence of fully connected layers in most parts of CNNs, which sets them apart from classical MLPs. However, modern CNNs often retain the core structure of MLPs, comprising input layers, convolutional layers, and output layers. The essential concept of CNNs is preserved in various deep learning architectures, hence they provide fundamental building blocks in the realm of computer vision and image processing. Furthermore, specialized CNNs have been recently developed to extend their application to time series data by using a variable filter size for their convolutional operations. Hence, they can also be employed to simultaneously identify patterns that appear on different timescales. Figure 2.4 shows the architecture of VGG16 (Visual Geometry Group)³³, an image classifier that achieved an increased depth by using small 3×3 convolutional filters.

Convolution and cross-correlation

The term convolution is widely used in the context of CNNs but technically is a misnomer since the actual mathematical operation performed in CNNs is the cross-correlation. Both convolution and cross-correlation combine two functions to produce a third function. The resulting function is an expression of how the shapes of the original functions influence each other. The key difference is that convolution involves reflecting one of the functions about the y-axis ("flipping") one of the functions before sliding it over the other and computing the integral of their dot product:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) \cdot g(t - \tau) d\tau \quad (2.5)$$

The cross-correlation performs the sliding dot product without reflecting and shifting one of the functions:

$$(f \star g)(t) = \int_{-\infty}^{\infty} f(\tau) \cdot g(t + \tau) d\tau \quad (2.6)$$

Although both functions are closely related, convolution is commutative, whereas cross-correlation is not. This non-commutative property of cross-correlation underlines the intuitive explanation of how CNNs utilize

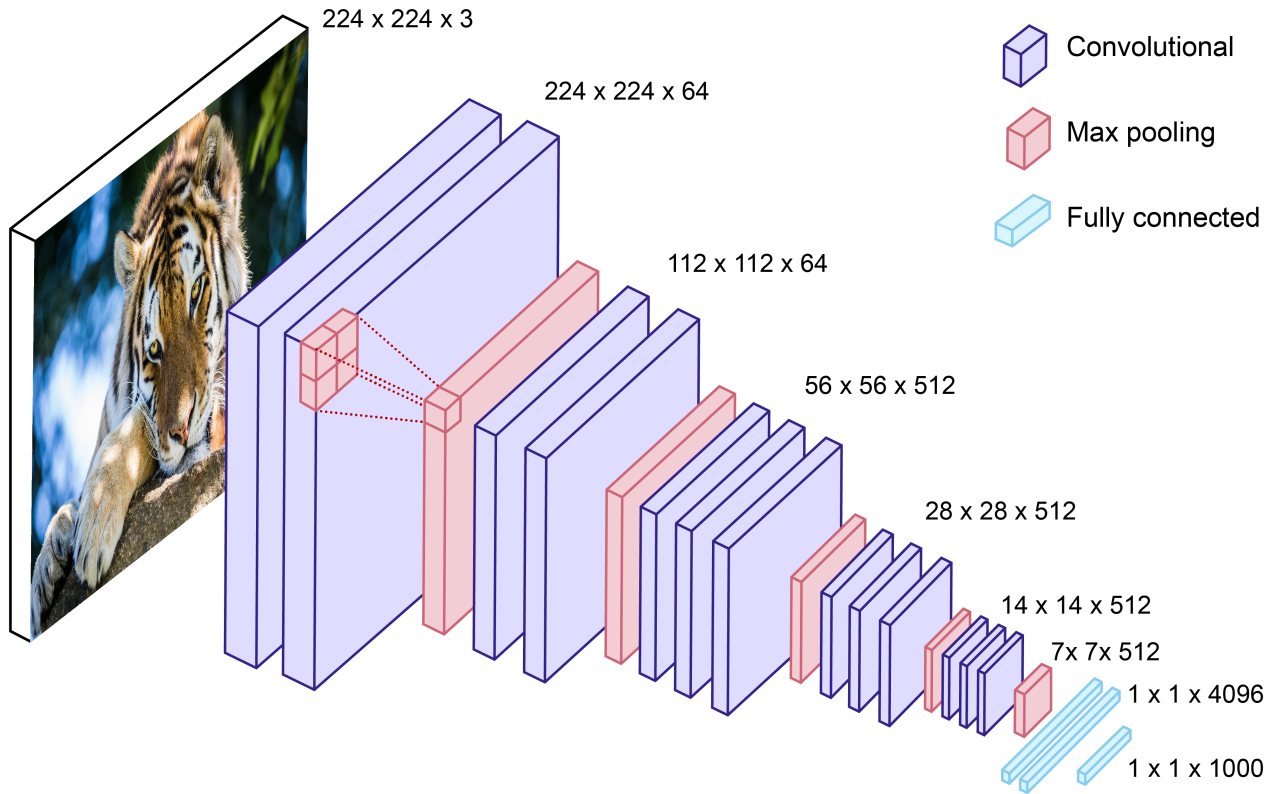


Figure 2.4 | CNN architecture of VGG16³³. The VGG16 model employs convolutional (blue), pooling (red) and dense (cyan) layers for image classification. The 2×2 pooling operation is illustrated in the first pooling layer. Each block of layers has specific dimensions, annotated to the right, showing the sequential compression of the data in the first two dimensions, i.e. width and height.

kernels to extract features from input data. Kernels are trainable weight matrices of fixed size and specified as a hyperparameter before training. Given an input feature map (I) and kernel (K) of size $m \times n$, the cross-correlation performed in CNNs can be expressed as:

$$(I \star K)(m, n) = \sum_m \sum_n I(x + m, y + n) \cdot K(m, n) \quad (2.7)$$

Here, the input feature map (I) can represent any multi-dimensional data such as colored images but also one-dimensional time series data. The cross-correlation operation is performed with a pre-defined step size across the input. The step size, also referred to as stride, and the kernel size directly influence the dimensionality of the output. Additionally, the input can be padded to increase the space for the kernel to cover the input such that the number of kernel operations and output dimensionality increases. In general, the output dimension of a CNN layer can be calculated as:

$$\text{Output Dimension} = \frac{\text{Input Dimension} - \text{Kernel Size} + 2 \times \text{Padding}}{\text{Stride}} + 1 \quad (2.8)$$

After cross-correlation, an activation function (section 2.2.1) is applied element-wise to introduce non-linearity. Assuming the rectified linear unit (ReLU) activation function is employed, the output feature map (O_j) of the j^{th} neuron in a given CNN layer can be expressed as:

$$O_j = \text{ReLU}(I \star K_j + b) \quad (2.9)$$

where K_j is the kernel matrix of the j^{th} neuron and b is the bias term. After scanning the whole input, the outputs of all neurons are concatenated and the next layer operates on the concatenated outputs. Upon reaching the final classification layer, the activated output of all neurons is typically flattened into a one-dimensional vector such that it can be passed through a fully connected layer and transformed into class probabilities via a softmax function (see 2.2.1 for details). An illustration of learned features by a CNN (VGG16³³) is shown in Figure 2.5 via the maximized activations of single convolutional filters. It becomes apparent that low level features are extracted first in the shallow layers and high level features are extracted last in the deep layers. By viewing the high level features one can draw conclusions about the data the DNN was trained on, e.g. a dataset containing various kinds of animals.

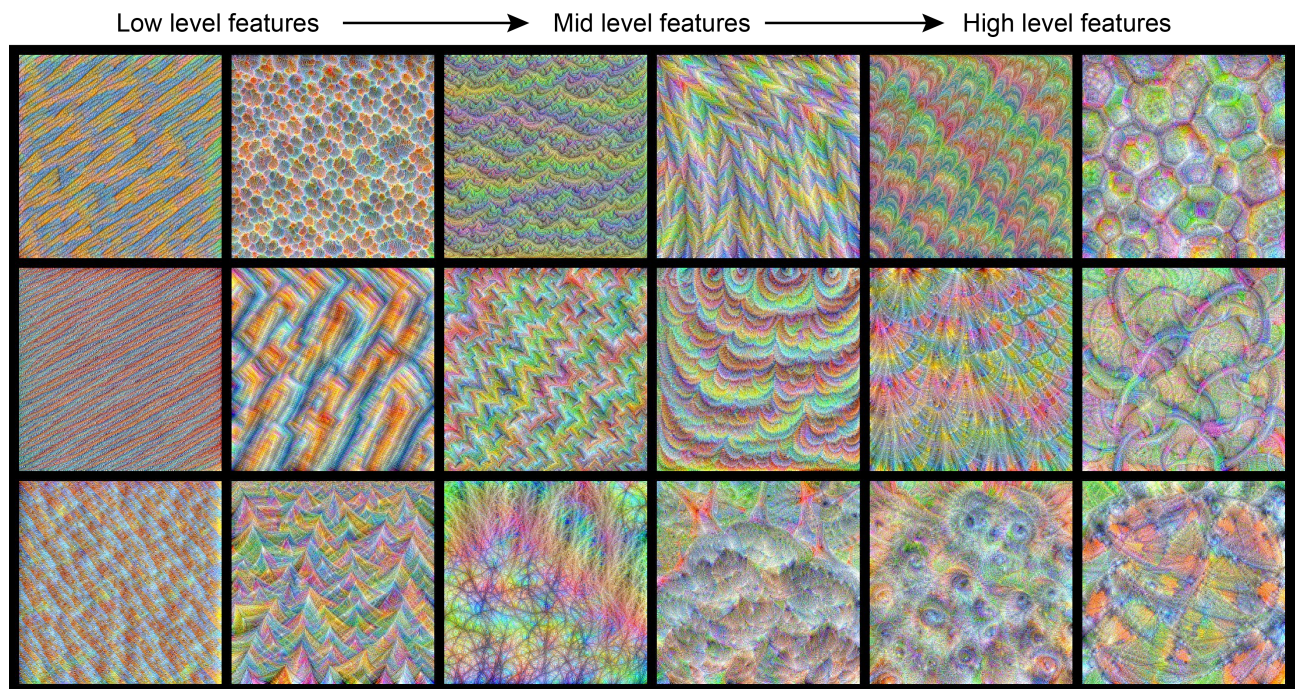


Figure 2.5 | Feature hierarchy of a CNN (VGG16³³). Visual representation of the hierarchical progression of features within VGG16. The network extracts increasingly abstract patterns from coarse to complex across its convolutional layers. The images represent the learned features of individual filters. Each image is generated separately through a training process with the objective of maximizing the activations of an individual filter.

Pooling operations

Pooling is a down-sampling technique used in CNNs to reduce the spatial resolution of feature maps while preserving important features. This operation helps in controlling the computational complexity of the network and increasing its ability to detect patterns at different scales. Two common types of pooling used in CNNs are Max Pooling and Average Pooling. In Max Pooling, for each local region (usually a small square) of the feature map, the maximum value within that region is retained while the rest are discarded. This process effectively emphasizes the most important features within each region. For a $k \times k$ window centered at position (x, y) in the input feature map (assuming a single channel), the max pooling operation is defined as follows:

$$\text{MaxPooling}(x, y) = \max_{p=0}^{k-1} \max_{q=0}^{k-1} (\text{Input}(x + p, y + q)) \quad (2.10)$$

Average Pooling, on the other hand, calculates the average value of the elements in a local region, providing a smoother down-sampling effect compared to Max Pooling. It is defined as:

$$\text{AveragePooling}(x, y) = \frac{1}{k^2} \sum_{p=0}^{k-1} \sum_{q=0}^{k-1} (\text{Input}(i + p, j + q)) \quad (2.11)$$

Note that these equations describe the process for pooling in a single channel. For multi-channel input feature maps, the same operation is applied independently to each channel, resulting in a pooled feature map with the same number of channels. There are several purposes for pooling layers. By reducing the spatial dimensions of feature maps, pooling helps decrease the number of parameters and computational load in subsequent layers. This is especially important when processing high-resolution images. Pooling also helps make the network invariant to small translations of the input. This is crucial for tasks like image recognition where the object's position in the image should not affect the network's ability to recognize it. By abstracting and summarizing information in the feature maps, pooling retains the most important features while filtering out less relevant details, making the network more robust to variations in the input. During training, pooling can reduce overfitting by preventing the network from memorizing exact details of the training data and promoting the learning of more general, high-level features. However, one trade-off when using pooling is the reduction in spatial resolution. While this is often beneficial for capturing high-level features, it can lead to a loss of fine-grained details, which may be especially important in timeseries data. To address this, some CNN architectures incorporate skip connections, which allow the network to access information from earlier layers with higher resolution. This type of model is called a Residual Neural Network (ResNet).³⁴ The core concept of using skip connections was already used in LSTM networks (discussed below) and is implemented in all modern deep neural networks such as ChatGPT, BERT³⁵ and AlphaFold¹⁰.

2.2.3 Long short-term memory (LSTM) networks

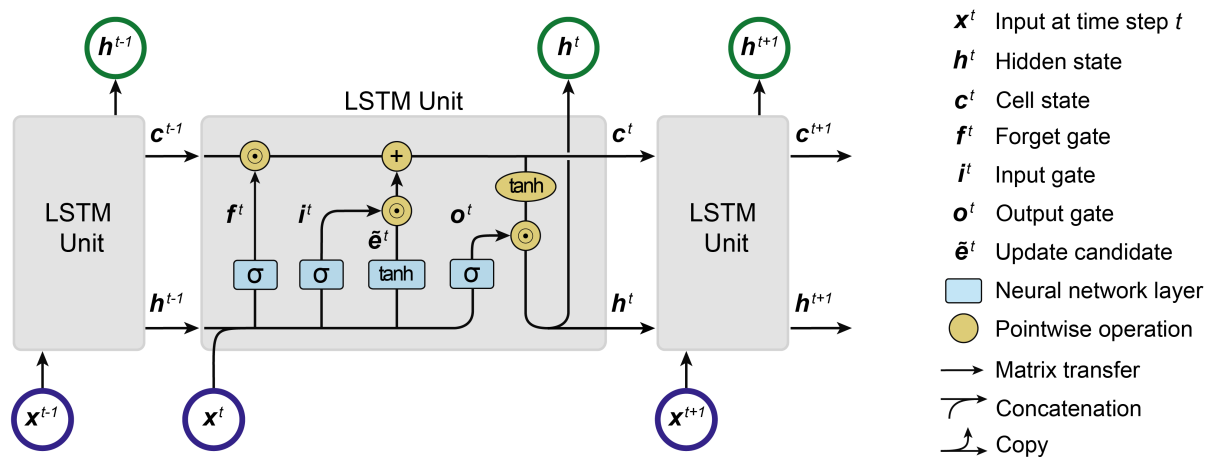


Figure 2.6 | Long short-term memory concept. The schematic illustrates the fundamental operations in a LSTM cell. The input data x is combined with the hidden state h of the previous time step and transformed via the forget gate and the input gate. The transformed data updates the cell state c , which can preserve and recall information across extended sequences. The cell state is again transformed in the output gate o , yielding the hidden state of the current time step. This hidden state serves as input for the next LSTM cell and for the next layer in the neural network.

Long short-term memory (LSTM) networks represent an advanced type of recurrent neural networks (RNNs), designed to model sequential data and capturing long-range dependencies.³⁶ In contrast to feedforward networks (FNNs, section 2.2.1), the emphasis of LSTMs lies on tasks where data unfolds over time. Within the architecture of an LSTM, a dynamic mapping is established, but the unique feature that sets LSTMs apart is their ability to preserve and selectively propagate information over extended sequences. LSTMs encompass

specialized recurrent layers that use gating mechanisms to facilitate the retention of important information and the discarding of irrelevant details. The flow of information within LSTMs mirrors the sequential nature of the data with each time step building upon the previous one. This allows the network to capture temporal patterns. Hence, in contrast to FNNs, the single LSTM units within one layer are interconnected and form feedback connections. In principle, a LSTM unit represents a neuron in traditional FNNs but it is composed of several interacting components:

- Cell state (c^t): The cell state acts as a memory cell that can store and retrieve information over long sequences. It runs straight down the entire chain of the LSTM, with only minor linear interactions. This cell state can store information that is considered important for the sequence.
- Hidden state (h^t): The hidden state is analogous to the output of the LSTM unit. It carries information that the network has deemed relevant up to the current time step. It is computed based on the cell state but filtered through the output gate as discussed below.
- Gates:
 1. Forget gate (f^t): The forget gate decides what information from the cell state should be discarded or kept. It takes input from the previous hidden state (h^{t-1}) and the current input (x^t) and produces values between 0 (completely forget) and 1 (completely keep).
 2. Input gate (g^t): The input gate controls what new information should be stored in the cell state. It consists of two separate layers with different activation functions (Figure 2.3): a sigmoid layer that decides which values to update (between 0 and 1) and a tanh layer that produces potential new values (between -1 and 1).
 3. Output gate (o^t): The output gate determines what the next hidden state (h^t) should be. It takes into account the cell state but filters it through a sigmoid layer.

The cell states and hidden states in each LSTM cell are continuously updated by the information passing through the forget gate, input gate and output gate. Each of these gates represent a neural network layer and have their own set of trainable parameters. These gate specific parameters are:

- U^{gate} : Weight matrix associated with the inputs x^t at time step t .
- W^{gate} : Recurrent weight matrix associated with the previous hidden state h^{t-1} at time step $t - 1$.
- b^{gate} : Bias term vector added to the weighted sums of inputs and weighted sums of the previous hidden state.

With the above defined gate parameters, the update equations can be formulated. Note that the following equations for the three gates represent the output of the j^{th} neuron within the corresponding gate layer. The forget gate (f^t) calculates a value for each element in the cell state, indicating the extent to which the corresponding information should be forgotten:

$$f_j^t = \sigma \left(b_j^f + \sum_i U_{i,j}^f x_i^t + \sum_i W_{i,j}^f h_i^{t-1} \right) \quad (2.12)$$

where σ represents the sigmoid activation function, x_i^t is the input of feature i at time step t and h_i^{t-1} is the i^{th} feature of the hidden state vector (h_{t-1}) of the previous time step.

The number of entries in the feature vectors of the hidden state and the cell state, i.e. their dimensions, is determined by the number of neurons employed in each layer and the number of features in each time step. The input gate (g^t) calculates a value for each element in the cell state, indicating the importance of new input information:

$$g_j^t = \sigma \left(b_j^i + \sum_i U_{i,j}^i x_i^t + \sum_i W_{i,j}^i h_i^{t-1} \right) \quad (2.13)$$

The update candidate (\tilde{c}^t) represents the new candidate values that could be added to the cell state at time step t . It is based on the input (x^t) and the previous hidden state (h^{t-1}):

$$\tilde{c}_j^t = \tanh \left(b_j^c + \sum_i U_{i,j}^c x_i^t + \sum_i W_{i,j}^c h_i^{t-1} \right) \quad (2.14)$$

where \tanh represents the hyperbolic tangent activation function.

The output gate (o^t) calculates a value for each element in the cell state, indicating the extent to which the corresponding information should contribute to the hidden state:

$$o_j^t = \sigma \left(b_j^o + \sum_i U_{i,j}^o x_i^t + \sum_i W_{i,j}^o h_i^{t-1} \right) \quad (2.15)$$

Finally, after the computation of all gates for all neurons, the cell state (c^t) and hidden state (h^t) are determined as follows:

$$c^t = f^t \odot c^{t-1} + g^t \odot \tilde{c}^t \quad (2.16)$$

$$h^t = \tanh(c^t) \odot o^t \quad (2.17)$$

where \odot symbolizes the Hadamard product (i.e. element-wise product) of two matrices. While the cell state is not directly transferred to the next layer it serves as an additional weight for the hidden state, which is the output of the LSTM. If the next hidden layer is a LSTM layer or the output is classified per time step via a fully connected layer, the hidden states are returned in sequence as depicted in Figure 2.6. However, LSTM models are also capable of classifying the complete time series as a single class by only returning the hidden state of the last time step. To date, many different LSTM variants have been developed, showing different advantages depending on the type of data. While the performance of a specific LSTM model cannot be predicted before training, numerous empirical studies and surveys have shown the strengths and weaknesses of all kinds of recurrent neural networks.³⁷

2.3 Learning Algorithms

A machine learning algorithm is characterized by its ability to learn patterns in data. However, the concept of learning in this context is extremely diverse and strongly depends on the specialized field within the realm of machine learning. Typically, learning algorithms propose a task for a computer program that is being fed with 'experiences', i.e. data of all kinds, and continuously advanced by using a particular performance metric. In the following sections, these principles are discussed in the context of training deep neural networks for classification tasks.

2.3.1 Initialization

The initialization of neural network parameters is a critical step in the training process, as it can significantly impact the convergence speed and the quality of the learned representations. A common practice is to initialize neural network weights using different probability distributions, each suited to specific network architectures and activation functions. Random initialization is the simplest and most widely used method. It initializes weights with values drawn from a random distribution, e.g. uniform or normal distribution. The choice of initialization method depends on the specific network architecture, activation functions, and the task of the model. Experimentation and hyperparameter tuning are often required to determine the most effective initialization strategy for a given neural network.

Glorot Initialization

Glorot initialization is a distribution-based method designed to address the problem of vanishing and exploding gradients, which can occur during training (section 2.3.2).³⁸ It is tailored to the sigmoid and hyperbolic tangent (tanh) activation functions. Glorot initialization samples weights from a uniform distribution defined as follows:

$$w_{i,j} \sim \text{Uniform}(-a, a) \quad (2.18)$$

with the limit a calculated as:

$$a = \sqrt{\frac{6}{n_{\text{input}} + n_{\text{output}}}} \quad (2.19)$$

where n_{input} and n_{output} are the number of input and output units, respectively.

He Initialization

He initialization is another distribution-based initialization method designed to alleviate the vanishing gradient problem for rectified linear unit (ReLU) activation functions.³⁹ It initializes weights by sampling from a normal distribution defined as follows:

$$w_{i,j} \sim \mathcal{N}(0, \sigma) \quad (2.20)$$

with the standard deviation σ calculated as:

$$\sigma = \sqrt{\frac{2}{n_{\text{input}}}} \quad (2.21)$$

where n_{input} is the number of input units.

Custom Initialization

In some cases, custom initialization schemes are designed based on domain-specific knowledge or network architecture requirements. Custom initialization can help accelerate training or achieve better convergence properties.

2.3.2 Backpropagation

Backpropagation, a core concept in neural network training, is the process that enables models to learn from data.⁴⁰ It involves the iterative adjustment of network parameters, namely weights and biases, to minimize prediction errors. Backpropagation is not limited to a specific network architecture, making it a universal tool for training various types of neural networks, including feedforward networks, convolutional neural networks, and recurrent neural networks.

Forward Pass

The first step is the forward pass, where input data is processed layer by layer through the neural network. This process involves calculating the weighted sum of inputs, applying activation functions, and producing predictions (section 2.2). The output of a neuron in the final classification layer can be expressed as:

$$p = \sigma \left(\sum_i (w_i \cdot x_i) + b \right) \quad (2.22)$$

where p represents the probability for a class, σ is the sigmoid activation function, w_i is the weight of the i^{th} connection associated with the input x_i and b is the bias term. With the output p and a given ground truth of the training data, an error can be calculated.

Loss Function

The loss function measures the error between the predicted output and the actual target values. The choice of the loss function depends on the specific task, such as regression or classification. Common loss functions include mean squared error (MSE) for regression and cross-entropy (CE) for classification. The CE Loss L is defined as follows:

$$L = - \sum_{i=1}^N y_i \log p_i \quad (2.23)$$

where y_i and p_i are the ground truth and the neuron output probability of the i^{th} , respectively, and N is the total number of classes. In the next step, the error is propagated backwards through the network.

Backward Pass

In the backward pass, gradients of the loss with respect to weights and biases are computed. The chain rule is applied for error propagation through the layers:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_i} \quad (2.24)$$

where L is the loss, w_i is the weight, a is the neuron's output and z is the weighted sum of all inputs of the neuron.

Parameter Update

The weights and biases are updated using an optimization algorithm, such as stochastic gradient descent (SGD)^{41,42} or adaptive moment estimation (ADAM)⁴³. In recent years, many different optimization algorithms have been published and continue to be developed.⁴⁴ In general, most methods adjust the parameters in the direction that minimizes the loss by computing the gradient of the loss with respect to a parameter.

$$\theta \leftarrow \theta - \alpha \cdot \frac{\partial L}{\partial \theta} \quad (2.25)$$

where θ is the parameter being updated in the network and α represents a step size, which is typically defined as a hyperparameter before training. Since ADAM is one of the commonly used optimization algorithms, it is discussed in more detail in section 2.3.3.

2.3.3 Adaptive Moment Estimation (ADAM)

ADAM is an advanced optimization algorithm that was introduced to overcome some of the limitations of traditional optimization methods such as stochastic gradient descent (SGD) or mini-batch gradient descent.⁴³ It belongs to the family of adaptive learning rate methods, where it dynamically adjusts the learning rates for each parameter during the training process. ADAM was built upon the principles of two other methods, namely Root Mean Square Propagation (RMSprop)⁴⁵ and Momentum⁴⁰. In essence, both methods are combined in ADAM by considering the first order moment (the gradient) and the second order moment (uncentered variance of the gradients). Additionally, ADAM introduces bias correction terms to further enhance its performance and stability during training. This adaptive learning rate mechanism allows ADAM to converge more efficiently and robustly when dealing with complex loss surfaces, making it particularly well suited for deep learning tasks. The individual steps of the algorithm start with initialization of all parameters

in the neural network:

$$t = 0$$

Initialize θ - Model parameters to be optimized.

α - Learning rate (typically a value around 0.001).

where θ are the parameters to be optimized. Note that θ represents a vector of possibly billions of parameters in a deep neural network and every optimization ADAM undertakes is specific to each parameter. The next steps involve the first forward pass and gradient calculation of the loss function with respect to the model parameters (see section section 2.3.2 for details):

$$\mathbf{g}_t = \nabla_{\theta} L(\theta_{t-1}) \quad (2.26)$$

where $\nabla_{\theta} L$ represents the gradient vector of the loss function with respect to the model parameters θ and \mathbf{g}_t is the gradient vector, consisting of the partial derivatives at time step t . Next, the first and second order moment estimates are calculated, i.e. the exponentially weighted moving average (EMA) of past gradients (\mathbf{m}_t) and the EMA of past squared gradients (\mathbf{v}_t):

$$\mathbf{m}_t = \beta_1 \cdot \mathbf{m}_{t-1} + (1 - \beta_1) \cdot \mathbf{g}_t = (1 - \beta_1) \sum_{i=1}^t \beta_1^{t-i} \cdot \mathbf{g}_i \quad (2.27)$$

$$\mathbf{v}_t = \beta_2 \cdot \mathbf{v}_{t-1} + (1 - \beta_2) \cdot \mathbf{g}_t \odot \mathbf{g}_t = (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \cdot \mathbf{g}_i \odot \mathbf{g}_i \quad (2.28)$$

where β_1 and β_2 are hyperparameters, representing the exponential decay rates for the first and second moment estimates. They are usually set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, which provides the best compromise between a smooth path to convergence and momentum to cross local minima or saddle points. In the early stages of training, the moment estimates are biased towards the initial estimate used in the first time step. Rather than using a random initial guess, the first and second order moments are initialized as $\mathbf{m}_0 = 0$ and $\mathbf{v}_0 = 0$, making them biased towards zero. However, this bias can be easily corrected by using the correction term $(1 - \beta_1^t)$:

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t} \quad (2.29)$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t} \quad (2.30)$$

With the bias-corrected moment estimates, the parameters of the previous time step (θ_{t-1}) are updated by the following rule:

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon} \quad (2.31)$$

where θ_t is the vector of resulting parameters and ϵ is a small constant (10^{-8}) added for numerical stability.

2.4 Regularization Techniques

In deep learning, regularization refers to a set of techniques used to prevent overfitting in neural network models. Overfitting occurs when a model becomes too complex and fits the training data too closely, leading to poor performance when encountering new data. Deep neural networks that not only fit training data well but also generalize effectively to new, unseen data is a central challenge. One of the fundamental issues faced in this endeavor is the need to strike a balance between complexity and simplicity in neural network architectures. Models that are too complex are prone to overfitting, where they memorize the training data

but fail to generalize. On the other hand, overly simplistic models may not capture the underlying patterns in the data. Regularization techniques play a pivotal role in addressing this challenge by imposing constraints on model parameters, thus preventing overfitting and promoting generalization. Specifically, the aim of regularizers is to find the best tradeoff between two key aspects: bias and variance. Bias refers to the error introduced by approximating a real-world problem, which may be complex, by a simplified model. In the context of regularization, introducing bias means making the model simpler. This can be achieved by adding constraints or penalties to the learning algorithm, making it less flexible or less capable of fitting noise in the data. Variance refers to the model's sensitivity to small fluctuations or changes in the training data. In the context of regularization, reducing variance means making your model less prone to overfitting by reducing its complexity. Figure 2.7 shows a visual representation of the tradeoff between bias and variance. It illustrates how increasing the bias of a model (by adding regularization) reduces its variance and vice versa. Furthermore, the graph indicates a straightforward technique of regularization, which is stopping the training process early when overfitting is detected. The goal is to find the right balance that minimizes the overall error on unseen data, which is the ultimate objective in deep learning. This cannot be achieved by training one particular model for an extended period of time due to the high degree of overfitting that neural networks are susceptible to. While underfitting models simply do not learn abstract features, overfitting models find features that do not correspond to a specific class but to an individual example or biases in the training data. Additionally, different classes often contain overlapping attributes. The training data may contain outliers or some examples in the data may be ambiguous. Since neural networks have plenty of parameters at their disposal, they are easily capable of "memorizing" the whole training data set and hence must be properly regularized to actually learn the class-relevant features.

2.4.1 L1 and L2 Regularization

L1 regularization, also known as LASSO (Least Absolute Shrinkage and Selection Operator), is a regularization technique that adds a penalty term to the loss function of a neural network model. The purpose of this penalty is to encourage the model to minimize the absolute values of its weights. Mathematically, L1 regularization can be expressed as:

$$\text{Loss}_{L1} = \text{Loss} + \lambda \sum_{i=1}^N |w_i| \quad (2.32)$$

where w_i is the model's weights and λ is the regularization strength controlling the variance-bias tradeoff. The main characteristic of L1 regularization is that it introduces sparsity in the model. It encourages some of the coefficients to become exactly zero, effectively performing feature selection. On the other hand, L2 regularization, referred to as Ridge regularization, achieves the opposite by penalizing the squared values of the model's coefficients. Hence, L2 regularization encourages the model to distribute the weight across all the features and is particularly useful when there are many correlated features in the dataset. Mathematically, L2 regularization can be expressed as:

$$\text{Loss}_{L2} = \text{Loss} + \lambda \sum_{i=1}^N w_i^2 \quad (2.33)$$

Both L1 and L2 regularization techniques provide a way to control the complexity of a machine learning model. The choice between them depends on the specific characteristics of the data and the desired trade-off between sparsity and stability. In practice, a combination of L1 and L2 regularization, known as Elastic Net, is often employed to leverage the advantages of both techniques.

2.4.2 Batch Normalization

One common problem that arises during training of a deep neural network is the issue of internal covariate shift, where the distribution of activations in hidden layers changes over time, making training slow and can

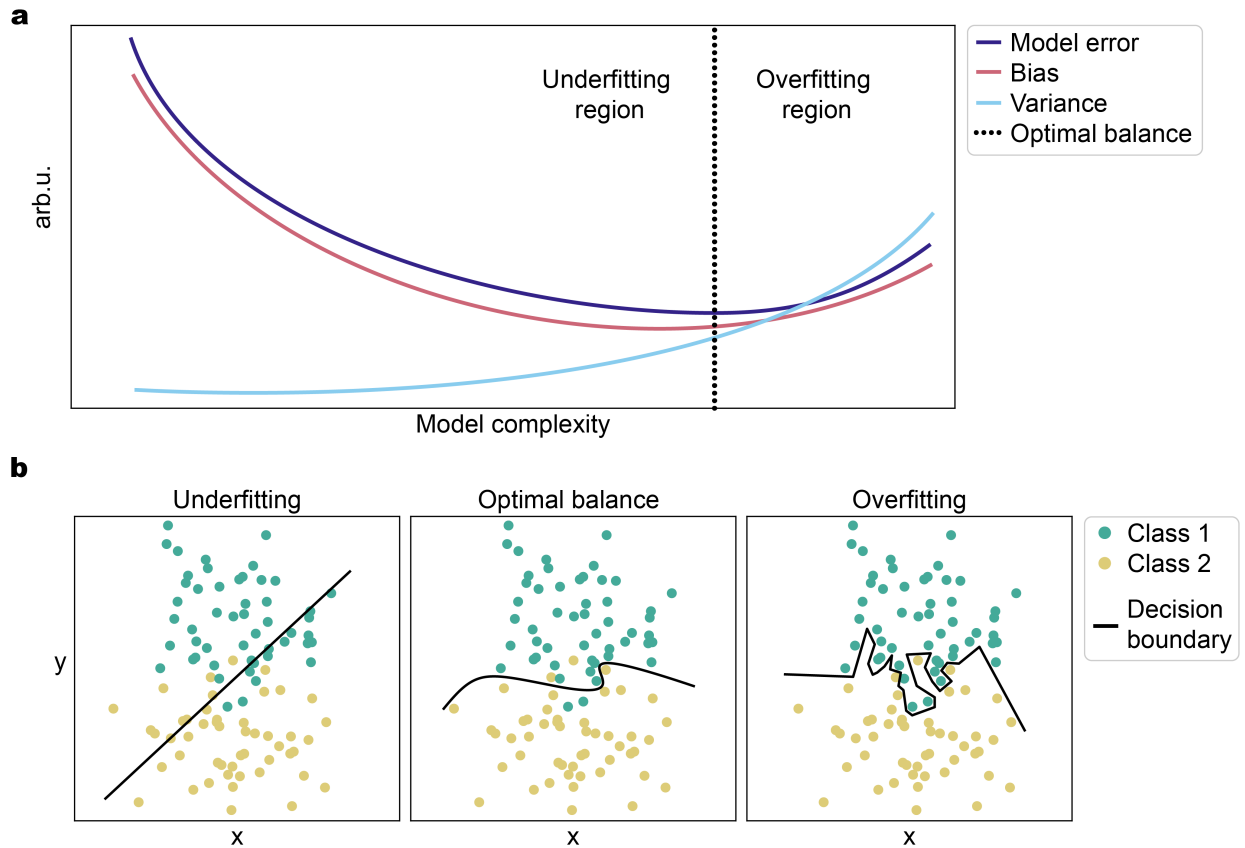


Figure 2.7 | The variance-bias tradeoff. (a) Illustration of the variance-bias tradeoff in a graph. As the model complexity increases during training, the error and bias decreases, whereas the variance increases. This progression typically reaches an optimal balance between the parameters followed by a continuous degradation of the model performance due to overfitting the training data. (b) Scatter plots showing two classes and decision boundaries of an underfitting (left panel), optimally balanced (middle panel) and overfitting (right panel) model. Due to outliers and inherent feature overlap, a balanced and regularized model outperforms an overfitting model, which merely 'memorized' the training data by capturing properties of individual samples that are not representative of the corresponding class.

lead to instability. This is commonly referred to as internal covariant shift. Batch Normalization (BatchNorm) is a technique designed to address this problem by normalizing the activations within each mini-batch. Here, mini-batch refers to the number of individual samples that are fed into the neural network at each iteration. Specifically, BatchNorm normalizes the input to have a mean of 0 and unit variance. For each mini-batch of size m , the mean (μ) and variance (σ^2) are calculated for each dimension of the input (\mathbf{x}):

$$\mu = \frac{1}{m} \sum_i^m x_i \quad \text{and} \quad \sigma^2 = \frac{1}{m} \sum_i^m (x_i - \mu)^2 \quad (2.34)$$

The normalized input of all features in one dimension (\hat{x}) is then given by:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (2.35)$$

where x are the original input features, μ is the mean of the mini-batch and σ^2 is the variance of the mini-batch. To restore the representational power and add flexibility in the network, the normalized data is transformed using two learnable parameters, namely the scaling factor (γ) and the shifting factor (β). This enables

the network to adjust the amplitude and center position of the output as follows:

$$\mathbf{y} = \gamma \hat{\mathbf{x}} + \beta \quad (2.36)$$

where \mathbf{y} is the transformed output, which is subsequently passed to other network layers. BatchNorm has become a standard component in modern deep learning architectures. It is commonly applied before activation functions in convolutional and fully connected layers. Overall, BatchNorm improves the training stability, regularizes the model, accelerates convergence and reduces the sensitivity to the choice of the weight initialization.

2.4.3 Dropout

Dropout is a simple yet effective regularization technique that combats overfitting by introducing random noise during training.⁴ The idea behind Dropout is to randomly deactivate, or "drop out", a fraction of neurons in a neural network during each forward and backward pass. This prevents the network from relying too heavily on any individual neuron or feature. In each iteration, only a subset of the total architecture is trained and hence the training process essentially draws from a distribution of network architectures. Therefore, Dropout provides an ensemble effect, emulating the behavior of training multiple models at once and taking their average. As a result, the final neural network model is more robust, generalized better and is less likely to overfit. In detail, Dropout randomly sets a fraction of neurons to zero. The fraction, referred to as the dropout rate, is typically set between 0.2 and 0.5. The choice of the dropout rate depends on the specific task and architecture. Mathematically, the input is multiplied by a binary mask with the same dimensions as the input. This mask is newly created for each iteration using the dropout rate, i.e. a new set of neurons are deactivated for each mini-batch.

2.4.4 Data augmentation

Data augmentation involves creating new training samples by making reasonable alterations to the existing data. The newly generated samples then become part of the entire training data set including the validation data set. Especially in case of purely synthetic training data, data augmentation is one of the most important methods to address biases and improve the performance of the neural network model. Depending on the type of data, the alterations can include various transformations such as:

- Geometric transformation: Rotations, translations, scaling and cropping.
- Noise: Adding random noise drawn from appropriate distributions.
- Flipping: Mirroring the input data horizontally or vertically.
- Perturbations: Replacing parts of the input data or implementing reasonable perturbations.

The specific choice of augmentations strongly depends on the type of data and the task. Therefore, data augmentation demands a comprehensive understanding of the dataset and expertise in the relevant field of application for neural networks.

2.4.5 Label smoothing

The idea behind label smoothing is to introduce a degree of uncertainty into the training labels used to supervise the neural network during training.⁴⁶ In supervised learning techniques, the data is not labeled as an ascending number sequence but with a binary vector. This vector has the same length as the number of unique classes and contains all zeros except for a single one at the index corresponding to the assigned class. This method is called (hard) one-hot encoding of categorical data. However, instead of assigning hard one-hot labels (i.e. a single class with a label of one and all others with labels of zero), label smoothing redistributes the

label probabilities among the classes. This prevents the neural network from becoming overly confident in its predictions since no single class can be predicted with 100 % certainty. Besides improving the generalization to unseen data, label smoothing reduces the penalty of erroneously labeled samples that may exist in the training dataset. Additionally, it allows neural networks to learn an overlap of classes and various probability distributions of the class labels in a supervised manner.

2.4.6 Pruning

Pruning in deep neural networks is inspired by synaptic pruning in the human brain, a natural process where axons and dendrites decay, leading to synapse elimination between early childhood and puberty.⁴⁷ Similarly, the goal of pruning is to eliminate redundant or less informative parts of a neural network, creating a more compact and streamlined architecture. The difference to Dropout (section 2.4.3) is that pruning the complete removal of certain parts of the neural network after training while dropout only temporarily deactivates neurons during training. However, this has to be achieved without significantly compromising the performance. The selection of which elements to prune can be guided by various criteria such as weight values close to zero and redundant activation patterns. Based on these metrics, entire neurons, filters or layers can be eliminated in the final model. In most cases, unnecessary variables in the model can only be identified during or after training. Therefore, pruning is an iterative process that can entail time-consuming fine tuning. While the technique is effective, the elimination process during pruning often leads to a loss in model performance that has to be recovered. Therefore, proper validation and evaluation are crucial to ensuring that pruned models maintain their performance while becoming more efficient.

2.5 Training and Monitoring

This section serves as a general guideline on how the training process of deep neural networks can be approached and describes an exemplary parameter distribution of a neural network trained on simulated single-molecule FRET intensity trajectories. Training neural networks is an iterative process and monitoring the learning progress is crucial to achieve the desired performance while prevent overfitting and ensuring that the model generalizes well to unseen data. Before training, the data and their labels are separated into a training and validation set. The training set is used for the actual learning procedure and the validation set is used as unseen data for testing the prediction accuracy after each epoch (training cycle). One epoch is complete when the full training set has been used once. During an epoch, the progression of the training loss can be monitored since the training data is fed into the neural network in small batches, which is an important aspect of introducing noise into the learning procedure and prevent overfitting. After one epoch, the validation loss is calculated by classifying the full validation data set. When the validation loss eventually plateaus, the learning rate can be reduced and/or the batch size increased to reduce noise and converge faster to the global minimum.

2.5.1 Hyperparameters

While the loss function can be visualized as a surface, the relationship between the network parameters and the loss function is high-dimensional. This loss landscape can be extremely complex, with multiple local minima, saddle points and regions of high curvature. Optimization algorithms (section 2.3.3), initialization strategies (section 2.3.1) and regularization techniques (section 2.4) are employed to navigate this high-dimensional landscape and find the global minimum. However, all these methods involve specifying hyperparameters, i.e. parameters that do not change during training one particular model. Any of these hyperparameters, can act as a bottleneck for a model's performance. Furthermore, every trained neural network is unique due to the large number of involved parameters and near infinite ways the error landscape can be traversed. Hence, a given model is trained not only once but multiple times using the same hyperparameters. Identifying limiting factors and finding the right combination of hyperparameters is a central task

that constitutes an additional, iterative learning process layered upon the training of the deep neural network itself. Note that hyperparameters can include parameters that define the architecture the deep neural network such as the depth, the number of neurons, dropout rates, convolutional strides or filter sizes and the initialization technique, and can also include parameters that only influence the training process such as mini-batch size, learning rates and the type of optimizer used. Several techniques can be used to search for optimal hyperparameters, for example:

- **Grid Search:** Grid search involves defining a set of values or ranges for each hyperparameter and exhaustively searching all possible combinations. While this method can be computationally expensive, it's a simple and systematic approach.
- **Random Search:** Random search selects hyperparameter values randomly from predefined ranges. It is more efficient than grid search and often provides good results with fewer evaluations.
- **Bayesian Optimization:** Bayesian optimization is a more sophisticated approach that models the objective function (e.g. validation accuracy) and uses probabilistic models to guide the search for optimal hyperparameters. It can be more efficient than grid or random search.
- **Genetic Algorithms:** Genetic algorithms use a population-based approach to evolve sets of hyperparameters over multiple generations. Through selection, crossover, and mutation operations, genetic algorithms adaptively explore the search space. Essentially, an evolutionary process is emulated by training multiple models and identify the best performer via natural selection.
- **Ensemble Methods:** Ensemble techniques combine predictions from multiple models trained with different hyperparameters. By leveraging the diversity of these models, ensembles can improve performance. However, this requires additional computation time for predictions since the data has to be fed into multiple models.

The choice of which technique to use depends on factors such as the computational resources available, the complexity of the problem, and the specific characteristics of the dataset. A reasonable approximation for the first range of hyperparameters can also be found in machine learning libraries and histories of similar tasks. Of course, since deep learning is rapidly evolving and naturally requires trial and error methodologies, staying informed about the latest publications and advances in the field is key to efficiently optimize the training process.

2.5.2 Model inspection

Monitoring the training of deep neural networks is crucial to ensure that models converge effectively and to diagnose issues when they arise. While the training and validation loss provide straightforward insights into a model's performance, they only scratch the surface of what can be understood about the training process. The distribution of model parameters, particularly weights and biases, are at the core of a neural network's functionality. Hence, examining the evolution of these distributions during training can reveal important aspects about the model's behavior. In general, the parameter distributions are influenced by various factors, such as the training data, the initialization and preceding layers. The parameter distributions alone cannot reveal an underfitting or overfitting model and need to be interpreted together with the training and validation metrics. However, they can help identify underperforming substructures in the neural network architecture and performance bottlenecks. The following graphs (Figures 2.8 - 2.11) show various parameters of the training procedure of a hybrid CNN-LSTM model, developed for classifying single-molecule intensity trajectories in FRET experiments.⁴⁸ For illustrative purposes, the size of the training data set was deliberately reduced to increase the number of necessary epochs for convergence.

Loss and accuracy

Figure 2.8 displays the training loss and accuracy for every epoch. The accuracy is only based on correct and false class predictions, whereas the loss encompasses the error of both false and accurate predictions. For example, a training sample can have a ground truth label of 0.95 for one particular class and the deep neural network can accurately predict this class but only with a probability value of 0.6. At the beginning of training, the validation loss is typically higher than the training loss and can be unstable. When the model continues to be well regularized, the validation loss follows the training loss since the model learns to generalize the desired features. However, even with several regularization employed, the validation and training loss may diverge after a number of epochs. The divergence point in Figure 2.8 is indicated at 62 epochs. While the validation loss still slightly decreases in the subsequent epochs, the model starts to overfit the training data, as can be seen by no further improvement in the predictions of the validation dataset.

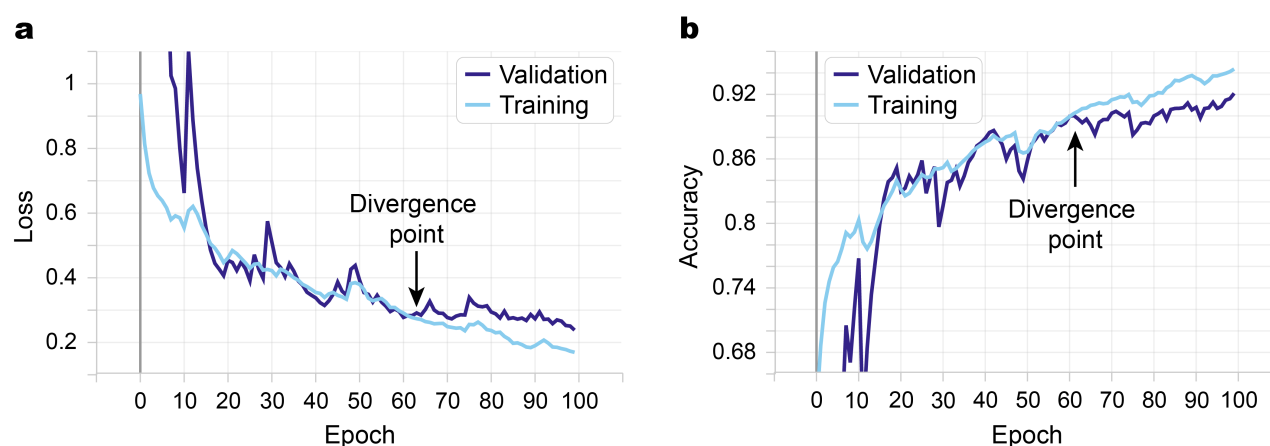


Figure 2.8 | Loss and accuracy during training. The loss (a) and classification accuracy (b) are monitored for each epoch. The training metrics (cyan) are calculated based on the last batch of the training data set while the validation metrics (blue) are calculated based on the complete validation data set that is classified after each epoch. For a well regularized model, the validation metrics closely follow the training metrics up a divergence point (black arrow). Training beyond this point leads to an overfitting model and does not improve its performance for classifying unseen data.

CNN weight distributions

The trained model is composed of three one-dimensional CNN layers followed by two LSTM layers and a fully connected classification layer. Here, one-dimensional refers to the directionality of the CNN and not the individual kernel sizes, which can still be two-dimensional. Since the training data set is based on 2-color ALEX data, there are three channels that are analyzed simultaneously by the model. Figure 2.9 shows the weight distributions of different kernel sizes in the first layer for each epoch. The purpose of using a range of kernel sizes at the same level of depth is to extract important features on different time scales. It can be seen that all weight distributions are approximately centered around zero and mostly show an increasing tail to negative values as the training progresses. This is a typical phenomenon for neural networks that learn to disregard most of the noise in the data and mainly focus on distinct features. The shape of the weight distribution approximates a normal distribution with increasing kernel size. The sparsely distributed weights of smaller kernels originates from the fact that single-molecule intensity trajectories are inherently noisy and only short events hold important information for these kernels. Sparse distributions indicate distinct and easily recognizable features and the information extracted by small kernels can be useful for the following LSTM layer. However, the distributions for small kernels in the first layer remain unstable throughout the training process. Hence, removing these layers in the next training cycle reduces the number of parameters that need to be trained and may improve the convergence time without sacrificing performance. On the other hand, larger kernels learn to recognize features within a larger time frame and

are capable of removing noise more effectively than smaller kernels. Their weight distributions are stable and show a continuously increasing variance. However, larger kernels are at a higher risk of becoming too specialized to certain features, exploiting biases in the training data and leading to a model that overfits the data. Depending on the number of employed neurons in a layer, overfitting kernels may not be identifiable by only viewing their weight distributions. A static variance and extreme values are typically the first indicators for a lack of learning and overfitting, respectively. In general, the weights have to be considered together with the bias, the activation function and the batch normalization parameters, which can significantly transform their values. For example, if the neural network shifts most of the weights to negative values via the batch normalization layer, it has learned to render this particular layer useless by turning all outputs to zero.

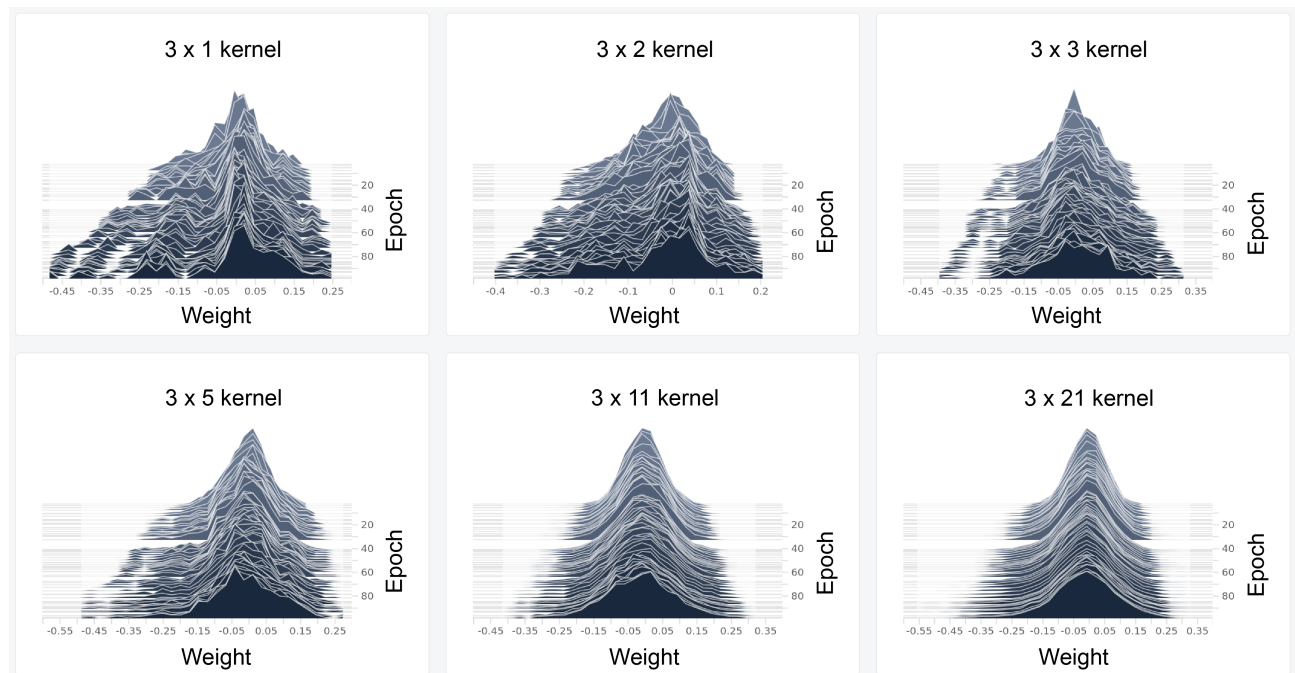


Figure 2.9 | CNN weight distributions. Weight distributions of differently sized kernels employed in the first layer of the neural network, shown for the whole training process. With increasing kernel size, the weight distribution approximates a normal distribution, showing a higher stability than the weights of smaller kernels. All distributions increase in variance and shift to negative values on average, signifying that the neural network assigns positive weights only to important features, filtering them from random noise.

LSTM parameter distributions

Figure 2.10 shows the parameter distributions of the last LSTM layer in the neural network. In this architecture, each LSTM layer is bidirectional, i.e. each sequence is processed from both directions. After the full time series is processed, the forward and backward layer outputs are concatenated before being transferred to the next layer. The biases in the input and output gates are initialized with zeros since their primary roles is to control the inflow and outflow of information. Here, the emphasis lies on importing and exporting only important features. During training, there is a shift in the distribution mean towards negative values, indicating that the network learned to focus on specific features and disregards most others. Conversely, the forget gate biases are initialized with values of one. This choice ensures that, in the initial stages, the network preserves all available information, learning to forget only when necessary. As training progresses, the distribution of forget biases gradually smooths out and the average bias values increase. This trend implies that the network deems every time step valuable throughout the entire sequence while adjusting their significance accordingly. When examining the weights across all layers, they exhibit a normal distribution centered around zero, displaying an increasing variance as training continues. Most importantly, the distributions are consistent and show a stable development over training epochs. For example, a static uniform

distribution of input weights would indicate that the LSTM layer does not capture important features and relies on other parameters for classification or prediction.

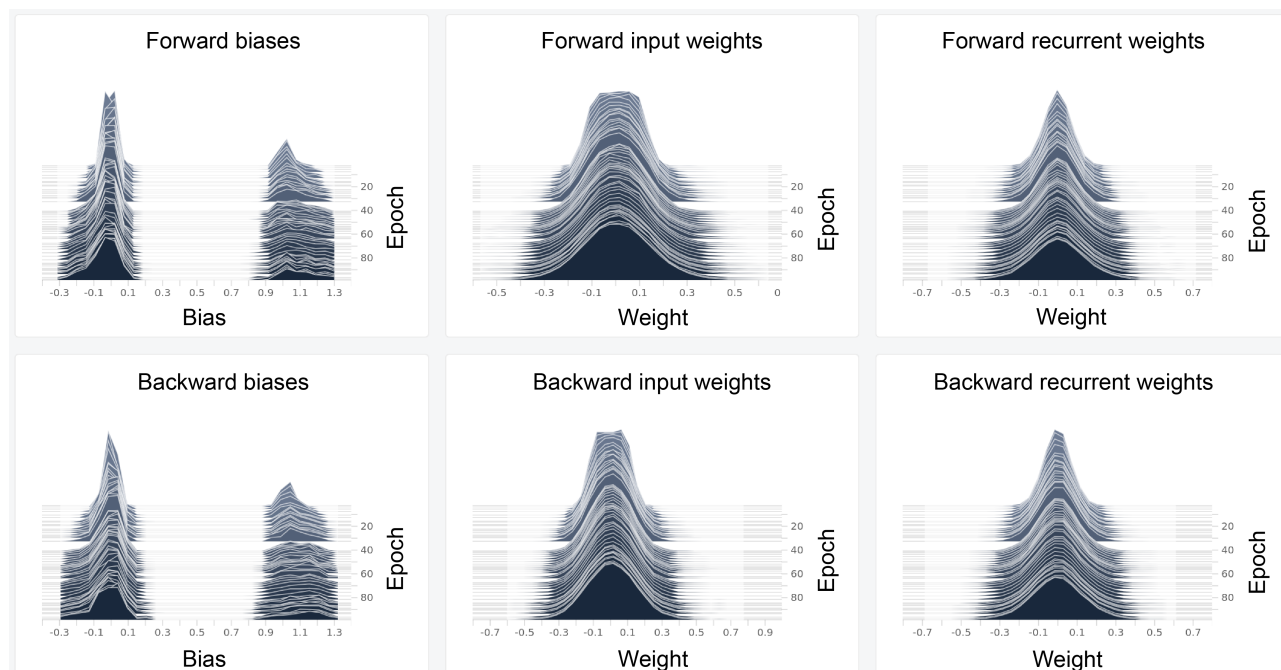


Figure 2.10 | BiLSTM parameter distributions. Evolution of biases and weights in a bidirectional LSTM layer (BiLSTM), processing a given sequence forward and backward. All distributions show a stable development with increasing variance during the training process. The biases of the input and output gates are initialized with zeros. During training, they show increasing negative values and hence an increase in suppression of redundant information. In contrast, the forget gate biases are initialized with ones and flattened during training to values between 0.8 and 1.3. This indicates that all information in the sequences is retained with a scaled importance. The weights are all approximating a normal distribution centered around zero, indicating a well balanced inclusion, suppression and retention of features.

Dense layer parameter distributions

Figure 2.11 shows the parameter distributions of the dense layer, serving as the final classification layer. It consists of eight neurons and is fully connected to the previous LSTM layer. The sparsity of the bias distribution, with values well separated between positive and negative, suggests that the individual neurons in the dense layer are serving distinct classification decisions. This also implies that the neurons are specialized in their response to different aspects of the input data, contributing to the overall discriminative power of the neural network. Positive bias values indicate that certain neurons are more inclined to activate for particular patterns or features in the input data. This could imply that these neurons have learned to respond strongly to specific cues relevant for the classification task. Negative bias values indicate neurons that are biased towards not activating for certain input patterns, suggesting they play a role in suppressing specific features or patterns. Note that each neuron only has one bias term, which means negative bias values make it more likely that a neuron will output low activation scores for specific input patterns, effectively reducing the likelihood of classifying an input as the associated class. The weight distribution of the dense layer resembles two overlapping normal distributions centered around zero. This reveals that the layer has evolved to have subgroups of neurons that are responsive to different subsets of features. With one population comprised of mostly negative values, it is likely that the neural network learned to recognize the random noise that is omnipresent in single-molecule intensity trajectories. By assigning negative weights to certain features, the

network can reduce the impact of noise or unwanted variations in the input data. The other population consists of more positive values and it is likely, that these weights are associated with features that correspond to a specific class and hence result in higher activations.

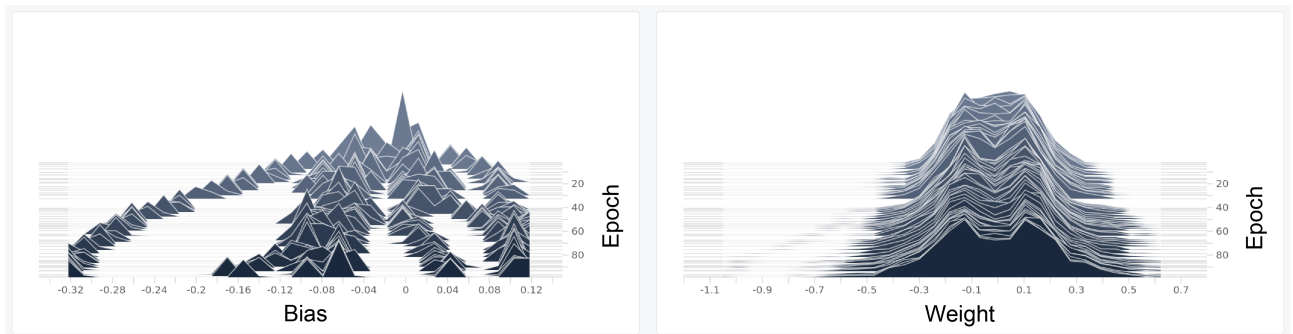


Figure 2.11 | Dense layer parameter distributions Bias distributions (left panel) and weight distributions (right panel) of the last classification layer for eight classes, consisting of eight neurons that are fully connected to the previous layer. During training, the zero-initialized biases are increasingly separated and contribute to the decision boundaries of the neural network for each class. Positive bias values indicate that the neuron is biased towards recognizing patterns associated with a specific class, while negative bias values indicate a bias against classifying input patterns as that class. The uniform initialized weights develop into two main populations resembling two overlapping normal distributions, suggesting that the neural networks learned to make a relatively clear distinction between important features and random noise.

Chapter 3

Fluorescence

3.1 Principles of fluorescence

Fluorescence is a versatile phenomenon that has emerged as a powerful tool in the realm of scientific research and technology over the past century. This phenomenon, which encompasses the emission of light by certain substances when they absorb photons of a specific wavelength, has found applications in a wide range of fields in chemistry, physics and biology. Its ability to provide valuable insights into the properties of matter at various scales has made fluorescence an important technique for investigating intricate processes at the molecular and cellular levels. The central feature of fluorescence is the interaction between incident light and a substance capable of fluorescing, often referred to as a fluorophore. When a fluorophore absorbs photons, it transitions into an excited state and, upon returning to its ground state, it releases energy in the form of lower-energy photons. This emitted light is characterized by its longer wavelength, typically within the visible spectrum, making it easily detectable. The specific spectral characteristics of the emitted light, including its intensity, lifetime, and polarization, hold information about the environment, structure, and dynamics of the fluorophore. Hence, fluorescence methods allow the observation of processes at the molecular level, making it particularly useful for studying cells, DNA and proteins.

3.1.1 Energy diagram

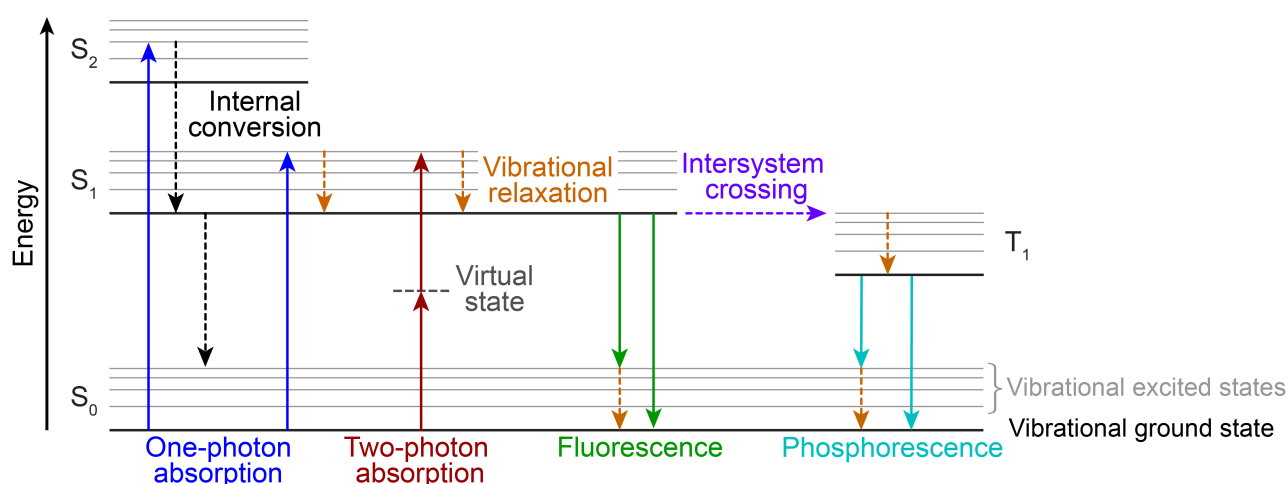


Figure 3.1 | Jablonski diagram. Excitation and relaxation pathways of one-photon (blue) and two-photon (red) absorption, vibrational relaxation (dark red), internal conversion (black), fluorescence (green), phosphorescence (orange) and intersystem crossing between the excited singlet and triplet states (purple). Radiationless transitions are represented by dashed lines.

Fluorescence refers to the spontaneous release of electromagnetic radiation shortly after excitation of a fluorescent molecule or fluorophore. Figure 3.1 depicts a Jablonski diagram of various processes.⁴⁹ At room temperature, the vast majority of fluorescent molecules will be in the ground state S_0 . When a fluorophore

absorbs a photon of a specific wavelength, a molecular orbital electron, typically a π electron, moves from the S_0 state to a higher electronic state such as S_1 or S_2 . While not specifically relevant in this thesis, the excited state can also be reached in other ways. For example, two photons with the combined energy matching the gap between the ground and excited state can be absorbed via an intermediate virtual state. At each electronic state, the electron can occupy multiple vibrational energy levels. Due to frequent collisions with surrounding molecules, the fluorophore relaxes radiationless to the vibrational ground state of the respective electronic state on a picosecond timescale. If the vibrational ground state is that of a high electronic state, e.g. S_2 , a transition to a lower electronic state occurs with high probability due to the closely spaced higher electronic energy levels. This process is termed internal conversion and describes radiationless transitions between energy states with the same spin. The emission of photons during relaxation from energy states of the same spin state, e.g. S_1 to S_0 , is termed fluorescence and occurs on a timescale of nanoseconds. In case of relaxation between electronic states with different spin states, e.g. T_1 to S_0 , the emission is termed phosphorescence. Since this spin forbidden process occurs with a lower probability than fluorescence, phosphorescence can have much longer lifetimes, typically in range of microseconds but lifetimes up hours are also possible.⁵⁰ The triplet state, T_1 or higher triplet levels, can be occupied via intersystem crossing, a radiationless transition between different spin states.

3.1.2 Franck-Condon principle and Stokes shift

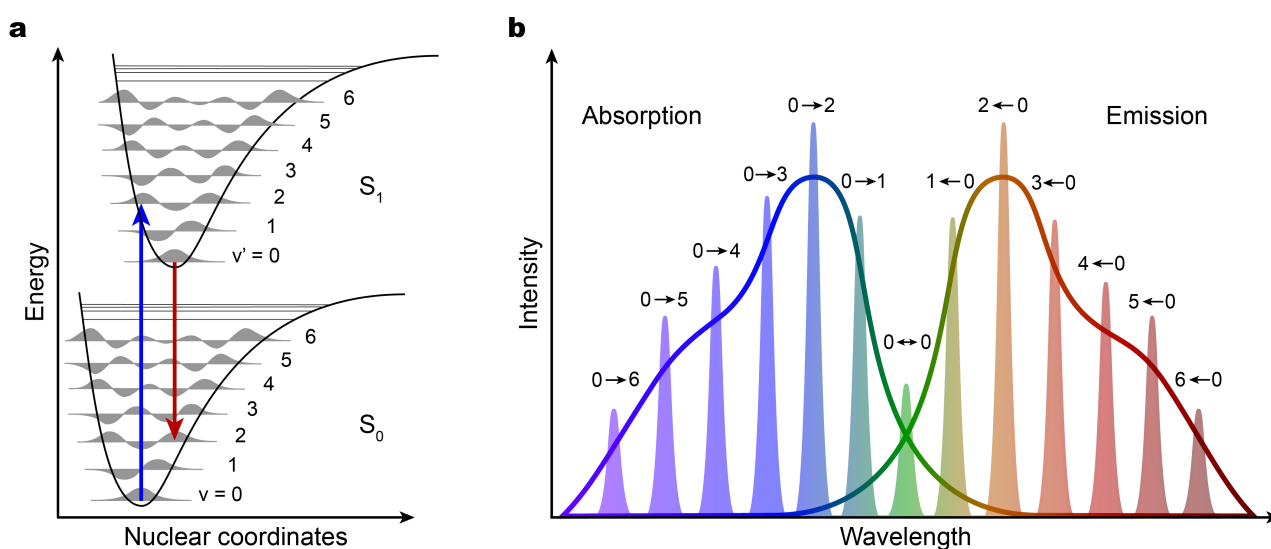


Figure 3.2 | The Franck-Condon principle. (a) The transitions depicted by arrows are favored due to higher overlap of the vibrational wave functions. (b) Illustration of vibrational transitions in absorption and emission spectra of dilute gases (transparent narrow lines) and liquids (solid lines).

The Franck-Condon principle, a quantum mechanical law, explains vibrational transition intensities and their relevance to photon absorption and emission.⁵¹⁻⁵³ This principle asserts that the likelihood of a vibrational transition depends on the degree of overlap between the vibrational wave functions corresponding to the initial and final electronic states following excitation. An essential assumption of the Franck-Condon principle is that when a molecule undergoes electronic transitions, which occur on the order of femtoseconds, the nucleus can be considered stationary. This assumption, known as the Born-Oppenheimer approximation, mathematically separates the electronic and vibrational wave functions. The states S_0 and S_1 shown in Figure 3.2a can be described as the product of the electronic, vibrational and spin wave functions:

$$\Psi = \Psi_e \Psi_v \Psi_s \quad (3.1)$$

The scalar product of the initial state $|\Psi\rangle$ and the final state $|\Psi'\rangle$ can be used to calculate the probability P of a transition:

$$P = |\langle \Psi' | \boldsymbol{\mu} | \Psi \rangle|^2 \quad (3.2)$$

where $\boldsymbol{\mu}$ is the molecular dipole operator, given by the sum of the charges and locations of the electrons and nuclei:

$$\boldsymbol{\mu} = \boldsymbol{\mu}_e + \boldsymbol{\mu}_N \quad (3.3)$$

Equation 3.2 can be written as:

$$P = |\langle \Psi'_e \Psi'_v \Psi'_s | \boldsymbol{\mu}_e | \Psi_e \Psi_v \Psi_s \rangle + \langle \Psi'_e \Psi'_v \Psi'_s | \boldsymbol{\mu}_N | \Psi_e \Psi_v \Psi_s \rangle|^2 \quad (3.4)$$

$$= \underbrace{|\langle \Psi'_v | \Psi_v \rangle|}_{\text{vibrational overlap integral}} \cdot \underbrace{|\langle \Psi'_e | \boldsymbol{\mu}_e | \Psi_e \rangle|}_{\text{orbital selection rule}} \cdot \underbrace{|\langle \Psi'_s | \Psi_s \rangle|}_{\text{spin selection rule}} + \underbrace{|\langle \Psi'_e | \Psi_e \rangle|}_{0} \cdot \underbrace{|\langle \Psi'_v | \boldsymbol{\mu}_N | \Psi_v \rangle|}_{\text{spin selection rule}} \cdot \underbrace{|\langle \Psi'_s | \Psi_s \rangle|}_{\text{orbital selection rule}} \quad (3.5)$$

The wave functions Ψ_e and Ψ'_e describe distinct electronic states, and they are orthogonal to each other with an integral $|\langle \Psi'_e | \Psi_e \rangle|^2$ of zero, in accordance with the Born-Oppenheimer approximation. However, vibrational changes can still be significant and are not restricted by this approximation. During electronic transitions, vibrational changes ($\langle \Psi'_v | \Psi_v \rangle$) can occur, allowing for non-zero vibrational overlap due to the movement of the nuclei. The intensity of a vibrational transition between different electronic states is proportional to the square of the vibrational overlap integral $|\langle \Psi'_v | \Psi_v \rangle|^2$, termed the Franck-Condon factor. An example of equally favored transitions is depicted in Figure 3.2a ($v = 0$ and $v = 2$). This principle extends to other transitions, resulting in an approximate mirror symmetry in absorption and emission spectra, as shown in Figure 3.2b. Cold and dilute gases exhibit sharp peaks, while inhomogeneous broadening in liquids and solids leads to broadening when the individual transitions are no longer observable.⁵⁴ Fluorescence emission spectra remain independent of the excitation wavelength due to the high degree of overlap in vibrational wave functions of the excited electronic states. Hence, the relaxation from any high electronic state to the S_1 state occurs on a much shorter timescale than the relaxation from the S_1 to the S_0 state, a principle termed Kasha's rule.⁵⁵ Following the vibrational relaxation, the emission spectrum of a fluorophore experiences an additional shift towards higher wavelengths and correspondingly lower energy levels. This is due to the realignment of dipole moments between the solvent and fluorophore after vibrational relaxation causes additional energy loss. The resulting energy difference between absorption and emission spectra, known as the Stokes shift, is observed experimentally as a shift in the maxima of the two spectra.⁵⁶

3.2 Förster resonance energy transfer

Förster resonance energy transfer (FRET) has established itself as a widely used method in molecular and cellular research to probe interactions and dynamics. FRET yields distances between two or more fluorophores at the nanoscale, providing insights into the intricacies of diverse systems. The phenomenon of FRET is named after the German physicist Theodor Förster. It was first described classically in 1946⁵⁷ and later formulated in a quantum-mechanical framework in 1948⁵⁸. Since then, it has become an important tool for studying molecular interactions, structural changes, and energy transfer processes.

3.2.1 Fundamentals of FRET

Förster resonance energy transfer (FRET) is a mechanism describing the radiationless energy transfer from an excited donor fluorophore (D) to an acceptor fluorophore (A). For FRET to occur, the energy level of the excited donor has to match the energy required for the acceptor to transition to a higher energy state, i.e. the two energy gaps have to be in resonance, as shown in Figure 3.3. The rate of the transfer process k_T is highly sensitive to the spatial separation r between the two fluorophores:

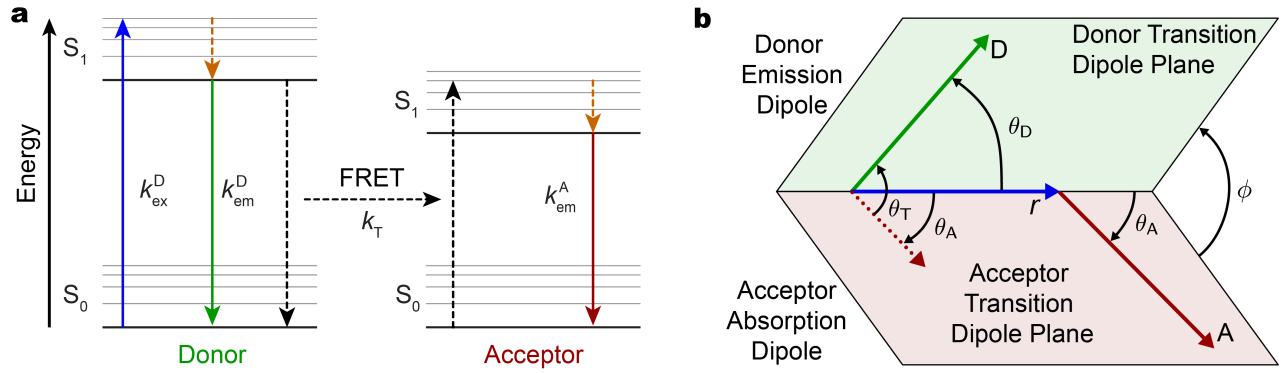


Figure 3.3 | Theory of FRET. (a) Jablonski diagram for FRET. Upon excitation of a donor fluorophore, a photon can be emitted or the energy can be transferred to an acceptor fluorophore via dipole-dipole interaction. (b) Relevant angles of the donor emission dipole and acceptor absorption dipole for the FRET efficiency.

$$wk_T = \frac{1}{\tau_D} \left(\frac{R_0}{r} \right)^6 \quad (3.6)$$

where τ_D is the fluorescence lifetime of the donor and R_0 is the Förster distance, representing the separation distance at which the energy transfer efficiency is 50%. The FRET efficiency is then calculated as:

$$E = \frac{k_T}{k_{em}^D + k_T + \sum k_i} = \frac{1}{1 + \left(\frac{r}{R_0} \right)^6} \quad (3.7)$$

where k_{em}^D and $\sum k_i$ are the rates of photon emission and all other radiationless relaxation pathways of the donor fluorophore. The Förster distance depends on various parameters and is given by:

$$R_0^6 = \frac{9000 (\ln 10) Q_D J(\lambda) \kappa^2}{128\pi^5 N_A n^4} \quad (3.8)$$

where Q_D is the quantum yield of the donor, $J(\lambda)$ is the dimensionless overlap integral as calculated from a normalized donor spectrum, κ^2 represents the relative orientation between the transition dipole moments of the donor and acceptor fluorophores, N_A is Avogadro's number and n is the refractive index of the medium. The orientation factor κ^2 depends on the angle between the two dipole vectors θ_T and the angles between the dipole vectors and the distance vectors, θ_D and θ_A :

$$\kappa^2 = (\cos \theta_T - 3 \cos \theta_D \cos \theta_A)^2 = (\sin \theta_D - \sin \theta_A \cos \phi - 2 \cos \theta_D \cos \theta_A)^2 \quad (3.9)$$

The value of κ^2 ranges between $[0, 4]$ with $\kappa^2 = 0$ and $\kappa^2 = 4$ implying the lowest transfer efficiency and highest transfer efficiency, respectively. In most FRET experiments, all orientations are assumed to be equally probable, resulting in the isotropically averaged value of $\kappa^2 = 2/3$. If the fluorophores are not freely rotating or their rotation speed is slow relative to the fluorescence lifetime, κ^2 can be estimated using the residual anisotropies of the donor and acceptor fluorophores. If the fluorophores are not freely rotating or their rotation speed is slow relative to the fluorescence lifetime, κ^2 can be estimated using the residual anisotropies of the donor and acceptor fluorophores.^{59,60} In this context, the angles that can be determined experimentally are θ_D , the angle between the transition dipole of the donor and a reference axis, θ_A , the angle between the transition dipole of the acceptor and a reference axis, and θ_T , the angle representing the rotational orientation between the donor and acceptor transition dipole moments. However, lifetime and anisotropy information are only obtainable using microscope setups that employ advanced techniques for excitation and photon detection, e.g. multiparameter fluorescence detection (MFD)⁶¹ combined with pulsed interleaved excitation (PIE)⁶², and time correlated single photon counting (TCSPC)⁶³.

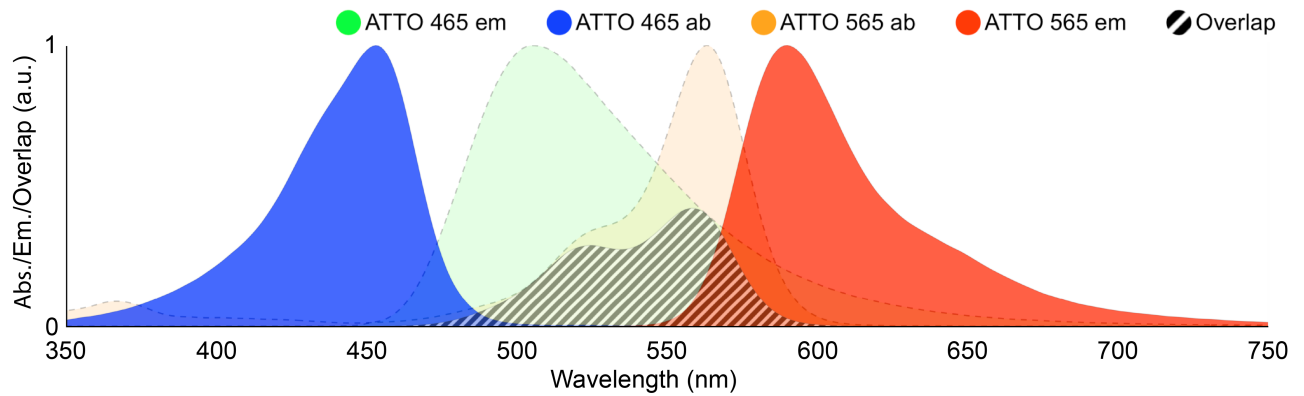


Figure 3.4 | Spectral overlap integral. Normalized absorption and emission spectra of the dye pair Atto465-Atto565. The striped area represents the overlap integral, normalized by the donor emission spectrum.

When designing FRET experiments, the choice of fluorophores has an influence on the Förster distance (equation (3.8)) due to different quantum yields of the donor fluorophores and due to the spectral overlap integral $J(\lambda)$. Example spectra of the dye pair Atto465-Atto565 and their corresponding overlap are shown in Figure 3.4. The overlap integral $J(\lambda)$ is calculated as:

$$J(\lambda) = \int_0^{\infty} F_D(\lambda) \epsilon_A(\lambda) \lambda^4 d\lambda \quad (3.10)$$

where $F_D(\lambda)$ is the normalized emission spectrum of the donor and $\epsilon_A(\lambda)$ is the extinction coefficient of the acceptor in $\text{mol}^{-1} \text{cm}$. The overlap integral in Figure 3.4 is normalized by the donor emission spectrum for illustration purposes.

3.2.2 Two-color FRET

In single-molecule experiments, the FRET efficiency is typically obtained by exciting the donor fluorophore and using two detection channels for measuring the donor and acceptor intensity, which can be achieved by spectrally separating the emitted fluorescence using a dichroic mirror. In all following equations, the background corrected channel intensities are labeled as I_{XY} and I_{XY}^{corr} for the fully corrected intensities including the correction factors. Here, 'X' refers to the excitation source, and 'Y' refers to the emission channel. For example, I_{DA} represents the background-corrected emission of the acceptor in the red channel (A) following the excitation at the wavelength used for the donor (D), excluding all other correction factors. The apparent FRET efficiency can then be expressed as:

$$E = \frac{I_{DA}}{I_{DD} + I_{DA}} \quad (3.11)$$

where I_{DD} and I_{DA} is the donor and acceptor intensity after donor excitation, respectively. Experimentally, a number of correction factors have to be considered to accurately determine the FRET efficiency. Due to the broad fluorescence emission spectra, a fraction of the donor fluorescence is usually detected in the acceptor channel (spectral crosstalk). Likewise, the broad absorption spectra result in non-negligible excitation of the acceptor fluorophore by the donor excitation laser (direct excitation). Lastly, donor and acceptor fluorophores have different quantum yields and the photon detection efficiency of the apparatus and detectors shows a spectral dependence. The necessary corrections and the determination of correction factors are discussed in subsections 3.2.4 and 3.2.5. If the lifetime information of donor is available, the accurate FRET efficiency can be calculated without performing any corrections:

$$E = 1 - \frac{\tau_{D(A)}}{\tau_{D(0)}} \quad (3.12)$$

where $\tau_{D(0)}$ and $\tau_{D(A)}$ are the donor lifetimes in the absence and presence of the acceptor, respectively. The lifetime approach also works if the acceptor labeling is incomplete, in which case the lifetimes of unquenched and quenched donor fluorophores can be extracted using a bi-exponential model function.

3.2.3 Three-color FRET

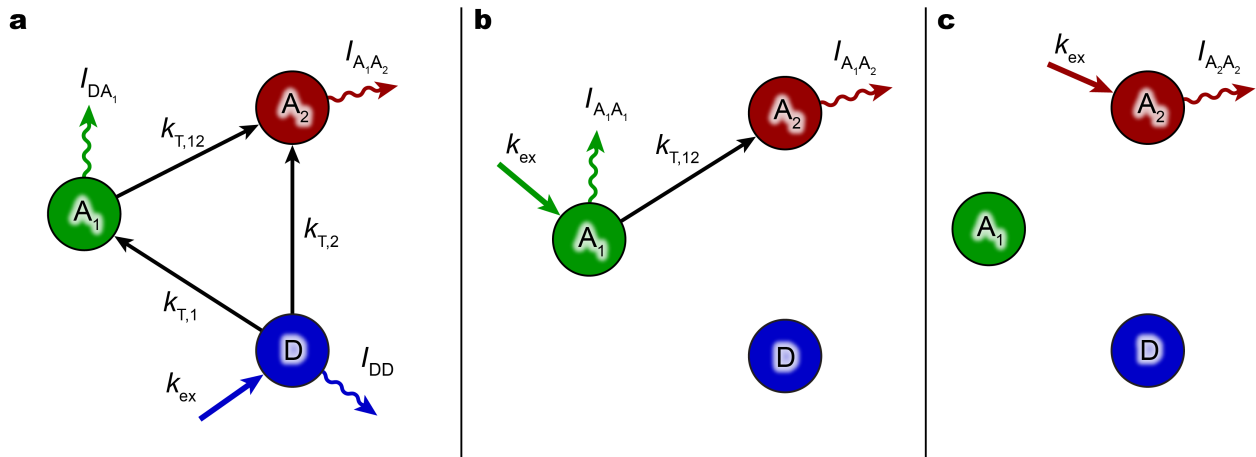


Figure 3.5 | Energy transfer in a three-color FRET system. (a) All possible transition pathways upon excitation of fluorophore D. **(b)** Excitation of fluorophore A₁ represents the two-color FRET case. **(c)** In ALEX measurements, the acceptor fluorophore A₂ is additionally excited to obtain information about the photophysical state of A₂ and thereby the stoichiometry and the direct excitation factor can be determined.

In a three-color FRET system as shown in Figure 3.5, the energy of an excited donor fluorophore (D) can be transferred to one of two acceptors (A₁ and A₂) or emitted as a photon (I_{DD}). If the energy is transferred to the first acceptor A₁, the energy can be transferred again to the second acceptor A₂ or emitted as a photon (I_{DA_1}). Upon excitation of the donor fluorophore (D) as shown in Figure 3.5a, the apparent FRET efficiencies are given by:

$$E_{DA_i} = \frac{k_{T,i}}{k_{em}^D + k_{T,1} + k_{T,2}} \quad \{i \in (1, 2)\} \quad (3.13)$$

Analogously to equation (3.7) and excluding radiationless relaxations, the FRET efficiency between acceptor A₁ and A₂ is given by:

$$E_{A_1A_2} = \frac{k_{T,12}}{k_{em}^{A_1} + k_{T,12}} \quad (3.14)$$

It is important to note that the above defined FRET efficiencies are not directly related to distances, thus they are termed apparent FRET efficiencies. The donor fluorophore (D) is quenched by two acceptors, which results in a distance dependent change of the donor quantum yield. When calculating the distance-related FRET efficiency of the donor to one of the acceptors, the distance to the other acceptor has to be considered. Assuming ideal fluorescence intensities, denoted as shown Figure 3.5, the distance-related FRET efficiencies can be calculated as:

$$E_{A_1A_2} = \frac{I_{A_1A_2}}{I_{A_1A_1} + I_{A_1A_2}} \quad (3.15)$$

$$E_{DA_1} = \frac{I_{DA_1}}{I_{DD} (1 - E_{A_1A_2}) + I_{DA_1}} \quad (3.16)$$

$$E_{DA_2} = \frac{I_{DA_2} - E_{A_1A_2} (I_{DA_1} + I_{DA_2})}{I_{DD} + I_{DA_2} - E_{A_1A_2} (I_{DD} + I_{DA_1} + I_{DA_2})} \quad (3.17)$$

While calculating the FRET efficiency $E_{A_1A_2}$ between acceptor A_1 and A_2 is straight forward, the equations for calculating the FRET efficiencies of the donor (D) to the other two acceptors (equations (3.16) and (3.17)) reveal the importance of determining $E_{A_1A_2}$ as accurately as possible. This can be achieved experimentally by exciting acceptor A_1 directly using alternating excitation techniques such as ALEX⁶⁴ or PIE⁶². However, the inherent uncertainties in $E_{A_1A_2}$ cause significant broadening of E_{DA_1} and E_{DA_2} . Careful consideration is required in three-color FRET experiments since the interdependence of all possible energy transfers and propagating errors in the analysis may lead to an ambiguous interpretation of the data. Obtaining the distance-related FRET efficiencies is further complicated by the necessary corrections applied to the measured intensities, which is discussed in subsection 3.2.4.

3.2.4 Accurate FRET efficiencies

This subsection focusses on the determination of accurate, distance-related FRET efficiencies in a three-color system. For two-color FRET, the necessary calculations simply revert to a system illustrated in Figure 3.5b. To obtain accurate FRET efficiencies, it is necessary to apply several corrections to the acceptor signal intensity. These corrections account for various factors that can affect the measurements. Specifically:

- Direct excitation (*de*): The intensity of the acceptor signal must be adjusted to account for the direct excitation of the acceptor fluorophore, which can occur for the wavelength used for donor excitation. This correction factor is denoted as de_{XY} , where X represents the excitation source, and Y represents the acceptor fluorophore.
- Spectral crosstalk (*ct*): Another correction is required to compensate for spectral crosstalk, where some of the donor's emission is detected in the acceptor channel. This correction factor is denoted as ct_{XY} , where X represents the fluorophore emitting the signal, and Y represents the emission channel.
- Detection sensitivity differences (γ): To account for variations in detection sensitivity between the donor and acceptor fluorophores, a correction factor denoted as γ_{XY} is applied.

These corrections are crucial to obtain FRET efficiencies that correspond to actual distances by accounting for various sources of signal contamination and differences in detection sensitivity. In 3-color FRET experiments employing a blue, yellow and red fluorophore, the crosstalk from the blue emission into the red detection channel and the direct excitation of the red fluorophore by the wavelength used for exciting the blue fluorophore is typically below 5%. On the other hand, the crosstalk and direct excitation factors can be much higher (up to 40% depending on the setup) for two channels that have higher spectral overlap (e.g. blue/yellow and yellow/red). The three gamma factors (γ) are given by:

$$\gamma_{DA_1} = \frac{\eta_{A_1}}{\eta_D} \cdot \frac{\Phi_{A_1}}{\Phi_D}, \quad \gamma_{A_1A_2} = \frac{\eta_{A_2}}{\eta_{A_1}} \cdot \frac{\Phi_{A_2}}{\Phi_{A_1}}, \quad \gamma_{DA_2} = \frac{\eta_{A_2}}{\eta_D} \cdot \frac{\Phi_{A_2}}{\Phi_D} \quad (3.18)$$

where Φ_X is the quantum yield of fluorophore X and η_X is the detection efficiency of channel X corresponding to the emitted wavelength of fluorophore X. From these definitions of γ it follows:

$$\gamma_{DA_2} = \gamma_{DA_1} \cdot \gamma_{A_1A_2} \quad (3.19)$$

Hence, if only two gamma factors are obtainable in a three-color FRET experiment the third gamma factor can be approximated using the relationship above. However, this relationship is only theoretically true since, in practice, the excitation and detection volumes do not match due to chromatic aberrations. There are various strategies to experimentally determine the correction factors (see subsection 3.2.5). Following the notation from Figure 3.5, i.e. the intensities are denoted as $I_{\text{fluorophore emission, detection channel}}$, the corrected intensities are calculated as:

$$I_{DA_1}^{\text{corr}} = I_{DA_1} - ct_{DA_1}I_{DD} - de_{DA_1}I_{A_1A_1} \quad (3.20)$$

$$I_{DA_2}^{\text{corr}} = I_{DA_2} - ct_{DA_2}I_{DD} - de_{DA_2}I_{A_2A_2} \quad (3.21)$$

$$- ct_{A_1A_2} (I_{DA_1} - ct_{DA_1}I_{DD}) \quad (3.22)$$

$$- de_{DA_1}I_{A_1A_1} (1 - E_{A_1A_2})^{-1} E_{A_1A_2} \quad (3.23)$$

$$I_{A_1A_2}^{\text{corr}} = I_{A_1A_2} - ct_{A_1A_2}I_{A_1A_1} - de_{A_1A_2}I_{A_2A_2} \quad (3.24)$$

It is clear that the detected acceptor signal, $I_{A_2}^D$, needs the most amount of corrections due to the different possible energy pathways. In detail, the first terms (3.21) accounts for crosstalk from the donor channel and direct excitation of acceptor A_2 . The next term (3.22) subtracts the additional crosstalk contribution of the crosstalk corrected fluorescent signal of acceptor A_1 . The last term (3.23) consider the direct excitation of acceptor A_1 and the subsequent energy transfer to acceptor A_2 , depending on the FRET efficiency $E_{A_1A_2}$. Using the corrected fluorescence intensities, the distance-related FRET efficiencies are given by:

$$E_{A_1A_2} = \frac{I_{A_1A_2}^{\text{corr}}}{\gamma_{A_1A_2}I_{A_1A_1} + I_{A_1A_2}^{\text{corr}}} \quad (3.25)$$

$$E_{DA_1} = \frac{I_{DA_1}^{\text{corr}}}{\gamma_{DA_1}I_{DD} (1 - E_{A_1A_2}) + I_{DA_1}^{\text{corr}}} \quad (3.26)$$

$$E_{DA_2} = \frac{I_{DA_2}^{\text{corr}} - E_{A_1A_2} (\gamma_{A_1A_2}I_{DA_1}^{\text{corr}} + I_{DA_2}^{\text{corr}})}{\gamma_{DA_2}I_{DD} + I_{DA_2} - E_{A_1A_2} (\gamma_{DA_2}I_{DD} + \gamma_{A_1A_2}I_{DA_1}^{\text{corr}} + I_{DA_2}^{\text{corr}})} \quad (3.27)$$

Due to the high number of correction terms, three-color FRET efficiencies are subjected to broader error distributions and inherent uncertainties. However, the detection of conformational changes does not rely on corrected signals and the apparent FRET efficiencies are well suited for extracting state dwell times, provided the duration of the time traces is sufficient. Moreover, the adoption of three-color FRET enhances the analysis by offering a higher-dimensional dataset. This increased dimensionality leads to a more detailed understanding of dynamic processes and enables the study of complex biological systems, especially if multiple states are involved. In contrast to two-color FRET, three-color FRET provides a solution to the challenge of degenerate states, enabling more robust identification of individual states and their transitions. For distance-related FRET efficiencies, states can be identified from the apparent FRET efficiencies, corrected individually and averaged when state transitions are analyzed.⁴⁸ In the case of diffusion-based confocal data, such as intensity bursts collected from single-molecule events, likelihood approaches have been developed to circumvent the direct calculation of three-color FRET efficiencies while still allowing the effective analysis of the underlying kinetics of molecular interactions.^{25,26}

3.2.5 Determination of correction factors

The correction factors can be obtained in various ways, depending on the type of experiment and microscope setup used for data acquisition. The most common methods make use of molecules without an active donor or acceptor, either due to being single-labeled or due to photobleaching events. From these so called 'Donor/Acceptor only' populations or regions in an intensity trace where the donor/acceptor is off, the correction factors can be determined. The contribution of spectral crosstalk (ct) from a donor channel (D) to an

acceptor channel (A) is calculated as:

$$ct = \frac{\langle I_{DA} \rangle}{\langle I_{DD} \rangle} \Big|_{\text{Donor only}} \quad (3.28)$$

where $\langle I_{DD} \rangle$ and $\langle I_{DA} \rangle$ represent the mean intensities of the donor and acceptor channels, respectively. Similarly, the correction factor for direct excitation of the acceptor during donor excitation can be quantified as:

$$de = \frac{\langle I_{DA} \rangle}{\langle I_{AA} \rangle} \Big|_{\text{Acceptor only}} \quad (3.29)$$

Here, $\langle I_{DA} \rangle$ and $\langle I_{AA} \rangle$ describe the mean acceptor emission after donor excitation or acceptor excitation, respectively. It is important to note that obtaining an accurate direct excitation factor requires an acceptor excitation laser in a well aligned microscope setup. The contribution of direct excitation can also be obtained by independent measurements.²² The differences in the quantum yield of the donor and acceptor fluorophores and differences in the detection sensitivity of the microscope channels are taken into account by the gamma factor (γ). Depending on the design of the experiments and available information, γ can be determined with several techniques.^{20,22,65-70} A popular method for calculating γ in confocal measurements with ALEX or PIE makes use of the relationship between the stoichiometry (S) and the apparent FRET efficiency (E_{app}) after correcting for crosstalk and direct excitation. In this context E_{app} is defined as:

$$E_{app} = \frac{I_{DA}^{corr}}{I_{DD} + I_{DA}^{corr}} = \frac{\gamma E}{1 + (\gamma - 1) E} \quad (3.30)$$

where I_{DA}^{corr} is the direct excitation and crosstalk corrected FRET intensity and E refers to the fully corrected FRET efficiency. The stoichiometry (S) can be calculated as:

$$S = \frac{I_{DA}^{corr} + I_{DD}}{I_{DA}^{corr} + I_{DD} + I_{AA}} \quad (3.31)$$

A linear relation between $1/S$ and E_{app} can be obtained via:

$$\frac{1}{S} = 1 + \gamma\beta + \beta(1 - \gamma) E_{PR} = \Omega + \sum E_{PR} \quad (3.32)$$

where Ω is the intercept and \sum is the slope of the linear fit in a plot of $1/S$ against E_{app} . The β factor is accounting for differences in excitation power of the acceptor (I_{Aex}) and the donor (I_{Dex}) as well as their corresponding absorption cross sections σ_{Aex}^A and σ_{Dex}^D :

$$\beta = \frac{I_{Aex}\sigma_{Aex}^A}{I_{Dex}\sigma_{Dex}^D} \quad (3.33)$$

From the linear fit described in equation 3.32, γ and β can be determined:

$$\beta = \Omega + \sum - 1 \quad (3.34)$$

$$\gamma = \frac{(\Omega - 1)}{\Omega + \sum - 1} \quad (3.35)$$

Since this method relies on a robust extraction of the intercept and slope, well separated populations are necessary for an accurate determination of γ and β . In measurements with the molecule of interest attached to the surface, γ can also be calculated via the intensity differences in each channel upon acceptor photo-bleaching:

$$\gamma = \frac{\langle \Delta I_{DA}^{\text{corr}} \rangle}{\langle \Delta I_{DD}^{\text{corr}} \rangle} \Big|_{\text{Acceptor photobleaching event}} \quad (3.36)$$

Here, $\langle \Delta I^{\text{corr}} \rangle$ and $\langle \Delta I^{\text{corr}} \rangle$ refer to the corrected intensity difference for the mean donor and acceptor emission after donor excitation before and after acceptor photobleaching. This approach necessitates intensity traces with a duration long enough to smooth out signal noise, a feature unattainable in typical diffusion-based measurements. For wide-field data, however, molecule-wise γ factors can be extracted and applied, provided that the acceptor bleaches before the donor fluorophore. This has the advantage of considering γ factor distributions throughout the field-of-view, such as those caused by a heterogenous illumination profile.

3.3 Data acquisition

The foundation of data acquisition in single molecule studies lies in the ability to detect and monitor the properties of single molecules with precision. Various experimental techniques have been developed for this purpose, including optical microscopy, atomic force microscopy, and nanopore sensing. These techniques not only enable the detection of single molecules but also offer valuable information on their positions, motions, conformational changes, and interactions. Within the scope of this thesis, the most relevant techniques are surface-based, i.e. the molecules of interest are attached to a solid surface instead of freely diffusing through the solution. This allows for the observation of single molecules for seconds to minutes, depending on the fluorophore stability and excitation power. In the context of fluorescence measurements, two common techniques are total internal reflection fluorescence (TIRF) and confocal microscopy.

3.3.1 Total internal reflection fluorescence (TIRF)

TIRF microscopy is based on the principle of total internal reflection, which occurs when a light beam traveling through a high refractive index material (e.g. glass) encounters a boundary with a lower refractive index material (e.g. air or an aqueous sample). Under specific conditions, the incident light is completely reflected back into the higher refractive index material, creating an evanescent wave that penetrates only a short distance into the lower refractive index medium. Snell's Law specifies the necessary conditions by relating the angles of incidence and refraction when light passes from one medium to another with different refractive indices:

$$n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (3.37)$$

where n_1 and n_2 are the refractive indices of the first and second medium, respectively, θ_1 is the angle of incidence of the incoming light with respect to the normal to the interface between the two media and θ_2 is the angle of refraction within the second medium. The critical angle (θ_c) is the angle of incidence above which total internal reflection occurs. It is determined by Snell's Law, and it marks the boundary beyond which light is no longer transmitted into the second medium and is instead reflected internally:

$$\theta_c = \arcsin \frac{n_2}{n_1} \quad (3.38)$$

In TIRF microscopy, this critical angle is carefully controlled by adjusting the angle of incidence. The intensity of the evanescent field ($I(z)$) decreases exponentially with distance from the interface. This exponential decay is characterized by:

$$I(z) = I_0 \cdot e^{-\frac{2z}{\delta}} \quad (3.39)$$

where I_0 is the initial intensity at the interface, z is the distance from the interface and δ is the penetration depth of the evanescent wave. This penetration depth indicates how deeply the illumination and excitation of fluorescent molecules occur within the sample. It depends on the wavelength of the incident light (λ), the

refractive indices of the first medium (n_1) and second medium (n_2), and the angle of incidence (θ_1):

$$\delta = \frac{\lambda}{4\pi\sqrt{n_2^2 \sin^2 \theta_1 - n_1^2}} \quad (3.40)$$

The penetration depth (δ) of the evanescent wave is typically on the order of 100-200 nanometers, thus creating a highly localized illumination region and reducing background noise from out-of-focus fluorescence. The dependence on the angle of incidence can be used to fine-adjust the illumination of the sample. Three commonly employed methods for achieving total internal reflection are objective TIRF (oTIRF), prism TIRF (pTIRF) and lightguide TIRF (lgTIRF), illustrated in Figure 3.6. Objective TIRF relies on an objective lens with a high numerical aperture (NA), e.g. oil immersion objectives. The objective lens is designed to direct the laser beam at the glass-substrate interface at an appropriate angle for total internal reflection. Due to this fixed geometry, the maximum angle of incidence is limited. Prism TIRF relies on the use of a glass prism placed in contact with the glass substrate containing the sample. The light from the laser source enters the prism at an appropriate angle and undergoes total internal reflection at the prism-substrate interface. In lightguide TIRF, a waveguide, usually made of glass, is placed in contact with the sample surface. The waveguide acts as an optical conduit, guiding the incident laser light along its length. The laser light entering the waveguide is typically coupled into the waveguide at a specific angle and undergoes internal reflection at the waveguide-sample interface.

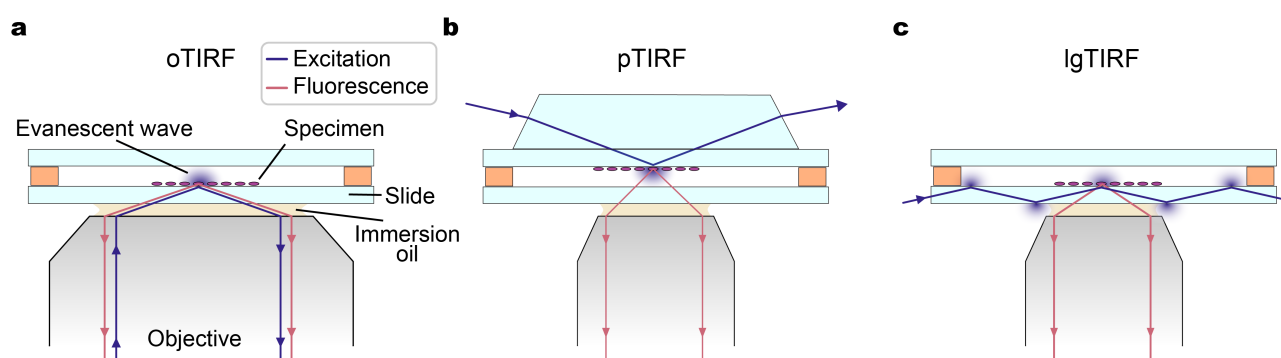


Figure 3.6 | Methods in TIRF microscopy. (a) Objective TIRF (oTIRF): The excitation and emission share the same path. TIRF can be achieved using high NA objectives and aligning the laser light such that it enters the objective off-center. (b) Prism TIRF (pTIRF): Excitation occurs by guiding the laser through a high refractive index prism, achieving TIRF at the top of the sample chamber. (c) Lightguide TIRF (lgTIRF): The laser light is confined within a thin, high refractive index layer, continuously undergoing TIRF at the interfaces.

TIRF microscope setups are equipped with wide-field detection using cameras such as EMCCD (Electron Multiplying Charge-Coupled Device) cameras or sCMOS (Scientific Complementary Metal-Oxide-Semiconductor) cameras. In the study of complex biological systems, camera-based approaches have become the predominant choice, primarily due to the necessity of acquiring substantial statistical data. Regarding the excitation techniques, Alternating Laser Excitation (ALEX) has gained prominence in single molecule studies.^{22,62} ALEX utilizes two different excitation lasers that alternate in illuminating the sample. One laser is used to excite a donor fluorophore, while the other excites an acceptor fluorophore. By carefully synchronizing the laser excitations, usually on a microsecond timescale, the presence of the acceptor fluorophore can be probed continuously. This provides information about the stoichiometry and allows for the calculation of the direct excitation factor of the acceptor by the donor laser. However, one disadvantage can arise when studying fast molecular kinetics close to the integration time since no FRET information (section 3.2.1) is available during direct excitation of the acceptor.

3.3.2 Confocal microscopy

In confocal imaging, a focused laser beam illuminates a diffraction limited point. For image collection, the beam is scanned across the sample or the sample itself is moved. This scanning process allows for the precise collection of light from the focal plane, providing improved optical sectioning. Critical components in confocal microscopy are the pinhole apertures placed in front of the excitation source and the detector. This pinhole selectively collects emitted fluorescence only from the focal plane while rejecting out-of-focus light. By allowing only the fluorescence emitted from the precisely defined focal volume to reach the detector, the pinhole significantly enhances the image contrast. This capacity for confocal microscopes to selectively capture light from a specific plane is what distinguishes them from wide-field fluorescence microscopy. Furthermore, the pinhole diameter is a critical parameter in confocal microscopy. A smaller pinhole size results in improved axial resolution but may reduce the amount of collected fluorescence signal. Conversely, a larger pinhole increases signal intensity but compromises axial resolution. Therefore, the selection of an optimal pinhole size involves a trade-off between resolution and signal intensity based on the specific requirements of the imaging experiment. For excitation, the laser light is typically coupled into an optical fiber in most modern microscope setups. The exit of this fiber acts as a point source, effectively serving the purpose of a pinhole and emitting a gaussian beam. However, it is worth noting that a gaussian beam can also be generated without the use of an optical fiber. This can be achieved, for instance, by focusing a laser beam on a small pinhole in the range of 50 μm . Alternatively, if the laser itself emits a collimated beam, a pinhole is not needed as the focussed laser beam acts a point source. The pinholes provide confocal microscopes with an advantage over TIRF microscopes (section 3.3.1), which is the in-built removal of background and scattered light independent of the penetration depth. While TIRF microscopes offer excellent background rejection near the surface, imaging deeper into the sample or visualizing molecules at various depths may require adjustments in laser power or other parameters. The specific laser power requirements depend on the sample characteristics and the depth of the structures of interest. For example, studying molecules within a cell may require a larger penetration depth whereas single molecules directly attached the surface may not. In confocal microscopy, the molecules of interest can be excited with lower laser power without sacrificing penetration depth. Lower laser powers, in general, yield longer fluorophore survival times and hence longer intensity trajectories per molecule due to the reduced phototoxicity. However, one disadvantage is that the data has to be collected for each molecule individually, whereas TIRF allows the recording of multiple molecules at once. Confocal microscopes usually employ single-photon counting units as detectors and the photon counts are time-binned after the measurement, allowing some flexibility the downstream analysis. The concept of a confocal microscope setup is illustrated in Figure 3.7.

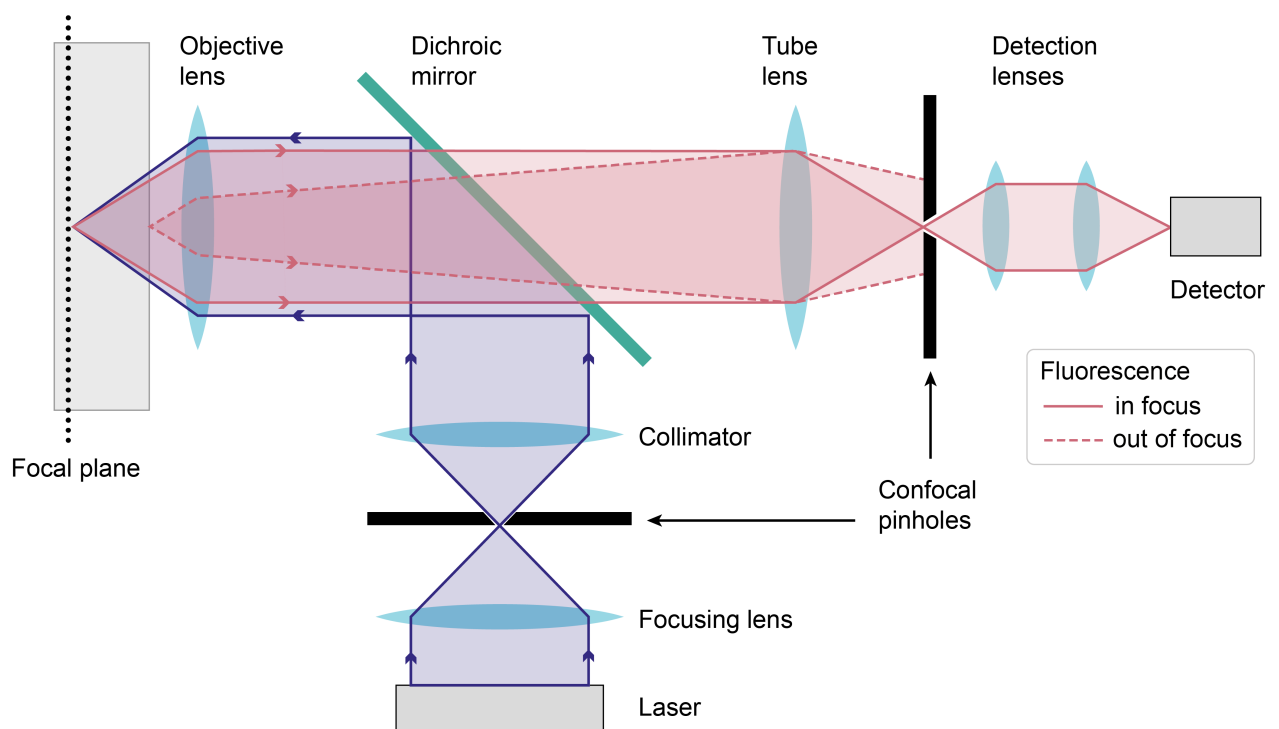


Figure 3.7 | Principle of confocal microscopy. The excitation laser light (blue) is focused on a pinhole, spatially filtering the beam. The second pinhole blocks out of focus fluorescence (red) and only in focus light guided to the detector.

Chapter 4

Publications

4.1 Paper 1: Deep-LASI:deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures

Deep-LASI:deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures

Wanninger, S., Asadiatouei, P., Bohlen, J., Salem, C., Tinnefeld, P., Ploetz, E., Lamb, D.C., *Nature Communications*, **14**, 6564 (2023).

DOI: <https://doi.org/10.1038/s41467-023-42272-9>

This paper presents a software suite that employs an ensemble of deep neural networks to automate the analysis of single molecule intensity trajectories.

4.1.1 Motivation and main results

Single-molecule spectroscopy has fundamentally transformed the investigation of nanoscale processes, particularly through the utilization of optical fluorescence imaging. This approach allows for non-invasive exploration of individual dynamic biomolecules within cellular environments, membranes, and solutions. The integration of single-molecule Förster resonance energy transfer (smFRET) with advanced microscopy techniques, such as confocal microscopy or total internal reflection fluorescence (TIRF) microscopy, enables precise probing of nanometer-scale distances and dynamics. While solution-based measurements yield insights into sub-millisecond dynamics, immobilized molecule experiments provide a temporal view of single molecules on the microsecond to minute timescale. The elimination of ensemble averaging permits direct measurements of biomolecular conformational states and dynamics.

Conventional FRET experiments typically employ two colors to investigate conformational distributions and distance changes. Multi-color FRET, involving three or more labels, extends the capability of FRET to explore molecular interactions between distinct binding partners and simultaneously measure multiple distances, reflecting correlated motion within the same molecule. Nevertheless, quantitative smFRET data analysis encounters various challenges, ranging from experimental limitations such as a scarcity of usable single molecule traces, low signal-to-noise ratio, and short traces due to photochemistry, to computational complexities associated with processing large data volumes. Overcoming these challenges requires meticulous data sorting, which intensifies the analytical efforts when performed manually. Low statistical significance arises from factors, i.e. slow kinetics, rare transition probabilities, low labeling inefficiencies, low signal-to-noise ratio, rapid photobleaching, or spurious background. Furthermore, arbitrary fluctuations introduced by unwanted interactions and aggregations between binding partners complicate the analysis of the underlying states and kinetics. After data categorization and extraction of potentially useful information, the main approach to analyze kinetics is Hidden Markov Models (HMM), which are statistical models commonly used to analyze time-series data. In the context of smFRET, HMMs are employed to identify distinct conformational states and transitions within single-molecule trajectories. However, the application of HMMs has limitations, particularly in dealing with multi-color FRET data, where the complexity of the data sets and the required effort for analysis increases.

Prior to this work, efforts were made to address some of these challenges through the development of Deep-FRET⁶ and AutoSIM⁷, which both use deep learning to analyze single molecule FRET data. However, the employed deep neural networks (DNN) are limited to classifying two-color FRET trajectories without offering auto-correction capabilities and kinetic analysis. To bridge these gaps and advance the field further, an ensemble of DNNs was developed in this project and integrated into the software suite called Deep-LASI (discussed in section 4.2), which also offers common methods for raw data extraction, manual categorization by the user, statistical evaluation techniques and HMM. The DNNs are tailored for fully automated analysis of single-color traces, as well as two-color and three-color single-molecule FRET data. Figure 4.1 illustrates the complete data pipeline, starting from extracted intensity traces and ending with transition density plots, from which the user can select and fit dwelltime populations. The DNNs were trained with large data sets of simulated trajectories of each data type and benchmarked using simulated validation data sets, achieving high prediction accuracies. In addition, the fully automated approach was extensively benchmarked with experimental data by comparing the DNN outputs to human evaluation and HMM. The DNN predictions achieved a high concordance between the automated and manual results while quantitatively analyzing the underlying kinetics of single-molecule intensity traces. This comprehensive auto-analysis can be performed within 100 ms per trajectory, reducing the total analysis time for a typical data set from days to minutes.

4.1.2 Brief description of the method

Deep-LASI employs an ensemble of pre-trained DNNs for the automated analysis of single-molecule data, encompassing one-, two-, and three-color scenarios, along with multi-color FRET correction and kinetic analyses. The DNNs process single-molecule fluorescence intensity traces obtained directly from confocal microscopy or extracted from movies using wide-field or TIRF microscopy. Two-color fluorescence data with continuous wave excitation or ALEX modalities, as well as three-color smFRET measurements using ALEX, can be analyzed. All available channels are fed into hybrid models, consisting of a convolutional neural network (CNN) using the omni-scale feature learning approach and long short-term memory (LSTM) layers. Deep-LASI simultaneously extracts spatial and temporal sequence features, categorizing each frame into specific classes, namely: dynamic, static, noisy, artifact, aggregate, and photobleached. The number of photobleached classes depends on the number channels, i.e. employed fluorophores, with each combination of active and inactive fluorophores representing a separate class. Traces with artifacts, aggregates, or high noise are excluded in the auto-analysis process. The output of the trajectory classifier DNN provides probabilities for each category, and the summed probabilities over non-photobleached frames serve as confidence levels for each trace. User-defined thresholds can be applied to control the inclusion or exclusion of non-ideal traces in further analyses. The trajectory classifier DNN detects photobleaching events, including frames in which the fluorophores are in a dark state. This information enables Deep-LASI to automatically calculate the correction factors for individual molecules, e.g. crosstalk and direct excitation can be corrected via the predicted dark states of the acceptor and donor, respectively. Since all correction factors are rarely obtainable from a single trajectory, all locally determined correction factors are used to calculate the corresponding global correction factors. While using the median values for global correction proved to be the most robust approach, the user can also use the mean or mode values. After the classification of a given data set, various features of any class can be histogrammed, including the apparent and corrected FRET efficiencies, signal intensities, fluorophore survival times, stoichiometry information, and all available correction factors. Trajectories that are classified as dynamic are fed into a DNN designed to detect the number of observed states for each molecule. Based on the predicted number of states, each dynamic trajectory is transferred to a DNN that predicts the transitions of the corresponding states. This state transition classifier DNN assigns each frame to a particular state, providing probability values for state occupancy. Hence, the state predictions effectively digitize all frames and produce state trajectories for each molecule. Using this information, a transition density plot (TDP) is generated and kinetic rates can be extracted by fitting the dwell-time distributions of each state. The TDP marks the first point of human intervention, involving manual selection of state transitions and the fitting procedure. Optionally, the user can generate TDPs of specific number of states classes and apply probability thresholds on the state occupancy before generating the TDP. The total

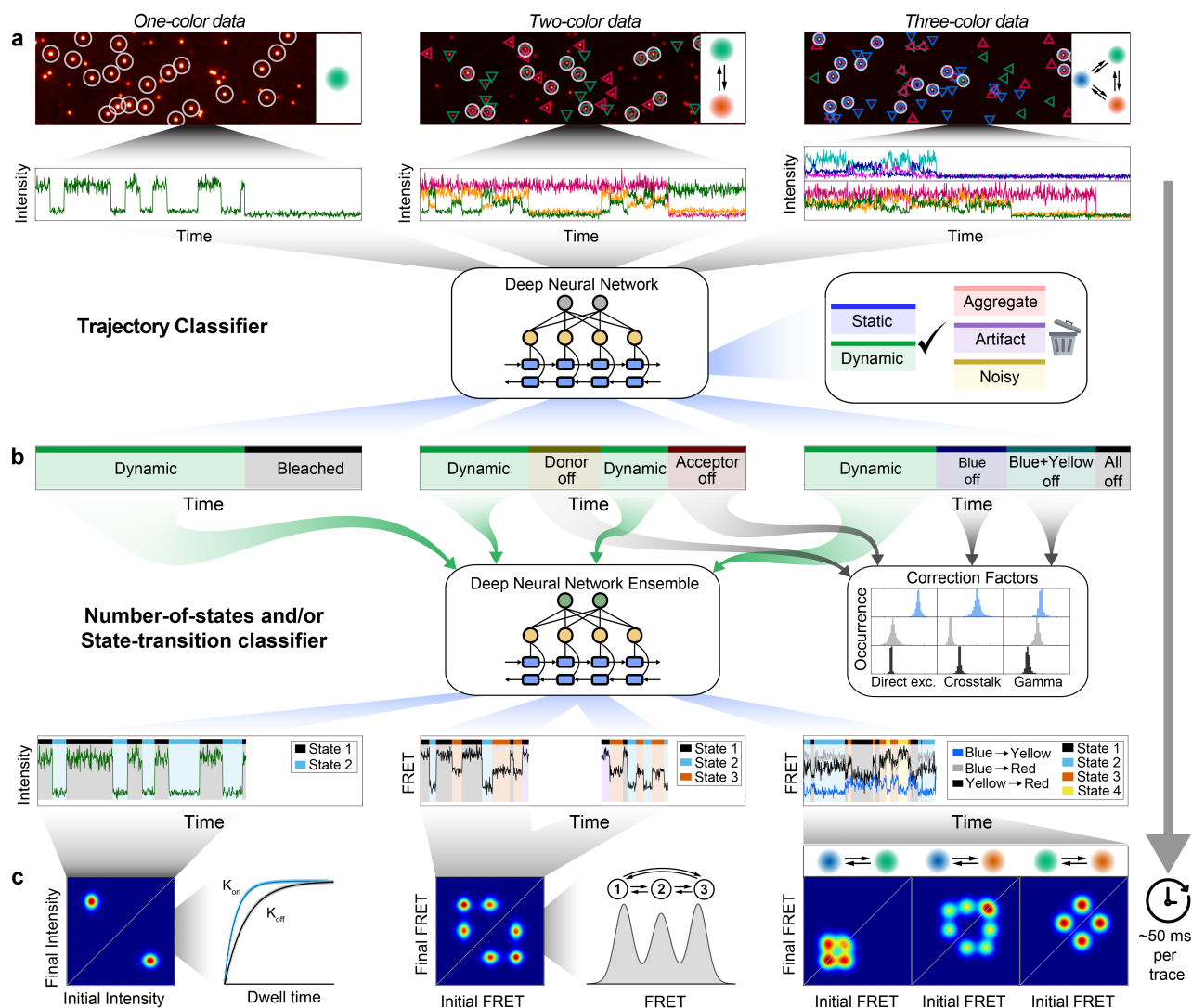


Figure 4.1 | Deep-LASI workflow. (a) Single-molecule data with up to 3 separate channels after direct and alternating laser excitation are identified, extracted, and presorted for further analyses. Each frame within the time traces is classified into categories using a hybrid CNN-LSTM. (b) A second hybrid CNN-LSTM evaluates the kinetics and state information in the presorted data. The photobleaching information can be used for determining the correction factors to obtain accurate FRET efficiencies for experiments with 2 and 3 fluorophores. (c) Next, the interconversion rates between underlying states and absolute, distance-related FRET values are extracted from multi-color data sets.

duration of the analysis depends on the data set characteristics with a typical prediction time of 20 ms to 100 ms per trace. The performance also depends on several other factors such as the computational resources and the total number of trajectories, frames and dynamic molecules. Since all employed DNNs are trained on synthetic data, they can be retrained and tailored to specific conditions that may have not been captured in the original training data set.

4.1.3 Outlook

The development and application of Deep-LASI represents a significant advancement in the automated analysis of single-molecule data, offering a comprehensive solution for one-, two-, and three-color scenarios, along with multi-color FRET correction and kinetic analyses. While the robustness of the neural network predictions were thoroughly demonstrated, Deep-LASI needs to be further tested with unseen data from multiple

sources to extend its applicability. The neural network architectures were specifically developed for time series data and can be readily used for other techniques such as DNA-PAINT, optical tweezers and electrophysiological techniques (e.g. patch-clamp). In terms of extending the capabilities of Deep-LASI, the main effort lies in finding a representative training data set for the desired task. If no labeled experimental data is available for training, optimizing the existing training data set or designing simulations from the ground up are viable options. The big advantage of applying deep learning methods in the field of single-molecule research is that, in most cases, the obtained experimental data is well understood and can be accurately modeled, enabling the generation of large training data sets. Experimental data sets are continuously growing in size and complexity as researchers strive to examine molecules in diverse environments and quantify their kinetics, which depend on the concentration of specific substrates. These studies necessitate high-throughput classification and analysis, making deep neural networks ideal candidates for addressing these challenges.

4.2 Paper 2: Deep-LASI, Single-molecule Data Analysis Software

Deep-LASI, Single-molecule Data Analysis Software

Asadiatouei, P., Salem, C., Wanninger, S., Ploetz, E., Lamb, D.C., *Unpublished Manuscript*.

In this paper, the complete software suite for analyzing single-molecule time trajectories is presented. The software description includes all preprocessing techniques before the integrated deep neural networks can be used and all analysis features that can be employed by the user outside of the automatic AI approach.

4.2.1 Motivation and main results

Single-molecule techniques have changed how we study molecules, providing a direct and highly detailed view of their behavior. These methodologies transcend ensemble averaging constraints, enabling a direct examination of sample heterogeneities, subpopulations, and dynamic processes, particularly when applied to immobilized molecules using modalities such as atomic force microscopy, optical and magnetic tweezers, or total internal reflection fluorescence (TIRF) microscopy. In the field of single-molecule research, fluorescence resonance energy transfer (FRET) experiments stand out as a non-contact method capable of detecting distances ranging from 2 to 10 nanometers and capturing dynamics across a spectrum of timescales from nanoseconds to kiloseconds. However, the transition from raw data to meaningful insights is complex. Additionally, advanced excitation techniques such as alternating laser excitation (ALEX) and the use of multiple fluorophores further complicate the data sets. As experiments embrace greater complexity, a strong demand for advanced analysis tools becomes apparent. While a plethora of software exists to facilitate single-molecule analysis, the majority caters to two-color FRET experiments. The software presented in this paper, called Deep-LASI (Deep-Learning Assisted Single-molecule Imaging analysis), features the data analysis of one, two and three-color FRET experiments and facilitates the analysis of these kind of experiments by providing auto-analysis methods using pre-trained deep neural networks. Recorded particles in multiple channels, corresponding to different emission wavelengths, can be localized and aligned with sub-pixel resolution using wavelet or intensity based algorithms. The sensitivity of the chosen particle detection method can be adjusted by the user, making it adaptable to different experimental conditions. The background is corrected individually for each particle during the signal extraction by using a custom or automatically determined mask for the point spread function (PSF) and the local background in a circular region surrounding the PSF. Following the trajectory extraction, Deep-LASI offers both manual and automatic analysis environments. These include various features such as framewise trace classification, determination of FRET correction factors and kinetic analyses. Additionally, the software provides fitting routines for distributions of important parameters such as background and fluorescence intensities, fluorophore survival times, and FRET efficiencies. The incorporation of deep-learning techniques presents a fully automated and rapid analysis of the complete data set, including the identification of relevant sections within each trajectory, the calculation of FRET correction factors, and quantitative analysis of multi-state kinetics. Since the software also offers multivariate HMM analysis of all data types, i.e. one, two and three-color data, the output of the deep neural networks can be cross-checked for consistency. The software also provides environments for simulating and training single-molecule time traces, complemented by example data sets and tutorials to accelerate proficiency. Representing a significant advancement in single-molecule analysis, Deep-LASI addresses the demands of progressively complex experiments, shaping the landscape of sophisticated methodologies in the fields of life sciences and microscopy.

4.2.2 Outlook

The development and implementation of Deep-LASI mark a significant milestone in the realm of single-molecule analysis, providing a user-friendly platform for efficient exploration of intricate molecular dynamics. The commitment to user-friendly interfaces mitigates the barrier between researchers and the powerful capabilities of single-molecule techniques. Looking ahead, Deep-LASI, being open source, invites continuous collaboration and improvement. Future development may involve extending its functionalities to fully

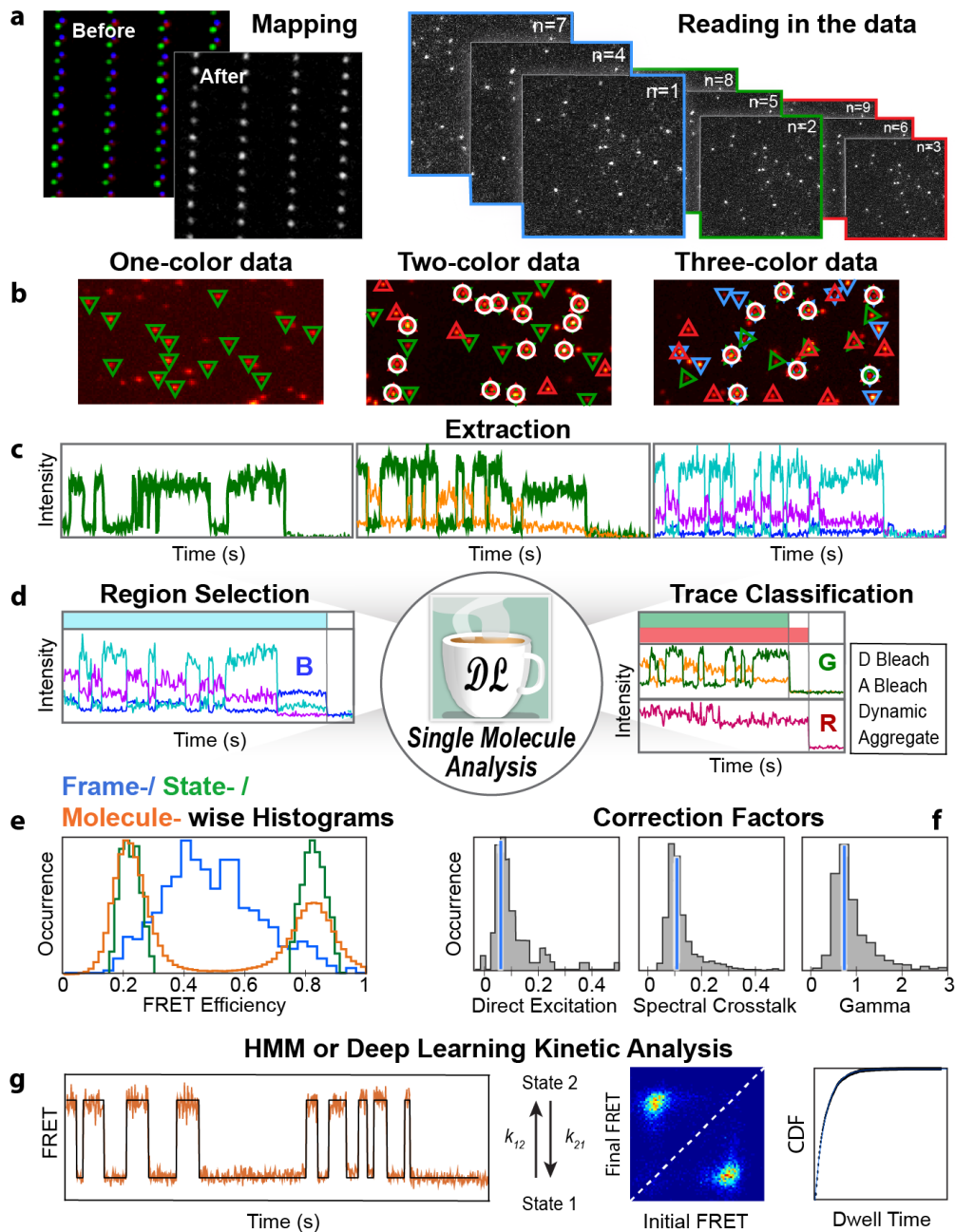


Figure 4.2 | Deep-LASI software overview. The main applications of the software package are to extract, sort and analyze intensity traces from single-molecule data. This process involves a series of key steps: **(a)** For multicolor experiments, the different channels need to be register to each other (i.e. mapped). Afterwards, the raw data is read-in for each channel from a stack of frames based on the excitation scheme. **(b)** Single molecules are localized and, when desired, co-localized across different channels based on the created map. **(c)** The intensity traces are extracted from each detected (and co-localized) particle and corrected for background. **(d)** The analysis of extracted intensity traces starts with trace classification and selection of the useful region of each channel where the corresponding fluorophores are active. **(e)** The results can then be visualized by the means of various histograms with frame-, state- and molecule-wise approaches. **(f)** Optionally, the method-specific correction factors are determined. **(g)** For dynamic traces, a kinetic analysis can be performed by Hidden-Markov Modeling (HMM) or deep learning approaches. The panels show a typical Viterbi path created by HMM and Transition-Density Plots (TDP) with state transition information and the cumulative dwell-time distribution function (CDF) determined by fitting, respectively.

accommodate other popular techniques such as binding assays using DNA-PAINT. Since the architecture of the employed deep neural networks is extendable and adaptable to other sources of data, Deep-LASI can be tailored to diverse experiments. With the strong emphasis on automation features, Deep-LASI empowers researchers to focus on the scientific essence of their work and allows them to quickly test new approaches without having to commit large amounts of time. Since the bottleneck of analysis time almost completely removed, new type of experiments are feasible and will give rise to new ideas on how to study single molecules.

4.3 Paper 3: A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories

A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories

Götz, M., Barth, A., Bohr, S.S.R., Börner, R., Chen, J., Cordes, T., Erie, D., Gebhardt, C., Hadzic, M., Hamilton, G., Hatzakis, N., Hugel, T., Kisley, L., Lamb, D. C., Lannoy, C., Mahn, C., Dunukara, D., Ridder, D., Sanabria, H., Schimpf, J., Seidel, C. A. M., Sigel, R. K. O., Sletfjerd, M. B., Thomsen, J., Vollmar, L., Wanninger, S., Weninger, K. R., Xu, P., Schmid, S. *Nature Communications*, **13**, 5402 (2022).

DOI: <https://doi.org/10.1038/s41467-022-33023-3>

This paper addresses the challenges associated with extracting quantitative kinetic information from single-molecule FRET (smFRET) data and compares the analysis results of leading software tools on multiple data sets.

4.3.1 Motivation and main results

The complexity of experimental single-molecule FRET (smFRET) data has led to the development of various analysis tools. However, a comprehensive comparison of these tools is lacking. This study presents the outcomes of a blind benchmark assessment involving eleven software tools. From these software tools, a total of 14 analysis techniques are used to infer kinetic rate constants from smFRET trajectories in seven different datasets. The results provide insights into the current strengths and limitations of these tools, offering concrete recommendations and identifying areas for future development. The evaluation encompasses simulated and experimental data and incorporates challenges commonly encountered in smFRET experiments such as diverse noise levels, varying model complexities, non-equilibrium dynamics and kinetic heterogeneity.

The main goal of this study is to enhance the understanding of biomolecular dynamics through the refinement of quantitative models while increasing the consistency and transparency of the employed analysis methods. The first two datasets, one simulated and one experimental, considered a kinetic two-state system with photobleaching events. In Wanninger *et al.*, the *Deep-LASI* software was added for these two data sets, shown in Figure 4.3.⁴⁸ The results of all analysis tools are in good agreement with the ground truth of the simulated dataset and produce consistent results for the experimental dataset with rate constants varying by 12% (k_{12}) and 16% (k_{21}). The third dataset is a simulation of a kinetic three-state model with all states interconnected in a circular flow, with intensity variations between individual dye molecules, emulating realistic experimental conditions. While all analysis tools found the three FRET efficiency populations, the extracted rates for each state showed larger variations than for the two-state datasets, with deviations ranging from only 9% to over 200%. The fourth dataset is a simulation of a four-state model with two degenerate FRET states and kinetic heterogeneities. Here, the challenge lied in the extraction of individual rates from the multi-exponential dwell time distributions and solving the connectivity of the four states. Only four analysis tools achieved a good agreement with the ground truth, whereas other tools that were restricted to mono-exponentially decaying dwell times showed the largest deviations or were not able to produce a kinetic model at all. The last benchmark represents the most challenging case in this study, involving three experimental datasets of the same system (protein binding to a fluorescently labelled DNA) with different experimental conditions and hence different kinetic behavior. Since these datasets are complex and lack a ground truth, the results from the different analysis methods and their accuracy are subjected to data interpretation. These datasets were also the only examples in which the employed analysis tools did not all agree on the involved number of states, which ranged from two states to more than four. The determined kinetic rates varied accordingly, making a meaningful comparison of all methods difficult. However, within the same assumptions regarding the number of states, the agreement among the analysis tools was within 25% of the average residence determined by these tools.

Regarding the different methodologies used for the analysis, eight tools are based on Hidden Markov Models (HMM) while the other tools employ correlation, step finding or clustering methods. While the underlying theory of the HMM based techniques is the same, the implementations can differ significantly, e.g. with

respect to the flexibility of selecting initial model parameters, constraints on learned parameters, thresholds and the type of input data (FRET or intensities).

Our analyses in this study were performed with the software package *Tracy*, which provides HMM functionalities for analyzing kinetics. While lacking the capability to determine the connectivity of degenerate and not excelling in one particular discipline, *Tracy* was consistently in the top 3 performing tools for the two-state and three-state data sets. As shown later in Wanninger *et al.*, *Deep-LASI* performed equally well for the simple two-state data sets. Although the fully automated analysis approach could be applied to the other data sets, the results were not included since it was impossible to analyse these data sets in a bias free manner.

In general, for methods that produce a state trajectory, the study revealed the importance of subtle differences in the approaches used for actually determining the kinetic rates, e.g. the parameters used for fitting the dwell time distributions. While these differences lead to only minor deviations for simple data sets, they can become an important factor for complex data sets and need to be thoroughly documented for ensuring reproducibility. Overall, the most significant factor affecting the final results of the analysis of complex datasets is the determination of the number of states, especially for generative models such as HMM that rely on this parameter as a prior assumption.

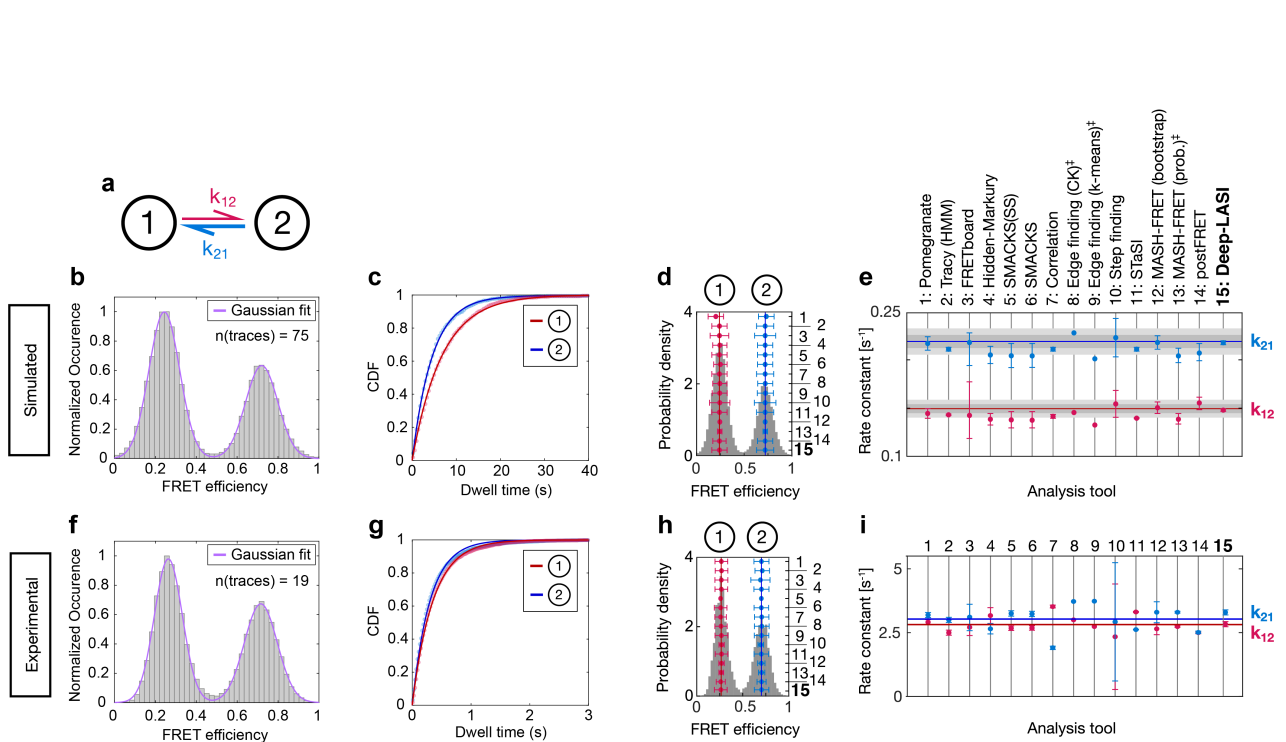


Figure 4.3 | Kinetic software challenge. (a) An illustration of the kinetic two-state model connected by forward and backward rate constants: k_{12} and k_{21} . (b) A framewise FRET efficiency histogram (gray) of the simulated data. A Gaussian fit to the two populations are shown in magenta. (c) Mono-exponential dwell time distributions of the data in (b) obtained from the state-transition classifier. (d) The ground truth FRET histogram (gray) with state assignments labeled at the top and the inferred average FRET efficiencies in red and blue. Numbers on the right axis refer to the analysis tools specified in (e). Vertical lines indicate the mean over all tools. The error bars represent the standard deviations returned from the different analysis routines. (e) Rate constants and uncertainties inferred from the data set in (d) by different labs using the respective analysis tools. The ground truth is indicated by the horizontal red and blue lines, the intrinsic uncertainty of the data set is represented by dark gray (1σ) and light gray (2σ) intervals. (f) A framewise smFRET efficiency histogram (gray) of the experimental data extracted by the trace classifier. (g) The dwell-time distributions and corresponding mono-exponential fits of the data in (f) obtained from the state-transition classifier. A Gaussian fit to the two populations is shown in magenta. (h) A smFRET histogram of preselected traces from panel (h) where photobleaching and photoblinking contributions have been removed. State 1 is labeled in red and state 2 in blue. The vertical lines indicate the average value returned from analysis routines 1-15. The legend for the analysis routines is given in (e). The error bars represent the standard deviations returned from the different analysis routines. (i) Inferred rate constants from the experimental data set in (h). The respective analysis tools are specified in (e). Horizontal red and blue lines indicate the mean of the inferred kinetic rate constants from analysis tools 1-15. The legend for the analysis routines is given in (e)

4.3.2 Outlook

Moving forward, there are several key areas that warrant attention and offer ways for future exploration. The complexity introduced by kinetic heterogeneity, especially with overlapping FRET states, underscores the need for enhanced model selection strategies. Analysis results need cautious interpretation of uncertainties in rate constants, considering variations in uncertainty measures reported by different tools. Future efforts could focus on establishing common standards for various parameters and fitting strategies, fostering consistency and comparability across diverse analysis approaches. A notable prospect lies in leveraging machine learning, particularly deep learning approaches, to enhance the characterization of individual states with distinct noise patterns. The application of machine learning for model-free kinetic analysis holds promise for improving model selection, providing a more accurate and unbiased understanding of biomolecular dynamics. This initiative is expected to facilitate future collaborations and accelerate the dissemination of theoretical developments within the single-molecule experimentalist community. Future benchmark studies could delve into assessing the 'data greediness' of analysis tools, exploring the amount of data and transitions-per-trace required for accurate rate inference. Understanding these requirements will contribute to a more informed selection of tools based on experimental conditions.

4.4 Paper 4: Single-molecule FRET reveals conformational changes of bacterial adhesin SdrG upon ligand binding

Single-molecule Förster Resonance Energy Transfer (FRET) reveals conformational changes of bacterial adhesin SdrG upon ligand binding

Bartnik, K., Wanninger, S., Milles, L., Rathnayak, I., Lamb, D.C., *Unpublished Manuscript*.

This paper analyses the conformational dynamics of the adhesin SD-repeat protein G (SdrG) in the absence and presence of the human fibrinogen β chain (Fg β).

4.4.1 Motivation and main results

The adhesin SD-repeat protein G (SdrG) plays a crucial role in the initiation of nosocomial infections caused by the opportunistic pathogen *Staphylococcus epidermis*. Single-molecule force spectroscopy and MD simulations revealed that the SdrG-Fg β interaction exhibits an unusually high binding strength, surpassing 2 nN.^{71,72} At that time, the SdrG-Fg β interaction represent the highest known binding force for non-covalent bonds, highlighting the extreme mechanostability of the SdrG adhesin. Regarding its binding mechanism, a dynamic "Dock, Lock and Latch mechanism" (DLL) was proposed based on crystal structures.⁷³ The DLL mechanism involves a host target, usually a peptide on the order of 15 residues, which is first bound (dock) and then buried (lock) between two Immunoglobulin-like fold domains N2 and N3. This mechanism was later supported by ensemble-level FRET experiments, qualitatively describing the conformational changes in the locking strand of SdrG upon peptide addition.⁷⁴ While crystallographic techniques were able to solve the structure of the SdrG-Fg β complex, the unbound state of SdrG remained elusive due to the undefined state of the locking strand, indicating inherent dynamics.⁷¹ Hence, this work aims to provide a detailed understanding of these dynamics at the single-molecule level and elucidate the unbound state of SdrG. This was achieved by combining single-molecule FRET (smFRET) experiments with molecular dynamics (MD) simulations, characterizing the conformational states of SdrG in the presence and absence of the human fibrinogen β chain (Fg β). In the absence of Fg β , both experimental and computational data revealed the dynamic nature of the locking strand on the sub-millisecond timescale, showing a preference for a specific orientation without spontaneous switching between open and closed states. Additionally, slow conformational changes on the second time scale are observed on the single-molecule level, providing additional insights into the low affinity between SdrG and Fg β . Contrary to earlier hypotheses, the data suggest that the presence of Fg β is a prerequisite for the transition from the open to the closed SdrG conformation.

4.5 Brief description of the method

One of the main objectives of this project was to structurally resolve the unbound state of SdrG since the position of the locking strand could not be solved by X-ray crystallography. We approached this open question by combining FRET experiments with MD simulations. The experimental distance data was obtained by measuring the FRET efficiency of multiple SdrG constructs with differently positioned fluorophores using PIE MFD⁶¹. For the computational data, the unbound state of SdrG needed for initializing the simulation was constructed starting from the well resolved crystal structure of SdrG bound to fibrinogen β . First, the fibrinogen β was removed and the locking strand was pulled away from the protein using OpenMM dynamics provided by the ChimeraX software.⁷⁵ After local energy minimization of the locking strand and equilibration procedures of the solvated protein, the all-atom MD simulation of the unbound state of SdrG was performed using the Gromacs 5.1.4 software.⁷⁶ The MD simulation involved a total number of approximately 120,000 atoms and a sufficient time duration of 1.2 μ s to sample a broad distribution of locking strand positions. All simulated structures were screened for specific SdrG conformations that are consistent with the experimentally determined distances from a total of five different FRET pairs. This screening process was performed using the FRET Positioning and Screening (FPS) software⁷⁷. By calculating the accessible volumes (AV) of each FRET pair for all simulated conformations, the FRET-averaged distances and expected FRET efficiencies

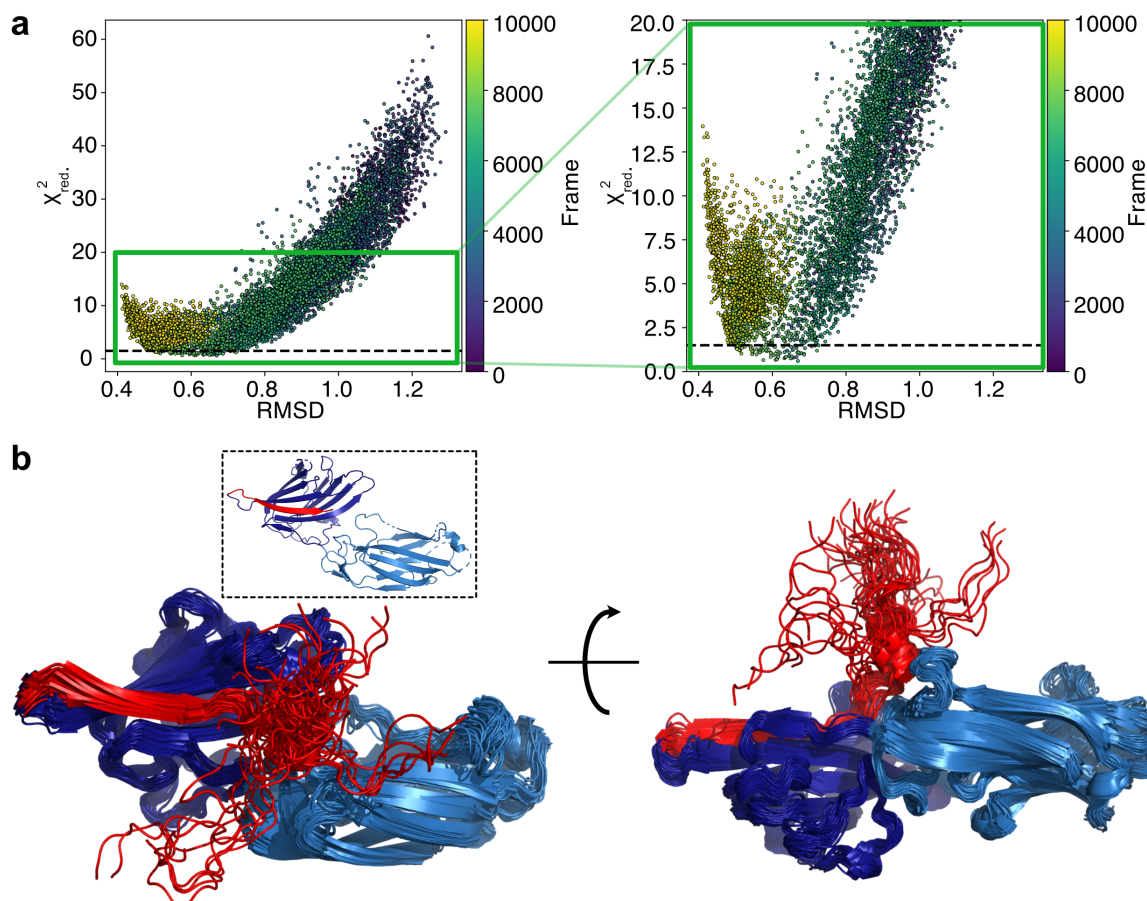


Figure 4.4 | FRET-restrained screening of simulated SdrG conformations. (a) FRET χ_{red}^2 values and RMSD, calculated from the superimposed crystal structure, of different conformations (points) sampled in the all-atom MD simulation of SdrG. Conformations below the χ_{red}^2 threshold of 1.5 represent the best fit to the experimentally determined inter-dye distances. (b) Structural model of the unbound SdrG based on smFRET data and all-atom MD simulations. Illustrated are two perspectives of 57 overlaid structures below the χ_{red}^2 threshold shown in panel (a). The subdomains N2 and N3 are colored in light and dark blue, respectively, and the locking strand (residues 569-596) is shown in red. The inset shows the previous structural model based on X-ray crystallography, which could not resolve the C-terminal part of the locking strand.⁷⁴

were obtained. Subsequently, the deviations of the experimental and theoretical distances of all five FRET pairs can be calculated and compared to the original crystal structure, which is illustrated in Figure 4.4a. Following the recommendations of Kalinin *et. al*⁷⁷, a χ_{red}^2 threshold of 1.5 was applied, yielding 57 conformations that can be considered good approximations to the experimental data (Figure 4.4b). Notably, the locking strand of SdrG is not bound to any domain in these conformations and multiple positions of the locking strand are consistent with the measured FRET efficiencies. This is in line with the intrinsic dynamics on the sub-millisecond timescale observed in the PIE MFD experiments.

4.5.1 Outlook

The synergy between single-molecule Förster Resonance Energy Transfer (smFRET) and all-atom molecular dynamics (MD) simulations has proven powerful in elucidating the conformational dynamics of SdrG. The structural model proposed for the unbound SdrG provides insights into the intrinsic dynamics of the locking strand, validating the DLL binding mechanism. In the context of developing therapeutic strategies, the central role of the locking strand acting as a "fishing rod" could be exploited by designing molecules that disrupt the binding mechanism. Exploring the forces and conditions that influence the stability of SdrG-Fg β interactions could provide valuable insights into the protein's behavior in complex milieus such as those

encountered during infection. This could involve experiments using relevant biological models to simulate the physiological conditions encountered during infection. The obtained insights could have implications not only for understanding bacterial adhesion but also for developing biomaterials and for bioengineering applications where mechanical stability is crucial.

Chapter 5

Summary and Conclusion

In the pursuit of unraveling the intricate dynamics of single-molecule systems, this thesis embarks on a multifaceted exploration, combining pre-existing methods and developing novel analysis approaches. The development of *Deep-LASI* (Deep-Learning Assisted Single-molecule Imaging analysis) emerged as a pivotal contribution, addressing the persistent challenges of time-consuming and potentially biased data analysis in single-molecule experiments. This software suite, powered by state of the art deep neural networks (DNNs), was extensively benchmarked using ground truth simulations and experimental data, demonstrating remarkable prediction accuracy and speed. The pre-trained ensemble of DNNs analyzed previously unseen data sets in a fraction of the time required by a human, achieving classifications comparable with the thorough classifications of expert users and outperforming traditional hidden Markov models in quantitative kinetic rate analysis. By removing of the time-consuming and complex analysis bottleneck, *Deep-LASI* unlocks the full potential of multicolor FRET studies, enabling studies previously deemed unfeasible. Designed with robustness and ease of use in mind, *Deep-LASI* facilitates the application of its deep learning techniques to a large variety of data. The flexibility of the integrated DNNs to various experimental conditions is demonstrated by their successful auto-analysis of HSP70 data and their consistent prediction of state dwell times of a DNA origami structure measured on different microscope configurations. Notably, all this was achieved without the need for retraining the DNNs, highlighting their ability to generalize their tasks and the advantage of using well-designed simulations as training data. An additional key advantage of using simulations is the possibility to refine and expand the capabilities of the software to meet even more diverse needs of researchers in the field. Since the DNN architectures are highly adaptable to other kinds of time-series data, the individual models can be readily trained for completely different classification tasks, even beyond fluorescence studies.

Besides software development, this thesis presents a collaborative effort to scrutinize numerous kinetic analysis tools designed for inferring quantitative kinetic rate constants from single-molecule FRET trajectories. While acknowledging the preference toward model-free approaches for unbiased data analysis, machine learning techniques, such as hidden Markov models (HMM), showed superior robustness toward data heterogeneity. Beyond the comparative analysis, this study identifies and addresses conceptual oversights in existing tools, contributes to our understanding of human influence and offers general recommendations for future studies. The findings further reveal that, while all tools accurately inferred the number of states when the FRET efficiencies were clearly separated, challenges arose in the presence of overlapping FRET states and kinetic heterogeneities. The issue of degenerate states could be solved using three-color FRET, as it increases the dimensionality of the data. However, the limited availability of tools for three-color FRET analysis emphasized the necessity for further advancements in this direction.

Within the framework of this thesis, a hybrid approach of combining computational methods with single-molecule FRET experiments is used to study the conformational dynamics of the adhesin protein SdrG. Recognizing the limitations of comparing FRET distance information to crystal structures, especially in dynamic systems, this hybrid approach leverages the strengths of both experimental and computational techniques. While MD simulations offer highly resolved structures and dynamics, their dependence on biased crystal structures poses challenges. Screening MD simulations for structures that are consistent with the measured FRET distances in real experiments addresses this issue. Following this methodology, new insights and more robust knowledge were obtained about the about the behaviour of SdrG and its dynamic locking strand.

In the absence of the target peptide, Fg β , the locking strand of SdrG shows intrinsic dynamics on the sub-millisecond timescale with no defined binding state, extending into solution. Additionally, the data reveals that the presence of Fg β is a prerequisite for the transition from the open to the closed conformation of SdrG.

Concluding the examination of the conformational dynamics of the adhesin protein SdrG, it is important to underscore the overarching emphasis on artificial intelligence (AI), which remains the primary focus of this thesis. The application of AI in single-molecule research holds great potential for addressing various challenges and unlocking new capabilities. One of the major advantages, as demonstrated by *Deep-LASI*, is the significant acceleration of data analysis. In the future, the prediction speed of DNNs could be leveraged to a point where real-time monitoring and classification of single molecules becomes feasible. Combined with other machine learning methods to automate and optimize data acquisition, single-molecule imaging studies could be conducted at a scale previously thought unimaginable. However, while AI technologies offer substantial benefits, researchers need to exercise caution and address several critical considerations to ensure responsible and reliable use. Despite common misconceptions, DNNs are not inherently bias-free. They heavily depend on the quality and representativeness of the training data, leading to inherent biases in these DNNs. Hence, DNNs are usually trained on very large data sets to mitigate these biases. While this is an appropriate solution in many fields, the question arises as to how much bias is still useful in single-molecule data analysis. The intricate nature of molecular behaviors, coupled with the limited information provided by noisy intensity trajectories, introduces complexities and potential ambiguities. For instance, a specific intensity pattern in one sample may be accurately classified as an artifact by a DNN, whereas the same pattern might contain useful information in other types of systems. Balancing the fine line between bias and bias-free outcomes in single-molecule data analysis is paramount and navigating this equilibrium is particularly challenging when training DNNs. One possible pathway for addressing this challenge involves moving beyond a one-size-fits-all model. Instead of relying on a singular model, a diverse ensemble of models could be trained, each with distinct biases deemed adequate and useful for specific scenarios. This approach acknowledges the nuanced nature of single-molecule systems and the varying contexts in which they operate. Training a multitude of models with tailored biases may enhance the adaptability of the AI system and provides researchers with a more comprehensive toolkit. However, implementing such a strategy requires careful consideration of the selection and validation of biases for each model within the ensemble. Striking the right balance between diversity and relevance is crucial to ensure that the biases introduced are indeed useful and contribute to the robustness of the overall AI system.

Since DNNs remain approximators of one particular function, it becomes evident that the broader trajectory of AI requires a shift away from depending solely on individual models. The future of AI lies not merely in the refinement of singular models but in the strategic orchestration of diverse ensembles. Developing ensembles of DNNs emerges as a promising strategy in all fields. Much like the varied skill sets and perspectives brought by individuals to a team, each DNN within an ensemble can contribute its unique strengths and biases to address different aspects of a given problem. Additionally, the concept of cross-validation within an ensemble, where different models validate and correct each other, parallels the peer-review and cross-verification process within a human team. The ensemble approach not only mirrors the collaborative nature of human teamwork but also underscores the importance of diversity in addressing complex problems. By acknowledging the intricacies of biases, harnessing diverse models and prioritizing transparency, the future of AI in science can advance securely and responsibly.

Bibliography

- [1] F. Rosenblatt. *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Report: Cornell Aeronautical Laboratory. Cornell Aeronautical Laboratory, 1957.
- [2] C. M. BERNERS-LEE. “Cybernetics and Forecasting”. *Nature* 219.5150 (July 1968), pp. 202–203.
- [3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. *Nature* 521.7553 (May 2015), pp. 436–444.
- [4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. *Journal of Machine Learning Research* 15.56 (June 2014), pp. 1929–1958.
- [5] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. *Journal of Machine Learning Research* 37 (July 2015).
- [6] Johannes Thomsen, Magnus Berg Sletfjerding, Simon Bo Jensen, Stefano Stella, Bijoya Paul, Mette Galsgaard Malle, Guillermo Montoya, Troels Christian Petersen, and Nikos S Hatzakis. “DeepFRET, a software for rapid and automated single-molecule FRET data classification using deep learning.” *Elife* 9 (Nov. 2020).
- [7] Jieming Li, Leyou Zhang, Alexander Johnson-Buck, and Nils G Walter. “Automatic classification and segmentation of single-molecule fluorescence time traces with deep learning.” *Nat Commun* 11.1 (Nov. 2020), p. 5833.
- [8] Elias Nehme, Daniel Freedman, Racheli Gordon, Boris Ferdman, Lucien E. Weiss, Onit Alalouf, Tal Naor, Reut Orange, Tomer Michaeli, and Yoav Shechtman. “DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning”. *Nature Methods* 17.7 (July 2020), pp. 734–740.
- [9] Benjamin Midtvedt, Saga Helgadottir, Aykut Argun, Jesús Pineda, Daniel Midtvedt, and Giovanni Volpe. “Quantitative digital microscopy with deep learning”. *Applied Physics Reviews* 8.1 (Feb. 2021).
- [10] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstern, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. “Highly accurate protein structure prediction with AlphaFold.” *Nature* 596.7873 (Aug. 2021), pp. 583–589.
- [11] Garrett B Goh, Nathan O Hodas, and Abhinav Vishnu. “Deep learning for computational chemistry.” *J Comput Chem* 38.16 (June 2017), pp. 1291–1307.
- [12] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. “Deep learning for computational biology”. *Molecular Systems Biology* 12.7 (2016), p. 878. eprint: <https://www.embopress.org/doi/pdf/10.15252/msb.20156651>.
- [13] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. “Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics”. *Physical Review Letters* 120.14 (Apr. 2018).

- [14] Ali Punjani, John L Rubinstein, David J Fleet, and Marcus A Brubaker. “cryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination”. *Nature Methods* 14.3 (2017), pp. 290–296.
- [15] Artur Speiser, Lucas-Raphael Müller, Philipp Hoess, Ulf Matti, Christopher J Obara, Wesley R Legant, Anna Kreshuk, Jakob H Macke, Jonas Ries, and Srinivas C Turaga. “Deep learning enables fast and dense single-molecule localization with high accuracy.” *Nat Methods* 18.9 (Sept. 2021), pp. 1082–1090.
- [16] Florian Schueder, Johannes Stein, Florian Stehr, Alexander Auer, Bianca Sperl, Maximilian T Strauss, Petra Schwille, and Ralf Jungmann. “An order of magnitude faster DNA-PAINT imaging by optimized sequence design and buffer conditions.” *Nature methods* (Oct. 2019), pp. 1–4.
- [17] Anna K Wozniak, Gunnar F Schröder, Helmut Grubmüller, Claus A M Seidel, and Filipp Oesterhelt. “Single-molecule FRET measures bends and kinks in DNA.” *Proceedings of the National Academy of Sciences of the United States of America* 105.47 (Nov. 2008), pp. 18337–18342.
- [18] Ucheor B Choi, Pavel Strop, Marija Vrljic, Steven Chu, Axel T Brunger, and Keith R Weninger. “Single-molecule FRET-derived model of the synaptotagmin 1–SNARE fusion complex”. *Nature Structural & Molecular Biology* 17.3 (Feb. 2010), pp. 318–324.
- [19] Eitan Lerner, Anders Barth, Jelle Hendrix, Benjamin Ambrose, Victoria Birkedal, Scott C Blanchard, Richard Börner, Hoi Sung Chung, Thorben Cordes, Timothy D Craggs, Ashok A Deniz, Jiajie Diao, Jingyi Fei, Ruben L Gonzalez, Irina V Gopich, Taekjip Ha, Christian A Hanke, Gilad Haran, Nikos S Hatzakis, Sungchul Hohng, Seok-Cheol Hong, Thorsten Hugel, Antonino Ingargiola, Chirlmin Joo, Achillefs N Kapanidis, Harold D Kim, Ted Laurence, Nam Ki Lee, Tae-Hee Lee, Edward A Lemke, Emmanuel Margeat, Jens Michaelis, Xavier Michalet, Sua Myong, Daniel Nettels, Thomas-Otavio Peulen, Evelyn Ploetz, Yair Razvag, Nicole C Robb, Benjamin Schuler, Hamid Soleimanejad, Chun Tang, Reza Vafabakhsh, Don C Lamb, Claus AM Seidel, and Shimon Weiss. “FRET-based dynamic structural biology: Challenges, perspectives and an appeal for open-science practices”. *eLife* 10 (Mar. 2021). Ed. by Olga Boudker, e60416.
- [20] Björn Hellenkamp, Sonja Schmid, Olga Doroshenko, Oleg Opanasyuk, Ralf Kühnemuth, Soheila Rezaei Adariani, Benjamin Ambrose, Mikayel Aznauryan, Anders Barth, Victoria Birkedal, Mark E Bowen, Hongtao Chen, Thorben Cordes, Tobias Eilert, Carel Fijen, Christian Gebhardt, Markus Götz, Giorgos Gouridis, Enrico Gratton, Taekjip Ha, Pengyu Hao, Christian A Hanke, Andreas Hartmann, Jelle Hendrix, Lasse L Hildebrandt, Verena Hirschfeld, Johannes Hohlbein, Boyang Hua, Christian G Hübner, Eleni Kallis, Achillefs N Kapanidis, Jae-Yeol Kim, Georg Krainer, Don C Lamb, Nam Ki Lee, Edward A Lemke, Brié Levesque, Marcia Levitus, James J McCann, Nikolaus Naredi-Rainer, Daniel Nettels, Thuy Ngo, Ruoyi Qiu, Nicole C Robb, Carlheinz Röcker, Hugo Sanabria, Michael Schlierf, Tim Schröder, Benjamin Schuler, Henning Seidel, Lisa Streit, Johann Thurn, Philip Tinnefeld, Swati Tyagi, Niels Vandenberg, Andrés Manuel Vera, Keith R Weninger, Bettina Wünsch, Inna S Yanez-Orozco, Jens Michaelis, Claus A M Seidel, Timothy D Craggs, and Thorsten Hugel. “Precision and accuracy of single-molecule FRET measurements—a multi-laboratory benchmark study.” *Nature methods* 15.9 (Sept. 2018), pp. 669–676.
- [21] Evangelos Sisamakidis, Alessandro Valeri, Stanislav Kalinin, Paul J Rothwell, and Claus A M Seidel. *18-Accurate Single-Molecule FRET Studies Using Multiparameter Fluorescence Detection*. Vol. 475. Single Molecule Tools, Part B: Super-Resolution, Particle Tracking, Multiparameter, and Force Based Methods. Elsevier Inc., Jan. 2010.
- [22] Nam Ki Lee, Achillefs N. Kapanidis, You Wang, Xavier Michalet, Jayanta Mukhopadhyay, Richard H. Ebricht, and Shimon Weiss. “Accurate FRET Measurements within Single Diffusing Biomolecules Using Alternating-Laser Excitation”. *Biophysical Journal* 88.4 (Apr. 2005), pp. 2939–2953.
- [23] Ganesh Agam, Christian Gebhardt, Milana Popara, Rebecca Mächtel, Julian Folz, Benjamin Ambrose, Neharika Chamachi, Sang Yoon Chung, Timothy D. Craggs, Marijn de Boer, Dina Grohmann, Taekjip Ha, Andreas Hartmann, Jelle Hendrix, Verena Hirschfeld, Christian G. Hübner, Thorsten Hugel, Dominik Kammerer, Hyun-Seo Kang, Achillefs N. Kapanidis, Georg Krainer, Kevin Kramm, Edward A.

- Lemke, Eitan Lerner, Emmanuel Margeat, Kirsten Martens, Jens Michaelis, Jaba Mitra, Gabriel G. Moya Muñoz, Robert B. Quast, Nicole C. Robb, Michael Sattler, Michael Schlierf, Jonathan Schneider, Tim Schröder, Anna Sefer, Piau Siong Tan, Johann Thurn, Philip Tinnefeld, John van Noort, Shimon Weiss, Nicolas Wendler, Niels Zijlstra, Anders Barth, Claus A. M. Seidel, Don C. Lamb, and Thorben Cordes. “Reliability and accuracy of single-molecule FRET studies for characterization of structural dynamics and distances in proteins”. *Nature Methods* 20.4 (2023), pp. 523–535.
- [24] Sungchul Hohng, Chirlmin Joo, and Taekjip Ha. “Single-molecule three-color FRET.” *Biophys J* 87.2 (Aug. 2004), pp. 1328–1337.
- [25] Anders Barth, Lena Voith von Voithenberg, and Don C Lamb. “Quantitative Single-Molecule Three-Color Förster Resonance Energy Transfer by Photon Distribution Analysis.” *J Phys Chem B* 123.32 (Aug. 2019), pp. 6901–6916.
- [26] Janghyun Yoo, John M. Louis, Irina V. Gopich, and Hoi Sung Chung. “Three-Color Single-Molecule FRET and Fluorescence Lifetime Analysis of Fast Protein Folding”. *The Journal of Physical Chemistry B* 122.49 (Sept. 2018), pp. 11702–11720.
- [27] Stefan Ernst, Monika G Düser, Nawid Zarrabi, and Michael Börsch. “Three-color Förster resonance energy transfer within single F_0F_1 -ATP synthases: monitoring elastic deformations of the rotary double motor in real time.” *Journal of Biomedical Optics* 17.1 (Jan. 2012), p. 011004.
- [28] Jaehyoun Lee and Tae-Hee Lee. “Single-Molecule Investigations on Histone H2A-H2B Dynamics in the Nucleosome.” *Biochemistry* 56.7 (Feb. 2017), pp. 977–985.
- [29] Lena Voith von Voithenberg, Anders Barth, Vanessa Trauschke, Benjamin Demarco, Swati Tyagi, Christine Koehler, Edward A. Lemke, and Don C. Lamb. “Comparative analysis of the coordinated motion of Hsp70s from different organelles observed by single-molecule three-color FRET”. *Proceedings of the National Academy of Sciences* 118.33 (2021), e2025578118. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.2025578118>.
- [30] John Sowa. “D. B. Lenat and R. V. Guha, Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.” *Artif. Intell.* 61 (Jan. 1993), pp. 95–104.
- [31] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation Learning: A Review and New Perspectives”. *IEEE Trans. Pattern Anal. Mach. Intell.* 35.8 (Aug. 2013), pp. 1798–1828.
- [32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Back-propagation Applied to Handwritten Zip Code Recognition”. *Neural Computation* 1.4 (1989), pp. 541–551.
- [33] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: [1409.1556](https://arxiv.org/abs/1409.1556) [cs.CV].
- [34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. *CoRR* abs/1512.03385 (2015). arXiv: [1512.03385](https://arxiv.org/abs/1512.03385).
- [35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.
- [36] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780.
- [37] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. “A survey on long short-term memory networks for time series prediction”. *Procedia CIRP* 99 (July 2021). 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, pp. 650–655.

- [38] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. PMLR. 2010.
- [39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015.
- [40] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. *Nature* 323.6088 (Oct. 1986), pp. 533–536.
- [41] Herbert E. Robbins. “A Stochastic Approximation Method”. *Annals of Mathematical Statistics* 22 (1951), pp. 400–407.
- [42] J. Kiefer and J. Wolfowitz. “Stochastic Estimation of the Maximum of a Regression Function”. *The Annals of Mathematical Statistics* 23.3 (1952), pp. 462–466.
- [43] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [44] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747 [cs.LG].
- [45] *RMSprop: normalize the gradient*. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf. Accessed: 2010-09-30. 2012.
- [46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the Inception Architecture for Computer Vision”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [47] Babak Hassibi and David G. Stork. “Second order derivatives for network pruning: Optimal brain surgeon”. *Advances in Neural Information Processing Systems (NeurIPS)*. 1993.
- [48] Simon Wanninger, Pooyeh Asadiatouei, Johann Bohlen, Clemens-Bässel Salem, Philip Tinnefeld, Evelyn Ploetz, and Don C. Lamb. “Deep-LASI: deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures”. *Nature Communications* 14.1 (2023), p. 6564.
- [49] A. Jabłoński. “Über den Mechanismus der Photolumineszenz von Farbstoffphosphoren”. *Zeitschrift für Physik* 94.1 (Jan. 1935), pp. 38–46.
- [50] Richeng Lin, Wei Zheng, Liang Chen, Yanming Zhu, Mengxuan Xu, Xiaoping Ouyang, and Feng and Huang. “X-ray radiation excited ultralong (>20,000 seconds) intrinsic phosphorescence in aluminum nitride single-crystal scintillators”. *Nature Communications* 11.1 (2020).
- [51] Edward U. Condon. “Nuclear Motions Associated with Electron Transitions in Diatomic Molecules”. *Phys. Rev.* 32.6 (Dec. 1928), pp. 858–872.
- [52] Edward Condon. “A Theory of Intensity Distribution in Band Systems”. *Physical Review* 28.6 (Dec. 1926), pp. 1182–1201.
- [53] J. Franck and E. G. Dymond. “Elementary processes of photochemical reactions”. *Transactions of the Faraday Society* 21. February (1926), pp. 536–542.
- [54] N. A. Nemkovich and A. N. Rubinov. “Spectral inhomogeneity and wavelength-dependent rotation of probe molecules in membranes”. *Journal of Fluorescence* 5.3 (Sept. 1995), pp. 285–294.
- [55] Michael Kasha. “Characterization of electronic transitions in complex molecules”. *Discussions of the Faraday Society* 9.0 (1950), pp. 14–19.
- [56] George Gabriel Stokes. “On the change of refrangibility of light”. *Philosophical Transactions of the Royal Society of London* 142 (Jan. 1852), pp. 463–562.
- [57] P. Debye and E. M. Purcell. “Rotational Brownian Motion and Dielectric Theory of Relaxation Processes”. *The Journal of Chemical Physics* 14.6 (1946), pp. 398–402.

- [58] Th Förster. "Zwischenmolekulare Energiewanderung und Fluoreszenz". *Annalen der Physik* 437.1-2 (Jan. 1948), pp. 55–75.
- [59] Vassili Ivanov, Min Li, and Kiyoshi Mizuuchi. "Impact of emission anisotropy on fluorescence spectroscopy and FRET distance measurements." *Biophysical Journal* 97.3 (Aug. 2009), pp. 922–929.
- [60] Simon Sindbert, Stanislav Kalinin, Hien Nguyen, Andrea Kienzler, Lilia Clima, Willi Bannwarth, Bettina Appel, Sabine Müller, and Claus A M Seidel. "Accurate distance determination of nucleic acids via Förster resonance energy transfer: implications of dye linker length and rigidity." *Journal of the American Chemical Society* 133.8 (Mar. 2011), pp. 2463–2480.
- [61] Volodymyr Kudryavtsev, Martin Sikor, Stanislav Kalinin, Dejana Mokranjac, Claus A M Seidel, and Don C Lamb. "Combining MFD and PIE for accurate single-pair Förster resonance energy transfer measurements." *ChemPhysChem* 13.4 (Mar. 2012), pp. 1060–1078.
- [62] Barbara K Müller, Evgeny Zaychikov, Christoph Bräuchle, and Don C Lamb. "Pulsed interleaved excitation." *Biophysical Journal* 89.5 (Nov. 2005), pp. 3508–3522.
- [63] W Becker, H Hickl, C Zander, K H Drexhage, M Sauer, S Siebert, and J Wolfrum. "Time-resolved detection and identification of single analyte molecules in microcapillaries by time-correlated single-photon counting (TCSPC)". *Review of Scientific Instruments* 70.3 (Jan. 1999), p. 1835.
- [64] Achillefs N Kapanidis, Ted A Laurence, Nam Ki Lee, Emmanuel Margeat, Xiangxu Kong, and Shimon Weiss. "Alternating-Laser Excitation of Single Molecules". *Accounts of Chemical Research* 38.7 (July 2005), pp. 523–533.
- [65] Tomasz Zal and Nicholas R.J. Gascoigne. "Photobleaching-Corrected FRET Efficiency Imaging of Live Cells". *Biophysical Journal* 86.6 (June 2004), pp. 3923–3939.
- [66] Bernhard Hochreiter, Markus Kunze, Bernhard Moser, and Johannes A. Schmid. "Advanced FRET normalization allows quantitative analysis of protein interactions including stoichiometries and relative affinities in living cells". *Scientific Reports* 9.1 (June 2019).
- [67] A Hoppe, K Christensen, and J A Swanson. "Fluorescence resonance energy transfer-based stoichiometry in living cells". *Biophysical Journal* 83.6 (Jan. 2002), pp. 3652–3664.
- [68] Christopher Thaler, Srinagesh V Koushik, Paul S Blank, and Steven S Vogel. "Quantitative Multiphoton Spectral Imaging and Its Use for Measuring Resonance Energy Transfer". *Biophysical Journal* 89.4 (Oct. 2005), pp. 2736–2749.
- [69] Srinagesh V. Koushik, Huanmian Chen, Christopher Thaler, Henry L. Puhl, and Steven S. Vogel. "Cerulean, Venus, and VenusY67C FRET Reference Standards". *Biophysical Journal* 91.12 (Dec. 2006), pp. L99–L101.
- [70] T Ha, T Enderle, D F Ogletree, D S Chemla, P R Selvin, and S Weiss. "Probing the interaction between two single molecules: fluorescence resonance energy transfer between a single donor and a single acceptor". *Proceedings of the National Academy of Sciences* 93.13 (June 1996), pp. 6264–6268.
- [71] Philippe Herman, Sofiane El-Kirat-Chatel, Audrey Beaussart, Joan A. Geoghegan, Timothy J. Foster, and Yves F. Dufrène. "The binding force of the staphylococcal adhesin SdrG is remarkably strong". *Molecular Microbiology* 93.2 (June 2014), pp. 356–368. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/mmi.12663>.
- [72] Lukas F Milles, Klaus Schulten, Hermann E Gaub, and Rafael C Bernardi. "Molecular mechanism of extreme mechanostability in a pathogen adhesin." *Science* 359.6383 (Mar. 2018), pp. 1527–1533.
- [73] K Ponnuraj, M G Bowden, S Davis, S Gurusiddappa Cell, and 2003. "A dock, lock, and latch structural model for a staphylococcal adhesin binding to fibrinogen". *Cell* 115 (Oct. 2003), pp. 217–228.
- [74] M G Bowden, A P Heuck, K Ponnuraj, E Kolosova, D Choe, S Gurusiddappa, S V L Narayana, A E Johnson, and M Höök. "Evidence for the "dock, lock, and latch" ligand binding mechanism of the staphylococcal microbial surface component recognizing adhesive matrix molecules (MSCRAMM) SdrG." *Journal of Biological Chemistry* 283.1 (Jan. 2008), pp. 638–647.

- [75] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Elaine C Meng, Gregory S Couch, Tristan I Croll, John H Morris, and Thomas E Ferrin. "UCSF ChimeraX: Structure visualization for researchers, educators, and developers." *Protein Sci* 30.1 (Jan. 2021), pp. 70–82.
- [76] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilard Pall, Jeremy C Smith, Berk Hess, and Erik Lindahl. "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers". *SoftwareX* 1-2 (Sept. 2015), pp. 19–25.
- [77] Stanislav Kalinin, Thomas Peulen, Simon Sindbert, Paul J Rothwell, Sylvia Berger, Tobias Restle, Roger S Goody, Holger Gohlke, and Claus A M Seidel. "A toolkit and benchmark study for FRET-restrained high-precision structural modeling". *Nature methods* 9.12 (Dec. 2012), pp. 1218–1225.
- [78] Markus Götz, Anders Barth, Søren S. -R. Bohr, Richard Börner, Jixin Chen, Thorben Cordes, Dorothy A. Erie, Christian Gebhardt, Mélodie C. A. S. Hadzic, George L. Hamilton, Nikos S. Hatzakis, Thorsten Hugel, Lydia Kisley, Don C. Lamb, Carlos de Lannoy, Chelsea Mahn, Dushani Dunukara, Dick de Ridder, Hugo Sanabria, Julia Schimpf, Claus A. M. Seidel, Roland K. O. Sigel, Magnus Berg Sletfjerding, Johannes Thomsen, Leonie Vollmar, Simon Wanninger, Keith R. Weninger, Pengning Xu, and Sonja Schmid. "A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories". *Nature Communications* 13.1 (Sept. 2022), p. 5402.

Acknowledgements

A long, long time ago my FAB lab journey started as a practicum for Wehne, introducing me, a naive chemistry student, to the world of laser microscopes, biological systems in motion and a lot of software bugs. Immediately, I was intrigued. Being fully aware that this was not the path of least resistance, I was committed to doing my PhD in this group and thanks to a unique mixture of extraordinary people, it has been the best time I had in my higher education.

Don, thank you for bestowing upon me the great honor (and great burden) of following in Anders's footsteps. I gained invaluable insights from you across all kinds of fields, even extending to the adventures of transporting laboratory sinks across the campus. Thank you for your rigorous writing, reading and correction of manuscripts, and your patience while doing so. Your way of supervision was incredibly motivating for me and made scientific research, dare I say it, *fun*. When I became a father midway through my PhD, you were nothing but supportive and understanding of all my decisions and new responsibilities. For that, you have my deepest gratitude.

Anders, I want to thank you for inspiring me to take on challenges by myself while simultaneously knowing exactly when to intervene and offer help. I know for sure you will be a perfect group leader. Thank you for planting ideas in my head and for filling me up with knowledge in the first few weeks of my PhD. Thank you for showing me the proper way of brewing coffee and, most importantly, thank you for convincing me to buy a mac instead of a windows laptop. It has been life-changing.

Of course, I want to thank all my collaborators for their hard work and giving me the opportunity to simply do interesting science. Evelyn and Pooyeh, thank you for your commitment and unwavering motivation to push the Deep-LASI project as far as it has gotten, it was truly a team effort. Thank you Evelyn for organizing so so much and getting the Tinnefeld lab involved. Philip and Johann, thank you for providing all the DNA origami structures, which were a perfect model system for benchmarking the software. Bässem, thank you not only for paving the way for my main project but all the good conversations I had with you. I would also like to thank Sonja Schmid for the much needed benchmark study and Kira for giving my PhD a touch of biology.

Lastly, I want to express my gratitude and appreciation to all members of AK Lamb for the great work environment. Special thanks to my previous office buddies Vanessa and Nader, and to my latest office buddies Irene and Virgile. I'll really miss the fun atmosphere.

Appendix A

Appended Papers

A.1 Paper 1: Deep-LASI: deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures⁴⁸

Wanninger, S., Asadiatouei, P., Bohlen, J., Salem, C., Tinnefeld, P., Ploetz, E., Lamb, D.C., *Nature Communications*, **14**, 6564 (2023).

DOI: <https://doi.org/10.1038/s41467-023-42272-9>



Deep-LASI: deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures

Received: 9 February 2023

Accepted: 5 October 2023

Published online: 17 October 2023



Simon Wanninger¹, Pooyeh Asadiatouei¹, Johann Bohlen¹, Clemens-Bässem Salem¹, Philip Tinnefeld¹, Evelyn Ploetz¹✉ & Don C. Lamb¹✉

Single-molecule experiments have changed the way we explore the physical world, yet data analysis remains time-consuming and prone to human bias. Here, we introduce Deep-LASI (Deep-Learning Assisted Single-molecule Imaging analysis), a software suite powered by deep neural networks to rapidly analyze single-, two- and three-color single-molecule data, especially from single-molecule Förster Resonance Energy Transfer (smFRET) experiments. Deep-LASI automatically sorts recorded traces, determines FRET correction factors and classifies the state transitions of dynamic traces all in ~20–100 ms per trajectory. We benchmarked Deep-LASI using ground truth simulations as well as experimental data analyzed manually by an expert user and compared the results with a conventional Hidden Markov Model analysis. We illustrate the capabilities of the technique using a highly tunable L-shaped DNA origami structure and use Deep-LASI to perform titrations, analyze protein conformational dynamics and demonstrate its versatility for analyzing both total internal reflection fluorescence microscopy and confocal smFRET data.

Single-molecule spectroscopy has revolutionized how we investigate the mechanism of processes on the nanometer scale. In particular, optical fluorescence imaging allows contact-free investigations of single, dynamic biomolecules, one at a time, in cells, membranes and in solutions. Single-molecule Förster Resonance Energy Transfer (smFRET) in combination with confocal microscopy or Total Internal Reflection Fluorescence (TIRF) microscopy probe distances on the nanometer scale (2.5–10 nm). While solution measurements can provide information on sub-millisecond dynamics, measurements with immobilized molecules give access to the temporal evolution of single molecules on the timescale of microseconds to minutes¹. By removing ensemble averaging, it is possible to directly measure the underlying conformational states and molecular dynamics of biomolecules. Its ability to measure accurate distances and kinetics turned smFRET into a powerful tool for deciphering molecular interaction mechanisms and structures of biomolecules^{1–3}. Typically, FRET experiments are

performed using two colors and used to probe conformational distributions and distance changes. However, also other single-molecule approaches can be used to investigate small distance changes or interactions (e.g., Metal-Induced Energy Transfer (MIET)⁴, Graphene Energy Transfer (GET)⁵, or Protein-Induced Fluorescence Enhancement (PIFE)^{6–8}).

When combining three- or more labels, multi-color FRET can probe molecular interactions between different binding partners and also measure multiple distances simultaneously, i.e. correlated motion within the same molecule^{9–11}. However, multi-color analyses remain challenging. Quantitative smFRET data analysis is strongly hampered by experimental restrictions due to, for example, a low number of usable single molecule traces, data with a low signal-to-noise ratio (SNR), or short traces due to photochemistry. Overcoming these limitations requires large data volumes as very few molecules contain the desired information with suitable quality, which significantly increases

¹Department of Chemistry and Center for NanoScience (CeNS) Ludwig-Maximilians-Universität München Butenandtstr. 5-13, 81377 Munich, Germany.

✉ e-mail: evelyn.ploetz@lmu.de; d.lamb@lmu.de

the efforts involved in sorting through the data when performed manually. Low statistics result from various reasons including molecular events exhibiting slow kinetics or rare transition probability, insufficient labeling efficiency, low SNR, quick photobleaching or spurious background. In addition, arbitrary fluctuations due to unwanted interactions and/or aggregations between binding partners hamper a concise analysis of the underlying state and kinetics.

Various approaches have been developed to overcome these time-consuming burdens, employing user-defined thresholds on the channel count rate, signal-to-noise ratio, FRET values, FRET lifetime, and donor/acceptor correlation^{12–19}. However, setting appropriate thresholds requires a substantial amount of expertise. Depending on the user, the data evaluation is prone to cognitive biases and poses a challenge to reproducible analysis results. Recently, software packages have been published that use deep-learning techniques to rapidly automate trace classification and keep user bias to a minimum^{20–22}. In particular, Thomsen et al. comprehensively demonstrated that artificial neural networks could match manual classifications and even outperform conventional methods of commonly used programs to extract valid single-molecule FRET traces²². So far, deep learning has been solely applied to single-channel and two-color FRET data to categorize the time trajectories for downstream analysis. To study structural dynamics, reflected by changes in intensity and FRET efficiencies, the kinetics are then analyzed separately typically using Hidden Markov Models (HMMs)^{23,24} approaches. Training an HMM requires knowledge of the number of states and modeling of the emission probabilities. Moreover, it assumes that the probability of a transition to the next state only depends on the current state. While the initial HMM settings are straightforward for simple systems, obtaining the optimal parameters for multi-color FRET becomes a challenging task. To date, only one software package, SMACKS¹³, allows an ensemble HMM for three-color FRET data. As the complexity of the datasets grows, the effort and the required knowledge about the system also grow.

To alleviate the shortcomings of HMM analyses, the hybridization of HMMs with Deep Neural Networks (DNN) has gained popularity^{25–29}. In contrast to HMMs, DNNs are capable of learning higher-order dependencies without prior assumptions about the number and properties of the states. A long-short-term memory (LSTM) neural network was developed to automate stoichiometry determination via photobleaching steps in fluorescence intensity traces³⁰. However, the use of DNNs for extracting quantitative kinetic information from single-molecule data has not yet been explored.

Here, we present the Deep-Learning Assisted, Single-molecule Imaging (Deep-LASI) approach, an ensemble of DNNs with architectures specifically designed to perform a fully automated analysis of single-color traces as well as two-color and three-color single-molecule FRET data. Deep-LASI begins with raw intensity traces and provides corrected FRET efficiencies, state determination, and dwell times without any prior knowledge or assumptions about the system. It classifies each time trace into different categories, identifies which fluorophores are active in each frame, which is then used for determining FRET correction factors for spectral crosstalk, direct acceptor excitation and detection efficiency, and performs a state transition analysis of the different states in dynamic traces. Deep-LASI also includes optional number-of-state classifiers to estimate the actual number of observed states within one trace. Since the pre-trained neural networks operate locally on each trace, they do not neglect rare events, which would be missed in global analysis approaches. We benchmark the performance of Deep-LASI using ground truth simulations and experimental one-, two- and three-color data using an L-shaped DNA origami structure with tunable dynamic behavior^{5,31}. The results are further compared to the manual evaluation of the data and the extracted dwell times obtained with HMM. Finally, we demonstrate the power of Deep-LASI with multiple applications: (1) titration

experiments, which would be unfeasible without Deep-LASI; (2) smFRET on a mitochondrial Hsp70 to extract substrate-specific dwell times and conformational states; and (3) the applicability of Deep-LASI to another experimental setup.

Results

The Deep-LASI approach

Deep-LASI utilizes an ensemble of pre-trained deep neural networks designed for the fully automated analysis of one-, two- and three-color single-molecule data including multi-color FRET correction and kinetic analyses (Fig. 1; Supplementary Note 1). The designed input for Deep-LASI is a single-molecule fluorescence intensity trace or traces measured directly using confocal microscopy or extracted from movies using wide-field or TIRF microscopy. In the case of two-color fluorescence data, continuous wave excitation or Alternating Laser EXcitation (ALEX) modalities can be analyzed. For three-color smFRET measurements, ALEX data is required. All available channels are fed into a combination of a Convolutional Neural Network (CNN) using the omniscience feature learning approach and a Long Short-Term Memory (LSTM) model (Supplementary Fig. 1.1).

Deep-LASI extracts spatial and temporal sequence features simultaneously and classifies every frame into a specific category (Fig. 1a). Building upon Deep-FRET for two-color FRET analysis²², we separate the traces into six categories: dynamic, static, noisy, artifact, aggregate as well as photobleached (see Supplementary Note 2 for details). The total number of categories depends on the number of input channels, i.e. the number of dyes (and alternating light sources) used in the experiment. Traces containing random artifacts, aggregates, or high noise are excluded from further analyses. The final output of the state classifier provides an estimation of the probability for each category. The summed probabilities over all non-photobleached frames serve as confidence levels for each trace. Here, user-defined thresholds can be set to increase or decrease the tolerance towards non-ideal traces to be included in further analyses. In contrast to previous networks, Deep-LASI detects photobleaching events of individual dyes and, therefore, allows the calculation of correction factors obtainable for that molecule. Traces showing no apparent state transition are classified as static and can be included, e.g. in the final corrected FRET histograms.

All sections in each trajectory identified as dynamic are transferred to the state classifier network (Fig. 1b), which is designed to detect transitions based only on the intensity data and not via the FRET efficiency. The state classifier assigns every frame to one of the multiple states present in a dynamic trace section and again provides a confidence value of state occupancy that can be used for additional thresholding. Given the state transition classifications, a Transition Density Plot (TDP) is calculated and the kinetic rates of all identified states can be extracted by fitting the corresponding dwell-time distributions (Fig. 1c). Starting from trace extraction, the TDP marks the first necessary point of human intervention, i.e., the manual selection of state transitions and the fitting procedure. Thus, user bias is kept to a minimum. No assumptions are needed regarding the number of states, state-specific emission probabilities, or other settings required for conventional methods such as Hidden Markov Models (HMM). Of course, as for any deep-learning algorithm, the output of the analysis is dependent on the quality and appropriateness of the training data used. Depending on (1) the total number of frames, (2) the yield of valid frames, (3) the computer performance, and (4) the desired confidence threshold, a given dataset can be fully categorized on a time scale of 20–100 ms per trace.

Training of Deep-LASI

To use Deep-LASI for analyzing single molecule data, we first trained the trace-classifier network on datasets appropriate for the corresponding network (i.e., one-color data, two-color data without ALEX,

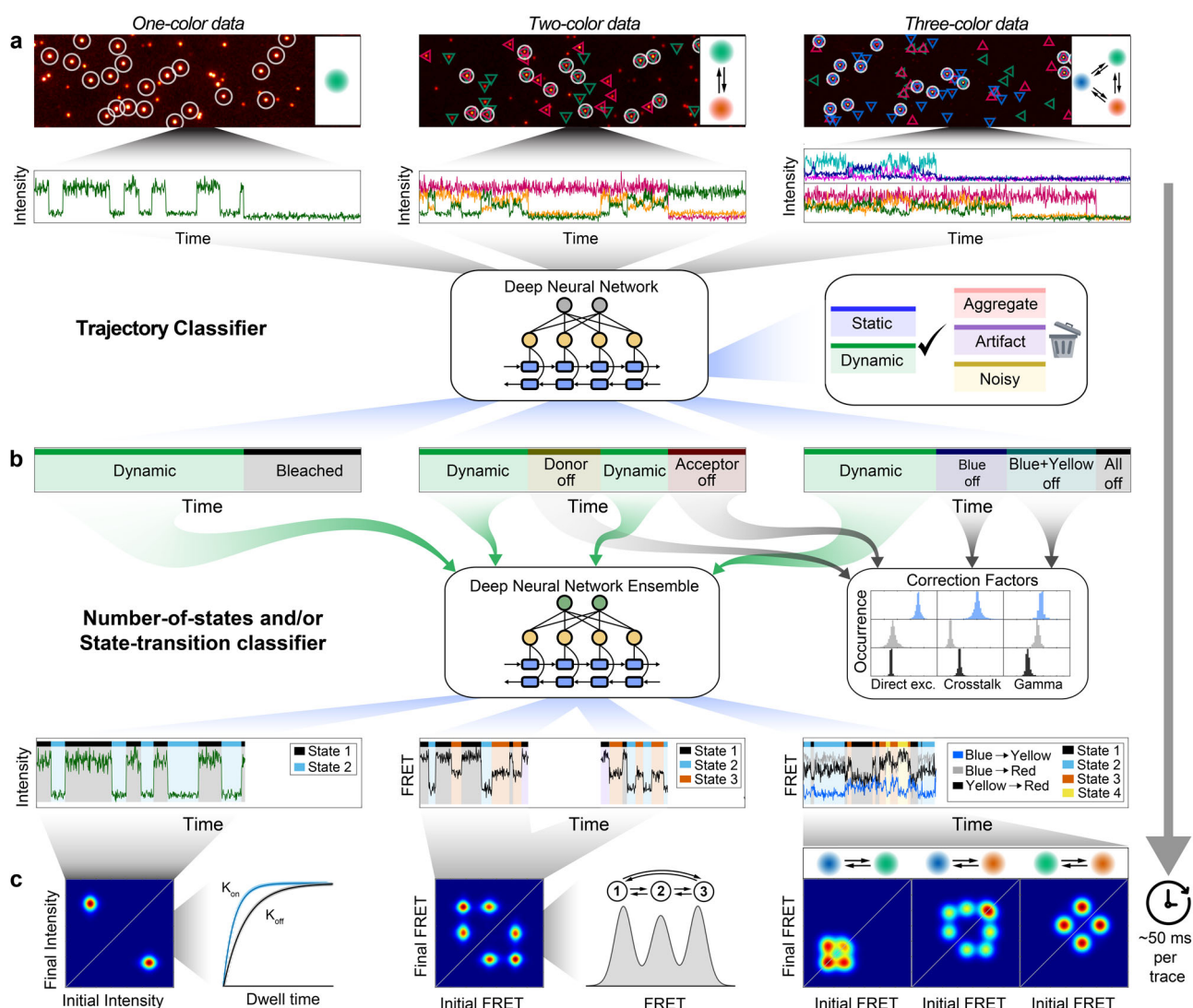


Fig. 1 | Overview of data extraction, evaluation, and analysis using Deep-LASI.

a Single-molecule data of up to three separate channels after direct and alternating laser excitation are identified, extracted, and presorted for further analyses. Each frame within the time traces is classified into categories using a hybrid CNN-LSTM. **b** A second hybrid CNN-LSTM evaluates the kinetics and state information in the

presorted data. The photobleaching information can be used for determining the correction factors to obtain accurate FRET values between two and three fluorophores. **c** Next, the interconversion rates between underlying states and absolute, distance-related FRET values are extracted from multi-color datasets.

two-color data with ALEX or three-color data with ALEX). As the noise sources in single-molecule fluorescence intensity data are well understood, simulated traces are well suited for training the neural network. In addition, it has the advantages of being able to minimize biases and quickly retrain neural network models to adjust for specific circumstances. The training datasets were designed to cover a wide range of experimental conditions and FRET efficiencies. Hence, no initial estimation of the number of states and expected FRET efficiencies are needed. A detailed description of the program architecture, simulations, training datasets and benchmarking can be found in the Methods section as well as in Supplementary Notes 1–4.

Deep-LASI contains a total of 16 pre-trained deep neural networks for state classification. Four models account for the classification and segmentation of time trajectories obtained from measurements using single-channel data acquisition, two-color FRET with continuous-wave excitation, two-color FRET with ALEX, and three-color FRET with ALEX. For each type of experiment, we provide three state-transition-classifiers trained on either two, three or four observed states, which take the output category dynamic as the input. Note that the acceptor intensity after direct excitation does not contain relevant kinetic

information and is not used in the state classifier networks. In addition, a deep neural network is provided that has been optimized for detecting the actual number of observed states and can be utilized for model selection. One network has been trained for each type of dataset (one-, two- and three-color data). The number-of-states neural networks are not essential in the automated analysis process but can serve as a safeguard against trajectories that may be out of the scope of the state transition classifiers.

Performance of Deep-LASI

A common approach to benchmark classifier models is using ground truth labeled data and calculating confusion matrices, which summarize the correct and incorrect predictions. For every trained model (using ~ 200,000 traces), we generated approximately 20,000 new traces for testing, which were not part of the training dataset. Each of the validation datasets was then fed into the corresponding model. The output predictions were compared to the ground truth labels for every frame to obtain the percentage values of true positive, false positive and false negative classifications. All trace classifier models achieve a minimum combined precision of 97% in predicting smFRET

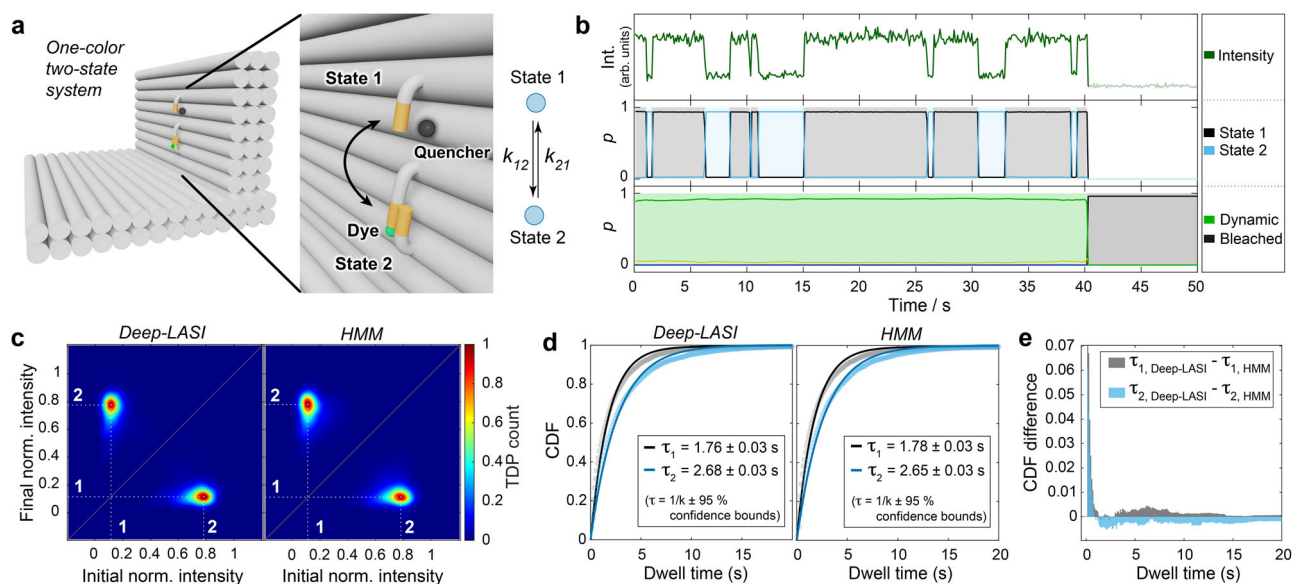


Fig. 2 | State analysis of single-color single-molecule data. **a** Sketch of the used L-shaped DNA origami structure with a single fluorophore (Cy3B) attached to a flexible tether, which changes position from state 1 \rightarrow 2 at the rate k_{12} and from state 2 \rightarrow 1 at the rate k_{21} . The zoom-in shows the two single-stranded binding sites (orange) in close and distant proximity to a quencher dye (Atto647N) bound to the DNA origami structure. **b** Representative time transient for a DNA origami structure with 7.5 nt binding strands after classification and kinetic evaluation by Deep-LASI. **c** Transition-density plots depicting the interconversion events between the two detected states 1 and 2 after trace kinetics evaluation by Deep-LASI (left, number of transitions $n = 25,948$) and by Hidden-Markov Modeling (HMM) analysis (right,

number of transitions $n = 19,390$). Both approaches identify identical states. **d** Cumulative probability Distribution Functions (CDFs) of the dwell times: The mono-exponential fits obtained by both methods reveal equivalent dwell times of approximately 1.75 and 2.65 s for the upper (State 1) and lower (State 2) binding sites, respectively. The errors in the dwell times are the 95% confidence intervals returned by the fitting procedure (estimated from the Jacobian matrix). **e** A comparison of the CDFs was determined using Deep-LASI and HMM. Deep-LASI is already sensitive at time scales on the order of the acquisition time. The average difference is less than 1% between both methods. Source data are provided as a Source Data file.

categories, i.e. static or dynamic, and 96% in predicting non-smFRET categories (Supplementary Figs. 3.1 and 3.2).

Our number-of-states and state-transition classifiers were benchmarked analogously. For the number-of-state classifiers, two states can be distinguished from multi-state trajectories with at least 98% precision whereas four states are predicted with the lowest precision of 86% for the single-channel model (Supplementary Fig. 3.3). For the state-transition classifiers, the states can be identified with accuracies of $\geq 98\%$, $\geq 90\%$ and $\geq 78\%$ for two-state, three-state and four-state models respectively (Supplementary Fig. 3.4). The comparison between all state-transition classifiers reveals a clear trend of decreasing accuracies with an increasing number of states and increasing accuracy with an increasing number of available channels. This is expected since a higher number of states have a larger probability of lower contrast, and a higher number of channels improves the robustness towards uncorrelated noise. Since confusion matrices do not reveal any underlying dependencies, we additionally benchmarked the state transition classifiers with HMM by calculating the precision of the state label prediction for a broad range of noise levels, FRET state differences and dynamic time scales (Supplementary Fig. 3.5). Overall, the performance of state classifiers is at least on par with HMM at low noise levels and outperforms HMM at high noise levels by up to 30%. To investigate the advantage of using the information in the entire dataset for the HMM analysis, we also compared the performance of Deep-LASI with a local and a global HMM on idealized synthetic data (Supplementary Fig. 3.6). Global HMM performs significantly better than local HMM in this case and is on par with Deep-LASI.

As a last test, we compared the performance of Deep-LASI with other kinetic analysis routines that have been recently published in a multi-laboratory study³². We chose to analyze the two-state datasets as these require no user input and the analysis can be performed without bias. Deep-LASI returned values corresponding to the ground truth for

the simulated dataset and close to the average values obtained for the experimental dataset (Supplementary Fig. 3.8).

Deep-LASI analyses of DNA origami structures

Next, we benchmarked the potential of Deep-LASI to automatically analyze experimental data obtained from DNA origami structures. DNA origami is extensively used in bio-nanotechnology and has the advantage of being programmable with high precision and controllability. In particular, we choose an L-shaped DNA nanostructure with a dynamic, fluorescently labeled 19 nucleotide (nt) single-stranded DNA pointer. The geometry of the DNA structure was originally designed for measuring energy transfer to a graphene surface^{5,31}. The single-stranded DNA pointer, along with two or three exchangeable docking strands of different complementary sequences, allows the number of states, position of the dyes, and kinetic rate to be programmed as desired. Hence, it is an ideal test system for measuring and extracting kinetic information from smFRET traces. FRET efficiencies and kinetic rates could be tuned by varying the position and complementary sequence length of binding strands on the DNA origami platform. We designed various DNA origami structures with one-, two-, and three-color labels and measured them on the single-molecule level.

In the first assay, we assessed Deep-LASI's capability to evaluate single-color data. For this, we probed one-color single-molecule kinetics where the flexible pointer was labeled with Cy3B at the 3'-end. Two complementary binding sites with 8 nt complementary nucleotides containing a 1 nt mismatch at the 5'-end (referred to as 7.5 nt) were placed about 6 nm below and above the pointer position (Fig. 2a). Binding occurred by spontaneous base-pairing to single-stranded protruding strands. A single red dye, Atto647N, acting as a quencher, was attached about 3 nm aside from the upper binding site (state 1). Figure 2b shows an exemplary intensity trajectory of Cy3B classified as dynamic until photobleaching was detected by the trace classifier with

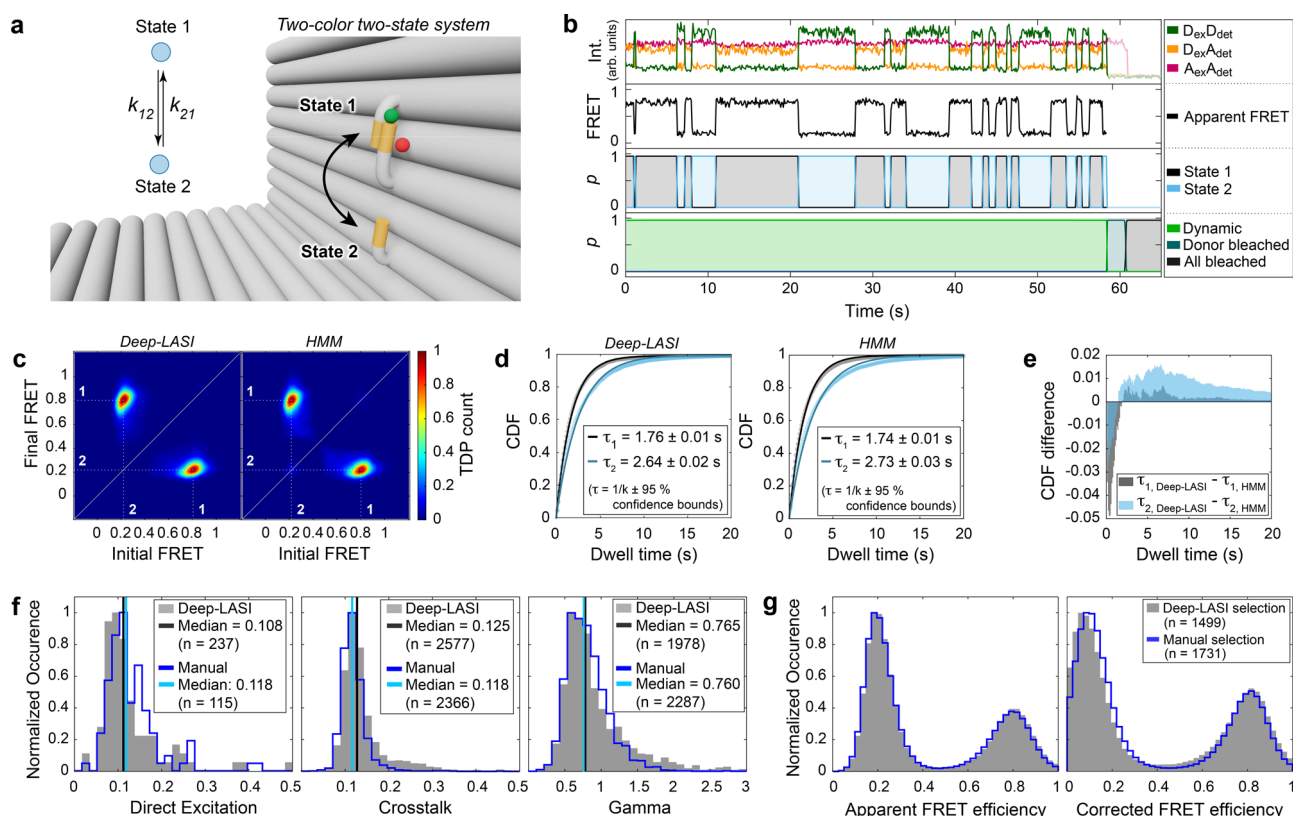


Fig. 3 | Single-molecule analysis of two-color FRET data. Experiments were performed with DNA origami structures exhibiting two FRET states. **a** Zoom-in of an L-shaped DNA origami structure labeled with Atto647N and Cy3B and corresponding kinetic scheme. The donor is attached to the flexible tether with a 7.5 nt overhang between the pointer and two single-stranded binding sites. FRET is expected between a high FRET state 1 (12 o'clock) and a low FRET state 2 (6 o'clock) interconverting at rates k_{12} and k_{21} . **b** Representative single-molecule and apparent FRET trace after alternating red-yellow (RY) laser excitation. Deep-LASI classifies the trace and determines the underlying state for each frame. D: donor; A: Acceptor; ex: excitation; det: detection. **c** TDPs determined using Deep-LASI (left) and HMM (right) are shown revealing two interconverting states with apparent FRET values of 0.8 and 0.2. The two states are labeled in white. Total number of transitions: $n_{\text{Deep-LASI}} = 15,958$, $n_{\text{HMM}} = 21,243$. **d** CDFs extracted from the TDPs

shown in (c) and mono-exponential fits yield dwell times of 1.76 s and 2.64 s, respectively. The errors in the dwell times are the 95% confidence intervals returned by the fitting procedure (estimated from the Jacobian matrix). **e** A comparison of the cumulative dwell-time distribution determined using Deep-LASI - HMM for τ_1 (gray) and τ_2 (cyan). **f** Histograms of trace-wise determined correction factors for direct excitation, crosstalk and detection efficiency, either derived automatically by Deep-LASI (gray histograms, median in black) or determined manually (blue lines, median in cyan) (see Supplementary Note 5). **g** Apparent (left) and corrected (right) frame-wise smFRET efficiency histograms for 1499 dynamic traces from a total of 6100 traces. The states have corrected peak FRET efficiencies of 0.07 and 0.81. The histograms from traces selected by Deep-LASI are shown in gray and by manual selection in blue. Source data are provided as a Source Data file.

two corresponding states determined by the state classifier as the linker moves up and down.

We compared the results from Deep-LASI with a Hidden-Markov-Model analysis (HMM) trained on the same dataset. Since the state classifier does not directly predict a pre-trained intensity value for each state, the TDP was generated by averaging the normalized intensity between transitions. Both methods yield identical TDPs (Fig. 2c). The residence time of the DNA tether in both states was determined by fitting the cumulative dwell-time distribution functions (CDFs) derived from the state-classifier of Deep-LASI with a mono-exponential fit and compared to the results from HMM. The dwell times of 1.76 s versus 1.78 s (State 1) and 2.68 s versus 2.65 s (State 2) for Deep-LASI and HMM, respectively, are in excellent agreement (Fig. 2d). The differences between the CDFs obtained by Deep-LASI and HMM (Fig. 2e) indicate that Deep-LASI identifies fast transitions close to the frame time more frequently than HMM. The overall difference at longer dwell times remains well below 1%, which proves that Deep-LASI obtains identical results to HMM with negligible differences in the extracted rates. Interestingly, although the DNA binding strands are identical in sequence and length, there are clear differences in the dwell times. We attribute this to an inherent bias in the equilibrium position of the DNA pointer and non-symmetric, non-specific dye-origami interactions. In

addition, it is unlikely that the distance to each docking strand and potentially induced stress upon binding are identical for the two binding sites, even though the binding sequence is the same. We note that the kinetics we measure here are not directly comparable to other DNA-hybridization experiments due to both interacting DNA strands being tethered to the DNA origami platform. This leads to a high local concentration of the binder strand, and multiple dissociation and rebinding events can occur before the tether switches binding sites.

In the next step, we studied Deep-LASI's ability to deal with two-color data. We investigated two-color FRET assays with two states and compared the results with a pure manual evaluation of the same data. Here, both donor and acceptor signals from the same DNA origami sample system as shown in Fig. 2a were analyzed (Fig. 3a). TIRF measurements were performed using msALEX³³ yielding donor signal (Cy3B, Channel $D_{\text{ex}}D_{\text{em}}$), sensitized emission (Channel $D_{\text{ex}}A_{\text{em}}$) and acceptor signal (Atto647N, Channel $A_{\text{ex}}A_{\text{em}}$) to obtain information about acceptor photobleaching and direct excitation. Figure 3b shows a fully classified example trace with the signals on top and the derived FRET trace below. From the trace classifier, Deep-LASI identified dynamic sections and individual photobleaching events (Fig. 3b; bottom). The dynamic section was further classified in the state transition classifier according to their state occupancy using only the two

channels of the donor and acceptor intensity after donor excitation (Fig. 3b; middle). The channel of acceptor excitation and detection does not serve as input for the state transition classifier since it does not contain valuable kinetic information. From a total of 6100 recorded traces in the dataset, 1499 traces were classified as dynamic smFRET trajectories with at least one transition.

The same traces were also sorted manually and the 1731 selected dynamic traces were analyzed using HMM³⁴ (see Supplementary Note 5 for details). TDPs from the state transition classifier and from the HMM analysis are nearly identical (Fig. 3c). Also, the corresponding dwell times, determined via mono-exponential fits to the CDFs, are similar (Fig. 3d) and correspond to the expected dwell times of the one-color sample shown in Fig. 2 (-1.75 s for state 1 and 2.68 s for State 2). A comparison of the CDFs from Deep-LASI and HMM indicates that manually selected traces contained more fast transitions than the traces selected by Deep-LASI in this case (Fig. 3e). We looked into the differences between manually selected traces and traces selected by Deep-LASI. The most common classification discrepancies between the two are discussed in Supplementary Note 4.1. Based on individual example traces, we observed a stronger influence of the non-ideal behavior of the traces outside the regions of interest (e.g. a noisy ALEX signal or nonconstant signal intensities in photobleached regions of the trace) on Deep-LASI's classification compared to that of manual selection. We also compared the output of Deep-LASI with that of a global HMM analysis executed on the same dataset (Supplementary Fig. 4.2). As expected, the global HMM was prone to miss transitions due to slight heterogeneities in the dataset.

Next, we investigated how sensitive Deep-LASI is to the training dataset. Hence, we trained two additional classifier networks using newly simulated datasets. Details are given in Supplementary Note 4.2. The consistency between the differently trained neural networks is ~90%, similar to what would be expected from analysis run on the validation datasets (Supplementary Fig. 3.1c). Interestingly, the consistency between the different neural networks is higher than between two independent users (Supplementary Fig. 4.3b).

To determine the distance between both dyes in the two FRET states, the smFRET data needs to be corrected. Deep-LASI uses the frames classified as photobleached to automatically derive the correction factors necessary for an accurate FRET calculation^{1,35,36}. In the manual analysis, the relevant regions are selected by hand (Fig. 3f, Supplementary Note 5). The correction factors agree within ~3%. Using the derived correction factors, the correct FRET efficiency is determined. The apparent (left) and corrected FRET histograms (right) of the Deep-LASI (gray histograms) and manually (blue lines) selected traces are shown in Fig. 3g. There is excellent agreement between the Deep-LASI and manually analyzed apparent FRET histograms. The difference between the corrected histograms is due to the difference in the correction factors determined and applied from the two analyses. In this case, as Deep-LASI classifies photobleaching on a per-frame basis, more frames can be used for determining the correction factors and are, thus, most likely, more accurate here. The corrected peak FRET efficiencies are 0.81 and 0.82 (State 1) and 0.08 and 0.14 (State 2) for Deep-LASI and manual evaluation, respectively, and correspond to distances of 53 and 53 Å, and 103 and 92 Å (assuming an R_0 of 68 Å⁷).

In the last step, we then tested the performance of Deep-LASI for analyzing three-color data by labeling the DNA origami structure with an additional blue dye, Atto488, at ~3 Å distance to the binding site for State 2 (Fig. 4a). The labeling sites of the yellow (Cy3b) and red (Atto647N) dyes were left unchanged to provide consistency with the previous two-color experiments. The use of three FRET pairs provides three distances simultaneously and allows the resolution of states that are degenerate for two-color FRET.

Using the six available intensity traces, each frame is categorized by the fluorophores that are active and whether the trace is static,

dynamic or should be discarded. As the acceptor intensity after acceptor excitation (R_{ex}, R_{em}) does not contain valuable kinetic information, the other 5 intensity channels for dynamic traces (before photobleaching) are given as input for the state transition classifier (Fig. 4b). Movement of the flexible tether results in an anti-correlated change in the FRET efficiency of blue to yellow (BY) and yellow to red (YR), visible in the apparent FRET panel of the example trace in Fig. 4b. For each FRET pair, a TDP can be calculated, which allows the assignment of the state number to the actual FRET populations (Fig. 4c). Note, the apparent FRET efficiency of blue to red (BR) varies with the YR FRET efficiency due to the different energy transfer pathways taken upon blue excitation. Deep-LASI classifies a state regardless of which dye is undergoing a transition, i.e. the extracted dwell time distribution of a given state is the same for all FRET pairs when there is no overlap of multiple states in the TDP. The dwell times for states 1 and 2 match with those for the one-color and two-color samples, which indicates that the transition rates are not influenced by the acceptor dyes close by (Figs. 2d, 3d, Supplementary Fig. 6.1). From a total of 2545 recorded molecules, 581 were classified as valid, dynamic three-color FRET traces. The uncorrected, framewise smFRET histograms of BY, BR and YR FRET pairs are very similar to those from the 694 manually selected traces (Supplementary Fig. 4.4a). A detailed comparison between the manual analysis of the results from Deep-LASI is given in Supplementary Notes 4.3 and 4.4.

As for two-color FRET, Deep-LASI automatically determines all correction factors obtainable per trace depending on which dyes are photoactive. The results of the automated extraction of correction factors are summarized and compared to manually derived correction factors in Supplementary Fig. 4.4b. The corresponding apparent and state-wise, corrected FRET efficiency histograms for each FRET pair are shown in Fig. 4d. While the YR FRET efficiency can be directly calculated, the corrected BY and BR FRET efficiencies are subjected to higher uncertainties due to the large number of correction factors involved (see Supplementary Note 5). In particular, their dependency on the YR FRET efficiency leads to the broadening of the distributions. To minimize this influence, we perform the correction using the state-averaged FRET efficiencies. After correction, the FRET efficiencies of State 1 (0.81) and State 2 (0.08) for the YR FRET pair are virtually identical as for the two-color system. For the BY FRET pair, State 1 and State 2 correspond to peak FRET efficiencies of 0.36 and 0.81, respectively. As expected, the two populations of the apparent BR FRET efficiency merge into one static population in the corrected histogram with a peak FRET efficiency of 0.36.

To probe the performance of the kinetic analysis from Deep-LASI, we used the tunability of the L-shaped DNA origami structure to vary the timescale of the dynamics. In addition to the 7.5 nt binding sites (Fig. 4a–d), we measured three samples using binding sites of length 7 nt with a 1 nt mismatch (referred to as 6.5 nt), 7 nt, and 8 nt (Fig. 4e). The summary of all extracted dwell times (Fig. 4f, Supplementary Figure 6.1) shows an exponential increase in the dwell times of both states with increasing binding site lengths ranging from 0.33 s to 9.5 s. Considering the camera exposure times of 32 ms (6.5 nt), 50 ms (7 nt) and 7.5 nt datasets) and 150 ms (8 nt dataset) and frame shift time of 2.2 ms, a dwell-time to frame-time ratio ranges from 9 (6.5 nt State 1) to 62 (8 nt, State 2).

To test Deep-LASI with more complex dynamics with multiple states, we constructed a three-state system with three-color labels using 7 nt binding strands at positions 6 and 12 o'clock and an additional 7.5 nt complementary binding strand at 9 o'clock (Fig. 5a). An example trace containing all possible transitions identified by Deep-LASI is shown in Fig. 5b. The TDP of the BY FRET pair (Fig. 5c, left panel) yields clearly distinguishable populations, while the TDP of the YR FRET pair (Fig. 5c, right panel) shows a degeneracy of state 3 transitions. Using the BY TDP, we determined the dwell time distributions with residence times between 0.65 s and 1.43 s (Supplementary

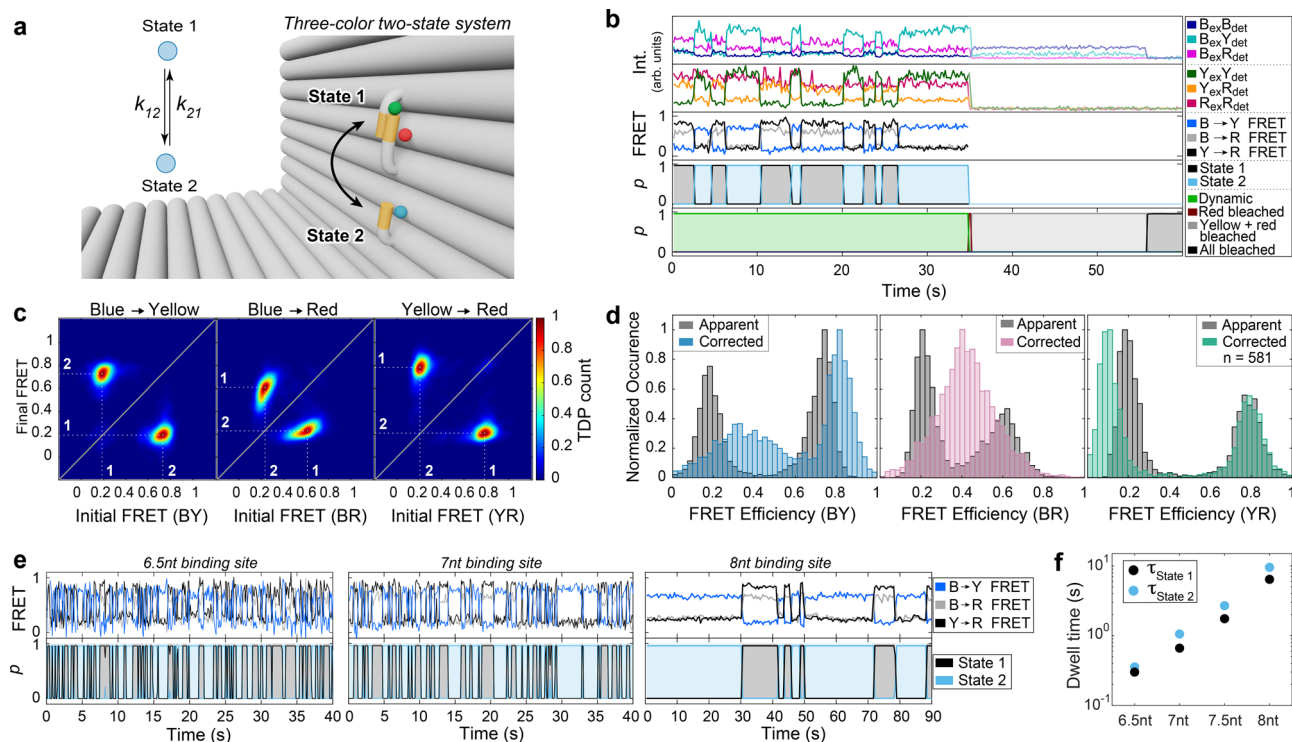


Fig. 4 | Single-molecule analysis of three-color FRET data. Experiments were performed on two-state DNA origami structures that were labeled with Atto647N, Cy3B and Atto488. While Cy3B is attached to a flexible tether, Atto647N and Atto488 are positioned close to the top (12 o'clock; state 1) and bottom (6 o'clock; state 2) binding sites, respectively. **a** Zoom-in of the L-shaped DNA origami structure and corresponding kinetic scheme. **b** Representative single-molecule intensity and FRET trajectories for binding sites with 7.5 nt overhang after alternating red-yellow-blue laser excitation. First panel: Intensities after blue excitation. Second panel: intensities after yellow and red excitation. Third panel: corresponding three-color FRET efficiencies. Fourth and fifth panels: Deep-LASI output for state transition and trace classification. B: blue; Y: yellow; R: red; ex: excitation; det: detection. **c** TDPs of the apparent FRET efficiency states reveal an apparent distance change for all three FRET pairs (BY (left), BR (middle), and YR channel (right) with dwell times of 1.75 s and 2.69 s for the upper and lower binding site, respectively, nearly

identical to the two-color DNA origami structures (Fig. 3c). Total number of transitions: 5,013. **d** Frame-wise weighted state-wise apparent (gray) and corrected (color) smFRET efficiency histograms of the BY (left), BR (middle), and YR (right) FRET pairs. As expected, the accurate FRET efficiency of the BR pair is static ($E = 0.36$). As the position of Cy3B changes from state 1 to state 2, the accurate FRET efficiency changes from 0.36 to 0.81 (BY pair) and from 0.81 to 0.08 (YR pair). **e** Upper panel: Representative three-color smFRET traces for binding sites with 6.5 nt (7 nt with 1 nt mismatch), 7 nt and 8 nt overhangs after alternating RYB laser excitation. Bottom Panel: The corresponding state determined by Deep-LASI. **f** Extracted dwell times from mono-exponential fits for the lower (blue) and upper positions (black) for 6.5 nt (τ_1 : 0.31 s, τ_2 : 0.4 s), 7 nt (τ_1 : 0.66 s, τ_2 : 1.05 s), 7.5 nt (τ_1 : 1.75 s, τ_2 : 2.69 s) and 8 nt overhangs (τ_1 : 6.41 s, τ_2 : 9.54 s) (see Supplementary Figure 6.1 for more details). nt: nucleotides. Source data are provided as a Source Data file.

Figure 6.2). The three states are well-resolved in the framewise apparent BY FRET histogram, while state 2 and state 3 are degenerate for the BR and YR FRET pairs (Fig. 5c). Applying all correction factors yields peak YR FRET efficiencies of 0.81 (state 1), 0.08 (state 2) and 0.19 (state 3). Upon correction, States 1 and 3 in the BY FRET histogram merge into a broad degenerate FRET population. However, using the state information for all three fluorophores allows us to separate out the BY FRET histograms of the individual states.

For three-color FRET, the corrected BY and BR FRET efficiencies depend on the YR FRET efficiency, and the additional corrections broaden the population. However, even though the data may be noisier, three-color experiments contain additional information, which typically allows one to resolve states that are degenerate in two-color experiments. This is exemplified in two-color FRET experiments on the same construct missing the blue fluorophore near the 6 o'clock binding site (Supplementary Note 6.3). For distinguishable states, the determined corrected FRET efficiencies and kinetic rates from two- and three-color experiments are the same. However, three-color FRET experiments enable the lifting of this degeneracy between states 2 and 3. To minimize the influence of the increased noise in three-color experiments, it is advantageous to analyze the data in proximity ratio and only convert it to corrected FRET efficiencies when necessary¹⁰. Deep-LASI can rapidly classify a large number of molecules and quickly

provide an overview of multi-state dynamics with easy access to the kinetic information.

Further applications of Deep-LASI

After extensive benchmarking, we applied Deep-LASI to various single-molecule datasets originating from biophysical assays, protein samples and experimental systems beyond TIRF microscopy. With the speedup in analysis time from days to minutes, experiments become possible that would have been unthinkable when performing the analysis manually. One example is a titration experiment where the biochemical conditions are varied. Here, we measured the influence of glycerol concentration on the dynamics of the 3-colored L-shaped DNA origami introduced in Fig. 4a with 7.5 nt overhangs. Interestingly, we observed a decrease in residence time in both states with increasing glycerol concentrations (Fig. 6a, b). Dwell times start at 1.75 s (state 1) and 2.69 s (state 2) for pure imaging buffer and decrease down to 0.62 s and 0.85 s in buffer containing 30% (v/v) glycerol. The multi-fold increase in binding kinetics can be explained by a reported destabilization of base-pairing due to changes in the ssDNA hydration shell³⁷ and concomitantly disturbed hydrogen bonding due to the osmolyte-DNA interaction. The melting enthalpy and melting temperature decreases linearly with glycerol concentration at about 0.2 °C per % (v/v)^{38,39} in line with our observations (Fig. 5b). With Deep-LASI at hand,

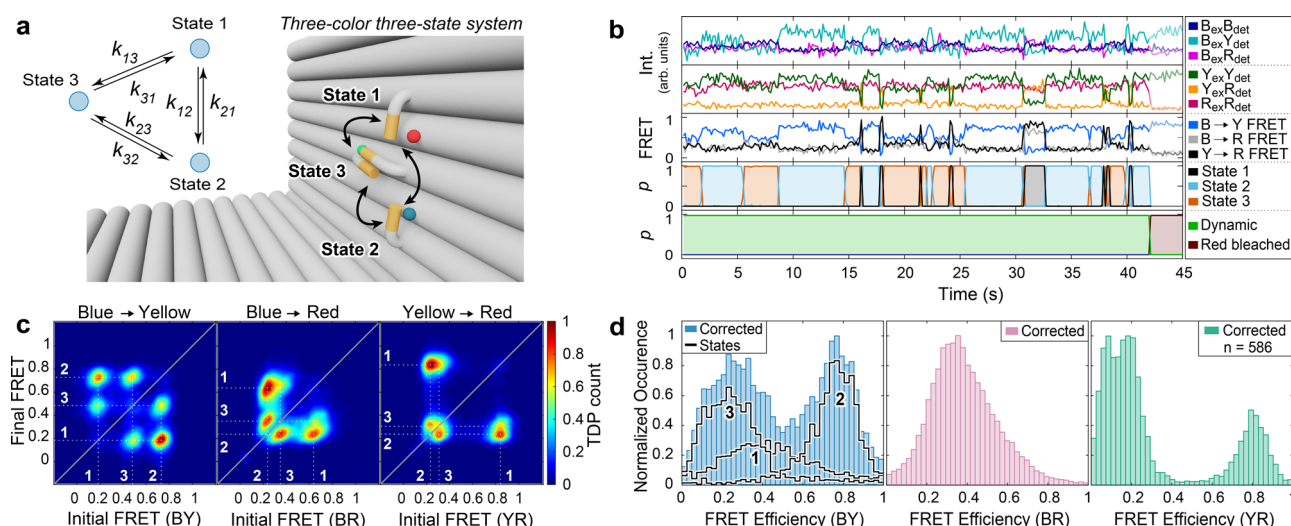


Fig. 5 | Single-molecule analysis of three-state, three-color FRET data. **a** Zoom-in of the L-shaped DNA origami structure with an additional binding site for the tether (state 3 at 9 o'clock) and corresponding kinetic scheme. **b** A representative single-molecule three-color FRET trace and apparent FRET for the 3-state system. The upper panel shows the intensity in the blue, yellow and red channels after blue excitation. The second panel shows the intensity in the yellow and red channels after yellow excitation and the red intensity after red excitation. The middle panel shows the corresponding FRET efficiencies for the three dye pairs. The fourth and fifth panels show the output of the Deep-LASI analysis for state transition and trace classification, respectively. B: blue; Y: yellow; R: red; ex: excitation; det: detection.

c Transition density plots of the apparent FRET efficiency data are shown for each FRET pair revealing an interconversion between 3 binding sites. Total number of transitions: $n = 17,136$. **d** Frame-wise weighted, state-wise corrected smFRET efficiency histograms. Corrected, distance-related FRET values are best resolved for the YR pair, showing three populations at 0.81, 0.19 and 0.09. The BY FRET shows one population at 0.8, corresponding to state 2, and a broad population at 0.3 for states 1 and 3. Individually-corrected states are indicated with the highlighted lines, showing the actual BY FRET efficiencies of state 1 (0.4) and state 3 (0.21). The apparent FRET states for the BR channel merge into one broad, static state with a value of 0.35. Source data are provided as a Source Data file.

local screening and time-consuming parameterization of imaging conditions become feasible.

Next, we applied Deep-LASI to smFRET measurements on proteins. We previously used dual-color FRET studies to probe the nucleotide-dependent conformational states⁴⁰ of Ssc1, a mitochondrial heat-shock protein Hsp70 in yeast. By fluorescently labeling both the nucleotide-binding domain and the substrate-binding domain, we investigated the influence of ADP on the inter-domain separation via smFRET. As the proteins were immobilized by incorporation in vesicles, a photostabilization buffer could not be used for the experiments. Hence, the signal-to-noise ratio in these experiments is lower than those exhibited by the photostabilized DNA origami structures. A comparison of traces selected manually and/or by the Deep-LASI analysis is discussed in Supplementary Note 6.5. For the different ADP concentrations, Deep-LASI identifies the underlying FRET states in line with the manually evaluated data⁴⁰ (Fig. 6c). It correctly identifies transitions between two distinct states, a loosely docked conformation with high FRET efficiency ($E = 81\%$) and a separated undocked state ($E = 50\%$), as shown in Fig. 6d). The automated data analysis of Deep-LASI confirmed the ADP-dependent kinetics of the domain sensor in good agreement with previous, manually evaluated results⁴⁰ (Fig. 6e, f). This demonstrates the proficiency of Deep-LASI for unsupervised data evaluation of smFRET data on proteins.

Finally, we tested the automated analysis of Deep-LASI applied to a different microscopy approach for smFRET, i.e. confocal single-molecule data on immobilized molecules that can be collected with microsecond time resolution. We chose the same double-labeled DNA origami structure introduced in Fig. 2a but with different combinations of docking strands. For these constructs, the measured FRET efficiencies will be the same but with different dynamics. By changing the hybridization length or by adding mismatched bases in the docking strand, the interaction time of each binding site can be tuned individually from ~1 ms to 10 s of seconds by adjusting the stabilization energy of DNA hybridization. As expected, shorter hybridization sequences lead to fast dynamics. Figure 6g shows a representative

intensity trajectory of a DNA origami structure (containing 6 nt complementary overhangs) that was classified as dynamic and the corresponding predictions of the state classifier. Although the unquenched state (state 2) shows a high variance in intensity, the state classifier predicts transitions with high accuracy and confidence. In the case of the 5 nt complementary overhangs, the dwell times approach 1 ms (Fig. 6h), and the output probability, p , of the state classifier decreases significantly due to the low signal-to-noise ratio of the trace. Thus, the probability value is an important parameter indicating the confidence the state classifier has in the assignment of the state and can be used as a threshold. Figure 6i (colored symbols) compares the mean dwell times extracted by Deep-LASI for all the confocal datasets with the results obtained by a newly developed shinking-gate fluorescence correlation spectroscopy (sg-FCS) approach⁴¹. In sg-FCS, a pulsed light source is used such that the fluorescence lifetime information can be incorporated into the analysis. By shrinking the analysis window of photons based on their detected arrival time after excitation, we vary the relative brightness of two species with different fluorescence lifetimes (e.g. the low FRET and high FRET states). This leads to a robust extraction of the kinetic rates between the two states from the auto-correlation analysis of the FCS data. For all binding site combinations with 6 nt to 7 nt complementary overhangs, dwell times obtained by both methods are in excellent agreement. The largest deviation was found for the 6 nt binding sites in the asymmetric 6 nt/7 nt sample (Fig. 6i, purple) (a factor of 2) where there is large heterogeneity and limited statistics⁴¹. The dwell times for the sample with 5 nt complementary overhangs follow the exponential trend observed for longer binding sites but the binning of 0.6 ms, together with the resulting low signal-to-noise ratio, reach the current limit of Deep-LASI's state classifier. For completeness, we have included the results from Fig. 4e, f in Fig. 6i (gray triangles). There is a shift in dwell times between TIRF and confocal data due to the different temperatures of the two laboratories (~19 °C confocal, ~22 °C TIRF, see Supplementary Note 6.4). Lower temperatures lead to a higher standard free energy and concomitantly longer binding time^{42,43}. In the case of the 6.5 nt

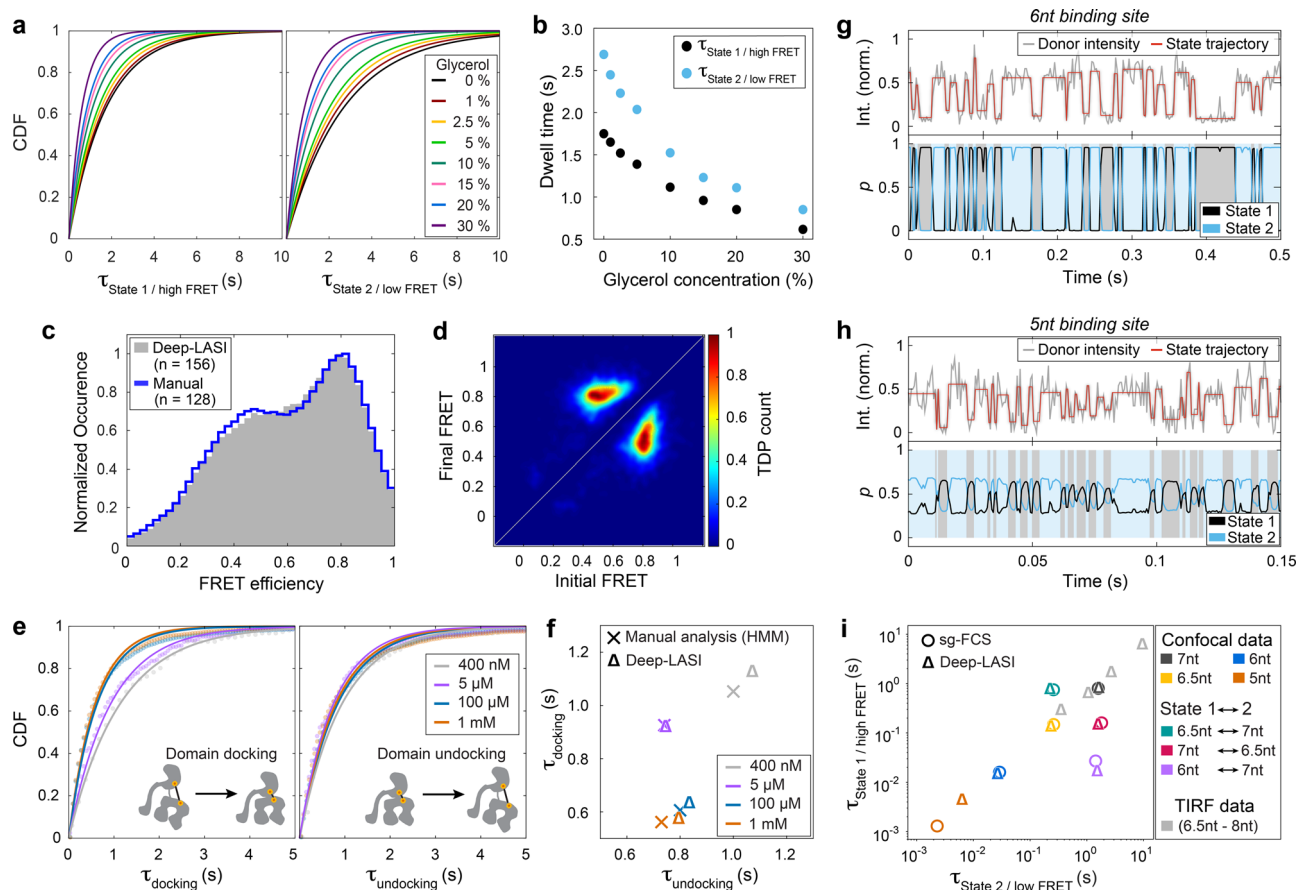


Fig. 6 | Use of Deep-LASI on titration experiments, protein data, and confocal data. **a, b** 3cFRET: Tuning the dissociation thermodynamics between protruding ssDNA strands by osmolytes. **a** CDFs of the dwell times, assessed through mono-exponential fits, for state 1 (left) and state 2 (right) of the L-shaped DNA origami structure from Fig. 4a decrease with increasing glycerol concentration. **b** Dependence of dwell times for both states versus glycerol concentration. **c–f** 2cFRET: Probing domain-domain interactions in Ssc1, a mitochondrial Hsp70. **c** Frame-wise smFRET distributions of Hsp70 molecules in the presence of 1 mM ADP classified as dynamic by Deep-LASI (gray) and evaluated manually (blue) from a total of 3534 traces. **d** The TDP generated by Deep-LASI aligns with the data plotted in (c), illustrating the interconversion between the undocked (~ 0.5) and docked (~ 0.8) conformations. Total number of transitions, $n = 3914$. **e** CDFs of the dwell times and mono-exponential fits to the dwell time distributions derived by Deep-LASI for domain docking (left panel) and domain undocking (right panel) depending on the ADP concentration. **f** Comparison between average dwell times extracted by Deep-

LASI (triangles) and by manual evaluation (crosses) using HMM. Deep-LASI matches the published trend with similar dwell times⁴⁰. **g–i** 1c-FRET: Deep-LASI analysis of ssDNA binding kinetics observed via confocal microscopy. **g** Confocal trace (with 2 ms binning) of the DNA origami structure from Fig. 2a with 6 nt binding sites and corresponding states predicted with high confidence. **h** Confocal trace (with 0.6 ms binning) of a DNA origami structure with 5 nt binding sites and predicted states. Due to the low SNR of the data, the confidence output of Deep-LASI reaches its lower limit. **i** Mean dwell times obtained from confocal data for various binding site lengths analyzed by sg-FCS⁴¹ (circles) and Deep-LASI (triangles). The results align well, except for dwell times extracted from the 5 nt sample, which was predicted with a low confidence distribution due to low SNR and a limited amount of information in the one-channel input. Dwell times obtained from TIRF data are displayed in light gray for comparison. nt: nucleotides. Source data are provided as a Source Data file.

binding sites sample (Fig. 6i, yellow), lower dwell times are consistently observed for the TIRF data. This discrepancy is due to the difference in temporal resolution of the two measurements (2 ms for confocal vs 30 ms for TIRF). The lower temporal resolution of the TIRF measurements led to a higher probability of fast transitions being averaged out and an underestimation of the actual transition time. This is a limitation of the real experimental data and is not attributable to Deep-LASI. On the contrary, Deep-LASI can back-trace shortcomings of either technique, identify rare events and monitor conformational changes over several time scales in an unsupervised manner.

Discussion

Deep-LASI is a deep-learning algorithm for the rapid and fully automated analysis of one-, two- and three-color single-molecule assays. Employing state-of-the-art neural network architectures optimized for time series data, we extended the classification of two-color FRET trajectories to include single- and three-color data, analyzed the

photobleaching information and incorporated a full state transition classification.

The utilization of deep-learning approaches for single-molecule analysis comes with both advantages and potential pitfalls. One major advantage is the ability of neural networks to capture intricate temporal dependencies and complex patterns in time-series data. This allows for improved classification accuracy and the identification of subtle transitions or states that may be challenging to discern using traditional analysis methods. Additionally, deep-learning models can learn from large amounts of data, reducing the dependence on prior assumptions that may introduce user bias.

It is essential to consider potential pitfalls when using deep neural networks for single molecule analysis. One challenge is the interpretability of the neural networks' decisions. While mathematical models and simpler thresholding techniques introduce user bias, they provide explicit confidence levels or probabilities derived from the user's modeling choices. In contrast, the output generated by neural

networks can be viewed as an artificial confidence level, minimizing user bias and increasing the consistency of the analysis. However, potential unknown biases inherent in the network itself may be introduced. Although neural networks can demonstrate high accuracy on validation datasets, understanding the underlying features and mechanisms influencing their predictions can be more challenging compared to conventional methods with explicit assumptions.

Ideal single-molecule traces are straightforward to model, and Deep-LASI is trained to be stringent when encountering non-ideal traces at a high confidence threshold. Hence, the default output of Deep-LASI when analyzing data with poor SNR is to discard the majority of the traces. By adjusting this threshold, users can instantaneously modify the accepted traces and monitor changes in the final results. This approach ensures a very low false positive classification rate when using a high confidence threshold and allows for gradual threshold reduction to increase statistical coverage. However, the outcome should be continuously monitored by the user. This iterative process effectively balances stringent classification and the need for increased statistical robustness in the analysis of new datasets.

Furthermore, Deep-LASI offers an advantage in terms of interpretability as it is trained solely on editable and extendable simulations. This characteristic provides users with greater control and knowledge over potential biases and enables them to tailor Deep-LASI to a wide range of experimental conditions. It is important to note that neural networks are data-driven models and heavily rely on the quality and representativeness of the training data. Therefore, careful consideration must be given to curating the training dataset to avoid biases and ensure the generalizability of the model to diverse experimental conditions. Regular validation and testing using independent datasets are crucial steps to assess the robustness and reliability of the model's performance. In addition, when measuring an unknown experimental system for the first time, it is helpful to visually inspect the traces that are being discarded to verify that the classification is still reasonable. By following these practices, researchers can enhance the trustworthiness and applicability of Deep-LASI in real-world scenarios.

In conclusion, Deep-LASI addresses the need for rapid, high-throughput screening of fluorescence intensity trajectories. This opens new possibilities for single-molecule assays and enables a timely analysis of complex experimental approaches thanks to the efficient and retrainable neural network architecture of Deep-LASI. It has a high potential for applications in a myriad of fields including biotheranostics, sensing, DNA barcoding, proteomics and single-molecule protein sequencing. We envision that deep-learning approaches, along with single-molecule sensitivity, will dramatically assist and accelerate analytics and be indispensable in the future.

Methods

Chemicals

Chemicals were purchased from Sigma-Aldrich and used without further purification, if not stated otherwise. Chemicals include acetic acid, agarose, ammonium persulfate, (3-aminopropyl-) triethoxysilane (APTES), biotin-poly(ethylene glycol)-silane (*biotin-PEG*, MW3000, PG2-BNSL-3k, Nanocs, NY; USA), bovine serum albumin (BSA; New England Biolabs, Ipswich, MA, USA), Blue Juice gel loading buffer (ThermoFisher Scientific), ethylene-diamine-tetraacetic acid sodium salt dehydrate ($\text{EDTA-Na}_2 \times 2\text{H}_2\text{O}$), glycerol, magnesium chloride ($\text{MgCl}_2 \times 6\text{H}_2\text{O}$), 2-[methoxy(polyethyleneoxy)propyl]trimethoxysilane (*mPEG*, #AB111226, abcr; Germany), phosphate-buffered saline (PBS), protocatechuate 3,4-dioxygenase from *Pseudomonas* sp. (PCD), protocatechuic acid (PCA), streptavidin, sodium chloride, Tris base, Tris HCl, and 6-hydroxy-2,5,7,8-tetramethylchroman-2-carboxylic acid (Trolox) and beta-mercaptoethanol (β ME).

All unmodified staple strands (Supplementary Note 7, Supplementary Table 7.2) used for DNA origami structure folding are

commercially available and were purchased from Integrated DNA Technologies®. Staple strands with modifications (Supplementary Tables 7.3 and 7.4) were obtained from Biomers (Supplementary Table 7.3: Biotin; Supplementary Table 7.4: Atto488) and Eurofines Genomics (Supplementary Table 7.4: binding sites, Cy3b and Atto647N).

DNA origami structures: assembly, purification and characterization

Preparation of the L-shaped DNA origami structures follows the procedures described previously by Tinnefeld et al.^{5,31}. In brief, the L-shaped DNA origami structures were folded with a 10-fold excess of 252 different, unmodified and labeled oligonucleotides to the complementary 8064 bp scaffold strand in folding buffer, which contained 40 mM Tris base, 20 mM acetic acid, 20 mM $\text{MgCl}_2 \times 6\text{H}_2\text{O}$, and 1 mM $\text{EDTA-Na}_2 \times 2\text{H}_2\text{O}$. A complete list with sequences of oligonucleotides used for producing the DNA origami structure is given in Supplementary Note 7. For folding, a nonlinear thermal annealing ramp over 16 hours was used⁴⁴.

After folding, the DNA origami solution was cleaned via gel electrophoresis in 50 mL 1.5% agarose-gel containing 1× gel buffer (40 mM Tris base, 20 mM acetic acid, 12 mM $\text{MgCl}_2 \times 6\text{H}_2\text{O}$, and 1 mM $\text{EDTA-Na}_2 \times 2\text{H}_2\text{O}$). The gel pockets were filled with a solution of 1× Blue Juice gel loading buffer and the DNA origami solution. The ice-cooled gel was run for 2 h at 60 V. When samples were to be recovered from the gel, the staining step was omitted and the Cy3b fluorescence was used instead to identify the correct DNA origami structures. Gel extraction was performed via cutting with a scalpel and squeezing the gel with a Parafilm® (Bernis®) wrapped glass slide. The concentration was determined by absorption spectroscopy on a NanoDrop 2000 device (ThermoFisher Scientific). Purified DNA origami structures were kept in storage buffer, i.e. in 1× TAE buffer (40 mM Tris base, 20 mM acetic acid and 1 mM $\text{EDTA-Na}_2 \times 2\text{H}_2\text{O}$) with 12.5 mM $\text{MgCl}_2 \times 6\text{H}_2\text{O}$ (pH = 8.4).

The correct folding of the DNA origami structures was confirmed using atomic force microscopy and transition electron microscopy (see Supplementary Figure 7.2).

Sample preparation for multicolor prism-type TIRF experiments

Labeled DNA origami molecules were immobilized in flow channels formed between a coverslip and a surface-functionalized quartz prism. The surfaces were sandwiched on top of each other and sealed by a molten, pre-cut Nesco film (Nesco) channel. The employed prism surface was functionalized before with a biotin-PEG/mPEG coating to achieve surface passivation and prevent unspecific binding. Before the TIRF experiments, the prisms were first flushed with PBS and then incubated with a streptavidin solution (0.2 mg/mL) for 15 min. Afterwards, the sample holder was washed 3× with PBS to remove free streptavidin and then with storage buffer (1× TAE, 12.5 mM MgCl_2 , pH = 8.4). Next, the DNA origami sample was diluted to 40 pM in storage buffer, added to the flow chamber and immobilized to the prism surface via the biotin-streptavidin linkage. After 5 min, untethered DNA origami structures were removed by rinsing the chamber 3× with storage buffer. Next, the attached fluorophores on the DNA origami structure were photostabilized by a combination of ROXS and an oxygen scavenging system based on PCA/PCD⁴⁵. The photostabilization buffer was mixed as follows: 1 μL of 100 mM Trolox/Ethanol solution was added to 97 μL storage buffer. The sample was then aged using a UV Lamp (M&S Laborgeräte GmbH, UVAC-6U, 2 × 6 W; 254 and 366 nm) until an equal ratio of Trolox and Trolox-quinone was formed (typically 6 minutes)⁴⁶. Immediately before starting the TIRF experiments, 1 μL of 100 mM PCA in methanol and 1 μL of 100 mM PCD solution (50% glycerol, 50 mM KCl, 100 mM Tris HCl, 1 mM $\text{EDTA-Na}_2 \times 2\text{H}_2\text{O}$, pH8) was added to the total volume. In the case of samples containing Atto488, 1 μL of 14.3 M β ME was added to the

photostabilization buffer. The sample chamber was flushed with photostabilization buffer and sealed to allow for enzymatic oxygen depletion. A minimum of 5 minutes waiting time preceded the experimental recordings. The photostabilization buffer was refreshed every 45 minutes until the end of the experiment.

All two- and three-color FRET experiments were carried out using msALEX³³, i.e. two- or three excitation lasers were alternated frame-wise. The lasers of different excitation wavelengths were synchronized using an acousto-optical filter (OPTO-ELECTRONIC, France) with the camera frame rate using an FPGA that synchronizes the excitation and simultaneous detection on the EMCCD cameras at 32 ms, 50 ms or 150 ms exposure times (depending on the sample) for 2000 (two-color) and 2400 (three-color) frames. The laser powers were set to 28 mW (0.022 mm², 491 nm), 16 mW (0.040 mm², 561 nm) and 10 mW (0.022 mm², 640 nm) for B-Y-R excitation.

Multi-color TIRF setup

Single-pair FRET experiments on surface-immobilized DNA origami structures were carried out on a home-built TIRF microscope with prism-type excitation as previously published⁴⁷. Three laser sources (Cobolt, Solna; Sweden) at 491 nm, 561 nm and 640 nm are available, and used for triple-color TIRF experiments with an alternation rate of 27 Hz (including a 2.2 ms frame transfer rate) between the B-Y-R laser excitation. The resulting emission was collected by a 60×water immersion objective (60×/1.27 WI Plan Apo IR, Nikon), cleaned up with a notch filter (Stoptone® Notch 488/647, AHF), and the red emission was separated from the blue/yellow emission by a dichroic mirror (630DCXR AHF; Germany) followed by separation of the blue and yellow emission (560DCXR AHF). The emission was spectrally filtered (AHF Analysetechnik, Tübingen, Germany) for the blue (ET525/50), yellow (HQ595/50) and red (ET685/40) collection channels and afterwards detected on three EMCCD cameras (Andor iXon (1×)/iXon Ultra (2×), Andor Technologies, Belfast; UK) via the supplier's software Andor Solis (Version 4.29.30005.0; Oxford Instruments). Synchronization and alternation of the exciting laser sources, as well as the frame-wise data acquisition on three separate cameras, were achieved using a LabView-written program that controls a field programmable gate array (FPGA; NI cRIO-9073). While the program starts the measurement, the FPGA synchronizes the execution of the hardware via TTL pulses, i.e. it controls switching on/off the excitation sources by direct modulation of the AOTF (491, 561, 640 nm), while simultaneously starting the data acquisition by the three cameras. The videos were analyzed afterward by a custom-written MATLAB program (Version 9.13.0.2166757; Mathworks, Massachusetts, USA).

Single-molecule data analysis, data evaluation and representation

Time traces of individual, fluorescently labeled DNA origami structures were extracted from measurements using one, two or three cameras for one-, two- and three-color experiments, respectively, using Deep-LASI. Deep-LASI is written in MATLAB (Version 9.13.0.2166757; Mathworks, Massachusetts, USA) and uses neural networks trained with the Python library TensorFlow (Version 2.8.0). All raw data were recorded by EMCCD cameras (iXon 897, i.e. frames with 512 × 512 pixels containing fluorescence intensity information) and stored as TIFF stacks using the supplier's software Andor Solis. The resulting traces are then analyzed either using the pre-trained neural networks (Supplementary Notes 2, 3) or manually (Supplementary Note 5). The regions of single-molecule traces that were classified as dynamic with photoactive fluorophores were selected for downstream analysis. In the automated analysis procedure, the state transitions and state dwell times were predicted by a neural network model. All manually selected traces were analyzed using Hidden Markov Models, locally fit to each intensity trace (1-color data) or FRET trace (2-color data) assuming two states with Gaussian emission distribution functions and using the

Baum-Welch algorithm. The Gaussian emission distribution functions serve as the prior for the HMM, which are iteratively updated during the analysis. The convergence threshold was set to 10⁻⁹ and the maximum number of iterations was set to 10⁸. All predicted transitions were extracted from a transition density plot and the corresponding dwell times were fit to an exponential function. All correction factors for calculating the corrected FRET efficiency were determined using the manual or automated classification of photobleaching steps. All employed methods, automated and manual, were performed using the Deep-LASI user interface. Final panels were all presented using MATLAB 2022b (Version 9.13.0.2166757; Mathworks, Massachusetts, USA), exported as vector graphics, and assembled into figures using Adobe Illustrator CS2022 (Adobe Inc.; USA). 3D representations of the DNA origami structures were rendered in Blender (Version 2.93.6), and further assembled and labeled in Illustrator. The AFM images were plotted using JPK Desktop Software (Version 6.1.200 A).

Statistics & reproducibility

For training the neural network, we used a simulated dataset with ~200,000 traces as it is sufficient to cover an extensive range of realistic experimental parameters and thereby avoid any bias in the analysis. This includes FRET efficiencies between 0.01 and 0.99, dwell times of 1 to 100 frames and SNR of ~0.3 to 50. Experimentally, we typically measured 100 movies for each condition, as this usually generates several thousand acceptable traces. The full datasets were analyzed. The program, as part of its function, determines which intensity traces are suitable for further analysis. The computer selection was tested against simulated traces as well as compared with human analyses.

The Deep-LASI software was trained on three independently generated datasets. Deep-LASI was also compared with two users who manually analyzed the same datasets. The number of states, FRET efficiency histograms and kinetic rates extracted from the different analyses are consistent and, when available, are within the confidence intervals from the fits. Experiments were not randomized. The researchers were not blinded as knowledge regarding the sample did not influence the manual selection or analysis of the data. For the neural network, the advantage is that it operates only based on the data that it has been trained with. Hence, blinding is not applicable.

For analysis of the dwell-time distributions, a mono-exponential function was fit to the cumulative distribution function in MATLAB 2022b. The optimal fit values, along with the 95% confidence intervals, are given in the text and figures.

Reporting summary

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

Data availability

The data for all figures and all supplementary figures have been deposited in the Zenodo database⁴⁸ [<https://zenodo.org/record/7561162>], with the exception of previously published data (HSP70 SSC1 and DNA origami confocal data in Fig. 6). Source data are provided with this paper.

Code availability

The program is available on GitLab [<https://gitlab.com/simon71/deeplasi>]. Extensive documentation for the Deep-LASI software package can be found at <https://deep-lasi-tutorial.readthedocs.io/en/latest/index.html>.

References

1. Lerner, E. et al. FRET-based dynamic structural biology: Challenges, perspectives and an appeal for open-science practices. *eLife* **10**, e60416 (2021).

2. Hellenkamp, B. et al. Precision and accuracy of single-molecule FRET measurements—a multi-laboratory benchmark study. *Nat. Methods* **15**, 669–676 (2018).
3. Agam, G. et al. Reliability and accuracy of single-molecule FRET studies for characterization of structural dynamics and distances in proteins. *Nat. Methods* **20**, 523–535 (2023).
4. Chizhik, A. I., Rother, J., Gregor, I., Janshoff, A. & Enderlein, J. Metal-induced energy transfer for live cell nanoscopy. *Nat. Photon.* **8**, 124–127 (2014).
5. Kamińska, I. et al. Graphene energy transfer for single-molecule biophysics, biosensing, and super-resolution microscopy. *Adv. Mater.* **33**, 2101099 (2021).
6. Ploetz, E. et al. Förster resonance energy transfer and protein-induced fluorescence enhancement as synergetic multi-scale molecular rulers. *Sci. Rep.* **6**, 33257 (2016).
7. Lerner, E., Ploetz, E., Hohlbein, J., Cordes, T. & Weiss, S. A Quantitative Theoretical Framework For Protein-Induced Fluorescence Enhancement—Förster-Type Resonance Energy Transfer (PIFE-FRET). *J. Phys. Chem. B* **120**, 6401–6410 (2016).
8. Ploetz, E. et al. A new twist on PIFE: photoisomerisation-related fluorescence enhancement. *Methods and Applications in Fluorescence* (in press), <https://doi.org/10.1088/2050-6120/acfb58> (2023).
9. Hohng, S., Joo, C. & Ha, T. Single-molecule three-color FRET. *Biophys. J.* **87**, 1328–1337 (2004).
10. Barth, A., Voith von Voithenberg, L. & Lamb, D. C. Quantitative single-molecule three-color Förster resonance energy transfer by photon distribution analysis. *J. Phys. Chem. B* **123**, 6901–6916 (2019).
11. Voith von Voithenberg, L. et al. Comparative analysis of the coordinated motion of Hsp70s from different organelles observed by single-molecule three-color FRET. *Proc. Natl Acad. Sci. USA* **118**, e2025578118 (2021).
12. Holden, S. J. et al. Defining the Limits of Single-Molecule FRET Resolution in TIRF Microscopy. *Biophys. J.* **99**, 3102–3111 (2010).
13. Schmid, S., Götz, M. & Hugel, T. Single-Molecule Analysis beyond Dwell Times: Demonstration and Assessment in and out of Equilibrium. *Biophys. J.* **111**, 1375–1384 (2016).
14. Van de Meent, J.-W., Bronson, J. E., Wiggins, C. H. & Gonzales, J. R. Empirical Bayes Methods Enable Advanced Population-Level Analyses of Single-Molecule FRET Experiments. *Biophys. J.* **106**, 1327–1337 (2014).
15. McKinney, S. A., Joo, C. & Ha, T. Analysis of Single-Molecule FRET Trajectories Using Hidden Markov Modeling. *Biophys. J.* **91**, 1941–1951 (2006).
16. Kaur, A., Ellison, M. & Dhakal, S. MASH-FRET: A Simplified Approach for Single-Molecule Multiplexing Using FRET. *Anal. Chem.* **93**, 8856–8863 (2021).
17. Preus, S., Noer, S. L., Hildebrandt, L. L., Gudnason, D. & Birkedal, V. iSMS: single-molecule FRET microscopy software. *Nat. Methods* **12**, 593–594 (2015).
18. Greenfeld, M., Pavlichin, D. S., Mabuchi, H. & Herschlag, D. Single Molecule Analysis Research Tool (SMART): An Integrated Approach for Analyzing Single Molecule Data. *PLoS ONE* **7**, e30024 (2012).
19. Juette, M. F. et al. Single-molecule imaging of non-equilibrium molecular ensembles on the millisecond timescale. *Nat. Methods* **13**, 341–344 (2016).
20. Ge, L., Liu, F. & Luo, J. Highly-efficient quantitative fluorescence resonance energy transfer measurements based on deep learning. *J. Innov. Opt. Health Sci.* **13**, 2050021 (2020).
21. Li, J., Zhang, L., Johnson-Buck, A. & Walter, N. G. Automatic classification and segmentation of single-molecule fluorescence time traces with deep learning. *Nat. Commun.* **11**, 5833 (2020).
22. Thomsen, J. et al. DeepFRET, a software for rapid and automated single-molecule FRET data classification using deep learning. *eLife* **9**, e60404 (2020).
23. Zarrabi, N., Schluesche, P., Meisterernst, M., Börsch, M. & Lamb, D. C. Analyzing the Dynamics of Single TBP-DNA-NC2 Complexes Using Hidden Markov Models. *Biophys. J.* **115**, 2310–2326 (2018).
24. Salem, C.-B., Ploetz, E. & Lamb, D. C. in *Spectroscopy and Dynamics of Single Molecules* (ed Carey K. Johnson) 71–115 (Elsevier, 2019).
25. Liu, D., Honoré, A., Chatterjee, S. & Rasmussen, L. K. Powering Hidden Markov Model by Neural Network based Generative Models. *24th European Conference on Artificial Intelligence (ECAI 2020)*, Santiago de Compostela, Spain, <https://doi.org/10.48550/arxiv.1910.05744> (2020).
26. Buys, J., Bisk, Y. & Choi, Y. Bridging HMMs and RNNs through Architectural Transformations. *32nd Conference on Neural Information Processing Systems (NIPS 2018)*, Montréal, Canada (2018).
27. Liu, L., Lin, Y.-C. & Reid, J. Improving the Performance of the LSTM and HMM Model via Hybridization. *ArXiv*, <https://doi.org/10.48550/arxiv.1907.04670> (2021).
28. Rallabandi, P. K. & Patidar, K. C. A Hybrid System of Hidden Markov Models and Recurrent Neural Networks for Learning Deterministic Finite State Automata. *Int. J. Comp. Sci. Inf. Eng.*, <https://doi.org/10.5281/zenodo.1109543> (2015).
29. Krakovna, V. & Doshi-Velez, F. Increasing the Interpretability of Recurrent Neural Networks Using Hidden Markov Models. *2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, New York, NY, <https://doi.org/10.48550/arxiv.1606.05320> (2016).
30. Xu, J. et al. Automated stoichiometry analysis of single-molecule fluorescence imaging traces via deep learning. *J. Am. Chem. Soc.* **141**, 6976–6985 (2019).
31. Krause, S. et al. Graphene-on-glass preparation and cleaning methods characterized by single-molecule DNA origami fluorescent probes and Raman spectroscopy. *ACS Nano* **15**, 6430–6438 (2021).
32. Gotz, M. et al. A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories. *Nat. Commun.* **13**, 5402 (2022).
33. Kapanidis, A. N. et al. Alternating-Laser Excitation of Single Molecules. *Acc. Chem. Res.* **38**, 523–533 (2005).
34. Schreiber, J. Pomegranate: fast and flexible probabilistic modeling in python. *J. Mach. Learn. Res.* **18**, 1–6 (2018).
35. Kudryavtsev, V. et al. Combining MFD and PIE for accurate single-pair Förster resonance energy transfer measurements. *Chem. Phys. Chem.* **13**, 1060–1078 (2012).
36. Lee, N. K. et al. Accurate FRET measurements within single diffusing biomolecules using alternating-laser excitation. *Biophys. J.* **88**, 2939–2953 (2005).
37. Diaz, A., Jothiraman, H. B. & Ramakrishnan, V. Effect of glycerol on free DNA: A molecular dynamics simulation study. *J. Mol. Graph. Model.* **114**, 108169 (2022).
38. Wang, C., Altieri, F., Ferraro, A., Giartosio, A. & Turano, C. The effect of polyols on the stability of duplex DNA. *Physiol. Chem. Phys. Med. NMR* **25**, 273–280 (1993).
39. Spink, C. H., Garbett, N. & Chaires, J. B. Enthalpies of DNA melting in the presence of osmolytes. *Biophys. Chem.* **126**, 176–185 (2007).
40. Sikor, M., Mapa, K., von Voithenberg, L. V., Mokranjac, D. & Lamb, D. C. Real-time observation of the conformational dynamics of mitochondrial Hsp70 by spFRET. *EMBO J.* **32**, 1639–1649 (2013).
41. Schroder, T. et al. Shrinking gate fluorescence correlation spectroscopy yields equilibrium constants and separates photophysics from structural dynamics. *Proc. Natl. Acad. Sci. USA* **120**, e2211896120 (2023).
42. Fornace, M. E., Porubsky, N. J. & Pierce, N. A. A Unified Dynamic Programming Framework for the Analysis of Interacting Nucleic Acid Strands: Enhanced Models, Scalability, and Speed. *ACS Synth. Biol.* **9**, 2665–2678 (2020).

43. Dirks, R. M., Bois, J. S., Schaeffer, J. M., Winfree, E. & Pierce, N. A. Thermodynamic Analysis of Interacting Nucleic Acid Strands. *SIAM Rev.* **49**, 65–88 (2007).
44. Nickels, P. C. et al. Molecular force spectroscopy with a DNA origami-based nanoscopic force clamp. *Science* **354**, 305–307 (2016).
45. Aitken, C. E., Marshall, R. A. & Puglisi, J. D. An oxygen scavenging system for improvement of dye stability in single-molecule fluorescence experiments. *Biophys. J.* **94**, 1826–1835 (2008).
46. Cordes, T., Vogelsang, J. & Tinnefeld, P. On the mechanism of Trolox as antiblinking and antibleaching reagent. *J. Am. Chem. Soc.* **131**, 5018–5019 (2009).
47. Kopperger, E. et al. A self-assembled nanoscale robotic arm controlled by electric fields. *Science* **359**, 296 (2018).
48. Wanninger, S. et al. Deep-Learning assisted, Single-molecule Imaging analysis of multi-color DNA Origami structures. *Zenodo*, URL: <https://zenodo.org/record/7561162> (2023).

Acknowledgements

We thank Julian Heeg for helping with data collection. We thankfully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 201269156 – SFB 1032 Project BO3 (to D.C.L.), Project A13 (to P.T.), individual grants to PL696/4-1 (to E.P.), TI 329/9-2, project number 267681426, TI 329/14-1, and TI 329/15-1 (to P.T.) and Germany's Excellence Strategy – EXC 089/1 – 390776260. D.C.L. and P.T. gratefully acknowledge funding from the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder through the ONE MUNICH Project Munich Multiscale Biofabrication. P.T. acknowledges the support of BMBF (SIBOF, O3VP03891). D.C.L., P.T., and E.P. gratefully acknowledge the financial support of the Ludwig-Maximilians-Universität München via the Department of Chemistry, the Center for NanoScience (CeNS) and the LMUinnovativ program Bioluminescence Network (BIN).

Author contributions

S.W. developed and implemented the deep-learning algorithm Deep-LASI and performed the Deep-learning assisted analyses, J.B. prepared the DNA origami samples under the supervision of P.T., P.A. collected the single-molecule TIRF data, P.A. and E.P. performed the manual analysis of the smFRET data, C.B.S. wrote the manual analysis software in which Deep-LASI was incorporated, S.W. and E.P. wrote the first draft of the manuscript, S.W. and E.P. designed the figures, all authors

contributed to revising the manuscript. E.P. and D.C.L. supervised the project.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

All authors (S.W., P.A., J.B., C.B.S., P.T., E.P., and D.C.L.) declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-023-42272-9>.

Correspondence and requests for materials should be addressed to Evelyn Ploetz or Don C. Lamb.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023

Supplementary Information

Deep-LASI: Deep-Learning Assisted, Single-molecule Imaging analysis of multi-color DNA Origami structures

Simon Wanninger¹, Pooyeh Asadiatouei¹, Johann Bohlen¹, Clemens-Bäsem Salem¹,
Philip Tinnefeld¹, Evelyn Ploetz^{1,*} and Don C. Lamb^{1,*}

¹Department of Chemistry and Center for NanoScience (CeNS)
Ludwig-Maximilians-Universität München
Butenandtstr. 5-13 81377 Munich,
Germany

* Corresponding authors: evelyn.ploetz@lmu.de; d.lamb@lmu.de

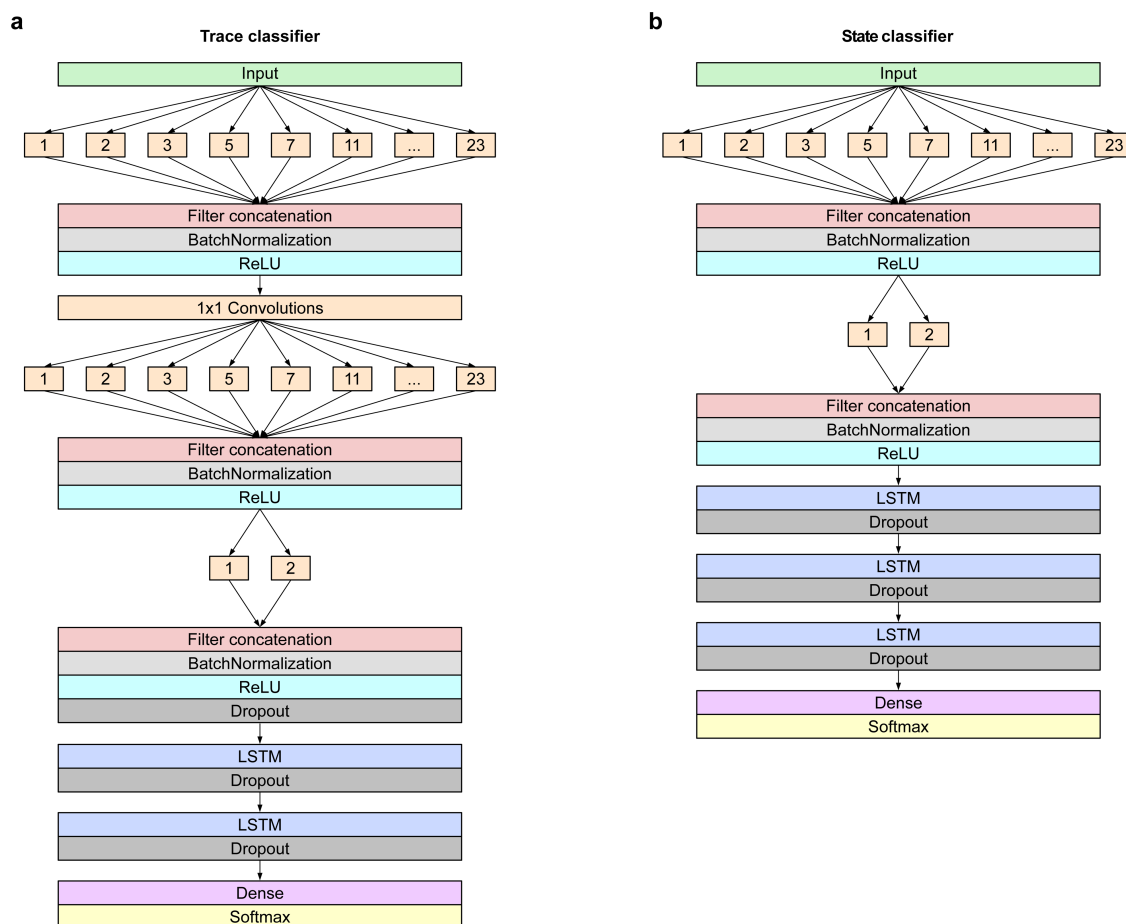
TABLE OF CONTENTS

Supplementary Note 1: Neural network	3
1.1. Architecture	3
1.2. Trace classifier architecture	4
1.3. State transition classifier and number of states classifier architecture.....	5
Supplementary Note 2: Training	6
2.1. Training procedure	6
2.2. Training dataset preparation	6
2.3. Simulation of single molecule traces.....	8
2.4. Simulation settings for training the state classifier network	12
Supplementary Note 3: Training validation.....	14
3.1. Trace classifiers.....	14
3.2. Number-of-states classifiers	16
3.3. State-transition classifiers	16
3.4. Limitations of the state classifiers and a comparison with HMM.....	17
3.5. Comparison of the state classifiers with local and global HMM.....	19
3.6. Training and validation loss.....	20
3.7. Analysis of kinetic data from the kinsoft challenge	22
Supplementary Note 4: Deep-LASI versus manual analyses	23
4.1 Comparison of Deep-LASI and manual analysis of 2-color DNA origami traces ..	23
4.2. Influence of different training datasets and comparison to user classification	24
4.3 Deep-LASI versus manual analyses for 3-color DNA origami samples.....	25
4.4 Comparison of 3-color DNA origami traces selected manually and/or via DNN classification.	27
Supplementary Note 5: Manual Analysis of Single-molecule TIRF Data	28
5.1. Work-flow	28
5.2. Camera mapping for FRET traces.....	29
5.3. Trace extraction and background subtraction	29
5.4. Manual trace selection and analysis.....	30
5.5. Accurate FRET determination.....	30
5.6. Hidden-Markov modeling	32
5.7. Parameters for Hidden Markov Modeling	34
5.8. Evaluation of involved FRET states and interconversion rates	34
Supplementary Note 6: Details of Deep-LASI Analyses.....	35
6.1 Results for the three-color, two-state DNA origami structure with different binding site lengths	35
6.2 Kinetics of the three-color, three-state DNA origami.	36
6.3 Results for the two-color, three-state DNA origami structure.....	37
6.4 Kinetics as a function of Temperature.....	39
6.5 Analysis of previous published 2-color Hsp70 Ssc1 using Deep-LASI	39
Supplementary Note 7: DNA sequences.....	41
Supplementary Note 8: Statistics Single-Molecule Data	50
Supplementary References.....	51

SUPPLEMENTARY NOTE 1: NEURAL NETWORK

1.1. Architecture

For the Deep-LASI software package, two different neural-network architectures are used. One architecture is for trace classification and another for the number of states and state transition classification ([Supplementary Figure 1.1](#)). Both architectures are hybrids of a convolutional neural network (CNN) and a long short-term memory (LSTM) model, which were designed using TensorFlow with Keras API.¹ The CNN framework was inspired by an omni-scale 1D-CNN, which elegantly solves the problem of finding the optimal kernel sizes by making it part of the training process.² Unlike traditional CNNs that operate on 2D spatial grids, the omni-scale CNN processes time series data directly by utilizing one-dimensional convolutions. These convolutional operations consider the temporal dependencies among data points, enabling the network to effectively model the sequential nature of time series data. Single-molecule Förster Resonance Energy Transfer (smFRET) data often exhibit complex patterns on different timescales, which can be crucial for accurate classification. The omni-scale CNN architecture incorporates multiple convolutional layers with different kernel sizes at the same level of depth, enabling the extraction of features at various temporal resolutions. However, the pure CNN architecture can only produce a single classification for the entire input sequence, i.e. one category per single molecule trace. The inclusion of LSTM after feature extraction of the CNN enables the sequential classification of each trace, producing a classification output at each time step. This characteristic of LSTM allows for the fine-grained detection of valid smFRET frames, photobleaching steps and state transitions. For all of our presented tasks, the omni-scale CNN LSTM hybrid architecture outperformed pure LSTM or ResNet³ models. We did not employ the full range of prime numbers suggested for the kernel sizes as we found the accuracy did not increase above 23. Hence, the number of trainable parameters was greatly reduced. In the trace classifier model, we added a 1x1 convolution layer for dimensionality reduction to further increase efficiency without a trade-off in validation accuracy. Any down-sampling of the time dimension was avoided since the loss of this information significantly decreased the validation accuracy. This was achieved by omitting any kind of pooling or averaging layers, by zero-padding all inputs for the convolutional layers and setting the stride of all convolutional filters to 1. For kernel initialization, we used the He Normal distribution⁴ as it showed the fastest convergence rate. Each convolution layer is followed by a batch normalization layer and an activation layer using the rectified linear unit (ReLU) activation function.⁵



Supplementary Figure 1.1: The deep neural network architectures used for the trace classifier (a), and for the state transition classifier and the number of states classifier (b).

1.2. Trace classifier architecture

The trace classifier consists of four convolution layers followed by two LSTM layers and one fully connected layer as the feature extraction module (Supplementary Figure 1.1a). In the first convolution layer, the input is fed into 10 layers with 32 filters each. The kernel filter size is varied between layers with sizes given by the prime numbers from 1 to 23. All layers are stacked sequentially, i.e. they operate on the same level of depth. The second convolution layer serves as a dimensionality reduction layer with 32 filters and a kernel size of 1. The third convolution layer has the same hyperparameters as the first layer. A fourth convolution layer is added, composed of two branches with 32 filters each and kernel sizes of one and two, which allows the receptive fields of the network to cover all possible integers. The output of the CNN is fed into a LSTM layer with 128 units, followed by a second LSTM layer with 32 units and the final dense layer for classification. For the training procedure, we placed a dropout layer at a rate of 0.22 before the first LSTM layer and two dropout layers at a rate of 0.5 after the two consecutive LSTM layers to maximize the validation accuracy and reduce overfitting.

1.3. State transition classifier and number of states classifier architecture

The main difference in architecture between the transition classifier and the trace classifier are the depths and widths of the CNN and LSTM structure. The state transition classifier is composed of two convolution layers, three LSTM layers, and one final dense layer ([Supplementary Figure 1.1b](#)). The kernel sizes of the first convolution layer are prime numbers in the range of 1 to 23 with 64 filters each. The second convolution layer has kernel sizes of one and two with 32 filters each. The CNN substructure is directly followed by three LSTM layers with 128 units. Dropout layers are placed after each LSTM layer using a rate of 0.5. At the end, a fully connected layer (or dense layer) is used to reduce the output of the network into the number of given categories.

SUPPLEMENTARY NOTE 2: TRAINING

Although an optimized architecture is important and improves the efficiency of a neural network, the functionality of the network rises and falls with the dataset used for training the network. There are a number of important factors to consider when training a neural network. Typical pitfalls include using a dataset that is too small in size or contains an intrinsic bias, or overfitting the training data. A neural network is biased towards features it has seen before. Hence, the training dataset should include the various possibilities (e.g. number of FRET states, kinetic rates, signal-to-noise ratios). If the training dataset includes any bias, this will also be reflected in the output of the algorithm. One way that bias can be introduced into the training dataset is from unbalanced sampling of categories. For example, for the trace classifier models, it is important that the training dataset includes the same number of traces from each category. It is also important to know when to stop the training process. Neural networks can be overtrained, meaning that they memorize the training data but do not learn the general principles behind it. Below we discuss the details of the training procedure and how we optimized the training process.

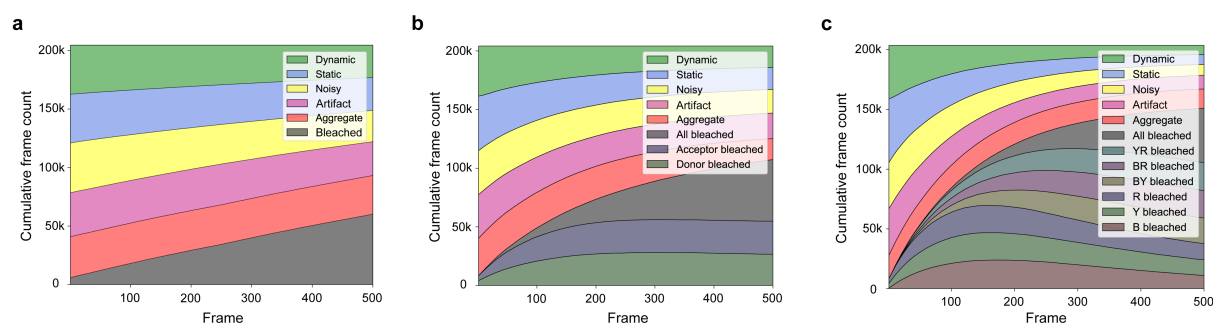
2.1. Training procedure

Both the trace classifier and transition classifier models were trained using the Adam optimizer with the default settings⁶. We used a hybrid method of increasing the batch size and lowering the learning rate during training. The entire training set of ca 200,000 traces is feed into the neural network in batch sizes of 32 traces until the network has seen all traces (referred to as an epoch). After the network has seen all traces once, some input units are randomly set to 0 using dropout layers and the dataset is fed again in batches to the neural network in the next epoch. The dropout layers reduce overfitting and allow generalization of the learned information. When the validation loss is not significantly lowered within 4 epochs, the batch size is doubled. An initial learning rate of 0.001 was decreased analogously by factors of 10 after a batch size of 512 was reached.

2.2. Training dataset preparation

To generate training datasets, we found the approach of using simulations, originally described in Thomsen et al.⁷, to be the most promising. This is especially true for three-color models capable of detecting state transitions or photobleaching events of each dye individually. A manual collection of labeled traces on a scale large enough for adequate training would be prone to biases and/or errors due to incorrect trace identification. In addition, the datasets would not be optimized for microscope setups with different characteristics. The signal and noise characteristics of smFRET data is well enough understood that simulated data can accurately reproduce the characteristics of real data. Beside the architecture itself, the main differences between our trace classifiers and the DeepFRET model⁷ is the ability to classify one-color and three-color data and to predict the photobleached frames of each fluorophore separately. We adopted the categories ‘dynamic’, ‘static’, ‘noisy’, and ‘aggregate’ while implementing

additional categories for all possible photobleaching events. The ‘artifact’ category includes false localizations, overestimated background and random perturbations of the intensity traces. For the simulated data, idealized intensity traces for each primary category (i.e. non-photobleaching category, 'dynamic', 'static', 'noisy', 'aggregate' and 'artifact') are generated and then photobleaching steps are added for the different dyes by randomly determining the survival time of the dye from a given exponential distribution. The addition of photobleaching as well as other processes in the simulation may lead to alterations in the label of the given trace. For example, a simulated 'dynamic' trace that does not undergo a transition before photobleaching or by the end of the trace would be recategorized as 'static'. To ensure that the network sees the same number of frames for the different categories including all the photobleached categories, the number of traces selected for the training set needs to be balanced. Hence, we begin by simulating $\sim 250,000$ traces of 500 frames for each primary category. After including photobleaching, the number of labeled frames for each category is determined. The category with the minimum number of frames determines a threshold at which additional traces, depending on their present classification, are added or excluded from the final training dataset. The typical number of traces included in the final training dataset is approximately 200,000. This balancing procedure ensures that no category is over- or underrepresented across all frames and minimizes biases of the trained deep neural networks. **Supplementary Figure 2.1** shows the cumulative distribution of category labels in each training set used for trace classification.



Supplementary Figure 2.1: Cumulative distribution of labeled categories in training datasets for one-color (a), two-color (b) and three-color data (c). The same dataset was used for training the continuous wave two-color network as for the two-color ALEX network with the exception that the ALEX channel was not included.

For training of the state classifiers, only frames where all dyes are photoactive are included and hence photobleaching can be ignored for training these classifiers. The number of categories in the training sets then equals the number of states in the model. The visible states are first counted and sorted according to their chronological order before the state of each frame is assigned. This results in the first observed state always receiving the first label regardless of the FRET efficiency value and hence a state label only corresponds to a particular FRET value when a given dataset is analyzed globally.

2.3. Simulation of single molecule traces

Single-molecule intensities traces are simulated by first initializing the number of traces to be simulated (250,000 in our case) and the probability of the trace being a single-molecule trajectory or an 'aggregate'. All parameters used for simulating single-molecule traces are given in [Supplementary Table 2.1](#). An idealized FRET efficiency trajectory is then generated using an HMM routine⁸ in one, two or three colors where the FRET efficiency (or efficiencies) is randomly selected between 0.01 and 0.99. For single molecule trajectories, the number of states in the trajectory is selected with a probability of containing only a single state (45 % in our case for 'static' traces) and the remaining probability (55 % here) is equally distributed between two, three and four states ('dynamic'). In the case of 'dynamic' traces, the FRET efficiency or efficiencies of each state are randomly selected from a uniform distribution between 0.01 and 0.99. The difference in FRET efficiency for the different states has to be above a given threshold (0.1 in our cases). If this is not the case, new FRET efficiencies are randomly selected until this criterion is fulfilled. The transition rates are generated by taking the inverse of the dwell time to exposure ratio drawn from a uniform distribution between 1 and 100. The generated transition matrices are then, in general, non-symmetric. Hence, we use the transition rate matrix to calculate the probability of which state is observed first. We do this by using the least-squares solution to the matrix equation $Ax = b$, where A is the transition matrix and x is the probabilities for observing the different states. While the calculation of the state equilibrium is not mandatory for the classification accuracy, it ensures that the output matches the ground truth input of a defined transition matrix, which was used for benchmarking the transition classifier. Once the initial state has been selected, the parameters are fed into an HMM routine⁸ and a state trajectory is generated. 'Aggregate' traces are always assumed to be static (uncorrelated dynamics are categorized as 'artifact'), but the number of dye-pairs is generated from a Poisson distribution with randomly selected FRET efficiencies between each pair. Next, the idealized FRET efficiency trace is converted into normalized fluorescence intensity traces for the donor and acceptor molecules based on the FRET efficiency (discussed in more detail below). Next, photobleaching of the fluorophores are included into the trajectories. The frame at which each fluorophore photobleaches is randomly drawn from an exponential distribution. Upon photobleaching, the affected channel intensities are either set to 0 for both channels for donor photobleaching, 1 for the donor intensity upon acceptor photobleaching or recalculated using the two-color FRET equations (for three-color simulations). Blinking is then added to a fraction of the traces where each dye has a probability of being in a short-lived dark state ([Supplementary Table 2.1](#)). At this point, 'artifact' traces are generated from 'static' or 'dynamic' traces with a given probability by subtracting a constant from the trace (to simulate overestimation of the background correction), adding random fluctuations to the total intensity (to simulate among other things new molecules or aggregates flowing through the observation volume or simulating molecules in the background mask), flipping the traces (to simulate molecules that turn on during the experiment) and/or adding non-correlated signal in the different channels. To account for non-uniform brightness of the individual molecules, all excitation channels are multiplied by a scaling factor that is randomly selected from uniform distributions. In particular, the red channel after red excitation, I_{RR} , can reach scaling factors

up to three times higher than the other two channels. This allows the trace classifier to correctly analyze datasets in which high red laser powers were used to increase statistics for the calculation of correction factors and for making sure only a single red fluorophore is present. Without intensity scaling, the trace classifier strongly favors the aggregate category for traces with imperfect stoichiometry even when no second bleach step is present. With a given probability, additional small fluctuations in the total intensity are also added to the traces to simulate experimentally observed system instabilities (assuming sinusoidal oscillations of randomly determined frequency and amplitude). Next, we incorporate spectral crosstalk, direct excitation and differences in detection efficiency into the data by randomly selecting the respective parameters from a uniform distribution (see [Supplementary Table 2.1](#)). In the last step, we add two or three types of random noise to the traces. The first component is intensity-independent background noise drawn from a Poisson distribution. The second component considers intensity-dependent noise contributions (i.e. shot-noise) by drawing values from Gaussian distributions. There are different descriptions of how to treat noise from EMCCD cameras. According to Basden et al., the variance in shot noise due to the EM gain is increased by a factor of two.⁹ This corresponds to a rescaling of the Gaussian distribution mentioned above. Hirsch et al describe the additional noise from EM-CCD cameras using a gamma distribution.¹⁰ Hence, with a given probability, we also add a third component to the noise modeled using a gamma distribution with random amplitude. After adding noise to the trajectories, we then recategorize traces with high noise as 'noisy'. This is done by recalculating the FRET efficiency trace or traces from the intensity data. When the standard deviation for static traces or individual states of a dynamic trace are above the given threshold ([Supplementary Table 2.1](#), we used 0.25), the trace is categorized as 'noisy'. Finally, the classification of the individual traces is checked and, if necessary (for example a dynamic trace that photobleaches before a transition is observed), recategorized. The dataset is then balanced, as discussed above, each trace normalized to its maximum value and then used for training. For one-color traces, we simulated the intensity of the donor molecule although, for a single channel, it does not make a difference. The donor intensity (we refer to it as YY here) is given by:

$$I_{YY} = 1 - E_{YR} \quad \text{Eq. 2.1}$$

Since only one dye and one channel is observed, there is only one photobleaching category and no correction factors are included. The photoactive state of the acceptor molecule is still calculated and its influence on the donor intensity incorporated into the trace. For calculation of 'aggregates', fluorescent dye-pairs are added to the trace but only the donor signal is considered. Furthermore, the amount of noise is not quantified by the standard deviation of the FRET efficiency but by the signal-to-noise ratio of the channel intensity, which is defined as:

$$\text{SNR} = \frac{\mu}{\sigma} \quad \text{Eq. 2.2}$$

where μ is the mean signal intensity of the observed state and σ is its standard deviation. When the signal-to-noise ratio falls below the given threshold, the trace is classified as 'noisy'.

For two-color FRET simulations, we use the same approach as described in the following section for 3-color FRET but we only consider the equations necessary for 2-color FRET, i.e. all equations including the yellow/red FRET pair.

For generating three-color FRET data, the distances and Förster radii between all three fluorophores need to be considered as they are interrelated. Assuming a minimum FRET efficiency of 0.01, the generated FRET states of the first two randomly drawn FRET pairs put constraints on the maximum possible distance for the third FRET pair. To guarantee a uniform distribution of possible FRET combinations, we randomly select two of the three FRET pairs and their corresponding FRET efficiencies. Using the two selected FRET efficiencies, a lower limit is calculated for the third FRET pair, which depends on the Förster radii of the first two FRET pairs. For example, when the yellow-red dye-pair is generated last, the dye-dye separation for r_{BY} and r_{BR} are calculated and then used to determine the minimum FRET efficiency (i.e. maximum separation for the third dye-pair) as given below:

$$r_{BY} = R_{0,BY} \left(\frac{1}{E_{BY}} - 1 \right)^{\frac{1}{6}} \quad \text{Eq. 2.3}$$

$$r_{BR} = R_{0,BR} \left(\frac{1}{E_{BR}} - 1 \right)^{\frac{1}{6}} \quad \text{Eq. 2.4}$$

$$E_{YR,min} = \frac{1 - 0.01}{1 + \left(\frac{r_{BY} + r_{BR}}{R_{0,YR}} \right)^6} + 0.01 \quad \text{Eq. 2.5}$$

where the Förster radii $R_{0,BY}$, $R_{0,BR}$ and $R_{0,YR}$ are sampled over values that are typically available using commercially available dyes pairs. The FRET efficiency $E_{YR,min}$ represents the lower boundary used to randomly scale the FRET trace of the yellow-red FRET pair in a correlated or anti-correlated manner. When a different dye-pair is generated last, the same equations are used where the indices are changed accordingly. For dynamic traces, this procedure is performed for all states. Once we have selected the FRET efficiencies for the different dye-pairs and states, we then convert them what would be observed for a two-color experiment. The YR dye-pair is already a two-color FRET efficiency and does not need to be corrected. When all three fluorophores are photoactive, the blue dye may be quenched by two acceptors. In this case, the distance-related FRET efficiencies E_{BY} and E_{BR} need to be converted into the apparent FRET efficiencies $E_{BY,app}$ and $E_{BR,app}$ via:

$$E_{BY,app} = \frac{E_{BY}(1 - E_{BR})}{1 - E_{BY}E_{BR}} \quad \text{Eq. 2.6}$$

$$E_{BR,app} = \frac{E_{BR}(1 - E_{BY})}{1 - E_{BY}E_{BR}} \quad \text{Eq. 2.7}$$

Since the input data for the neural networks are normalized, the channel intensities are initialized as follows:

$$I_{BB} = 1 - E_{BY,app} - E_{BR,app} \quad \text{Eq. 2.8}$$

$$I_{BY} = E_{BY,app}(1 - E_{YR}) \quad \text{Eq. 2.9}$$

$$I_{BR} = E_{BR,app} + E_{BY,app}E_{YR} \quad \text{Eq. 2.10}$$

$$I_{YY} = 1 - E_{YR} \quad \text{Eq. 2.11}$$

$$I_{YR} = E_{YR} \quad \text{Eq. 2.12}$$

$$I_{RR} = 1 \quad \text{Eq. 2.13}$$

Upon photobleaching of one of the dyes in the three-color experiments, the system then reverts into the two-color case:

$$I_{BY,2c} = -\frac{I_{BB}E_{BY}}{E_{BY} - 1} \quad \text{Eq. 2.14}$$

$$I_{BR,2c} = -\frac{I_{BB}E_{BR}}{E_{BR} - 1} \quad \text{Eq. 2.15}$$

I_{BB} is still determined by using

$$I_{BB,2c} = 1 - E_{BY} \quad \text{Eq. 2.16}$$

$$I_{BB,2c} = 1 - E_{BR} \quad \text{Eq. 2.17}$$

where FRET to the blinking fluorophore is set equal to zero in two-color sections of the trace. Blinking events are treated the same way as photobleaching during the frames where the one dye is off and the channel intensities are either set to 0 or recalculated using Eq. 2.16/2.17.

In three-color experiments, each channel has its own set of correction factors for differences in detection efficiency and quantum yield, γ , direct excitation, de , and spectral crosstalk, ct . The values are randomly drawn from a wide uniform range and implemented in the following order. First, the FRET channels are multiplied by the corresponding γ -factor:

$$I_{BY} = \gamma_{BY}I_{BY} \quad \text{Eq. 2.18}$$

$$I_{BR} = \gamma_{BR}I_{BR} \quad \text{Eq. 2.19}$$

$$I_{YR} = \gamma_{YR}I_{YR} \quad \text{Eq. 2.20}$$

The crosstalk of the blue fluorophore leaking into the yellow and red channel after blue excitation are given by:

$$I_{BY}^{ct} = \frac{ct_{BY}}{1 + (ct_{BY} + ct_{BR})} I_{BB} \quad \text{Eq. 2.21}$$

$$I_{BR}^{ct} = \frac{ct_{BR}}{1 + (ct_{BY} + ct_{BR})} I_{BB} \quad \text{Eq. 2.22}$$

where ct_{BY} and ct_{BR} denote the randomly drawn crosstalk factors ([Supplementary Table 2.1](#)). Spectral crosstalk of the yellow fluorophore into the red channel is calculated using:

$$I_{(BY)R}^{ct} = \frac{ct_{YR} I_{BY}}{(1 + ct_{YR})} \quad \text{Eq. 2.23}$$

$$I_{YR}^{ct} = \frac{ct_{YR} I_{YY}}{(1 + ct_{YR})} \quad \text{Eq. 2.24}$$

The observed intensities including all correction factors are determined by:

$$I_{BB,obs} = I_{BB} \quad \text{Eq. 2.25}$$

$$I_{BY,obs} = \gamma_{BY} I_{BY} + de_{BY} I_{YY} - I_{(BY)R}^{ct} \quad \text{Eq. 2.26}$$

$$I_{BR,obs} = \gamma_{BR} I_{BR} + de_{BR} I_{RR} + E_{YR} \frac{de_{YR} I_{YY}}{(1 - E_{YR})} \quad \text{Eq. 2.27}$$

$$I_{YY,obs} = I_{YY} - I_{YR}^{ct} \quad \text{Eq. 2.28}$$

$$I_{YR,obs} = \gamma_{BY} I_{YR} + de_{YR} I_{RR} \quad \text{Eq. 2.29}$$

where de_{BY} , de_{BR} and de_{YR} are the randomly drawn direct excitation factors from a uniform distribution ([Supplementary Table 2.1](#)).

While the non-smFRET categories ‘noisy’ and ‘aggregate’ mimic experimental data, the category ‘artifact’ is primarily designed to increase the robustness of the trace classifier. It is important to note that the accuracy of a trained neural network to distinguish between an ‘artifact’ and any other category depends on the number of traces which are labeled as ‘artifact’ but maintain a strong resemblance to the original trace. For the goal of increasing robustness, it is therefore not desirable to achieve 100% prediction accuracy as it would be caused by too easily identifiable perturbations in the training dataset.

2.4. Simulation settings for training the state classifier network

Sixteen pre-trained deep neural networks are provided for state classification. Four models account for the classification and segmentation of time trajectories obtained from measurements using single-channel data acquisition, two-color FRET with continuous wave excitation, two-color FRET with ALEX, and three-color FRET with ALEX. For each type of experiment (one, two and three-color), we provide three state-transition-classifiers trained on either two, three or four observed states. The state classifier networks only use traces as input

that are categorized as dynamic. Hence, the training datasets only contain valid FRET traces with at least one transition, a minimum state difference of 0.1 in FRET efficiency and no photobleaching. The transition rates are generated by drawing random dwell time to exposure ratios between 1 and 100 from a uniform distribution. Traces with a state-wise FRET distribution width above 0.25 on average are excluded from the training dataset. After a dynamic trace is simulated, it is labeled according to state occupancy. Here, the first observed state always receives the first label regardless of the FRET efficiency, followed by the next observed states until the maximum number of states is reached. For three-color FRET data, every transition regardless of the dye is treated as a new state. For a two-state model, transitions of one dye can be described whereas the multi-state model also considers transitions of two dyes. Thus, we have trained the network to recognize four different states. For a system with three independently moving dyes, a minimum of 9 states would be possible in one trace. Expert users can generate a corresponding training dataset by setting the algorithm parameter ‘static dyes’ to ‘None’.

Supplementary Table 2.1. Simulation parameters for the training datasets.

Number of Frames	Maximum number of states	Minimum state difference	Dwell time / exposure ratio	Artifact probability	Stoichiometry tolerance
500	4	0.1	1-100	0.25	0.1-0.9
Aggregate probability	Blue intensity scaling	Yellow intensity scaling	Red intensity scaling	Mean bleaching frame	Blinking probability
0.1	0.7-1.3	0.7-1.3	0.5-2	400	0.2
Gamma blue/yellow	Gamma blue/red	Gamma yellow/red	Direct excitation blue/yellow	Direct excitation blue/red	Direct excitation yellow/red
0.7-2	0.49-2.6	0.7-1.3	0-0.4	0-0.2	0-0.3
Crosstalk blue/yellow	Crosstalk blue/red	Crosstalk yellow/red	Noise scaling factor	Gamma noise prob.	Noisy threshold
0-0.6	0-0.2	0-0.3	0-0.9	0.8	0.25 (SNR: 1.5)

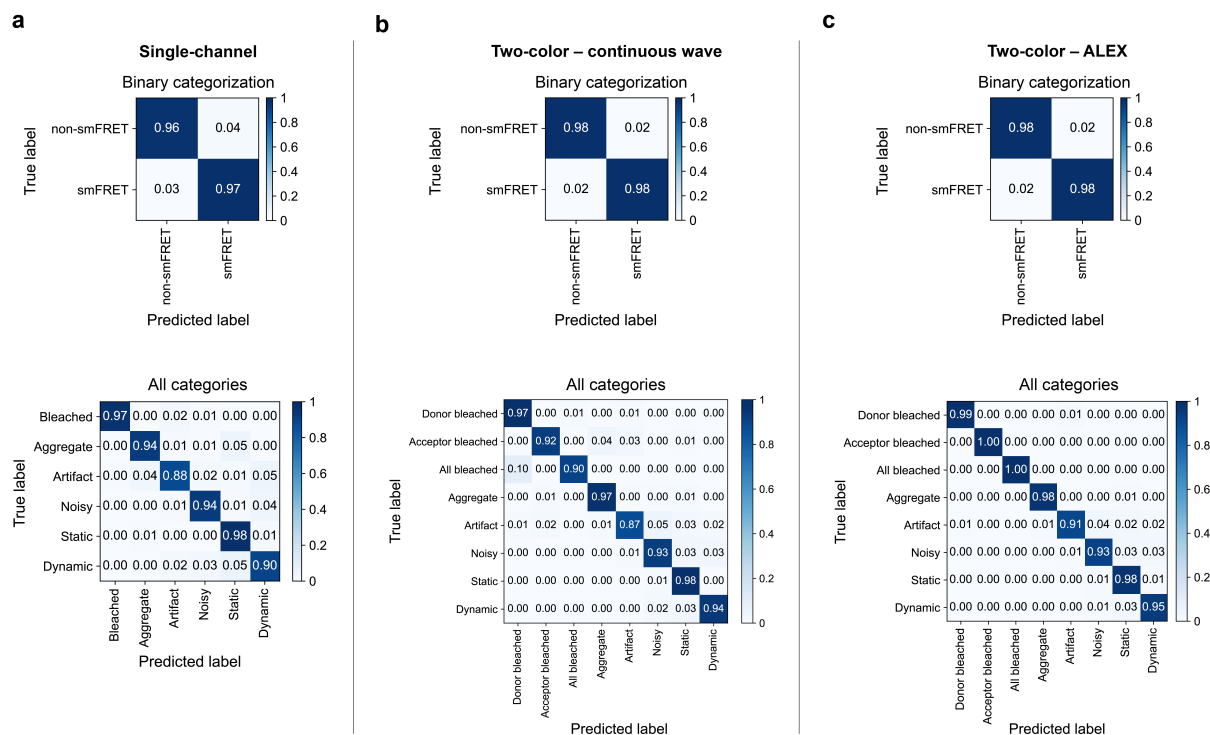
SUPPLEMENTARY NOTE 3: TRAINING VALIDATION

In the following sections (3.1–3.3), the final validation of every deep neural network is shown via confusion matrices. Approximately 20,000 new simulated traces were generated and fed into each trained model. Each row of the confusion matrices represents the instances in a ground truth category while each column represents the instances in a predicted category. The diagonal values report the percentage of true positives and true negatives whereas the off-diagonal values are the false negatives and false positives.

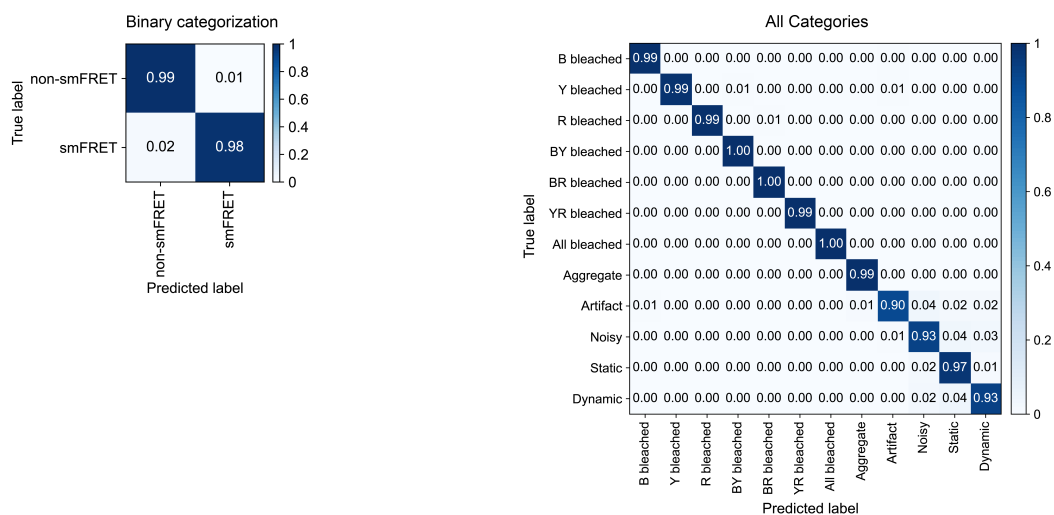
3.1. Trace classifiers

Confusion matrices for the trace-classifier networks are shown in [Supplementary Figures 3.1 and 3.2](#). The single-channel classifier has the lowest overall performance, in particular, due to a higher rate of falsely classifying random perturbations in ‘artifact’ frames (88 % precision) and misinterpreting ‘dynamic’ traces as ‘static’ (5 % false negative rate). The two-color and three-color models achieve similar accuracies for recovering smFRET frames with at least 93 % precision in correctly predicting ‘dynamic’ frames and 96 % precision for ‘static’ frames. In general, most of the false predictions concerning smFRET categories come from the high resemblance of ‘static’ frames, ‘dynamic’ frames with low contrast between states and ‘noisy’ frames close to the defined threshold. Here, the tolerance towards noise, defined as the mean standard deviation of the observed FRET efficiencies for all states, was set to 0.25. The highest sensitivity for detecting photobleached dyes (>98 %) is achieved by ALEX-enabled models for two- and three-color data. The continuous wave models depend on the contrast in intensity between the quenched and photobleached dyes, causing a significant decrease in sensitivity down to 91 % for detecting a photobleached acceptor. However, falsely predicted ‘acceptor bleached’ frames were mostly misclassified as either ‘aggregate’ or ‘artifact’ and would still be excluded from further analysis.

In addition to the confusion matrix for all available categories, we also calculated a binary trace classifier confusion matrix where we separated the frames into those that were accepted for further analysis (i.e. from ‘static’ and ‘dynamic’ traces without photobleaching) and those that were rejected (‘photobleached’, ‘aggregate’, ‘artifact’ and ‘noisy’ traces and/or frames). All trace classifier models achieve a minimum combined precision of 97 % in predicting smFRET categories, i.e. ‘static’ or ‘dynamic’, and 96 % in predicting non-smFRET categories ([Supplementary Figures 3.1 and 3.2](#)).



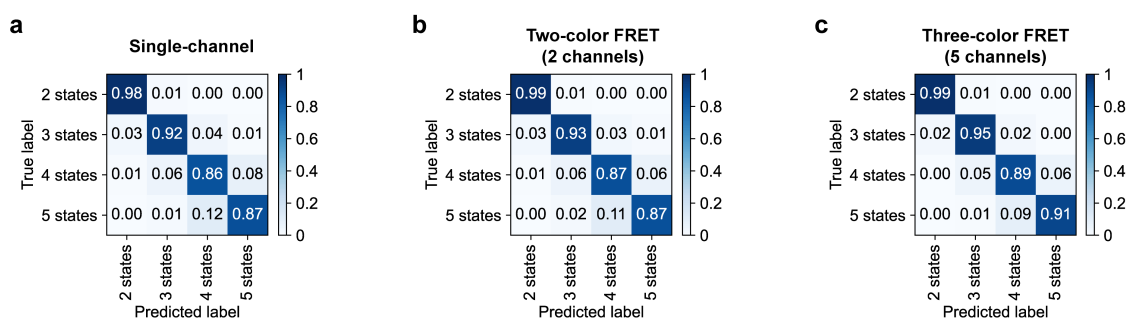
Supplementary Figure 3.1: Confusion matrices for 1-color and 2-color trace classification. Prediction accuracies depicted as confusion matrices for the (a) single-channel, (b) two-color continuous wave and (c) two-color ALEX models. The upper panels show the binary assignments into valid smFRET and non-smFRET categories. The detailed categorization is shown in the lower panels.



Supplementary Figure 3.2: Confusion matrices for 3-color trace classification. The left panel shows the binary assignments into valid smFRET and non-smFRET categories. The detailed categorization is shown in the right panel.

3.2. Number-of-states classifiers

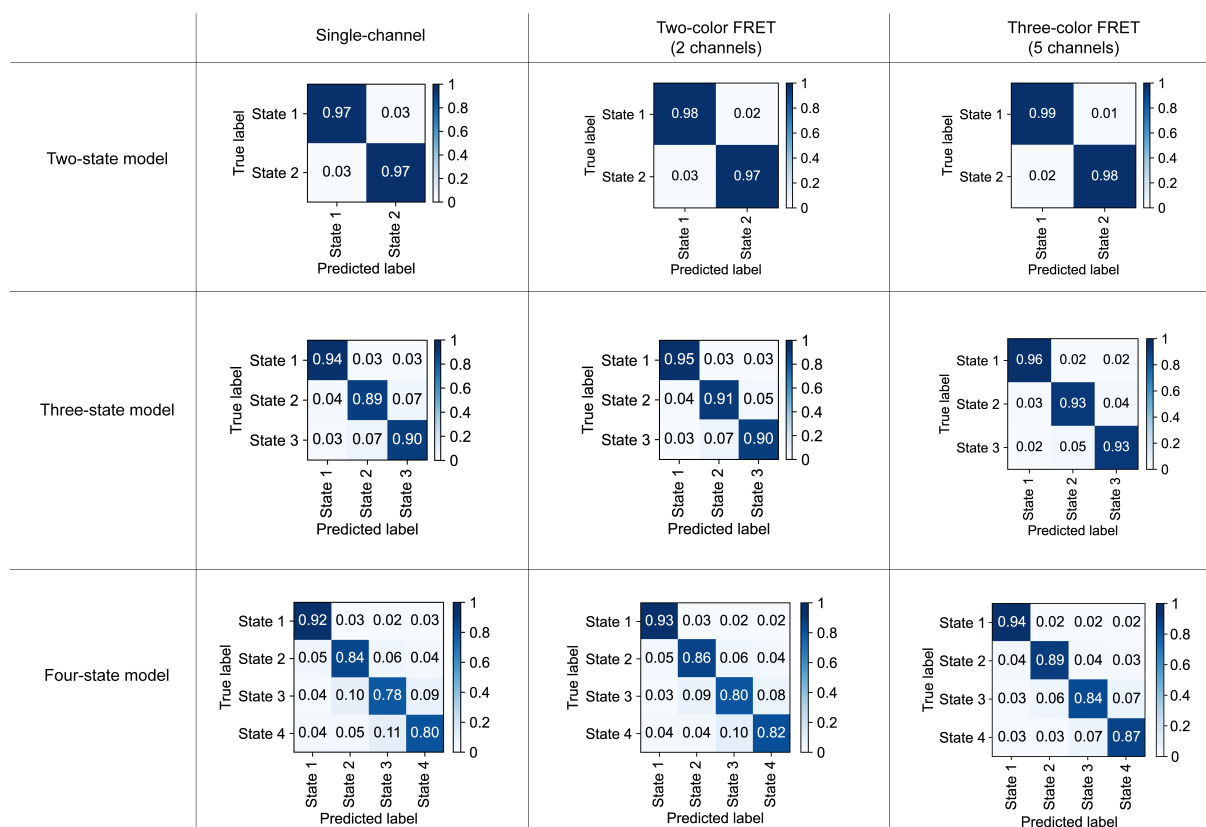
After classifying the individual traces, the dynamics are analyzed. One option is to classify the number of states in a particular trace, i.e. to run the number of states classifier for the type of data measured. [Supplementary Figure 3.3](#) shows validation of the deep neural networks trained on traces containing the given number of observed states. Only traces classified as ‘dynamic’ by the trace classifiers serve as input, hence the first category is for two observed states. The category of five observed states serves as a safeguard against traces that may be out of the scope of the pretrained state transition classifiers. All models achieve a high accuracy of at least 98 % in distinguishing two-state from multi state traces. The lowest accuracies are achieved in separating four-state from five-state traces, ranging from 86 % (single-channel) to 89 % (5-channels). The overall performance increases with increased number of available channels. As only dynamic information is considered in the state classifiers, the presence of the ALEX channel, though very useful for the trace classification, is no longer relevant.



Supplementary Figure 3.3: Confusion matrices for number of states classification. Confusion matrices for the (a) single-channel, (b) two-color FRET and (c) three-color number of states classifiers.

3.3. State-transition classifiers

After estimating the number of states in a dataset, the state trajectories of the individual dynamic traces are determined. This section summarizes the validation of the deep neural networks trained on the state occupancy and therefore also on the state transitions ([Supplementary Figure 3.4](#)). The performance does not differ significantly for the two-state models with a minimum of 97 % precision for predicting the correct state and a minimum of 84 % precision for four-state systems. The performance for multi-state models increases when more channels are available

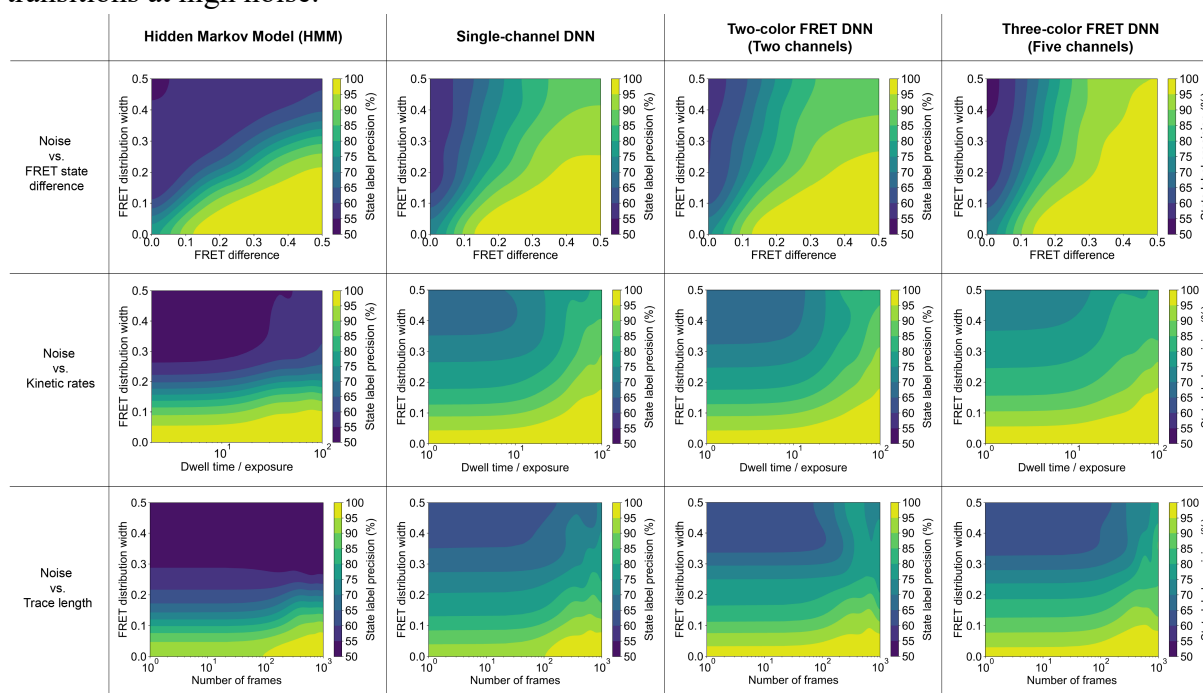


Supplementary Figure 3.4: Confusion matrices for state classification. Confusion matrices for the single-channel (first column), two-color FRET (second column) and three-color state classifiers (third column) and their corresponding two-state (first row), three-state (second row) and four-state models (third row).

3.4. Limitations of the state classifiers and a comparison with HMM

This section provides additional benchmarks and a comparison of the results from the state classifiers with HMM. First, we investigated how the performance of HMM and our state classifiers depends on noise (i.e. the width of the FRET distribution), difference between FRET states, the kinetic rates (dwell time to exposure ratio) and the length of observation time (number of frames) for dynamic transitions between two FRET states. For the three-color simulations, only the FRET distribution width of the yellow-red dye pair was used as the ground truth parameter to keep the continuity with two-color FRET traces and avoid averaging inconsistencies. [Supplementary Figure 3.5](#) shows interpolated maps of the precision of state label recovery for all models and were generated using approximately 300,000 simulated traces for each condition. The precision is the fraction of true positives divided by the sum of true positives and false positives for the state label predictions. Each map shows the precision dependency on the amount of signal noise with two of three parameters being fixed, namely the FRET state contrast (0.2), the transition rate (0.05/frame) and the number of frames (500). In general, at a fixed transition probability and number of frames (top row), the precision decreases with broader FRET distributions and smaller differences between the FRET states. All models are able to achieve a precision of at least 90% for FRET differences above 0.2 and FRET distribution widths below 0.10 with the state classifiers outperforming HMM only at high noise

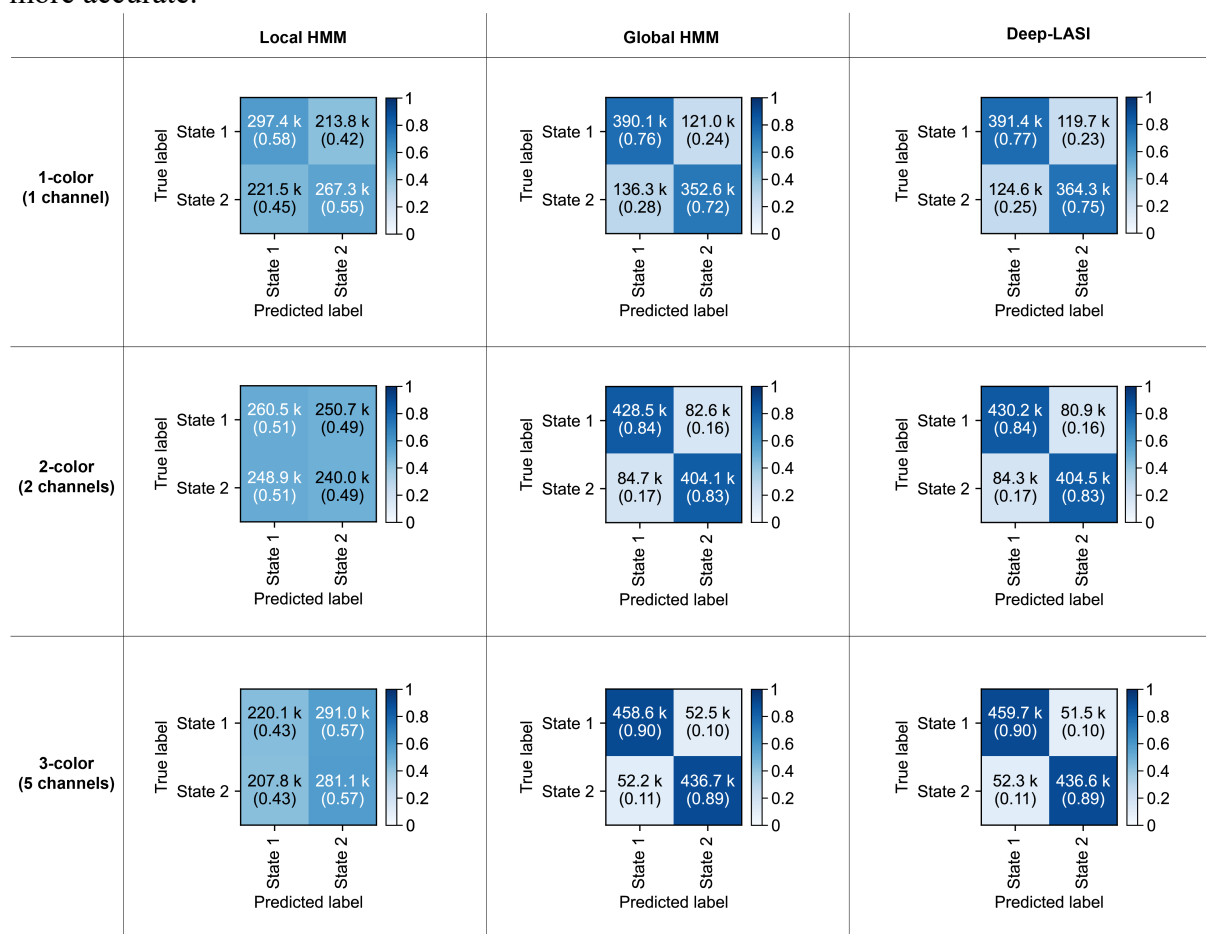
levels above 0.25. For a fixed contrast between FRET states (0.2) and total number of frames (500) (Supplementary Figure 3.5, middle row), the precision of HMM remains largely independent of the dwell time to exposure ratio at a constant noise level. All DNN state classifiers show a similar overall performance but achieves a higher precision than HMM at higher noise levels for larger dwell time to exposure ratios. For fixed FRET states and kinetic rates (Supplementary Figure 3.5, bottom row), trace length has little influence on the precision of all models below ~ 100 frames and the precision slightly increases for all models/classifiers above 100 frames. Again, the DNN outperform HMM at higher noise levels. In summary, while the precision does not differ significantly between the single-channel and two-channel state classifiers, the five-channel model used for three-color FRET shows an increased performance of up to $\sim 10\%$ at high noise levels. Due to the five available channels, the signal-to-noise ratio is effectively increased which leads to higher precision and accuracy as soon as the signal noise becomes a limiting factor for the other models. In addition, DNN models still predict transitions in high noise trajectories, however with decreased confidence, whereas HMM no longer finds transitions at high noise.



Supplementary Figure 3.5: Deep-LASI state prediction compared to HMM. Precision of the state-label recovery for HMM and for the state transition classifiers as a function of noise (i.e. width of the FRET distribution), contrast between FRET states and the kinetic rates (dwell time to exposure ratio). Each map shows the precision dependency on the noise and one additional parameter: the contrast between FRET states (top row), the kinetic rate (middle row) and the number of frames (bottom row). One dataset with $\sim 300,000$ traces was generated for each row with the corresponding two of the three parameters fixed (FRET efficiency for yellow/red: 0.4 and 0.6, transition probability: 0.05/frame, and number of frames: 500). The noise is defined as the mean standard deviation of the FRET signal from both states. The lower limit of the precision is set to 50 % since it represents the highest amount of uncertainty for two states.

3.5. Comparison of the state classifiers with local and global HMM

We also compared the performance of Deep-LASI with a local and global HMM analysis (Supplementary Figure 3.6). In a global HMM analysis, the algorithm can learn from a large dataset rather than being limited to the number of frames in a single trace. For the comparison, we simulated 2000 traces of 500 frames (1 million data points) of three-color data having two states. To avoid difficulties for the global HMM analysis, traces were simulated with time-independent, normally-distributed noise. From the three-color data, we analyzed the yellow channel alone (1-color), yellow and red together (2-color) as well as the full three-color data (3-color). As the ground truth is known, we were able to initialize a three-color HMM analysis such that it could eventually converge. Whereas local HMM struggles to analyze the traces yielding an almost random guesstimate of the state, both global HMM and Deep-LASI perform similarly well in all cases. As more channels become available, the state classification becomes more accurate.



Supplementary Figure 3.6: Confusion matrices of frame-wise state predictions for local HMM, global HMM and Deep-LASI, performed on a global data set containing 2000 traces with 500 frames each. Three-color traces were simulated and the yellow channel with yellow excitation (1-color), yellow and red channels with yellow excitation (2-color) and all five channels for the three-color data with blue and yellow excitation were analyzed (3-color). In all cases, local HMM is unable to learn states and transitions due to the limited amount of training data. Global HMM and Deep-LASI are both able to effectively remove background noise, showing similar accuracies. All models show increased performance with higher number of channels due to effectively decreasing the signal-to-noise ratio. The data set was simulated based on 3-color FRET using the following parameters: Two states with only the yellow dye transitioning between $E(YR) = 60\%$ and $E(BY) = 40\%$ to $E(YR) = 40\%$ and $E(BY) = 60\%$, a static blue to red FRET efficiency with $E(BR) = 20\%$, a symmetric transition probability of 0.05 per frame, and

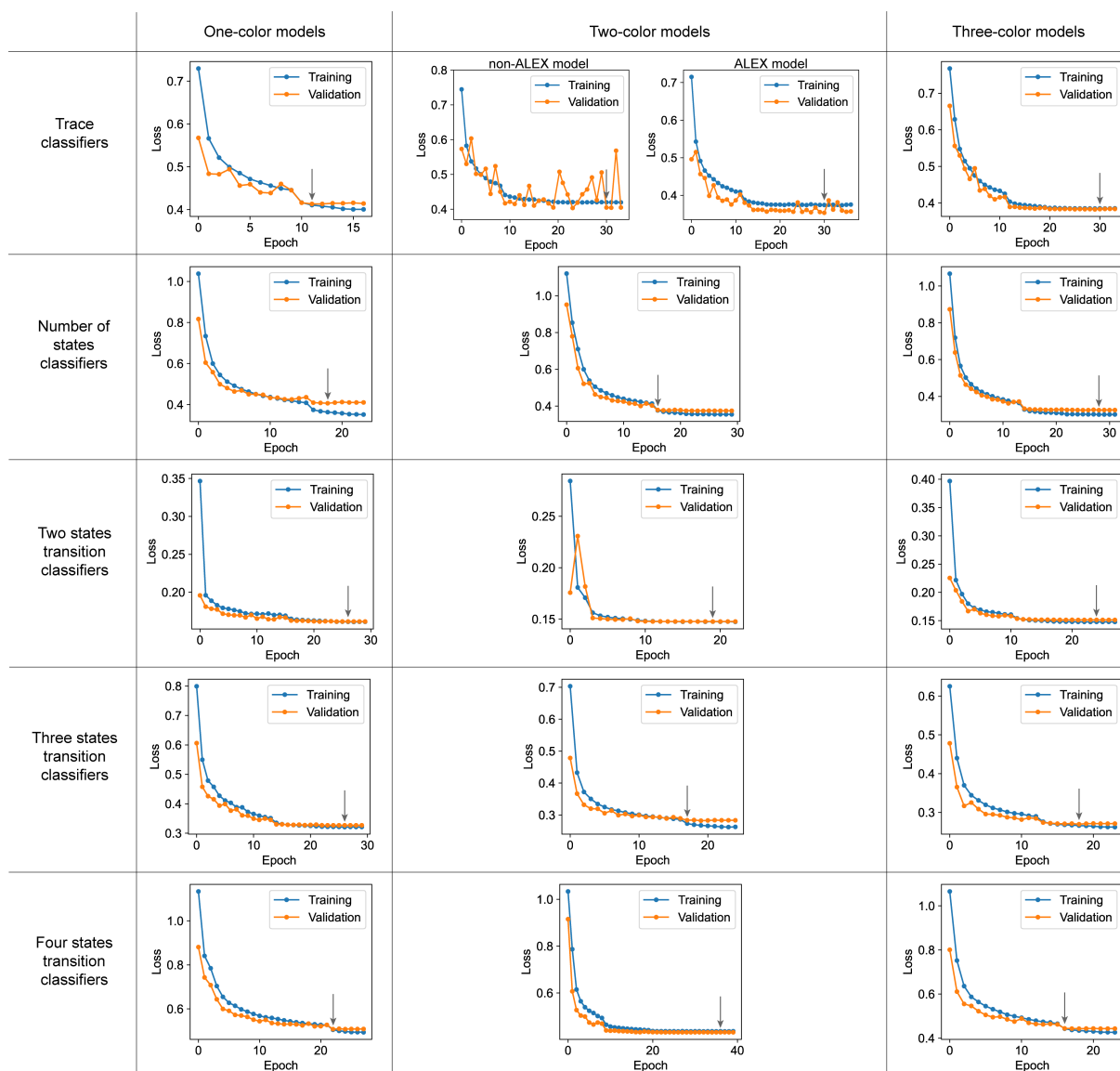
the addition of normally distributed noise resulting in a mean FRET distribution width of 0.23 (averaged over the yellow-red FRET efficiency states).

3.6. Training and validation loss

The training and validation loss for all pre-trained deep-neural networks are shown in [Supplementary Figure 3.7](#). Of the ~200,000 traces generated for training, ~160,000 were used in each epoch for training and then the capacity of the network to generalize what it learned was tested with the remaining ~40,000 traces. The error was calculated using the categorical cross-entropy, i.e. the loss function:

$$Loss = - \sum_i^C t_i \log(f(s)_i) \quad \text{Eq. 3.1}$$

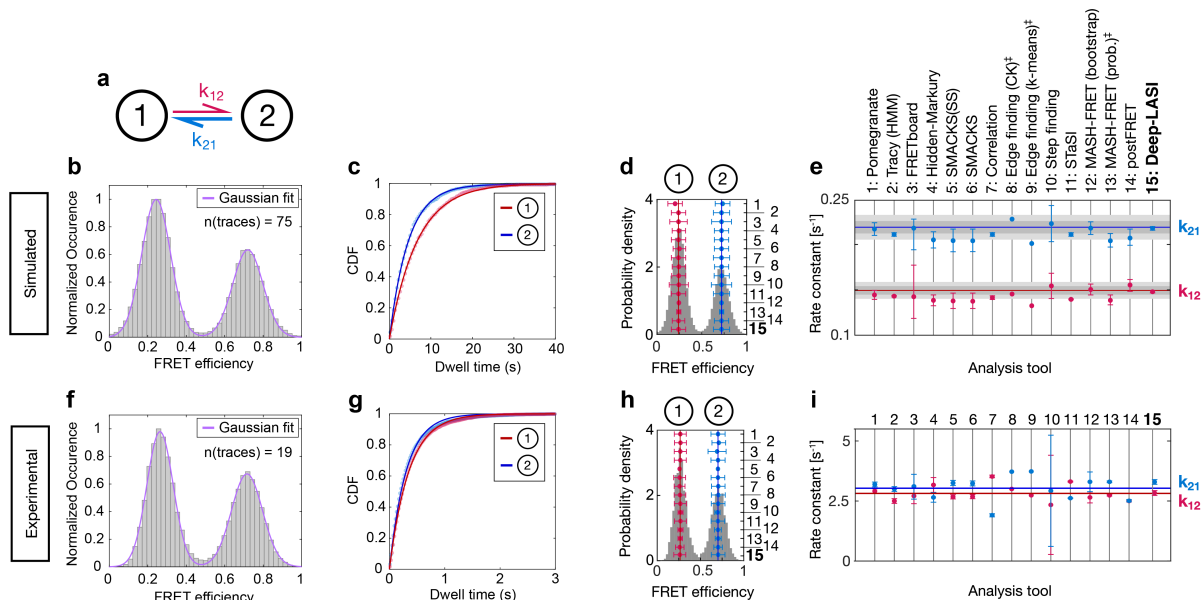
where C is the total number of classes, t is the target vector and $f(s)$ is the one-hot encoded vector of scores.¹¹ The categorical cross-entropy is specifically designed for multi-class classification problems, where each input belongs to exactly one class out of multiple mutually exclusive classes. It calculates the dissimilarity between the predicted class probabilities and the true class labels, providing a measure of how well the model captures the correct class assignments. Therefore, the model is able to produce probabilistic outputs in the form of class probabilities. By optimizing the model to minimize the cross-entropy loss, it learns to assign higher probabilities to the correct class and lower probabilities to the incorrect classes. During training of the model, the loss should decrease but maintain similar values for the training dataset as for the validation set. If a lower loss is observed for the training dataset, then for the validation dataset, the network is overfitting (i.e. it is memorizing the traces rather than learning the features of the categories). All models show no or a minimal amount of overfitting. While the full number of epochs are displayed in each plot, the model with the lowest validation loss and lowest amount of overfitting was saved and implemented (indicated with an arrow).



Supplementary Figure 3.7: Training and validation loss of all Deep-LASI models. Each row refers to the type of classifier and each column refers to the corresponding data type. For two-color data, there are two trace classifier models, one for ALEX and a second for non-ALEX measurements. Black arrows mark the saved model used when following epochs did not decrease the validation loss and indicated overfitting.

3.7. Analysis of kinetic data from the kinsoft challenge

We tested the performance of Deep-LASI on datasets provided by a recently published multi-laboratory software comparison study for extracting kinetics from smFRET data. As we contributed to this study using conventional HMM, we chose to analyze the datasets that did not require additional human input for interpretation of the data. The results are shown in [Supplementary Figure 3.8](#). Deep-LASI returned values corresponding to the ground truth for the simulated dataset and close to the average values obtained for the experimental dataset.

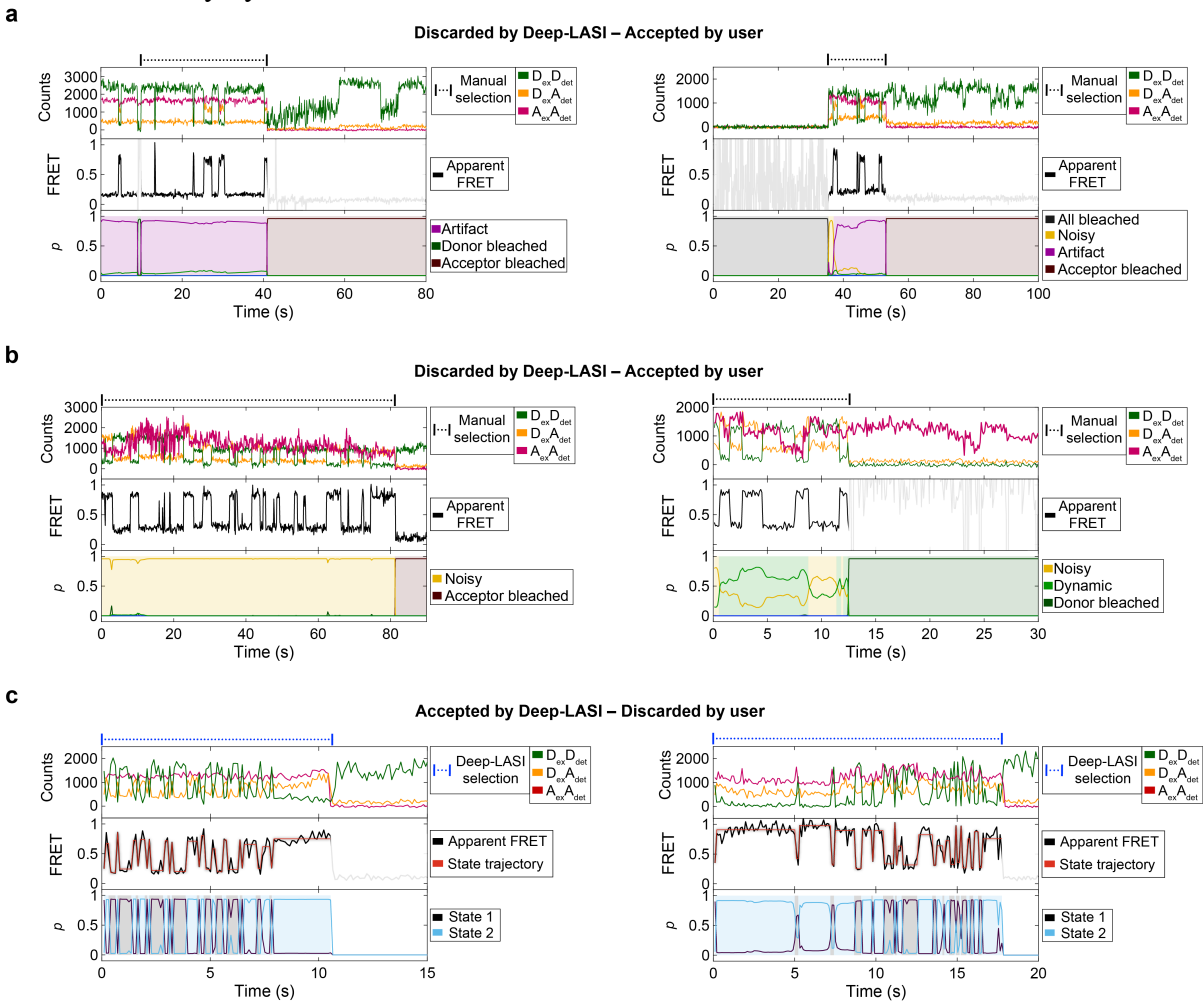


Supplementary Figure 3.8: Kinetic analysis of datasets from the kinetic software challenge. (a) An illustration of the kinetic two-state model connected by forward and backward rate constants: k_{12} and k_{21} . (b) A framewise FRET efficiency histogram (gray) of the simulated data extracted by the trace classifier. A Gaussian fit to the two populations are shown in magenta. (c) A mono-exponential dwell time distributions of the data in (b) obtained from the state-transition classifier. (d) The ground truth FRET histogram (gray) with state assignments labeled at the top and the inferred average FRET efficiencies in red and blue. Numbers on the right axis refer to the analysis tools specified in (e). Vertical lines indicate the mean over all tools. The error bars represent the standard deviations returned from the different analysis routines. (e) Rate constants and uncertainties inferred from the dataset in (d) by different labs using the respective analysis tools. The ground truth (GT) is indicated by the horizontal red and blue lines, the intrinsic uncertainty of the dataset is represented by dark gray (1σ) and light gray (2σ) intervals. (f) A framewise smFRET efficiency histogram (gray) of the experimental data extracted by the trace classifier. (g) The dwell-time distributions and corresponding mono-exponential fits of the data in (f) obtained from the state-transition classifier. A Gaussian fit to the two populations are shown in magenta. (h) A smFRET histogram of preselected traces from panel (h) where photobleaching and photoblinking contributions have been removed. State 1 is labeled in red and state 2 in blue. The vertical lines indicate the average value returned from analysis routines 1-14. The legend for the analysis routines is given in (e). The error bars represent the standard deviations returned from the different analysis routines. (i) Inferred rate constants from the experimental dataset in (h). The respective analysis tools are specified in (e). Horizontal red and blue lines indicate the mean of the inferred kinetic rate constants from analysis tools 1-14. The legend for the analysis routines is given in (e)

SUPPLEMENTARY NOTE 4: DEEP-LASI VERSUS MANUAL ANALYSES

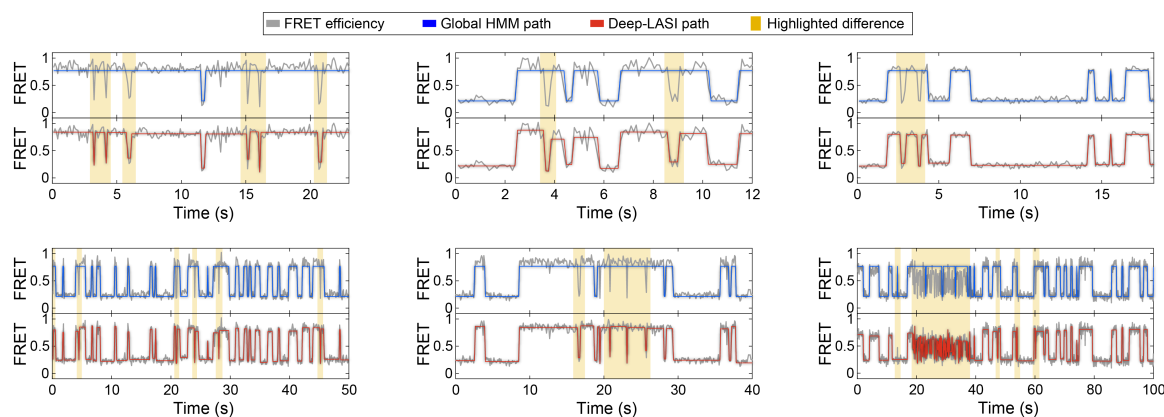
4.1 Comparison of Deep-LASI and manual analysis of 2-color DNA origami traces |

We investigated the disagreements between Deep-LASI and manual classification and summarize the primary causes, illustrated through specific examples in **Supplementary Figure 4.1**. First, traces that exhibit non-ideal behavior after photobleaching are often thrown out by Deep-LASI as the entire traces is than categorized as an artifact whereas users may ignore characteristics of the traces in the non-accepted regions. Secondly, in cases where the leading frames are photobleached and the molecules reactivate, Deep-LASI tends to classify the entire trace as an artifact instead of extracting potentially useful information from the middle section of the trace. This is due to the fact that the training dataset does not yet include valid single molecule FRET traces starting with inactive dyes. Thirdly, Deep-LASI categorizes high noise or intensity fluctuations in the acceptor channel as noisy or classifies them as dynamic, but with insufficient confidence ($> 70\%$) to include in further analyses. In such cases, the user has the flexibility to adjust the confidence threshold or consider the unfiltered dynamic category based on the maximum confidence output. Lastly, Deep-LASI may select traces that were discarded by manual evaluation, particularly short traces with fast dynamics that may be overlooked or considered noisy by the user.



Supplementary Figure 4.1: Representative 2-color DNA origami traces with disagreements between manual and Deep-LASI selection. (a) Non-ideal donor signals. In the case of non-ideal donor intensities after photobleaching of the acceptor (left panel) or leading photobleached frames (right panel), Deep-LASI tends to classify the whole trace as an artifact while the manual selection still includes the middle region between the photobleached frames. (b) Non-ideal acceptor signals. Even though the FRET efficiency trace looks ideal, due to the high noise or intensity fluctuations in the acceptor channel, Deep-LASI classifies the trace as noisy (left panel) or is not confident enough ($> 70\%$) to classify the trace as dynamic (right panel). In the latter case, the user could influence the selection of this trace by lowering the confidence threshold. (c) Traces selected by Deep-LASI that were discarded by manual evaluation. Short traces (pay attention to the timescale on the x-axes) with fast dynamics can be either overlooked or deemed noisy by the user.

We further compare representative FRET traces analyzed using global HMM and Deep-LASI (Supplementary Figure 4.2). Global HMM tends to struggle in accurately capturing fast transitions, which can be attributed to the difficulties of HMM in distinguishing between fast transitions and noise or due to the inherent (non-Markovian) heterogeneities commonly encountered in single molecule experiments. Local HMM analyses can more easily deal with these heterogeneities. In comparison, Deep-LASI exhibits enhanced performance in detecting and characterizing these fast kinetics, suggesting its potential advantages over global analysis approaches.

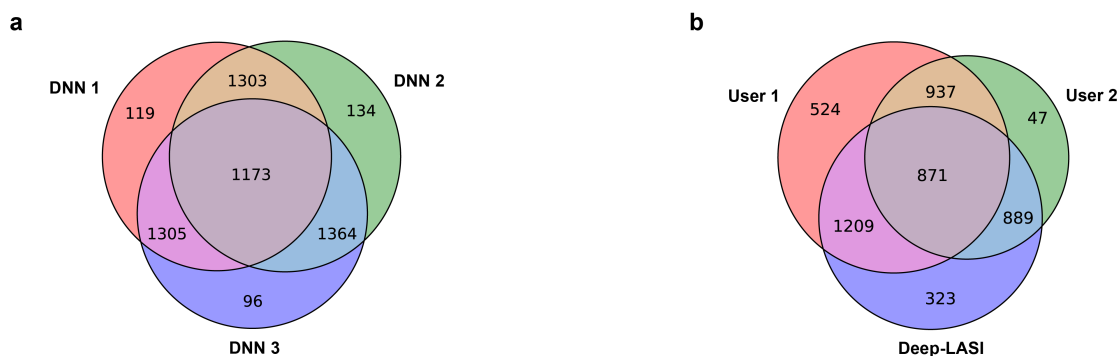


Supplementary Figure 4.2: Comparison of global HMM and Deep-LASI analyses. Representative 2-color DNA origami smFRET traces highlighting the differences between a global HMM and Deep-LASI analyses. In contrast to Deep-LASI, global HMM frequently misses fast transitions due to heterogeneities in the single molecule data.

4.2. Influence of different training datasets and comparison to user classification

The quality of a neural network rises and falls with the data by which it has been trained. To see the influence of different training datasets, we simulated three different datasets using the same parameters and used them to train Deep-LASI. The three different networks were then used to classify the two-color, two-state data shown in Figure 3. The results are shown in Supplementary Figure 4.3a. From the confusion matrix shown in Supplementary Figure 3.1, one would expect a consistency on the order of 95 %, provided the experimental data are similar to the training datasets. Each pair of networks agree within ca 93 %. Interestingly, the consistency between the neural networks is higher than that from two individual users (Supplementary Figure 4.3b). Here, user 1 tried to maximize the statistics and selected subsections of traces whereas user 2 was very conservative, only classifying the best traces as

dynamic. This suggests that neural networks may be more consistent in the analysis than different users.



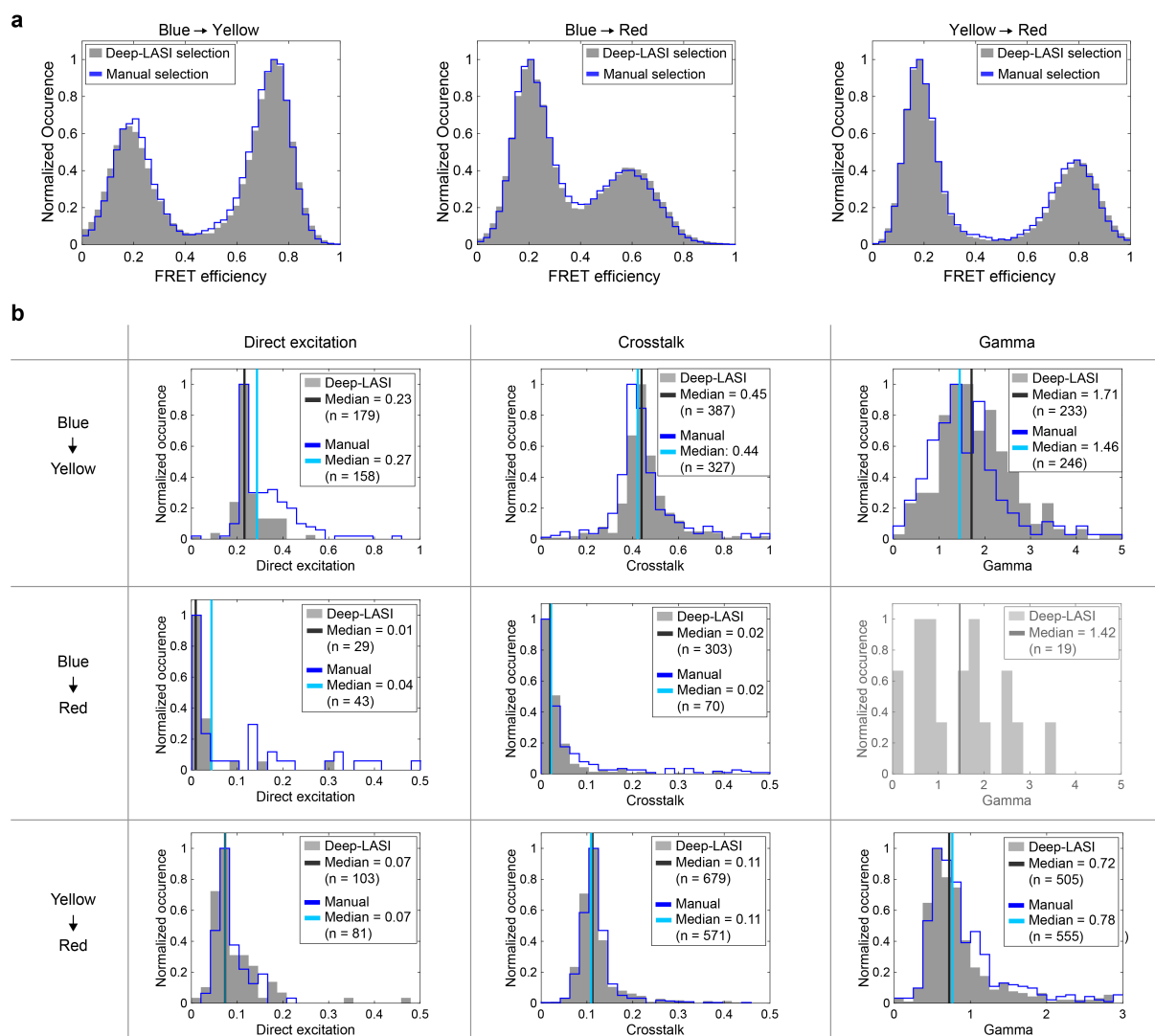
Supplementary Figure 4.3: Venn diagrams for differently trained neural networks and users. a) A Venn diagram showing the consistency between traces classified as dynamic (with a 70% confidence threshold) for three networks trained using different datasets (DNN1, DNN1 and DNN3) and applied to the experimental data shown in Figure 3. b) A Venn diagram comparing the number of traces classified as dynamic by two different users as well as DNN1 (with a 70% confidence threshold) for the same dataset used in panel a.

4.3 Deep-LASI versus manual analyses for 3-color DNA origami samples

We emphasize the importance of also using experimental data for testing deep learning methods trained on synthetic data since the simulations used for validation are usually generated by the same algorithm as the training dataset. Deep neural networks can easily learn biases of any kind in the training data, which may have no relationship to the respective category under new conditions. Hence, the prediction of categories with respect to ground truth simulations can produce high accuracies, which may not be directly translatable to real-world examples. Therefore, we compared the performance of our network models on real data with that of experts who manually analyzed the same dataset.

We benchmarked the three-color performance of Deep-LASI by comparing the automated analysis with traces manually selected by an expert user (Supplementary Note 5). We used the three-color L-shaped DNA origami structure with two binding locations spaced at 6 and 12 o'clock with complementary binding regions of 7.5 nt (Figure 4). Deep-LASI yielded 581 usable smFRET traces versus 694 for manual selection out of a total of 2545 extracted traces (Supplementary Figure 4.4a). The two uncorrected, framewise smFRET histograms are almost identical. The automatically extracted FRET correction factors, which are based on the predictions of the three-color trace classifier, were compared to those determined manually. The expert user selected the relevant regions of the traces for determining various FRET correction factors by hand. Very similar distributions and median values were obtained for the YR correction factors (Supplementary Figure 4.4b). For BR, both direction excitation and spectral crosstalk terms are small and the differences are not significant here. Due to the high stability of the yellow fluorophore, it is challenging to collect enough statistics to directly derive the detection correction factor. Hence, it is calculated from the product of the BY and YR γ factors. For BY, the distributions for spectral crosstalk from Deep-LASI and manual selection are consistent. However, for both direct excitation and the detection correction factor, there are

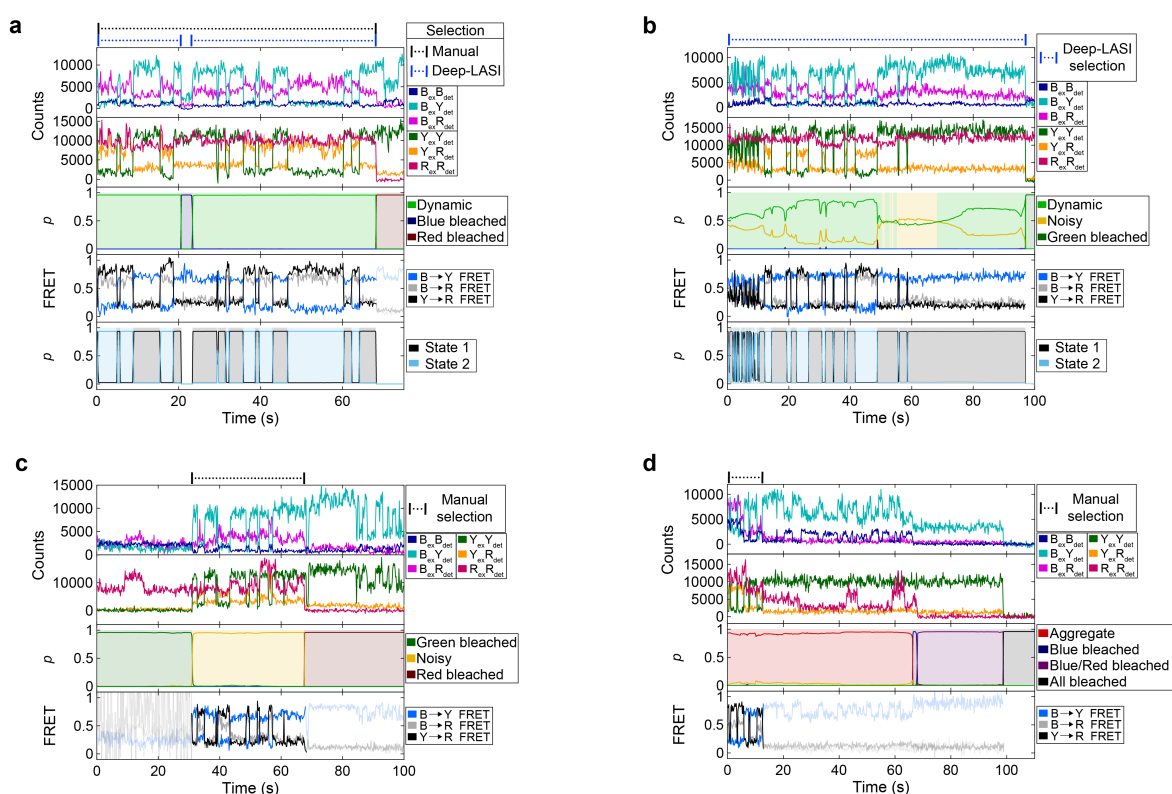
differences of $\sim 15\%$. Manual selection with the blue fluorophore is difficult because of the low fluorescence intensity of the blue dye. In the manually selected regions for direct excitation, a second population is visible due to difficulties of distinguishing between a Y only fluorophore and a dim B fluorophore undergoing high FRET. Similarly, there are differences in the detection correction factor distribution. As Deep-LASI has more flexibility in choosing relevant regions of the traces for determining the correction factors, it is most likely that Deep-LASI is more accurate in these cases. FRET correction-factor determination is a potential source of human bias in the analysis of smFRET data and we demonstrate here an advantage of using a well-trained neural network for automated analysis.



Supplementary Figure 4.4: Uncorrected smFRET histograms and correction factors extracted by Deep-LASI for the 3-color 2-state DNA origami. (a) Uncorrected framewise smFRET histograms for BY, BR and YR calculated from traces selected manually ($n=694$, blue line) versus the histograms determined by Deep-LASI ($n=581$, gray histograms). There is excellent correspondence between the histograms. **(b)** Each panel displays the normalized distribution of available correction factors from all traces categorized as ‘dynamic’ by Deep-LASI (gray filled histograms) or manually labeled as dynamic (blue histogram line). Due to the high stability of the yellow dye compared to the blue and red dyes, the number of usable traces to calculate blue/red detection correction factor was too low to be determinable. Therefore, we used the theoretical value of 1.23 (for Deep-LASI compared to 1.15 for manual selection) for the blue/red gamma factor determined from the product of the gamma factors for blue/yellow and yellow/red.

4.4 Comparison of 3-color DNA origami traces selected manually and/or via DNN classification.

To gain insights into the selection criteria of traces performed manually and via Deep-LASI, we examined the traces in detail that were selected differently. Examples are shown in [Supplementary Figure 4.5](#). In general, similar differences arise as observed in the two-color FRET classifications ([Supplementary Figure 4.1](#)). In case of three-color FRET, blinking and dark states of the blue dye are inherently difficult to spot during manual inspection of the trace, whereas Deep-LASI predicts dark frames with high accuracy. Single-molecule traces often exhibit ambiguity. Deep-LASI tends to disregard traces displaying non-ideal intensities due to the way it was trained. Conversely, when manually selecting traces, users may incline towards including more non-ideal traces to improve the statistics and their selection can be subjectively influenced by a myriad of parameters.



Supplementary Figure 4.5: Representative 3-color DNA origami traces with disagreements between manual and Deep-LASI selection. (a) Short blinking events. In this trace, Deep-LASI and manual selection agree in general but Deep-LASI excludes frames with an inactive blue dye. Blinking events of the strongly quenched blue dye in 3-color experiments can be easily missed during manual selection. (b) Regions of traces with high-noise or fast dynamics. The trace shown here was excluded manually due to seemingly high noise at the beginning of the trace. Deep-LASI predictions show the competing categories of ‘dynamic’ and ‘noisy’. The summed confidence for the ‘dynamic’ classification exceeds the user defined threshold of 70 % and the state classifier predicts state transitions with high confidence. (c) Initial dark frames and non-ideal intensities. In the training datasets, we currently do not start with photobleached molecules that begin to fluoresce during the traces. Here is an example of Deep-LASI’s tendency to classify valid sections as noisy due to leading bleached frames or erratic intensities during bleached frames. (d) Short traces. Short section in the beginning of the trace is manually selected whereas Deep-LASI classifies the whole trace as an aggregate due to intensity spikes in the acceptor channel after the valid section.

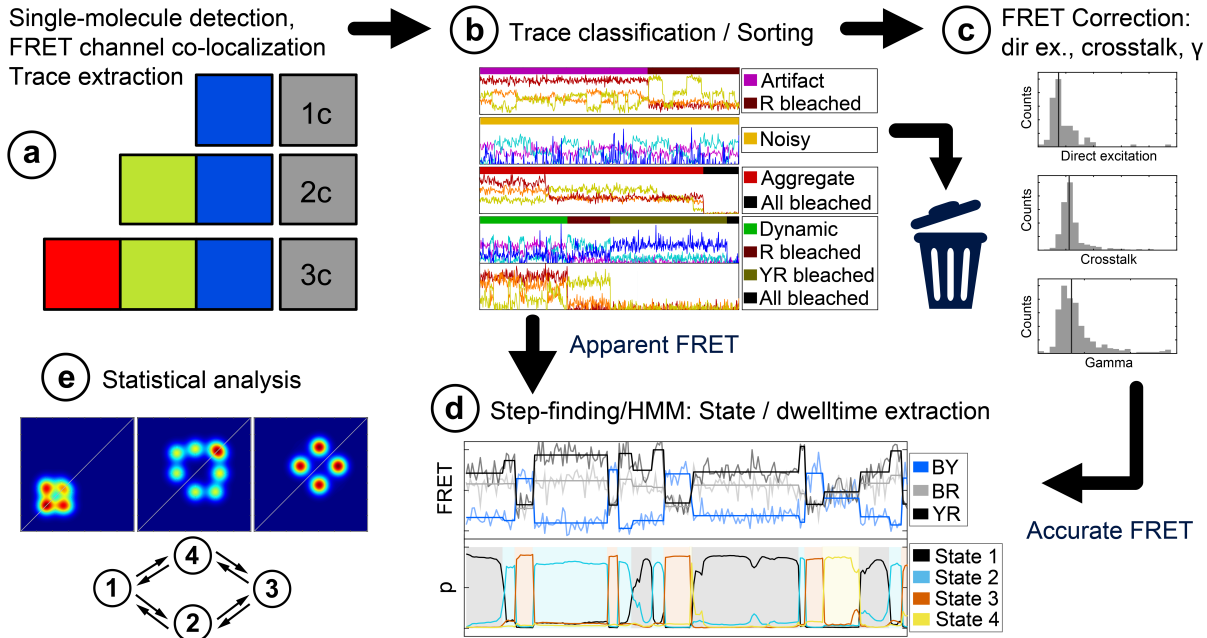
SUPPLEMENTARY NOTE 5: MANUAL ANALYSIS OF SINGLE-MOLECULE TIRF DATA

5.1. Work-flow

We benchmarked the performance of Deep-LASI by comparing it to manually analyzed single-molecule data from an expert user. Starting from individual movies, the procedure for extracting the intensity information over time is highlighted in **Supplementary Figure 5.1**.

The procedure begins with:

- (1) a pixel-wise mapping of the position between two or three cameras for two- and three-color experiments,
- (2) camera-wise localization and excitation-cycle dependent assignment of intensities, and
- (3) extraction of intensities and background correction for each detection channel.



Supplementary Figure 5.1: Work-flow of data extraction, sorting, analysis and evaluation. a) One-, two-, or three-color data is collected with various excitation schemes and the time-dependent intensity traces extracted and corrected for background. b) The traces are then visually inspected and sorted either for further analysis or marked as junk. Regions of the trace can be selected for smFRET evaluation or for correction factor determination. c) After manual selection, trace-wise correction factors are extracted. d) For the dynamic traces selected for further analysis, the dwell-time distributions are determined using a Hidden Markov Model approach. e) From the HMM analysis, transition density plots are extracted from the smFRET data.

Next, the recorded traces were analyzed (**Supplementary Figure 5.1b-e**) either manually (**Section 5.4-5.9**) or assisted by neural networks (cf. **Supplementary Note 1**). Manual evaluation of single-molecule or multi-color FRET traces involves:

- (4) the pre-sorting of traces suitable for 1c, 2c-, and 3c- smFRET analyses
- (5) determining consecutive regions of the trace for evaluation and for determination of local and global correction factors

- (6) Hidden Markov Modeling of smFRET traces to identify underlying states and dwell-times
- (7) kinetic evaluation of transition rates and states using transition density plots (TDP), state-wise histograms and dwell-time analyses.

5.2. Camera mapping for FRET traces

In order to extract the fluorescent intensity traces of individual, fluorescently labeled DNA origami structures detected in various channels, an accurate localization and mapping of the detected emission channels across the three cameras or detection areas needs to be achieved. To compensate for potential chromatic aberrations and non-ideal alignment, an image transformation was used to map the corresponding pixels between different cameras/regions onto each other. The associated transformation matrix describing the potential shifts, tilts, etc. was obtained by imaging a calibration pattern on all three detection channels ([Supplementary Figure 5.1a](#)). As a calibration pattern, we typically use a zero-mode waveguide array.

5.3. Trace extraction and background subtraction

After mapping the different detection channels, the location of the individual single emitters needs to be determined and the intensity extracted. When msALEX excitation is used, the alternating laser excitation scheme needs to be taken into account and the intensity traces separated based on both the detection and excitation channels. The most blue-shifted detection channel serves as the reference channel (Channel 1). This refers to the blue excitation, blue channel (BB) for BY-, BR- and BYR-labeled samples or the yellow detection channel with yellow excitation (YY) for YR-labeled samples ([Supplementary Figure 5.1a](#)).

Individual molecules in the reference channel are identified by searching for the brightest spot in the summed projection of the movie. After calculating the central position of the molecule using a wavelet approach¹², the corresponding position in all other channels is calculated using the transformation matrix. Molecules in the projection images exhibiting detectable intensity in all desired channels are then selected. The intensities and background are extracted using different masks. For the signal, the pixels within an approximate circle of roughly 3 pixels radius around the central coordinates of the molecule are summed together. With a pixel size of 124 nm, the fluorescence signal of a single molecule is accumulated within an area of $614 \times 614 \text{ nm}^2$. For the background, a mask representing roughly a circle with radius of 7.5 pixels (850 nm) and width of 2 pixel centered on the molecule is used. The background is calculated as the median value of all pixels inside the ring-shaped mask and averaged over a five-frame sliding window depending on the excitation cycle and the detection channel. Afterwards, the determined background is scaled to the signal mask and subtracted from the framewise intensity per each channel for each molecule. When analyzing single molecule traces from hand, trajectories which contain molecules within the background mask are discarded. In DeepLASI, these traces are typically discarded in the 'artifact' category during the first characterization step.

5.4. Manual trace selection and analysis

The background-corrected fluorescence intensity traces of the individual molecules are then inspected and sorted (Supplementary Figure 5.1b). The properties of the extracted traces are generally very heterogeneous. This stems from different sources including photochemistry, dye blinking, aggregates and impurities within the sample of different brightness. In all cases, molecules were rejected automatically if they exhibited (1) a low SNR or (2) a brightness that is significantly higher than expected for a single fluorophore (aggregates or impurities). We further classified traces according to

- (1) their static and dynamic behavior
- (2) the existence of photobleaching steps in the different intensity channels
- (3) the order of bleaching steps between the different intensity channels
- (4) the degree of labeling efficiency

With the presorted trajectories at hand, we next prepared the data either for (1) correction factor determination to obtain accurate FRET efficiencies (Supplementary Note 5.5) or (2) directly to kinetic and state evaluation based on background-corrected trajectories (1-color data) or apparent FRET efficiencies (2/3-color data). In the first case, we first derived the correction factors per trace and marked regions for trace evaluation by HMM afterwards (Supplementary Note 5.6). In the second case, we manually marked the regions in traces to be analyzed and added them to the ‘HMM’ category.

5.5. Accurate FRET determination

In real smFRET experiments, the intensity of the acceptor signal needs to be additionally corrected for direct excitation of the acceptor fluorophore and spectral crosstalk from donor into the acceptor channel. In addition, the one needs to correct for the difference in the detection sensitivity between the donor and acceptor fluorophores. The correction factors are denoted as:

de_{XY} for direct excitation of the acceptor fluorophore Y during excitation with X ,

ct_{XY} for spectral crosstalk from the fluorophore X in the detector channel Y ,

and γ_{XY} compensates for differences in detection sensitivities between channels.

We denote the background-corrected intensities as I_{XY} and the corrected Intensity as $I_{XY,corr}$, where x stands for the excitation source and y for the emission channel, i.e. $I_{BR,corr}$ denotes the background corrected emission of the acceptor within the red channel (R) after donor excitation in the blue channel (B).

Trace-wise and global correction factors

¹³Depending on when individual fluorophores photobleach, some of the correction factors can be extracted from the trace itself. However, in the vast majority of the traces, one cannot extract all correction factors individually. When a trace-wise correction factor is unavailable or unreasonable, the *median* value of the corresponding distribution of trace-wise correction factors for the particular correction factor is used to calculate the accurate FRET values, i.e. a global correction factor. Using traces that were presorted and categorized as ‘Blue dye

bleached' (or 'yellow / red dye bleached', respectively), we first determined the trace-wise correction factors for direction excitation de_{XY} and spectral crosstalk ct_{XY} . Having corrected the background-corrected intensities against both contributions, we next determined the trace-wise correction factor γ_{XY} .

To derive the contribution of spectral crosstalk from the donor channel X in the acceptor channel Y , we determine the trace-wise correction factor ct_{XY} using the intensity information after photobleaching of the acceptor:

$$ct_{XY} = \frac{\langle I_{XY} \rangle}{\langle I_{XX} \rangle} \Big|_{no\ acceptor} \quad \text{Eq. 5.1}$$

Here, $\langle I_{XX} \rangle$ refers the mean donor intensity and $\langle I_{XY} \rangle$ to the mean acceptor intensity after donor excitation in the region of the trace where there is no acceptor fluorescence.

Similarly, we determined the correction factors for direct excitation of the acceptor during donor excitation using traces in which the donor fluorophore bleached first:

$$de_{XY} = \frac{\langle I_{XY} \rangle}{\langle I_{YY} \rangle} \Big|_{no\ donor} \quad \text{Eq. 5.2}$$

where $\langle I_{XY} \rangle$ and $\langle I_{YY} \rangle$ describes the mean acceptor emission after donor excitation or acceptor excitation, respectively.

Lastly, we determined the detection correction factors γ_{XY} compensating for differences in detection sensitivities between different channels. For this, we used traces where the acceptor photobleaches before the donor. The acceptor intensity is first corrected for direct excitation de_{XY} and spectral crosstalk ct_{XY} . We then derive the detection correction factor γ_{XY} per trace from the ratio of changes in donor and acceptor emission before and after photobleaching of the acceptor. The correction factors are denoted as:

$$\gamma_{XY} = \frac{\langle \Delta I_{XY,cor} \rangle}{\langle \Delta I_{XX,cor} \rangle} \Big|_{A\ bleaches} \quad \text{Eq. 5.3}$$

where $\langle \Delta I_{XX,cor} \rangle$ and $\langle \Delta I_{XY,cor} \rangle$ refer to the intensity difference for the mean donor and acceptor emission after donor excitation before and after acceptor photobleaching.

Data Correction

Once all correction factors are determined, every trace is corrected using the local, trace-wise correction factors, when available and suitable. Otherwise, the global correction factor is used. In three-color experiments, the corrected FRET efficiency for E_{YR} is calculated first since it is required for subsequent corrections. Upon yellow excitation, the same approach is used as for two-color FRET experiments:

$$I_{YY,corr} = I_{YY} \quad \text{Eq. 5.4}$$

$$I_{YR,corr} = I_{YR} - ct_{YR}I_{YY} - de_{YR}I_{RR} \quad \text{Eq. 5.5}$$

The corrected FRET efficiency is then given by the ratio of both corrected intensities

$$E_{YR} = \frac{I_{YR,corr}}{\gamma_{YR}I_{YY,corr} + I_{YR,corr}} \quad \text{Eq. 5.6}$$

For the BY FRET pair, the fully corrected intensities after blue excitation read as:

$$I_{BB,corr} = I_{BB} \quad \text{Eq. 5.7}$$

$$I_{BY,corr} = I_{BY} - ct_{BY}I_{BB} - de_{BY}I_{YY} \quad \text{Eq. 5.8}$$

The accurate BY FRET efficiency follows equation 5.5 with an additional term which takes into account the reduction in brightness of the yellow dye due to the FRET process between the YR pair:

$$E_{BY} = \frac{I_{BY,corr}}{\gamma_{BY}I_{BB,corr}(1 - E_{YR}) + I_{BY,corr}} \quad \text{Eq. 5.9}$$

The intensity of the red fluorophore after blue excitation needs to be corrected against direct excitation, contributions of both the blue and yellow dye due to crosstalk into the red channel and due to cascading of FRET from the blue dye over the yellow dye into the red channel:

$$I_{BR,corr} = I_{BR} - de_{BR}I_{RR} - ct_{BR}I_{BB} - ct_{YR}(I_{BY} - ct_{BY}I_{BB}) - de_{BY}E_{YR}(1 - E_{YR})^{-1}I_{YY} \quad \text{Eq. 5.10}$$

The accurate FRET efficiency of the BR FRET pair is then given by:

$$E_{BR} = \frac{I_{BR,corr} - E_{YR}(\gamma_{YR}I_{BY,corr} + I_{BR,corr})}{\gamma_{BR}I_{BB,corr} + I_{BR,corr} - E_{YR}(\gamma_{BR}I_{BB,corr} + \gamma_{YR}I_{BY,corr} + I_{BR,corr})} \quad \text{Eq. 5.11}$$

5.6. Hidden-Markov modeling

The kinetics and underlying states within the selected trajectories, i.e. either smFRET or intensity traces, were evaluated using Hidden Markov Modeling. The input data of both assays vary between 0 and 1. We anticipate that every molecule undergoes transitions between a fixed numbers of conformations described by a discrete number of states q_i ($i = 1, \dots, Q$). The behavior of the system can be captured by the joint distribution of the observed data $\mathbf{x} = (x_1, x_2, \dots, x_T)$ and the corresponding hidden state sequence $\mathbf{q} = (q_1, q_2, \dots, q_T)$. The joint distribution can be factorized as follows:

$$p(\mathbf{x}, \mathbf{q}) = p(x_1|q_1) \cdot p(q_1) \cdot \prod_{t=2}^T p(x_t|q_t) \cdot p(q_t|q_{t-1}) \quad \text{Eq. 5.12}$$

Here, $p(x_t|q_t)$ represents the conditional probability of observing x_t given the system is in the hidden state q_t , $p(q_1)$ represents the probability of being in the initial state, q_1 , and $p(q_t|q_{t-1})$ represents the conditional probability of transitioning from state q_{t-1} to state q_t .

For a system with Q states in total, the transition probability matrix $\bar{\mathbf{K}}$ comprises $Q \times (Q - 1)$ independent transition probabilities k_{ij} describing the likelihood for going from state i to state j . Here, it is a prerequisite for the Markovian process, that the row-wise sum of transition probabilities is normalized to 1. For a Hidden-Markovian process, the state sequence is not directly observable but buried in random noise of the system. It can only be inferred from measured observables \mathbf{x} , i.e. the single-molecule trajectory, with a length of T data points. Here, the emission probabilities $f_{q_i}(x_t|\theta_q)$ serve as parameter to represent the relative likelihood for observing a specific FRET value (or intensity value) for a given set of model parameters θ_q and the molecule being in state q_i . For intensity measurements and single molecule FRET traces, it is appropriate to model the emission probability of a state q_i as a Gaussian distribution:^{14,15}

$$f_{q_i}(x_t|\theta_q = \{\mu_{q_i}, \sigma_{q_i}\}) = \frac{1}{\sqrt{2\pi}\sigma_{q_i}} \cdot e^{-\frac{(x_t - \mu_{q_i})^2}{2\sigma_{q_i}^2}} \quad \text{Eq. 5.13}$$

The parameters estimators are: the mean value μ_{q_i} and covariance σ_{q_i}

$$\langle \mu_{q_i} \rangle = \frac{\sum_{t=0}^T w_{q_i,t} x_t}{\sum_{t=0}^T w_{q_i,t}} \quad \text{Eq. 5.14}$$

$$\sigma_{q_i}^2 = \frac{\sum_{t=0}^T w_{q_i,t} x_t^2}{\sum_{t=0}^T w_{q_i,t}} - \langle \mu_{q_i} \rangle^2 \quad \text{Eq. 5.15}$$

For this, we introduce the relative occurrence probability $w_{q_i,t}$, i.e. the conditional probability $w_{q_i,t}$ of being in state q_i given the data x_t at a time t , which is linked to the fraction of time spent in state q , W_q

$$W_q = \frac{1}{T} \sum_{t=1}^T w_{q_i,t} \quad \text{Eq. 5.16}$$

and emission probability $f_{q_i}(\mathbf{x}|\theta_{q_i})$.

$$w_{q_i,t} = \frac{W_q f_{q_i}(x_t|\mu_{q_i}, \sigma_{q_i})}{\sum_{i=1}^Q W_{q_i} f_{q_i}(x_t|\mu_{q_i}, \sigma_{q_i})} \quad \text{Eq. 5.17}$$

As equations 5.16 and 5.17 are recursive, we have to optimize them iteratively. To do this, we indirectly maximize the likelihood function by optimizing an expectation-maximization (EM) criterion function. For the EM criterion, we use the log likelihood function as it allows for an efficient estimation of the model parameters, even in cases where the likelihood function is intractable or difficult to optimize directly. The log likelihood function for determining a sequence of states \mathbf{q} given the observed FRET trajectory \mathbf{x} , is computed as the product of the emission probabilities weighted by the relative occurrence probabilities summed over all trajectories and is given by¹⁵:

$$\log L = \sum_{i=1}^Q \sum_{t=1}^T w_{q_i,t} \log(f_{q_i}(x_t|\mu_{q_i}, \sigma_{q_i})) \quad \text{Eq. 5.18}$$

To optimize the log likelihood function, we employ the Baum-Welch algorithm, also known as the forward-backward algorithm. During each iteration of the Baum-Welch algorithm, an expectation step (E-step) and a maximization step (M-step) are performed. In the E-step, the algorithm calculates the expected values of the hidden states given the observed data and the current parameter estimates. These expected values are then used in the M-step to update the model parameters, aiming to improve the fit between the model and the observed data. The training of the HMM continues until the relative improvement in the log likelihood between consecutive iterations falls below a predefined threshold. In all of our HMM training procedures, we set the convergence threshold to 10^{-9} . To derive the transition density matrix, the HMM needs to be trained on the dataset to be analyzed. This working step can be carried out in two different approaches: the analysis can be carried out either (1) trace-wise or (2) globally. In the first case, the transition probability and emission probability are optimized for each individual trace while, in the second case, one uses a shared single transition probability matrix and parameters for the emission probability for all trajectories together. When analyzing 1-color and apparent FRET traces, we use a local HMM as the exact values of the states can be shifted due to the above-mentioned background contributions.

5.7. Parameters for Hidden Markov Modeling

FRET efficiencies of dynamic DNA Origami structures were conventionally analyzed by HMM using the HMM Pomegranate toolbox written by Jacob Schreiber (2016). Molecules were manually classified as dynamic or static, and time windows were selected for the data analysis. For molecules showing dynamic transitions, we choose an HMM model with two or three states, depending on the designed Origami structures. Start parameters were chosen assuming a self-adapting width μ , and a standard deviation σ of 0.05 with random uniform distribution. Convergence between the experimental data and the fit was assumed, when the change in likelihood during consecutive iterations was less than 10^{-9} . For visualization via TDP, each transition as superimposed as a 2D Gaussian function with a fixed width.

5.8. Evaluation of involved FRET states and interconversion rates

The last step involves the visualization of determined rates, i.e. dwell times, and states determined from the FRET and / or normalized intensity traces. We employed so called transition density plots (TDPs), which depict each transition that was identified by the HMM algorithm or the Deep-LASI state classifier in the recorded time traces as a single event in a 2D diagram. The diagram, hence, depicts and links the FRET value before and after an identified transition visually. In the case of 1-color data, we normalized the traces between the minimal and maximal value of observed counts of all measured single traces. The TDPs were generated as described by McKinney et al., i.e. all transitions are depicted as summed up two-dimensional Gaussian functions with an amplitude equal to the total number of transitions and a fixed variance of 0.0005.¹⁴

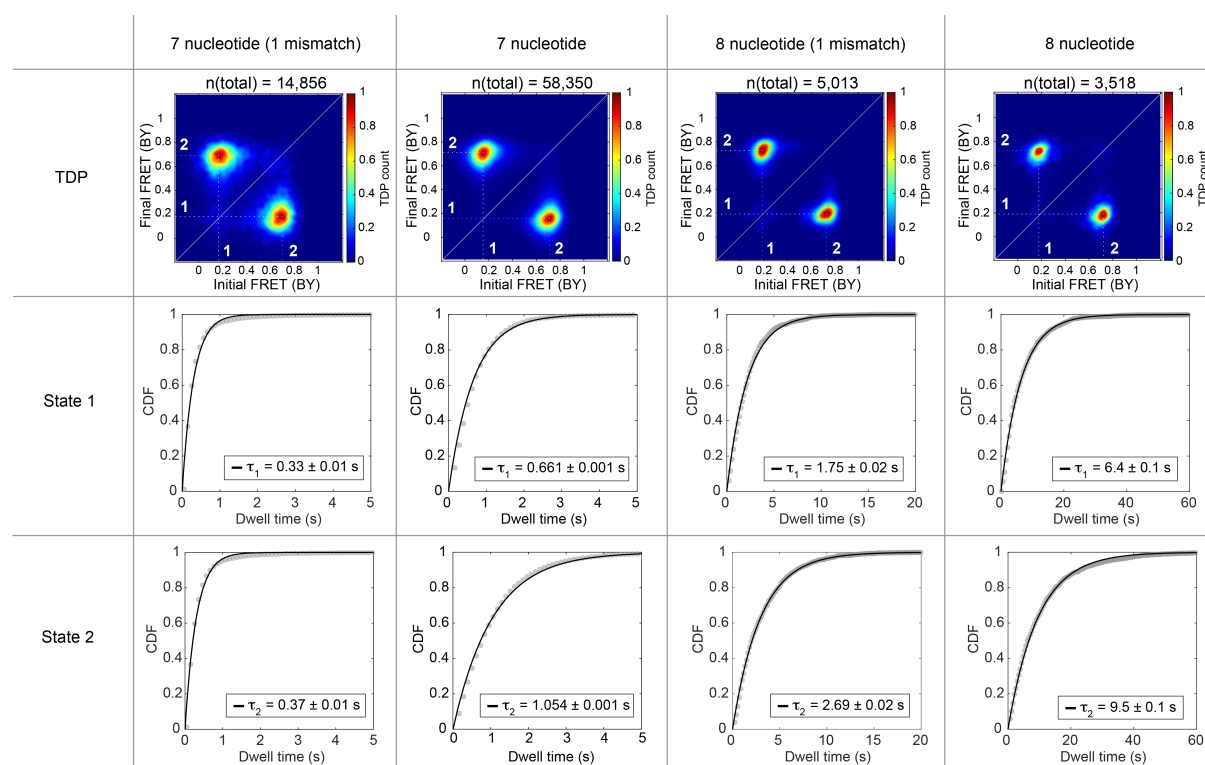
SUPPLEMENTARY NOTE 6: DETAILS OF DEEP-LASI ANALYSES

6.1 Results for the three-color, two-state DNA origami structure with different binding site lengths

The three-color DNA origami structures were measured with four different lengths of complementary DNA for the two binding sites. The two binding sites contained the identical DNA sequence and lengths. The dwell time distributions determined from the state classifiers of Deep-LASI for the different three-color DNA origami structures are shown in [Supplementary Figure 6.1](#). The same analysis workflow was followed for each sample: a fully automated categorization and prediction of state occupancy in traces labeled as ‘dynamic’ were performed with Deep-LASI followed by a manual selection of the different states and fit to a mono-exponential function:

$$f(x) = 1 - e^{-b \cdot x} \quad \text{Eq. 6.1}$$

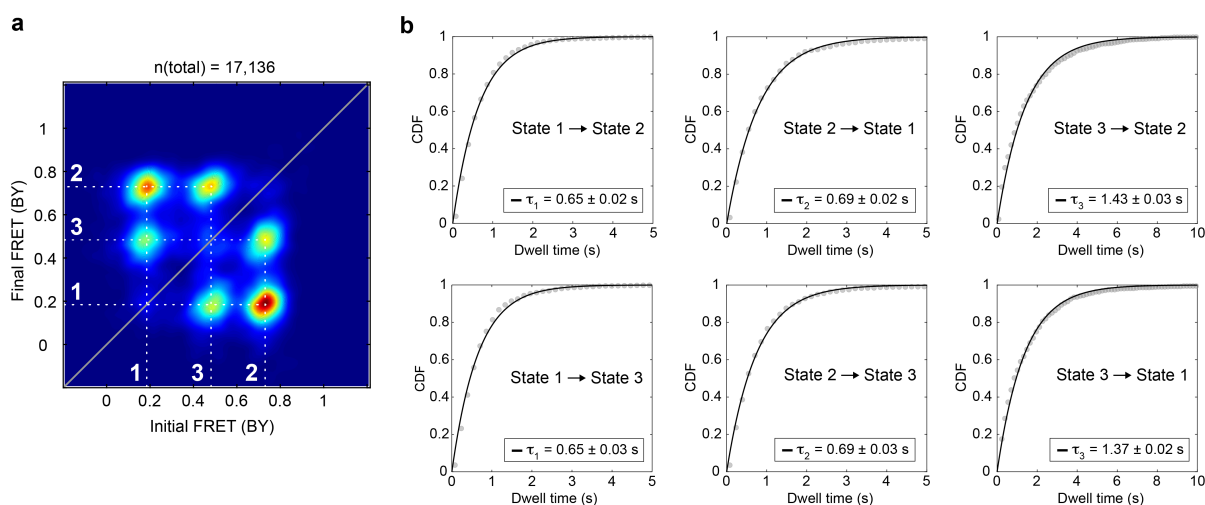
These experiments confirm that Deep-LASI is capable of extracting mono-exponentially distributed dwell times over a large range of kinetic rates.



Supplementary Figure 6.1: Dwell-time distributions of the three-color, two-state DNA origamis with different binding site lengths. Each row corresponds to a specific state and each column depicts the TDPs (top) and dwell-time distributions (middle, bottom) extracted from the uncorrected blue-yellow transition density plots and fitted with a mono-exponential for each binding site length. The errors on the dwell times are the 95% confidence intervals returned by the fitting procedure (estimated from the Jacobian matrix).

6.2 Kinetics of the three-color, three-state DNA origami.

From the three-color, three-state DNA origami with 7 nt binding strands at positions 6 and 12 o'clock and a 7.5 nt complementary binding strand at 9 o'clock. Three populations were extracted automatically from the traces identified by Deep-LASI as dynamic. The dwell-time distributions of all 6 populations observed in the blue/yellow TDP plot (Figure 5c) were extracted manually and fit with an exponential function (Supplementary Figure 6.2). The dwell times of each state are in excellent agreement with the two-color, three-state DNA origami sample (Supplementary Figure 6.3), indicating that the additional blue dye in close proximity of state 2 does not influence the kinetic rates.



Supplementary Figure 6.2: Dwell-time distributions of the 3-color 3-state DNA origami. (a) The blue/yellow transition density plot and (b) the dwell-time distributions extracted from the BY-TDP and fit using a mono-exponential. The errors on the dwell times are the 95% confidence intervals returned by the fitting procedure (estimated from the Jacobian matrix).

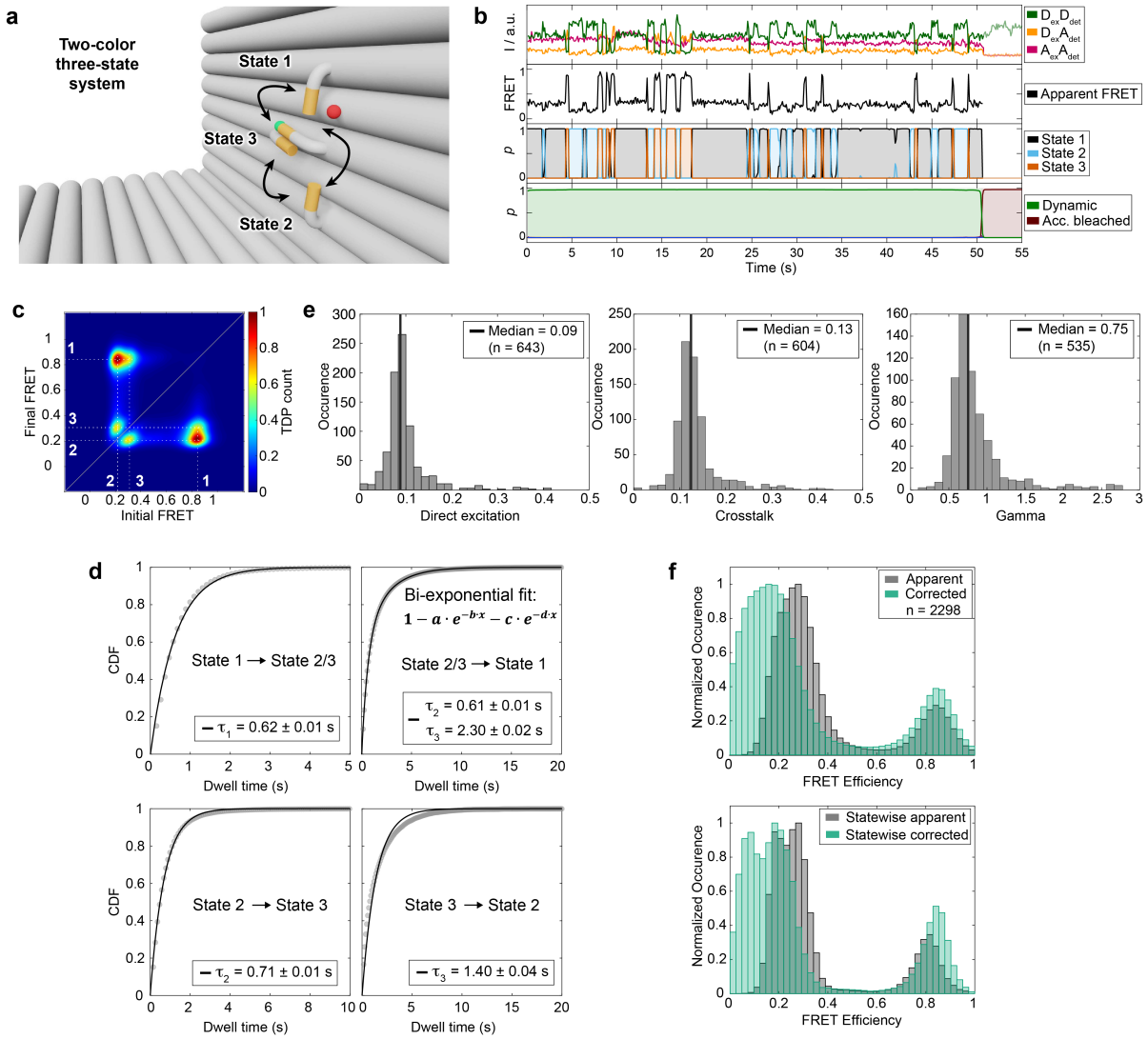
6.3 Results for the two-color, three-state DNA origami structure

Next, we tested the performance of Deep-LASI on a more complex, two-color, multi-state system by introducing a third binding site on the DNA origami ([Supplementary Figure 6.3a](#)) and increasing the average transition rates. In contrast to the two-state system described above, State 1 and State 2 at the 6 o'clock and 12 o'clock positions are now characterized by 7 nt binding sites in the three-state DNA origami. The added State 3 at 9 o'clock has a 7.5 nt overhang. In the example trace shown in [Supplementary Figure 6.3b](#), Deep-LASI extracts the dynamic section and identifies all transitions between the three states summarized in the TDP of apparent FRET efficiencies ([Supplementary Figure 6.3c](#)). As expected, the FRET efficiency of state 1 (0.83) and state 2 (0.21) do not change significantly compared to the two-state system. In addition, a third state with an apparent FRET efficiency of 0.31 is observed. However, as states 2 and 3 show a similar distance to the acceptor, the states and thereby the transitions are not easily separable. When looking at the dwell-time distributions, the transition out of state 1 is not affected by the degeneracy of states 2 and 3. However, the transition rates from state 2 or state 3 to state 1 differ significantly due to the different binding site lengths and can only be extracted using a bi-exponential fit ([Supplementary Figure 6.3d](#)):

$$f(x) = 1 - a \cdot e^{-b \cdot x} - c \cdot e^{-d \cdot x} \quad \text{Eq. 6.2}$$

From the TDP, we can also extract the transitions between states 2 and 3. The transition from state 2 to state 3 can be well described by a mono-exponential distribution whereas the reverse transition from state 3 to state 2 has a second component due to the difficulties of clearly separating the different states.

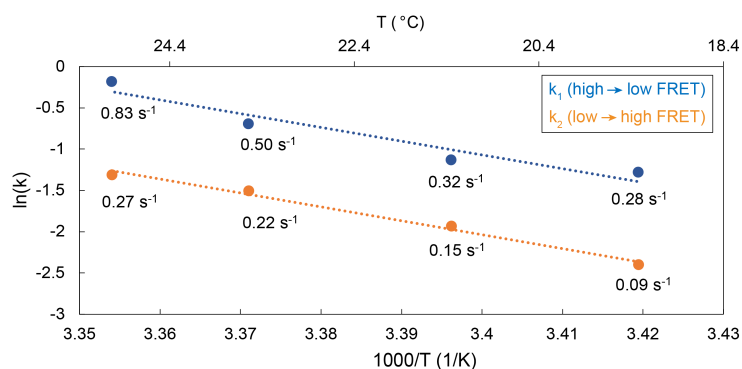
From the single molecule trajectories, Deep-LASI also extracts the regions of the trace that can be used for determining the different correction factors. The FRET correction factor distributions determined by Deep-LASI are shown in [Supplementary Figure 6.3e](#) and are consistent with the correction factors of the two-state DNA origami dataset shown in [Figure 3f](#). The framewise apparent smFRET histogram is shown in [Supplementary Figure 6.3f](#) (top, gray). In this histogram, states 2 and 3 merge into one degenerate state (0.27) due to heterogeneous broadening of the two populations. After correction ([Supplementary Figure 6.3f](#), top, orange), the degeneracy is decreased and the low-FRET peak broadens. However, they are still not clearly separable. It is only after using the state-label information, which allows us to average the state FRET efficiencies that the two low-FRET populations become distinguishable and the individual FRET populations observed ([Supplementary Figure 6.3f](#)).



Supplementary Figure 6.3: Analysis of 2-color, 3-state DNA origami measurements. (a) Zoom-in of the L-shaped DNA origami structure with three binding sites. FRET is expected between a high FRET state 1 (12 o'clock), a low FRET state 2 (6 o'clock), and an intermediate FRET state 3 (9 o'clock). (b) A representative single molecule intensity trace and FRET trajectory. The upper panel shows the intensity in the yellow and red channels after yellow excitation and the red intensity after red excitation. The middle panel shows the corresponding FRET efficiencies for the dye pair. The third and fourth panels show the output of the Deep-LASI analysis for state-transition and trace classification respectively. (c) The TDP of the apparent FRET efficiency states are shown. Interconversion between three conformations with apparent FRET efficiencies of 0.21, 0.31 and 0.83 are observed. The three states are labeled in white. Total number of transitions: 174,697. (d) Exponential fits of the dwell time distributions for all states are plotted. The transitions from state 2 and 3 to state 1 were pooled together due to the high overlap and fit with a bi-exponential function. While the dwell time of state 2 in the bi-exponential fit is close to the dwell time extracted from the single population (state 2 to state 3), the dwell time of state 3 is significantly overestimated compared to the single population of transitions from state 3 to state 2. The errors on the dwell times are the 95% confidence intervals returned by the fitting procedure (estimated from the Jacobian matrix). (e) Correction factors for direct excitation, crosstalk and gamma extracted by Deep-LASI. (f) *top* Frame-wise weighted state-wise smFRET histograms of apparent and accurate smFRET efficiencies. A broadening of the low-FRET population is observed as the correction of the FRET efficiency begins to lift the degeneracy. *bottom* Plotting the frame-wise-weighted statewise smFRET histograms of apparent and accurate FRET efficiencies improves the contrast. Three peaks are now observable with corrected FRET efficiencies of 0.09 and 0.84 (in line with the two-state system), and a new third state at 0.19.

6.4 Kinetics as a function of Temperature

To investigate the influence of temperature on the binding kinetics, we used the two-color two-state L-shaped origami structure with minor changes (exchanged staple strands are noted with asterisks in [Supplementary Table 7.2](#)). Single molecule dynamics were measured between 19.3 °C and 25.0 °C. The transitions rates between state 1 and state 2 are given in [Supplementary Figure 6.4](#). The rates change roughly by a factor of 2 per 2 °C.



Supplementary Figure 6.4: Binding kinetics as a function of temperature. An Arrhenius plot of the transition rates between state 1 and state 2 of the two-color, two-state L-shaped DNA-origami structure. Linear fits (dotted lines) are shown to guide the eye.

6.5 Analysis of previous published 2-color Hsp70 Ssc1 using Deep-LASI

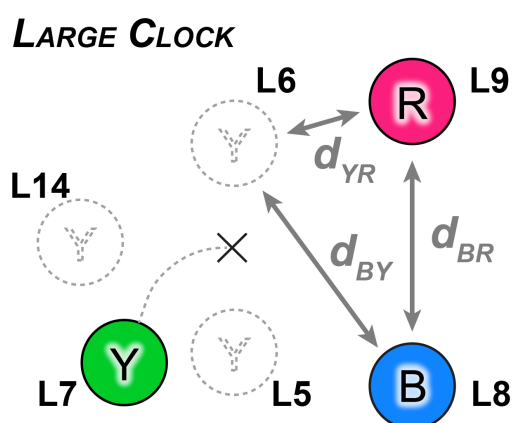
To test Deep-LASI on single-molecule FRET data from proteins, we reanalyzed data that we published previously¹⁶. These data were collected of proteins that were encapsulated in ~200 nm liposomes. Due to vesicle encapsulation, a photostabilization buffer could not be used. Hence, the protein data on this system had a lower signal-to-noise ratio than we typically had with the photostabilized DNA origami structures. In [Supplementary Figure 6.5](#), we show examples of individual traces that were evaluated similarly or differently by the user and Deep-LASI. From these comparisons, we see that: 1) the user and Deep-LASI agree on traces that have sufficient SNR and otherwise show no anomalies, 2) manually selected dynamic traces with erratic intensities after photobleaching or broad FRET distributions tend to be categorized by Deep-LASI as noisy rather than dynamic, 3) Deep-LASI tends to include short traces that are trashed in the manual analysis and 4) traces that show clear features of overestimated background, multiple bleaching steps or other anomalies are discarded by both Deep-LASI and the user.



Supplementary Figure 6.5: Representative 2-color SSC1 traces with disagreements between manual and Deep-LASI selection. (a) Two true positive examples of manual and Deep-LASI selection agreeing on the validity of dynamic sections. The state trajectory (red) and confidence levels of the state transition classifier are shown in the two lower panels. (b) Two example traces that were included in the manual analysis but classified by Deep-LASI as noisy. The user could include these traces by lowering the confidence threshold of the ‘dynamic’ category. (c) Two examples of short traces classified as ‘dynamic’ by Deep-LASI but not selected manually. The left panel shows a valid dynamic trace, which was likely missed during evaluation. The right panel shows an apparent false positive classification by Deep-LASI. The low confidence of the state transition classifier (lower panel) allows this trace to be easily excluded the user. (d) True negative examples excluded from further analysis by both manual evaluation and Deep-LASI. The left panel shows a trace with overestimated background correction and artifact prediction. The right panel shows a trace two bleaching steps of the donor and aggregate prediction.

SUPPLEMENTARY NOTE 7: DNA SEQUENCES

Here, we describe the details of the L-shaped DNA origami structures. The structures were previously published by Tinnefeld et al.^{17,18}. As a scaffold, we used the p8064 scaffold derived from M13mp18 bacteriophages. An overview of all designed DNA origami structures including name, the strand IDs of the introduced modified staple strands as well as the binding sites is given in [Supplementary Table 7.1](#). The unlabeled staple strands are specified in [Supplementary Table 7.2](#), staple strands with biotin modifications for surface immobilization are listed in [Supplementary Table 7.3](#) and staple strands with fluorescent modifications for single-molecule FRET are summarized in [Supplementary Table 7.3](#).

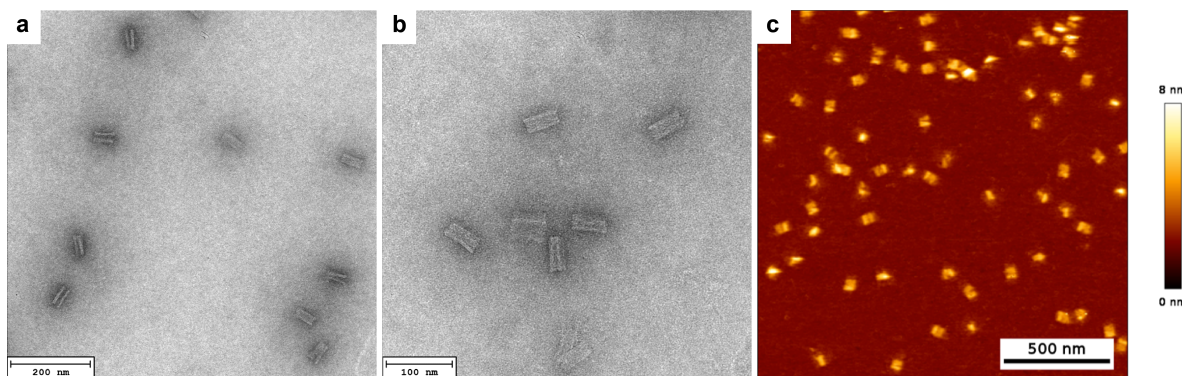


Supplementary Figure 7.1: Schematic of the replacement staple strands forming the 3 state, 3color FRET clock on the L-origami. The strands either carry one of the three fluorophores (L7, L8, and L9) or represent a binding site at 6, 9 and 12 o'clock (L5, L14 and L6 respectively).

The L-shaped DNA origami structures are made of 252 ssDNA staple strands annealed to a circular complementary ssDNA scaffold strand of 8064 nucleotides. The three fluorophores ATTO488, Cy3b and ATTO647N are introduced into the structures, by replacing the unlabeled ssDNA strands L7, L8 and L9 ([Supplementary Figure 7.1](#)) with strands containing the appropriate label ([Supplementary Table 7.4](#)). Binding sites for the L7-attached tether strands consisting of different lengths are introduced at position L5 and L6 for the 2 state systems with low and high FRET values and different binding rates, even with identical sequences. We refer to the binding site for staple strand L5 as 6 o'clock and staple strand L6 as 12 o'clock. For generating a 3 state FRET system, an

additional binding site was introduced on staple strands L14 at 9 o'clock. In addition, for the implementation of the 9 o'clock binding site, the staple strands L12 and L13 are replaced by L12-13-I, L12-13-II and L12-13-III ([Supplementary Table 7.4](#)). All samples share biotinylated attachment sites at positions L1-L4 ([Supplementary Table 7.3](#)).

After folding and purification of the origami samples, the correct folding was confirmed via transmission electron microscopy (TEM) and atomic force microscopy (AFM) as shown in [Supplementary Figure 7.2](#). They form compact structures of roughly $60 \times 20 \times 20 \text{ nm}^3$.



Supplementary Figure 7.2: Structural characterization of L-shaped DNA origami structures. (a-b) Transmission electron microscopy and (c) atomic force microscopy images indicated the efficient folding of the L-shaped DNA origami structures. Color table: Dark Gold.

AFM imaging was performed on a NanoWizard® 3 ultra AFM (JPK BioAFM AG/Bruker; Germany) in solution using 1x folding buffer. The DNA origami structures were immobilized on a freshly cleaved Mica surface (Quality V1, Plano GmbH; Germany) by Ni^{2+} ions, which were incubated on the Mica plate for 5 minutes with a 10 mM NiCl_2 solution. Afterwards, the mica was washed three times with ultra-pure water and dried by pressurized air. 10 μL of 1 nM DNA origami solution were then added and incubated for 5 minutes. Measurements were performed with a USC-F0.3-k0.3-10 cantilever (Nano World; Switzerland).

TEM imaging of the L-Origami structures was carried out on Ar-plasma cleaned TEM grids (Formvar/carbon, 400 mesh, Cu, TedPella, Inc.; USA). The DNA origami structures were stained with a 2 % uranyl formate solution. The imaging was performed on a JEM-1100 microscope (JEOL GmbH; Japan) with an acceleration voltage of 80 kV.

Supplementary Table 7.1. The applied nomenclature used for the designed L-shaped DNA origami structures with the corresponding staple strand IDs that carry the fluorescent dyes or the attachment of the pointer. The laser excitation scheme for the 3cFRET B-Y-R samples involves excitation at 488, 561 and 640 nm.

#	Name	Blue	Yellow	Red	Binding sites			Replaced
1	BYR-Pos6/12-6.5nt	488-L9	561-L7	640-L8	6.5nt-L5	6.5nt-L6	---	---
2	BYR-Pos6/12-7nt	488-L9	561-L7	640-L8	7nt-L5	7nt-L6	---	---
3	BYR-Pos6/12-7.5nt	488-L9	561-L7	640-L8	7.5nt-L5	7.5nt-L6	---	---
4	BYR-Pos6/12-8nt	488-L9	561-L7	640-L8	8nt-L5	8nt-L6	---	---
5	BY-Pos6/12-7.5nt	488-L9	561-L7	---	7.5nt-L5	7.5nt-L6	---	---
6	BR-Pos6/12-7.5nt	488-L9	---	640-L8	7.5nt-L5	7.5nt-L6	---	---
7	YR-Pos6/12-7.5nt	---	561-L7	640-L8	7.5nt-L5	7.5nt-L6	---	---
8	YR-Pos6/9/12	---	561-L7	640-L8	7nt-L5	7nt-L6	7.5nt-L14	L12-13-I-III
9	BYR-Pos6/9/12	488-L9	561-L7	640-L8	7nt-L5	7nt-L6	7.5nt-L14	L12-13-I-III

Supplementary Table 7.2. Unmodified staple strands used for the L-shaped DNA origami structure given from the 5' to 3' end. All oligonucleotides were purchased from Integrated DNA Technologies. *In the origami used for the measurements shown in **Supplementary Figure 6.4**, the staple strands L141, L153 and L165 were replaced with the sequences highlighted with the asterisks at the end of the table.

Staple ID	Sequence (5' to 3')
L1	ATCCAGAACAATATTAGTCCATCAGGAACGGT
L2	CGTGCCTGTTCTTCGCATCCAGCGCCGGGTTA
L3	ATAATCAGAAAAGCCCAACATCCACTGTAATA
L4	CATAGGTCTGAGAGACAAATCGTCGAATTACC
L5	AGAAACAGCTTTAGAAAGGAAGAAAAATCTACGATTTTAAGCATATAAC
L6	GCACCCCTCCGTCAGGTACGTTAGTAAATGAATAGTTAGCGTCAATCAT
L7	ACGATAAACCTAAAACAAAGAATACACTAAAACATTACCCAACAAAGC
L8	TGCTCATTCTTATGCGTTAATAAAACGAACTATATTCATTGGCTTTTG
L9	AAGGGAACCGGATATTCACTCATCTTTGACCCGTAATGCCATCGGAAC
L10	CGGAATCTCAGGTCTGTTTTAAATATGCATGCGAACGAATCATTG
L11	TGAATTACCAGTGAATGGAATTACGAGGCATATAGCGAGAGAATCCCC
L12	CATTATACGGTTTACCATAACCCTCGAAATACAATGTTTAAACAGGG
L13	ATTCATATCAGTGATTTGGCATCAGGACGTTGTAACATAAAACCAGACG
L14	TAATAAGAAGAGCCACCCTTATTAGCGTTTGCCATTCAACAATAGAAA
L15	GGCACCAAAACCAAAAGTAAGAGCAACACTATAGCAACGTAAATCGCC
L16	ATAAAAATATCGCGTTTCCTTTTGATAAGAGCTATAT
L17	TACCAGTAACGCTAACAGTTGCTATTTTGCACCCCATCCT
L18	GAGGGTAGTTGCAGGGTGCTAAACAACCTTACGCCTGGAAAGAG
L19	AGAGCCGCAAACAAATGAGACTCCTCAAGAGATTAGCGGGCAGTAGCA
L20	AGTTGATTAGCTGAAAAGAGTACCTTTAATTGTTAATTCCGGACCATAA
L21	TCGATAGCAGCACCGTAAAATCACGTTTTGCT
L22	AAAGACAAATTAGCAAGTCACCAATGAAACCA
L23	ATATTCACCGCCAGCATTGACAGGCAAAATCA
L24	TTTTCCCTTACACTGGTTGC
L25	CTCCAATCGTCTGAAATTTT
L26	TTTTTGCTGAGTAGAAGAA
L27	TTTTCCCGACTTACAAAATAAACAGTTTT
L28	ATACGCAAAGAAAATTATTCATTAAGGTGAATTTT
L29	TTTTCTTTACAAACAATTCG
L30	TTTTAAGTTACCAGGGTAATTGAGCTTTT
L31	TTTTTAAACGATGCTGATGG
L32	TTTTCAGGGTGGTTTTTCTT
L33	ACAAAGTATGAGGAAGCTTTGAGGACTAAAGATTTT
L34	CCGAATCTAAAGCATCTTTT
L35	AGATGAAGGGTAAAGTTTTT
L36	TTTTCGCAAATGGTCAATAAACCATTAGATGC
L37	TCGAAGATGATGAAACTTTT
L38	AGAGCAAATCCTGTCCAGATACCGACAAAAGGTAATTTT
L39	TTCCGGAATCATAATTTTTT

L40	TTTTTGGATTATTTACAGAA
L41	TGCGGCCAGAATGCGGTTTT
L42	TCAGCAGCAACCGCAATTTT
L43	TTTTAGAGCGGGAGCTAGAT
L44	TTTTGCTAATATCAGAGAGATAACCCCGCCACCGCG
L45	ACCTCGTCATAAACATTTTT
L46	TTTTTTGAGGGGACGACGAC
L47	TTTTAACAGTACCTTTTACA
L48	TTTTTGGCCTTCCTGTATAA
L49	TTTTGGCGCATAGGCTGGCTAACGGTGTTAAATTGT
L50	TCACCGGAAGCATAAAATTTT
L51	TTTTTATCATCGCCTGAACAGACCATTTT
L52	TTCATAGGGTTGAGTGTTTT
L53	TTTTTAGCCCGGAATAGCCTATTTCTTTT
L54	TTTTCCCTCAGAGCCACCACCCTCAGAAAGCGCTTA
L55	TAGTAATAACATCACTTTTT
L56	TTTTTTGTTCCAGTTTGAACAAGA
L57	TTTTCGGGCCGTTTTACGG
L58	TTTTATTGCTGAATATAATACATTTTTTTT
L59	TTTTAGTAATTCAATCGCAAGACAATTTT
L60	TTTTGAATGCCAACGGCAGC
L61	CAGATGAATATACAGTTTTT
L62	TTTTCCATATTATTTATCCCAATCCAAAGTCAGAGA
L63	TTTTTTATCACCGTCACAGCGTCAGTTTT
L64	TTTTCTTTTTTACAACGGAGATTTGTTTT
L65	TTTTGTGTAGGTAAAGATTC
L66	TTTTTTTTTTTTAAAAC TAG
L67	TTTTGATTAAGACGCTGAGA
L68	TTTGCGTATTGGGCGCTTTT
L69	ATTATAGCGTCGTAATAGTAAAATGTTTTTTT
L70	TAGTCAGAAGCAAAGCGGATTTT
L71	TTTTTAGACTGGCATCAGTTGAGATTTTTT
L72	CATAATAATTCGCGTCTTTT
L73	ATATATATAAAGCGACGACATCGGCTGTCTTTCCTTATCATTTTTT
L74	AAAACGGTAATCGTTTTTTTTT
L75	ACAAATTATCATCATATTTT
L76	TTTTTTCCTGATTATCACGT
L77	TTTTCATATAAAAAGAAAGCCGAACATTTT
L78	TTTTGTGTAAAGCCTGGCGG
L79	TTTTAAACATCAAGAAAAAA
L80	AATGCAATAGATTAAGGGCTTAGAGCTTATTTT
L81	ACATAGCGATAGCTTATTTT
L82	TTTTTTGCATCAAAGCCTGAGTAATTTT

L83	CTGATAGCCCTAAAACCTTTT
L84	GAAAGGAGCGGGCGCTAGGTTTT
L85	TTTTGCCTCAGAGCATAAAGAAAATTAAGCAATAAATTTT
L86	TTTTACTGTAGCCTCAGAACCGCCATTTT
L87	TTTTACCGTTCCAGTAAGCGTCATACATGGCTTCAGTTAAT
L88	AGTGTGCTGCAAGGCGTTTT
L89	TTAATTA AACCATACATACATAAAGGTGGCAATTTT
L90	CCGTCATCTGCCAGTTTTT
L91	TTTTTTAGGAATACCACAGTAGTAATTTT
L92	TTTTATTGGGCTTGAGATGGCCAGAACGATT
L93	TTTTGAACA ACTAAAGGAACACTGATTTT
L94	TTTTACTAGAAAAAGCCTGTT
L95	TTTCGACTTGATCGAGAGGGTTGATATAAGTATTTT
L96	ATTTAGAAGTATTAGATTTT
L97	TTTTACCTTGCTGAACCAGG
L98	TTTTTCCAAGAACGGGTGCGAACCTTTTT
L99	TTTTACGCATAATGAGAATAGAAAAGTTTT
L100	TTTTAGAACGCGAGAAAACCTTT
L101	TTTTGTTTCGTCACCAGTACTGTACCGTAAT
L102	CATGTTTACCAGTCCCTTTT
L103	TTTTGGAATTTGTGAGAGAT
L104	TTTTATTAAGTTGGGTACGC
L105	TTTTGGAACCTAAGTCTCTGAATTTTTTTTTTT
L106	TTTTATCGCCATTA AAAATA
L107	GAGCCGATATAACAACAACCATCGCCCTTTTTTTT
L108	CGGCCTCGTTAGAATCTTTT
L109	TTTTGCGCTGGCAAGTGTAG
L110	TAGTTGCCAGTTGCGGGAGGTTTTGAAGATCAATAA
L111	ATGGCTACAATCAACTGAGAGCCAGCAGCAAATGAAAAACGAACCTAATGCGCTTGGCAGA
L112	TCATCAACAAGGCAAATATGTACCCCGGTTG
L113	TTCAAATTTTTAGAAAAAACAGGAGCAAACAAGAGAATCGATGAAGGGTGAGATATTTTA
L114	CAACTAATGCAGACAGAGGGGCAACTACTG
L115	GTA CTATGGTTGCTTTTTTTAGACACGCAAATT
L116	TGTAGCTCAACATTTACCCTCGAAAGAC
L117	ATCAAAAAGTCATAAAACGGAACAACATTATCAACTTTAGTAGAT
L118	AACGTCAATAGACGGGGAATACCCAAAAGAACAAGACTCCGTTTTTAT
L119	GGAGGGAAGAGCCAGCAATCAGTAGCGACAGACCAGAACCGCCTC
L120	AGCGAACCAGAAGCCTGGAGAATCACAAAGGCTATCAGGT
L121	GCCCCCTGGTGTATCACCGTACTC
L122	TACAGGCATTA AATTAACCAATAGGAACGCCATCAAAGTCAATCAGAATTAGCCTAAATCG
L123	TATTTTTGAGAGATCTGCCATATTCCTCTACTCAATTGA
L124	CATTGCCTGAGAGTCTTTATGACCATAAATCATTTCATTT
L125	CCAGCCAGCTTCCGGGTAATGGGGTAACAAC

L126	ATCGGCAAAATCCCTTACGTGGACTCCAACGT
L127	CCTGCAGCCATAACGGGGTGTCCAGCATCAGC
L128	GGGCTCTTCGCTATTACGTTGTACCTCACCG
L129	ACCTCATGCCCTCATTTTCTGTATGGGATTTAGTTAAAGCAGCTTGA
L130	GTTGTACCACCCTCATAAAGGCCGGAGACAG
L131	TCTTTAGGCTGAATAATGCTCATTAGTAACAT
L132	CTCTCACGGAAAAAGAACGGATAAAAACGACG
L133	ACGCCAGATGACGGGGCGCCGCTAGCCCCAGC
L134	TTAATTTTCATGTTCTATAACTATATGTAATGCTGATGTCAATAGAATCCTTGACAAAATT
L135	TTTCATCGAATAATATCCAGCTACAATACTCCAGCAATTTCTTTACAG
L136	AATAAGTTAGCAAAAACGCAATAATAACGAGAATTTAAAGCCCAA
L137	GACCGTGTGATAAATACAAATTCT
L138	ACAAGAACCGAACCTGATGTTACTTAGCCGGAAAAGACAGCACTACGAA
L139	ATCAAACCTAAATTTCTGGAAGGGCCATATCA
L140	CGCTGGCACCACGGGAGACGCAGAAACAGCGG
L141	GAGAAACATTTAATTTTACAGGTAGAAAAG
L142	CTGCGCGGCTAACTCACAATTCCACACAACATACGAGTACCGGGGCTCTGTGGGTGTTTCAG
L143	CCGAGTAAGCCAACAGGGGTACCGCATTGCAA
L144	AAACGGCGCAAGCTTTGAAGGGCGATCGGTGC
L145	CAAAGAATAAAATACCCAGCGATTATACCAAGCGCGAA
L146	CTTAATTGAGACCGGAAACAGGTCAGGATTAGAGGTGGCA
L147	GCCAGTGCATTGACCCACCGCTTCTGGTGCC
L148	CCAGAATGGAGCCGCCAATCAAGTTTGCC
L149	CCCCCTGCGCCCGCTTAGCTGTTTCTGTGT
L150	GGAAACCAGGCAAAGCGTACATAAGTGAGTGA
L151	AAATCAACACGTGGCATCAGTATTCTCAATCC
L152	AGGAGGTGGCGGATAAGTATTAAGAGGCTAAATCCTCTACAGGAG
L153	GACAGATGGACCTTCATCAAGAGCCCTGAC
L154	CTGAGCCAACGGCTACAGAGGTTCCATT
L155	ATAACCTTATCAACAAAAATTGTATAACCTCC
L156	AAATCAGCTCATTTTTGTGAGCGAATAGGTCA
L157	CACAGACATTTACGGGATCTCCAAAAAAAAGGTTCTTAAAGCCGCTTT
L158	TAATAGTATTCTCCGTGCATTAAATTTTTGTT
L159	CGTTGGTAGTCACGACGCCAGCTGGCGAAAGGGGGATATCGGCCTGCGCATCGGCCAGCTT
L160	CTTCTGACCTAAATTTGCAGAGGCCAGAACGCAATTTACG
L161	GCTGCGCAACTGTTGGCAGACCTATTAGAAGG
L162	AGAACGTTAACGGCGTAATGGGTAAAGGTTTCTTTCGCTCGGTGGTGTCTGGTCTTGCCGTT
L163	TTAGTTTGCTGTTTAGGTCATTTTTGCGGATAGGAAGCCGACTATTA
L164	AATTACATAGATTTTCAATAACGGATTTCGCC
L165	AAACGGGGTTTTGCTACATAACGCCAAAAAAGGCTTGTAATCTTG
L166	TGCGAATAATAATCGACAATGTTTCGGTTCG
L167	TTATACTTAGCACTAAAAAGTTTTGTGCCGCCA
L168	GCCGTCACAATATAAAAGAAACCACCAGAAGGAGCGGACTCGTATTACATTTGTCAAATAT

L169	ATTGCGTTTAAACAACATTTCAATTACCTGAGCAAAAAGGGAGAAACAGGTTTAAGATGATGG
L170	GGAGCCTTCACCCTCAGAGCCACC
L171	CCAGCTTACGGCTGGAAACGTGCCCGTCTCGT
L172	TTCGTAATCATGGTCATCCATCAGTTATAAGT
L173	AACAGAGGTGAGGCGGCAGACAATTTAAAAGGG
L174	TTGAGTAAGCCACCCTCAGAACCG
L175	CAGTATGTTTATTTTGCGAAGCCCTTTTAAATTGAGTTCTGAACA
L176	GCCTGTTTGCTTCTGTTACCTTTTAAACGTAA
L177	ATAACAATCCCTTAGTGAATTTATCAAAAT
L178	CAAAGGGCCTGTCGTGTGGCCCTGAGAGAGTT
L179	CTCAAAATGTCAGAAATGGAAGTTTCACGCGCATTACTTCAACTGGCT
L180	CCCGCCGCGCTTAATGAAAGCCGGCGAACGTG
L181	TTCACCAGGTAGCAATGGCCTTGCTGGTAAT
L182	GTCGAAATCCGCGACCTGCTCCACCAACTTTTAGCATTC
L183	TGATTGCTTTGAATACAAACAGAATGTTTGGA
L184	TTCTGAAACATGAAAGTGCCGGCCATTTG
L185	AACCGTTTCACACGGGAAATACCTACATTTTGACGCTAAACTATCACTTCTTTAACAGGAG
L186	CGTTGAAAATAGCAAGCCCAATA
L187	CTTTTGC GTTATTTCAATGATATTCAACCGTT
L188	AAATCCCGTAAAAAACGTTTTTTGGACTTGT
L189	TATCATTTTGC GG AACATCCTGATATAAAGAA
L190	AAATTATTTGGAAACAGCCATTTCGAAAATCGC
L191	GCAGCAAGCGGTCCACAAGTGTTTTGAGGCCA
L192	CCAACATGACGCTCAATGCCGGAGGAAATACC
L193	TATTTTGTTAAAATTCGGGTATATATCAAAAC
L194	TGTTGCCCTGCGGCTGATCAGATGCAGTGTCA
L195	TGCGGGATAGCAGCGACGAGGCGCAGAGAAAACGGCCGCGTAACGATC
L196	TACCGATAGTTGCGCTTTTTCA
L197	TCAAATCACCATCAATACGCAAGG
L198	GTAAGAATAGTTGAAACTTTTCGAAAACACCGC
L199	ATTGCCCTTCACCGCCCAGCTGCTTGC GTT
L200	AAGCGCATAAATGAAACAGATATAGAAGGCTTAGCAAGCCTTATTACG
L201	GGAATTAGGTAAATTTTCGGTCATAGCCCCACCGGAACCACCACC
L202	GTTTTCCCGTAGATGGCAGGAAGATCGCACT
L203	GCGAGAAAAGGGATGACGAGCACGTATAACGTGCTTTTTACGCTGAAGAAAGC
L204	GGGGCGCGCCCAATTCATAAAGTACGGTGTACGAGAATAGCTTCAA
L205	GAAATTGTTATCCGCTCACATTAATTAATGA
L206	TTTTTTAATGCACGTACAAGTTACCCATTCAG
L207	CAATTCATATAGATAATAAATCCTTTGCCCG
L208	CCTCAGAGCACAAGAAGAAAAGTAAGCAG
L209	CGCTCACTATCAGACGGTCCGTGAGCCTCCTC
L210	GCAGAGGCGAATTATTTTTCATTTGCTATTAA
L211	TTAGAGCTATCCTGAGGCTGGTTTCAGGGCGC

L212	GCCAGTACGTTATAAGGCGTTAAATAAGAATAAACACAAAT
L213	AACGTTATTAATTTTACAACATAATCAGTTGGC
L214	GCCGGGCGCGGTTGCGCCGCTGACCCCTTGTG
L215	CTGCAACAGTGCCACGTATCTGGTAGATTAGA
L216	TAAAGTTTAGAACCGCTAATTGTATCGCGGGTTTAAAGTTTGGCCTTG
L217	GAAACAACGCGGTCGCCGCACAGGCGGCCTTTAGTGACTTTCTCCACGTACAGACGCCAGG
L218	GTCCACTAAACGCGCGGACGGGCAACAGCTG
L219	GGAACCCAAAACATAAAAACAGTTTCAGCG
L220	ATCGGCCTTAAAGAATAAATCAAAAAGAATAGCCCCGAGACCAGTGAGGGAGAGGGGTGCCTA
L221	ACAGTTGAGGATCCCCAGATAGAACTGAAAGC
L222	CCGGAACCGCAAGAAAGCAATAGCTATCTTACTCACAATCCGATTGAG
L223	GCAGTTGGGCGGTTGTCCAGTTATGGAAGGAG
L224	GCCGATTAAGGAAGGGCGCGTAACCACCACA
L225	TGTACTGGTAATAAGTTCAGTGCC
L226	CAAATCGTCAGCGTGGTGCCATCCCACGCAA
L227	TCTTACCATAAAGCCATAATTTAGAATGGTTTAGGGTAGC
L228	AGGCGAAAATCCTGTTGTCTATCACCCCGAT
L229	GCCTAATTATCATATGATAAGAGATTTAGTTAATTTTCAT
L230	TTTTCATCGGCATATTGACGGCACCACGG
L231	CTAGCTGATAAATTAACAGTAGGG
L232	CCCTGAACAAATAAGAAACGCGAGGCGTT
L233	CACATCCTCAGCGGTGGTATGAGCCGGGTCAC
L234	CAGGAAAAACGCTCATAACCAGTAAATTTTTGA
L235	CCACCTCTGTTAGGAAGGATCGTCTTCCAGCAGACGATTATCAGCT
L236	CAAACCCTTTAGTCTTACCAGCAGAAGATAA
L237	GGCTTAGGTTGGGTTAAGCTAATGATTTTCGA
L238	CCGTCGGAGTAGCATTCAAAAACAGGAAGATT
L239	ATGAGTGACCTGTGCAGTTTCTGCCAGCACG
L240	CCGGCAAATCGGCGAAGTGGTGAAGGGATAG
L241	ACAAGAAATAGGAATCCAATAGCAAGCAAATATAGCAGCATCCTGAA
L242	CCATTACCAAGGGCGACATCTTTTCATAGGCAGAAAGAATAGGTTGAG
L243	TGGAGCCGGCCTCCGGGTACATCGACATAAAA
L244	CACTCATGAAACCACCTTAAATCAAGATTGAGCGTCTTTTTGTTT
L245	GTATAAGCAAATATTTTAGATAAGTAACAACG
L246	AGGAAACCGAGGACGTAGAAAAAGTACCG
L247	CGGGAAACGAAAAACCTGATGGTGGTTCCGAA
L248	AGCATGTACGAGAACAAATCCGGTATTCTAAGAACGATTTTCCAGA
L249	ACATTCTGAAGAGTCTCCGCCAGCAGCTCGAA
L250	GGGGTCATTGCAGGCGGGAATTGACTAAAATA
L251	TGCTTTGAGGTTGAATCTCCAAAA
L252	CAGTACCATTAGTACCCAGTGCCCGTATAAATTGATGAATTAAG

*L141	TGCCCTGACGAGAAACATTTAATTTTACAGGTAGAAAAG
*L153	GACAGATGGACCTTCATCAAGAGTAATCTTG
*L165	AAACGGGGTTTTGCTACATAACGCCAAAAAAGGCT

Supplementary Table 7.3. Modified staple strands given from the 5' to 3' end for the L-shaped DNA origami structures used. The biotin was used for surface-immobilization via a biotin/avidin interaction. All oligonucleotides were purchased from Biomers.

ID	Sequence (5' to 3')	Function	Replace
Bio-L1	Biotin -ATCCAGAACAATATTAGTCCATCAGGAACGGT	Attachment Biotin at 5'	L1
Bio-L2	Biotin -CGTGCCTGTTCTTCGCATCCAGCGCCGGGTTA	Attachment Biotin at 5'	L2
Bio-L3	Biotin -ATAATCAGAAAAGCCCAACATCCACTGTAATA	Attachment Biotin at 5'	L3
Bio-L4	Biotin -CATAGGTCTGAGAGACAAATCGTCTGAATTACC	Attachment Biotin at 5'	L4

Supplementary Table 7.4. Modified staple strands given from the 5' to 3' end for the fluorescently-labeled L-shaped DNA origami structures. The complementary docking sequences are highlighted in orange. The docking strands have a three-base extension from the DNA origami structure and the pointer contains nine single-stranded thymine bases plus GC and the docking sequence, highlighted in grey.

ID	Sequence (5' to 3')	Supplier	Function	Replace
488-L8	TGC TCA TTC TXA TGC GTT AAT AAA ACG AAC TAT ATT CAT TGG CTT TTG; X = dT- <i>Atto488</i>	biomers	Lower Label – V1	L8
640-L9	AAG GGA ACC GYA TAT TCA CTC ATC TTT GAC CCG TAA TGC CAT CGG AAC; Y = dT- <i>Atto647N</i>	Eurofines Genomics	Upper Label – V1	L9
561-L7	GGCACAAAACCAAAAGTAAGAGCAACACTATA GCAACGTAAATCGCCTTTTTTTTTCGGGCATTTA - <i>Cy3b</i>	Eurofines Genomics	Pointer - dye at 3'	L7
6.5nt-L6	GCACCCTCCGTCAGGTACGTTAGTAAATGAATAG TTAGCGTCAATCATTTTCAAATGC	Eurofines Genomics	Pos 12 catching site 7 nt <i>mismatch</i>	L6
7nt-L6	GCACCCTCCGTCAGGTACGTTAGTAAATGAATAG TTAGCGTCAATCATTTTCAAATGC	Eurofines Genomics	Pos 12 catching site 7 nt	L6
7.5nt-L6	GCACCCTCCGTCAGGTACGTTAGTAAATGAATAG TTAGCGTCAATCATTTTCAAATGCC	Eurofines Genomics	Pos 12 catching site 8 nt <i>mismatch</i>	L6
8nt-L6	GCACCCTCCGTCAGGTACGTTAGTAAATGAATAG TTAGCGTCAATCATTTTCAAATGCC	Eurofines Genomics	Pos 12 catching site 8 nt	L6
6.5nt-L5	AGAAACAGCTTTAGAAGGAAGAAAAATCTACGAT TTAAGCATATAACTTTCAAATGC	Eurofines Genomics	Pos 6 catching site 7 nt <i>mismatch</i>	L5
7nt-L5	AGAAACAGCTTTAGAAGGAAGAAAAATCTACGAT TTAAGCATATAACTTTCAAATGC	Eurofines Genomics	Pos 6 catching site 7 nt	L5
7.5nt-L5	AGAAACAGCTTTAGAAGGAAGAAAAATCTACGAT TTAAGCATATAACTTTCAAATGCC	Eurofines Genomics	Pos 6 catching site 8 nt <i>mismatch</i>	L5
8nt-L5	AGAAACAGCTTTAGAAGGAAGAAAAATCTACGAT TTAAGCATATAACTTTCAAATGCC	Eurofines Genomics	Pos 6 catching site 8 nt	L5
7.5nt-L14	TGCCATTCAACAATAGAAAATTCATATGGTTTTCA AATGCCTTCAAATGCC	Eurofines Genomics	Pos 9 catching site 8 nt <i>mismatch</i>	L14
L12-13-I	CATTATACCAGTGATTTGGCATCAGGACGTTGTA ACATAAACCAGACG	Eurofines Genomics	Replacement for Pos 9	L12 and L13
L12-13-II	TAATAAGAAGAGCCACCCTTATTAGCGTT	Eurofines Genomics	Replacement for Pos 9	L12 and L13
L12-13-III	TTACCCATAACCCTCGAAATACAATGTTTAAACA GGG	Eurofines Genomics	Replacement for Pos 9	L12 and L13

SUPPLEMENTARY NOTE 8: STATISTICS SINGLE-MOLECULE DATA

The following section summarizes the results for all Hsp70 and origami datasets, newly designed, presented, and analyzed in this work. For each dataset, the total number (#) of recorded traces, the confidence interval, the number of dynamic traces, the retrieved number of states, and dwell times are specified. For the origami datasets, we refer to State 1 as the 12 o'clock position and State 2 as the 6 o'clock position for the 2-state systems. In 3-state systems (as presented in Figure 5), we refer to States 1, 2, and 3 as the 12 o'clock, 9 o'clock, and 6 o'clock positions, respectively. For the Hsp70 datasets, we refer to State 1 as the undocked conformation (low FRET) and State 2 as the docked conformation (high FRET). More details can be found together with the raw data on Zenodo [<https://zenodo.org/record/7561162>].

Supplementary Table 8.1. Statistics summarizing the single-molecule results obtained for the origami and Hsp70 datasets. N.A.: not applicable; max: highest confidence of all possible classes.

Fig.	Name	Binding Site	# of Traces	Confidence	Frame time (ms)	# of Frames	# of Dynamic Tracers	# of States	Dwell time t_1 (s)	Dwell time t_2 (s)	Dwell time t_3 (s)
2	1c origami 2 states	8 nt / 1 MM	7448	70 %	52.2	998	2510	2	1.75	2.65	N.A.
3	2c origami 2 states	8 nt / 1 MM	6100	70 %	52.2	1998	1499	2	1.76	2.64	N.A.
4	3c origami 2 states	7 nt / 1 MM	5731	20 %	34.2	2997	482	2	0.33	0.40	N.A.
		7 nt	5093	70 %	37.5	2997	1885	2	0.66	1.05	N.A.
		8 nt / 1 MM	2545	70 %	52.5	2397	581	2	1.75	2.69	N.A.
		8 nt	8097	70 %	152.2	1797	1545	2	6.41	9.54	N.A.
5	3c origami 3 states	7 nt @ 6 / 12h	7990	70 %	52.5	2997	586	3	0.65	0.69	~1.40
		8 nt / 1MM @ 9h									
6i	2c origami 2 states	5 nt	95	N.A.	N.A.	N.A.	95	2	0.0045	0.0063	N.A.
		6 nt	104	N.A.	N.A.	N.A.	104	2	0.015	0.026	N.A.
		7 nt / 1 MM	99	N.A.	N.A.	N.A.	99	2	0.14	0.23	N.A.
		7 nt	97	N.A.	N.A.	N.A.	97	2	0.84	1.62	N.A.
		6 nt \leftrightarrow 7 nt	11	N.A.	N.A.	N.A.	11	2	0.017	1.524	N.A.
		7 nt \leftrightarrow 7 nt / 1 MM	102	N.A.	N.A.	N.A.	102	2	0.79	0.22	N.A.
		7 nt / 1 MM \leftrightarrow 7 nt	21	N.A.	N.A.	N.A.	21	2	0.15	1.57	N.A.
Fig.	Name	Glycerol concentration (%)	# of Traces	Confidence	Frame time (ms)	# of Frames	# of Dynamic Tracers	# of States	Dwell time t_1 (s)	Dwell time t_2 (s)	Dwell time t_3 (s)
6a-b	3c origami 2 states 8 nt / 1 MM	1 %	1167	50 %	52.2	2397	176	2	1.65	2.45	N.A.
		2.5 %	1087	30 %	52.2	2397	150	2	1.52	2.23	N.A.
		5 %	1814	50 %	52.2	2397	382	2	1.38	2.02	N.A.
		10 %	1040	50 %	52.2	2397	120	2	1.11	1.52	N.A.
		15 %	1006	50 %	52.2	2397	204	2	0.96	1.23	N.A.
		20 %	1295	50 %	52.2	2397	101	2	0.85	1.11	N.A.
30 %	1207	30 %	52.2	2397	101	2	0.62	0.85	N.A.		
Fig.	Name	ADP concentration (M)	# of Traces	Confidence	Frame time (ms)	# of Frames	# of Dynamic Tracers	# of States	Dwell time t_1 (s)	Dwell time t_2 (s)	Dwell time t_3 (s)
6e-f	Hsp70 SSC1	100 nM	12319	max	32.5	2000	466	2	1.13	0.93	N.A.
		5 μ M	9246	max	32.5	2000	155	2	0.92	0.75	N.A.
		100 μ M	5035	max	32.5	2000	258	2	0.64	0.83	N.A.
		1 mM	3534	max	32.5	2000	156	2	0.58	0.79	N.A.

SUPPLEMENTARY REFERENCES

- 1 Abadi, M. *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems. *ArXiv*, doi:10.48550/arXiv.1603.04467 (2015).
- 2 Tang, W. *et al.* Omni-Scale CNNs: a simple and effective kernel size configuration for time series classification. *ArXiv*, doi:10.48550/arXiv.2002.10061 (2022).
- 3 He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *ArXiv*, doi:10.48550/arXiv.1512.03385 (2015).
- 4 He, K., Zhang, X., Ren, S. & Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile*, 1026-1034, doi:10.1109/ICCV.2015.123 (2015).
- 5 Fukushima, K. Cognitron: a self-organizing multilayered neural network. *Biol. Cybern.* **20**, 121-136, doi:10.1007/BF00342633 (1975).
- 6 Smith, S. L., Kindermans, P.-J., Ying, C. & Le, Q. V. Don't Decay the Learning Rate, Increase the Batch Size. *ArXiv*, doi:10.48550/arXiv.1711.00489 (2018).
- 7 Thomsen, J. *et al.* DeepFRET, a software for rapid and automated single-molecule FRET data classification using deep learning. *eLife* **9**, e60404, doi:10.7554/eLife.60404 (2020).
- 8 Schreiber, J. Pomegranate: fast and flexible probabilistic modeling in python. *J. Mach. Learn. Res.* **18**, 1-6, doi:10.48550/arxiv.1711.00137 (2018).
- 9 Basden, A. G., Haniff, C. A. & Mackay, C. D. Photon counting strategies with low light level CCDs. *Mon. Not. R. Astron. Soc.* **345**, 985-991, doi:10.1046/j.1365-8711.2003.07020.x (2003).
- 10 Hirsch, M., Wareham, R. J., Martin-Fernandez, M. L., Hobson, M. P. & Rolfe, D. J. A Stochastic Model for Electron Multiplication Charge-Coupled Devices – From Theory to Practice. *PLoS ONE* **8**, e53671, doi:10.1371/journal.pone.0053671 (2013).
- 11 Murphy, K. P. *Probabilistic machine learning: an introduction.* (MIT press, 2022).
- 12 Messer, P. K., Henss, A. K., Lamb, D. C. & Winterlin, J. A multiscale wavelet algorithm for atom tracking in STM movies. *New J. Phys.* **24**, 14, doi:10.1088/1367-2630/ac4ad5 (2022).
- 13 Wanninger, S. *et al.* Deep-Learning assisted, Single-molecule Imaging analysis of multi-color DNA Origami structures. *Zenodo*, doi:10.1101/2023.01.31.526220 (2023).
- 14 McKinney, S. A., Joo, C. & Ha, T. Analysis of Single-Molecule FRET Trajectories Using Hidden Markov Modeling. *Biophys. J.* **91**, 1941-1951, doi:10.1529/biophysj.106.082487 (2006).
- 15 Zarrabi, N., Schluesche, P., Meisterernst, M., Börsch, M. & Lamb, D. C. Analyzing the Dynamics of Single TBP-DNA-NC2 Complexes Using Hidden Markov Models. *Biophys. J.* **115**, 2310-2326, doi:10.1016/j.bpj.2018.11.015 (2018).
- 16 Sikor, M., Mapa, K., von Voithenberg, L. V., Mokranjac, D. & Lamb, D. C. Real-time observation of the conformational dynamics of mitochondrial Hsp70 by spFRET. *The EMBO Journal* **32**, 1639-1649, doi:10.1038/emboj.2013.89 (2013).
- 17 Krause, S. *et al.* Graphene-on-glass preparation and cleaning methods characterized by single-molecule DNA origami fluorescent probes and Raman spectroscopy. *ACS Nano* **15**, 6430-6438, doi:10.1021/acsnano.0c08383 (2021).
- 18 Kamińska, I. *et al.* Graphene Energy Transfer for Single-Molecule Biophysics, Biosensing, and Super-Resolution Microscopy. *Adv. Mater.* **33**, 2101099, doi:10.1002/adma.202101099 (2021).

A.2 Paper 2: Deep-LASI, Single-Molecule Data Analysis Software

Deep-LASI, Single Molecule Data Analysis Software

Pooyeh Asadiatouei*, Simon Wanninger*, Clemens-Bäsem Salem*, Evelyn Ploetz[§], Don C. Lamb[§]

*These three authors contributed equally to the work

[§]Corresponding authors

Department of Chemistry and Center for NanoScience (CeNS)

Ludwig-Maximilians-Universität München

Butenandtstr. 5-13

81377 Munich, Germany

Emails: evelyn.ploetz@lmu.de; d.lamb@lmu.de

Keywords

Single-molecule spectroscopy, single molecule FRET, multi-color FRET, machine learning, deep neural networks

Abstract

By avoiding ensemble averaging, single molecule methods provide novel means of extracting mechanistic insights in to function of material and molecules at the nanoscale. However, one of the big limitations is the vast amount of data required for analyzing and extracting the desired information, which is time consuming and user dependent. Here, we introduce Deep-LASI, a software package for the manual and automatic analysis of single molecule traces, interactions and the underlying kinetics. The software can handle data from one-, two- and three-color fluorescence data, and was particularly designed for the analysis of two- and three-color smFRET experiments. The functionalities of the program include: the registration of multiple-channels, trace sorting and categorization, determination of the photobleaching steps, calculation of FRET correction factors and kinetic analyses based on hidden Markov modeling or deep neural networks. After a kinetic analysis, the ensuing transition density plots are generated, which can be used for further quantification of the kinetic parameters of the system. Each step in the workflow can be performed manually or with the support of machine learning algorithms. Upon reading in the initial data set, it is also possible to perform the remaining analysis steps automatically without additional supervision. Hence, the time dedicated to the analysis of single molecule experiments can be reduced from days/weeks to minutes. After a thorough description of the functionalities of the software, we also demonstrate the capabilities of the software via the analysis of a previously published dynamic three-color DNA origami structure fluctuating between three states. With the drastic time reduction in data analysis, new types of experiments become realistically possible that complement our currently available palette of methodologies for investigating the nanoworld.

Significance

Single molecule experiments are very powerful but, at the same time, the analysis can be very time intensive. Here, we present a program that eases the analysis of single-molecule time traces. We have incorporated machine learning methods to support the data analysis. The software performs all steps required for such an analysis either manually or automatically starting from data extraction through to the final graphical outputs. Hence, the time investment needed for the analysis of single molecule data can be reduced from days or even weeks to minutes.

Introduction

The ability to detect individual molecules has revolutionized the way we investigate the physical world. When measurements are no longer limited by ensemble averaging, sample heterogeneities, subpopulations and dynamics processes are directly observable. With such high sensitivity, a minimal amount of sample is necessary, and as the analysis is done one molecule at a time, high purification of the sample can be performed in the analysis[1]. When performing measurements on immobilized molecules with methods such as atomic force microscopy, optical and magnetic tweezers[2], or total internal reflection fluorescence (TIRF) microscopy, the dynamic processes of a single molecule can be observed as a function of time[3, 4]. As a result of such measurements, a huge number of trajectories are typically produced that need to be analyzed to extract the desired information from the system of interest.

Among the various valuable single-molecule techniques, fluorescence resonance energy transfer (FRET) experiments stand out as a non-contact method that can detect distances on the 2-10 nm scale and measure dynamics processes from nanoseconds to kiloseconds. Recent studies have shown that single-molecule FRET (smFRET) experiments are reproducible with an accuracy of 0.6 nm[5, 6]. With smFRET, it becomes possible to gain insights about the structural features and dynamics of materials, such as the structural fluctuations in biomolecules resulting from complex biological interactions[7]. Notably, smFRET promises to be an important method for the upcoming age of dynamic structural biology[8]. For many experiments, it is possible to detect subpopulations and measure dynamics directly from the collected data. For a detailed quantitative analysis, there are additional steps that need to be performed. Here, it is useful to monitor the fluorescence of the acceptor directly, which can be done using alternating laser excitation (ALEX)[9]. In ALEX, the donor and acceptor molecules are excited alternately. Hence, the photophysical state of the acceptor can be probed during the smFRET experiment, and correction factors for the determination of accurate FRET efficiencies can be extracted[10-12]. When analyzing ALEX data, the excitation scheme needs to be determined and incorporated into the analysis. As experiments grow in complexity, the intricacy of the analysis increases as well, and the availability of more advanced analysis tools becomes increasingly important.

Numerous software packages have been developed to aid in the analysis of single-molecule measurements. The choice of the optimal program for analysis depends upon the specifics of the experimental system and analysis required for the study. For smFRET experiments, Lerner et al. recently summarized and published an extensive list of analysis programs that were released until the year 2021, including tools to analyze time trajectories from surface experiments [8]. In a later study, Götz et al. compared the performance of eleven widely used single-molecule FRET analysis tools regarding the determination of kinetic models and extraction of the rate constants[13]. With respect to smFRET experiments, the vast majority of software has been

developed for two-color FRET experiments [14-19]. Our group has also developed a MATLAB-based software for data analysis on two-color FRET systems for surface-immobilized molecules called Tracy [13, 20].

Upon expanding our single-molecule TIRF setup to accommodate three-color FRET experiments, we needed to expand our analysis software. With this paper, we introduce our new software, Deep-LASI (Deep-Learning Assisted Single-molecule Imaging analysis), an open-source software package using MATLAB (but also available as a runtime version) which incorporates Python and C++ routines. The Deep-LASI software offers both manual and automatic analysis environments for a wide range of one-, two- and three-color single-molecule experiments [21]. The features of the program include mapping of multiple detection channels, extraction and background correction of one-, two- and three-color FRET data, trace classification and selection of relevant time points for the analyses, determination of the correction factors for the calculation of accurate FRET efficiencies, histogram generation of various parameters and kinetic analyses using Hidden Markov Models (HMM) and neural networks. Deep-learning techniques are emerging in virtually all data-driven fields and are having a big impact in the life sciences, in particular in microscopy [22-27]. Inspired by these developments, we incorporated deep-learning to help in trace classification, determination of the relevant regions of relevant traces, automated FRET correction and kinetic analyses.

Deep-LASI supports various data file formats with extendible support to read-in new formats into the program. Although originally written for smFRET data, the software is adept at handling any data as long as it results in time traces. Furthermore, the program offers environments for simulating and training single-molecule time traces. We also provide example datasets and tutorials to quickly gain proficiency in using the software [28].

Results and Discussion

Deep-LASI is a user-friendly software package with a high degree of automation and compatibility for the analysis of time-resolved single-molecule intensity traces. It is designed to help with the data analysis of one-, two- and three-color FRET experiments with interactive graphical user interfaces (GUIs) to provide enough freedom so that the user can extract the desired information based on their analysis needs. The source code is available such that the program can be adapted and further developed by expert users and software developers [29]. A description of these program features is given in the following sections.

The information to be extracted from the intensity traces of single molecules and the necessary steps will vary depending on the measurement assay and question of interest. An overview of the most common procedures in single-molecule data analysis is summarized in [Figure 1](#). The main analysis steps include reading-in the raw data, mapping the detection channels, (co-)localizing the particles and extracting the intensity information over the measurement time. In the next step, the software allows for classifying traces, determining usable regions within each time trace, plotting the distributions of the extracted parameters (such as FRET values,

labeling stoichiometry, or dwell-times), calculating the necessary correction factors and performing a kinetic analysis in the case of a dynamic system. In the case of the kinetics analyses, Transition Density Plots (TDP) are automatically generated and provide access to the Cumulative Dwell-Time Distribution functions (CDFs).

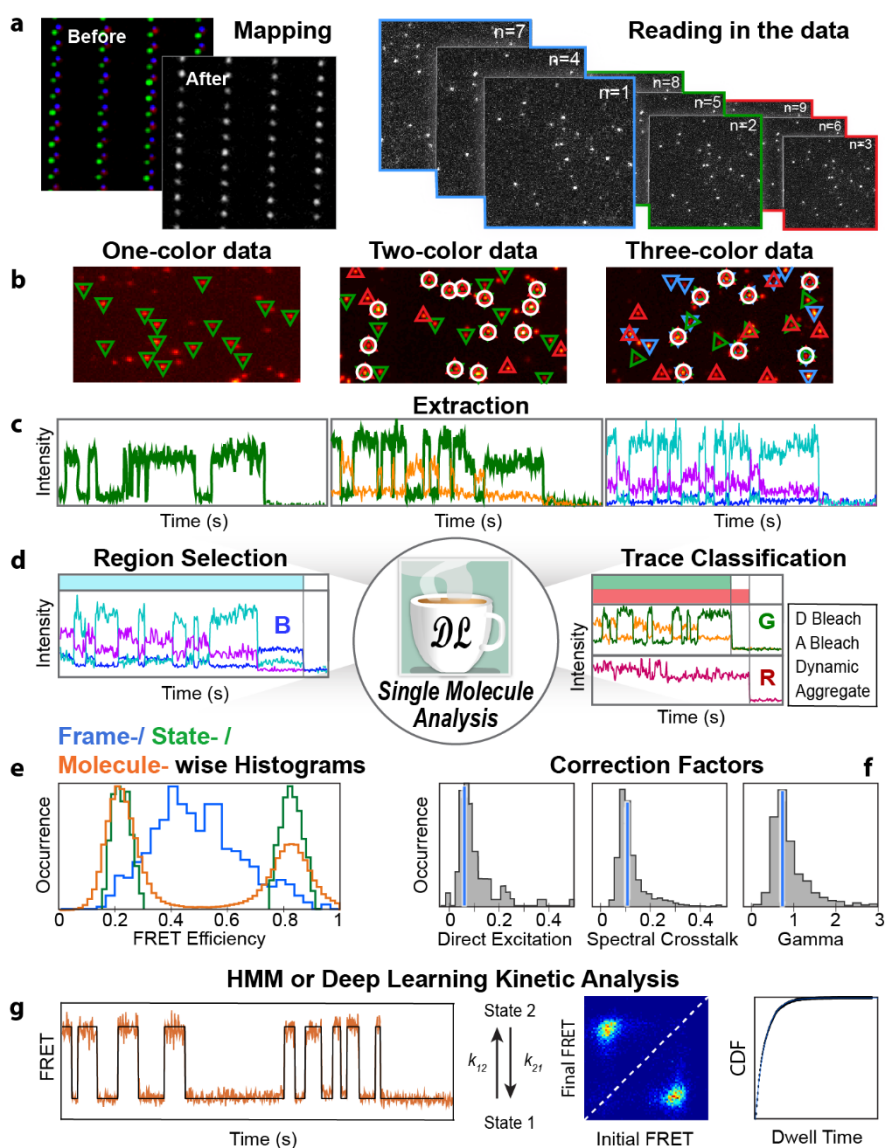


Figure 1. Overview of the functionalities of the Deep-LASI software package. The main applications of the software package are to extract, sort and analyze intensity traces from single-molecule data. This process involves a series of key steps: **(a)** For multicolor experiments, the different channels need to be register to each other (i.e. mapped). Afterwards, the raw data is read-in for each channel from a stack of frames based on the excitation scheme. **(b)** Single molecules are localized and, when desired, co-localized across different channels based on the created map. **(c)** The intensity traces are extracted from each detected (and co-localized) particle and corrected for background. **(d)** The analysis of extracted intensity traces starts with trace classification and selection of the useful region of each channel where the corresponding fluorophores are active. **(e)** The results can then be visualized by the means of various histograms with frame-, state- and molecule-wise approaches. **(f)** Optionally, the method-specific correction factors are determined. **(g)** For dynamic traces, a kinetic analysis can be performed by Hidden-Markov Modeling (HMM) or deep learning approaches. The panels show a typical Viterbi path created by HMM and Transition-Density Plots (TDP) with state transition information and the cumulative dwell-time distribution function (CDF) determined by fitting, respectively.

Deep-LASI also offers the opportunity for expert users to simulate multi-color single-molecule FRET traces and to train neural networks for new single-molecule assays. To ensure flexibility and accessibility overall, the extracted and analyzed data can be saved and reloaded at any time, and can be additionally imported and exported, from and into standard data files. In the following section, we discuss the individual working steps and underlying mechanisms that define the program's functionality.

1. Main functionalities of Deep-LASI

Typically, the initial step in analyzing single-molecule experiments involves read-in the raw data. The most elaborate features of the program are designed to work with images or movies from cameras like emCCD (electron-multiplying charge-coupled device) or sCMOS (scientific complementary metal oxide semiconductor) cameras. However, it also accommodates the direct read-in of custom, non-image data file formats encoding a time series (Figure 2; blue boxes). For detailed information and the latest list of supported file formats, please refer to the Online Tutorial available for *Deep-LASI* on Read the Docs [28].

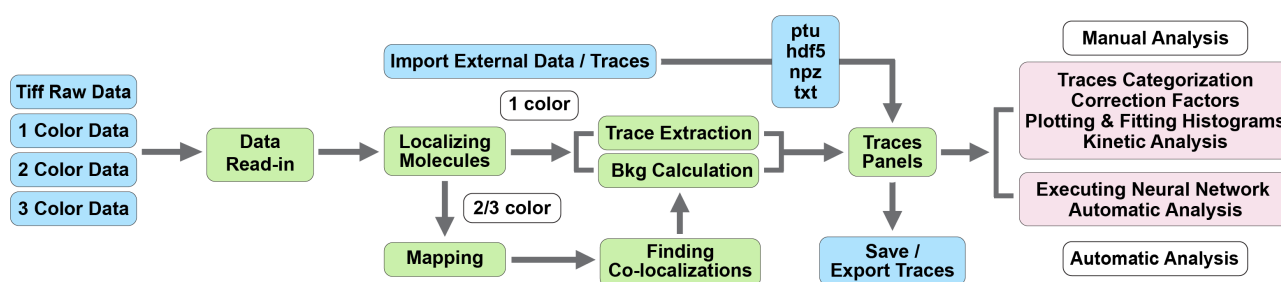


Figure 2. Schematic representation of the data handling workflow using the Deep-LASI software. The blue boxes show the raw movie data and intensity trace files that can be loaded into the program as well as the saving routines for storage and export of the analyzed data. The green boxes show the initial steps for data extraction and preparation with the Deep-LASI software. The pink boxes summarize the tools for data analysis and representation of results available through both manual and automatic modalities.

Given the diversity of fluorescence-based assays and methods resulting in time traces suitable for analysis through Deep-LASI, we will focus on the main functionalities of the program. This includes importing data files, detecting and mapping molecules, extracting traces, calculating background, and manually or automatically sorting the collected data (Figure 2; green boxes). Following these steps, the software provides a set of different tools to spectrally correct, kinetically analyze and summarize the single-molecule data (Figure 2; pink boxes). These tools can be used for (1) determining correction factors, (2) plotting representative properties of the results via their distributions (e.g. of apparent or accurate FRET values of single molecules, of states or frames), (3) allowing unsupervised, kinetic analysis of selected regions of the appropriate traces using hidden-Markov modeling (HMM) or deep-learning algorithms and, finally, (4) visualizing the data using TDPs and CDF plots. For an in-depth understanding of each feature, including the algorithms involved, the philosophy behind

the Graphical User Interfaces (GUIs), and tutorials featuring specific analysis examples, please refer to the comprehensive software documentation and manual [28].

Channel mapping

For multiple-color experiments using separate detection channels, a registration of the different detection channels is needed. When measuring at different wavelengths, perfect alignment between channels in terms of shift, magnification, and rotation can be challenging and time-consuming. Hence, a mapping process between cameras or regions of interest is required to ensure that the fluorescence signatures visible in the different channels originate from the same immobilized molecule. When performing fluorescence-based single-molecule experiments using imaging, the optimal pixel size is usually in the range of 40 - 100 nm. Thus, the fluorescence emitted by a single molecule spans multiple pixels and alignment within a single pixel is sufficient. The mapping is performed using a set of emitters well distributed across the detectors' field of view. We commonly employ a zero-mode waveguide pattern or a surface covered with emitting or scattering particles, such as fluorescent beads. Alternatively, mapping can be accomplished using the actual single-molecule data. One channel is selected as a reference channel. Our software then utilizes a phase-correlation algorithm to estimate the geometric transformation necessary to align the other channels to the reference image (Figure 3a)[30]. This geometric transformation involves scaling, rotation, and translation of the read-in images. Individual emitters are detected based on a user selected threshold and their localizations are utilized to further refined the mapping using a 2D polynomial of order up to 3. The prerequisite for this refinement operation is the colocalization of individual particles within 2 pixels after application of the geometric transformation. Their positions are determined using a stationary wavelet algorithm with adjustable sensitivity [31]. A transformation matrix is generated, which is then used to map the respective coordinates between channels. The mapping step corrects small misalignments between the cameras originating from tilts and shifts of cameras, different magnifications as well as aberrations in the detection paths. Notably, the mapping function is only used to find the corresponding pixels in the various detection channels corresponding to the location in the reference channel. The actual single-molecule analysis is performed separately on the raw data. No mapping of the images via the transformation matrix is performed except for inspection of the quality of the transformation matrix.

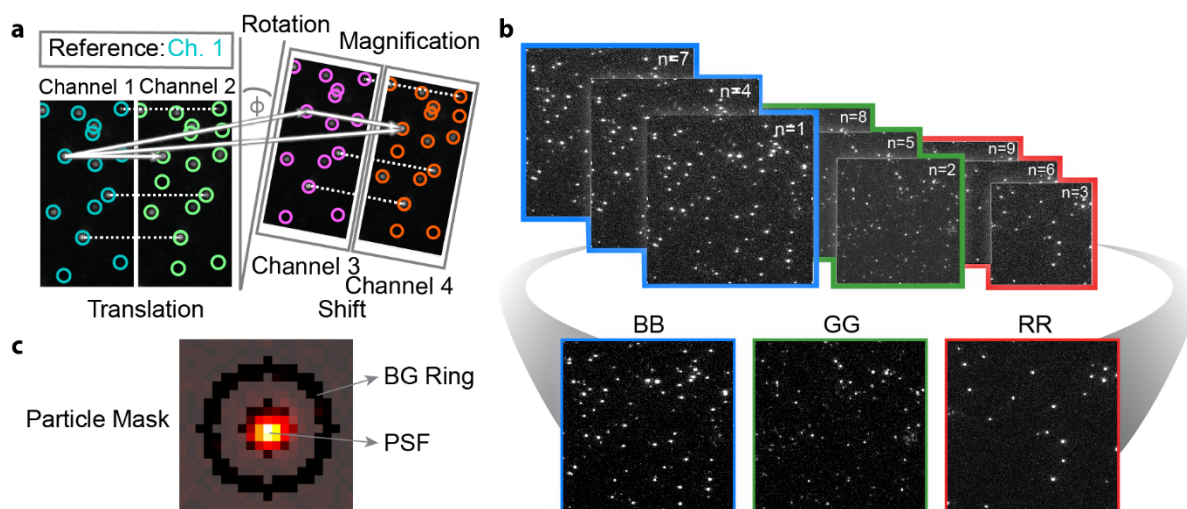


Figure 3. Particle detection, localization, and mapping. (a) Mapping process with Deep-LASI for multi-color experiments: A reference channel is chosen and all other channels are registered with respect to the reference channel. Here, the chosen reference channel is the blue channel. **(b)** Deep-LASI first opens the images registered on a specific camera and segregates the detection channels according to the excitation scheme. Here we show the BB, YY and RR channels. To identify the position of single emitters, we use the cumulative image taken from the direction excitation frames for the respective channels. **(c)** To calculate the measured intensity coming from a detected particle, a mask function is selected. We typically use a particle detection mask having a circular geometry of 7 pixels in diameter. The outer dark ring with a width of 2-pixels is used to determine the background contribution.

Loading imaging data collected using various excitation schemes

Once the detection channels have been mapped onto each other, the actual single molecule data one wishes to analyze is loaded. Upon loading the data, the frames are segregated based on the excitation scheme used (when necessary). For accurate smFRET experiments using camera based data acquisition, it is advantageous to use millisecond alternating laser excitation (ALEX) schemes [5, 32]. For two-color experiments, alternating frames are collected using donor and acceptor excitation respectively. Acceptor excitation is used to probe the presence and photoactive state of the acceptor molecule enabling the calculation of labeling stoichiometry. However, frames with acceptor excitation have to be excluded when calculating the FRET efficiency. When expanding to three-color experiments, ALEX is essential for analyzing the data and three excitation lasers are alternated respectively. There are also experiments where one wishes to detect the presence of one color at the beginning of the experiment, but then performs a smFRET experiment with different colors. One example would be measuring the conformation of DNA using smFRET in the absence or binding of a DNA-binding protein, which is labeled with a third color. In this case, the first few frames are used to detect the presence of the third color at the beginning of the measurement and used for selecting the traces that are to be analyzed. The remainder of the selected traces are then used to extract the smFRET information.

Typically, a series of consecutive measurements is performed using the same measurement parameters (excitation scheme, detection channels, exposure time, etc.) to gather sufficient statistics. This results in a

collection of data files originating from each camera. To initiate the analysis of the entire experiment, the Deep-LASI read-in process begins by collectively selecting all files from a single camera at once, usually starting with the most blue-shifted detection channel. Next, the first movie of the chosen file set is loaded. Here, the user is prompted to define the frame range, excitation sequence, and detection channel. Next, a cumulative intensity is displayed over the user-selected range (Figure 3b) in order to facilitate a user-friendly, interactive parameterization for the trace extraction. The loading procedure is then repeated for the corresponding movies of the remaining channels. The detection method and threshold for each detection channel are then determined. In the last step, the extraction parameters are then provided by the user: particle and background mask (Figure 3c), molecule selection criteria and the frame range used for extraction. Once the detection and extraction thresholds are established for all channels, the corresponding sets of files are sequentially loaded and single molecule traces are extracted according to the given selection mode: Deep-LASI extracts the trajectories either (1) for all detected molecules, (2) for colocalizing molecules only, or (3) for molecules detected in a given detection channel. At this point, all the extracted traces from the experiment are saved into a single datafile with a filename adapted from the first movie filename with the extension of .tdat.

Particle detection

To extract single-molecule trajectories, Deep-LASI provides three different techniques for single-molecule localization. For each technique, a sensitivity threshold is applied based on the normalized metrics of the wavelet coefficients or intensity values. Based on the selected threshold, a binary image is generated that encodes the detected particles. The position of each particle is determined by the center of mass of the pixels associated with the particle.

Wavelet. Wavelets are filters that can be applied to images (or time series) to enhance features with particular spatial (or temporal) frequencies. As the fluorescence signal coming from single molecules are diffraction limited, the detected fluorescence should be symmetric with the size given by the point-spread-function. By applying different wavelet filters, the original image is decomposed into a finite number of wavelets where particular spatial features are enhanced and others suppressed. More specifically, by mathematically applying low-pass and high-pass filters on the signal and repeating the procedure, a set of wavelet planes are generated at different resolutions [33-35]. Based on the median absolute deviation of the wavelet coefficients for each plane, insignificant features are removed automatically.

Intensity Thresholding. Another approach is to use intensity thresholding to detect molecules emitting intensities higher than a user-defined level. When enough adjacent pixels are above the threshold, the area would be considered as a particle and the central point taken for trace extraction. This rather easy method

works well as long as the signal and background are homogenous over the field of view and the signal from the molecules is sufficiently stronger than the background.

Regional Maxima. An alternative method based on intensity thresholding is the regional maxima approach. With this method, a Gaussian filter of nine pixels is first convoluted with the image and then the MATLAB function *imregionalmax* is used to find the local maxima. This is done by locating pixels where all eight neighboring pixels have lower intensities. It then returns all the regional maxima pixels as a binary mask to be considered as single emitting particles. The pixel intensities that are below the user-defined threshold are set to zero to allow adjustment of the overall sensitivity. This selection criteria has an advantage over normal intensity thresholding when analyzing data with heterogeneous single-to-noise within a frame.

Trace extraction

From the binary image generated from the detected particles, the particle positions are extracted using the MATLAB in-built function *regionprops*. This calculates the centers-of-mass for connected pixels. Using the central position of the individual particles, the particle mask (Figure 3c) is then used to determine the total number of detected photons for the particle as well as the background contribution. Typically, we use a circular particle mask with a diameter of 7 pixels. The size is chosen to optimize collection of photons within the point-spread-function of the molecule while minimizing the inclusion of additional pixel and hence potential overlaps between neighboring particles. The user can also adjust the particle mask settings based on their specific needs.

The particle positions are then linked in consecutive frames to generate time trajectories. To extract the intensity traces from each detected single emitter, frame-wise intensities for each channel are determined, and plotted over the whole measurement or selected frame range.

Background determination

The size and shape of the particle mask surrounding each particle's point-spread-function (PSF) (Figure 3c) and the method of background determination have a considerable impact on the signal-to-noise ratio, the quality of traces and, finally, on the resulting histograms. There are multiple approaches to background correction. Fortunately, the number of pixels that can be used to calculate the background intensity far outnumber the number of pixels within the PSF and hence can be subtracted with high accuracy. Deep-LASI extracts frame-wise intensities for each molecule detected in the various channel(s). To avoid any potential heterogeneity from the illumination profile, a non-constant background level within a frame or differences between cameras, the background signal is calculated and subtracted from the accumulated intensity within each particle mask. At any time during the analysis, the user can view the raw intensity traces without background subtraction.

As the signal is averaged in the background mask, no molecules should be present in region used for determining the background. For densely populated surfaces, the default mask can be adjusted, e.g. by reducing the radius of the mask. To decrease the uncertainty in the background estimation, the background is measured in approximately twice as many pixels as the signal. In addition, as the background does not typically change strongly with time, an eleven-frame sliding window (\pm five frame) is used to average the background value. The average background signal (scaled to the number of pixels in the particle mask) is then subtracted from the total measured intensity. The total measured intensity and the local background are determined for each frame and the background corrected intensity traces stored. By visually checking the intensity level of a trace after photobleaching of all fluorophores, the quality of the background subtraction routine can be controlled.

Trace Read-in Options

At this point in the analysis, one has extracted and saved the single-molecule time traces from one or more channels for a given excitation scheme. Here, it is also possible to reload the traces as well as to directly import intensity traces extracted using other software for any type of single-molecule time-series data. Several data importing options are incorporated include ptu, hdf5, npz and txt files. For example, we have also used Deep-LASI to analyze single-molecule intensity traces collected one at a time on a confocal microscope [21]. The txt file format is provided [13, 28] such that users can convert their data into a format that can be read into Deep-LASI.

Analysis Options

Deep-LASI offers diverse tools for analyzing and presenting information derived from single-molecule time traces, irrespective of the methods employed for data acquisition, ranging from one- to three-color measurements: the software facilitates both manual and automatic processes for (1) trace categorization into, for example, usable static and dynamic traces, and (2) selection of specific regions within individual traces for further analysis. Additionally, (3) Deep-LASI provides an overview of parameters characterizing selected regions in the intensity traces including brightness, background intensity, signal-to-noise ratio and photobleaching time. Beyond these basic functionalities, Deep-LASI supports manual and automatic analyses of one-, two- and three-color FRET assays. Moreover, the software enables (4) extraction of kinetic information from dynamic traces. Two distinct approaches are available for kinetics analyses: the first involves conventional hidden Markov modeling (HMM) with selectable algorithms for up to three channels, as detailed below [36, 37]. The second approach employs neural networks for automated data analysis wherein Deep-LASI outputs a confidence level of the time trajectory being in a specific state for each frame. In addition to

kinetic analyses, Deep-LASI allows for (5) obtaining accurate FRET efficiencies by extracting the necessary FRET correction factors from the data. Lastly, the software provides (6) state-of-the-art tools for summarizing the FRET states and kinetics extracted during the analysis. These include histograms illustrating distributions of, e.g., FRET efficiencies (apparent and accurate FRET) of static and dynamic traces, stoichiometry, or FRET correction factors. Furthermore, Deep-LASI provides transition density plots (TDPs) and cumulative dwell-time functions (CDFs) for summarizing the kinetics information found in the single-molecule data.

The subsequent paragraphs provide a brief introduction to manually using Deep-LASI for categorizing single-molecule traces, selecting regions and analyzing static experimental parameters. Subsequently, we discuss how to obtain accurate FRET measurements and extract kinetic information from single-molecule data.

Trace categorization and static analysis

After extracting or loading single-molecule traces, the next step involves the categorization and sorting of the molecules. In a typical single-molecule experiment, the dataset can easily comprise several thousands of traces. Many of the traces may be non-informative due to rapid photobleaching, the presence of aggregates, incomplete labeling or inadequate signal-to-noise ratios. Hence, the primary objective in trace categorization is to select the suitable regions of appropriate traces for further analysis. This starts by separating out traces that are unsuitable. For this, Deep-LASI provides dedicated panels and GUIs for systematically reviewing and categorizing traces (see [Figure 4a](#)). Typical categories include “static”, “dynamic”, and “trash” although users have the flexibility to add custom categories as needed for their experiment. Furthermore, Deep-LASI facilitates the sorting of traces based on the number of photoactive fluorophores by considering which fluorophores are active in each frame. For two-color FRET assays, for instance, traces can be sorted into categories like “donor bleach” and “acceptor bleach”, which proves instrumental in determining FRET correction factors at a later stage. Notably, users have the flexibility to assign multiple categories to individual traces, allowing classifications such as “static” and “acceptor bleach” simultaneously. This functionality becomes particularly advantageous in three-color FRET experiments where additional statistics for FRET correction factors can be obtained from analyzing constructs that contain only two of the three fluorescent dyes.

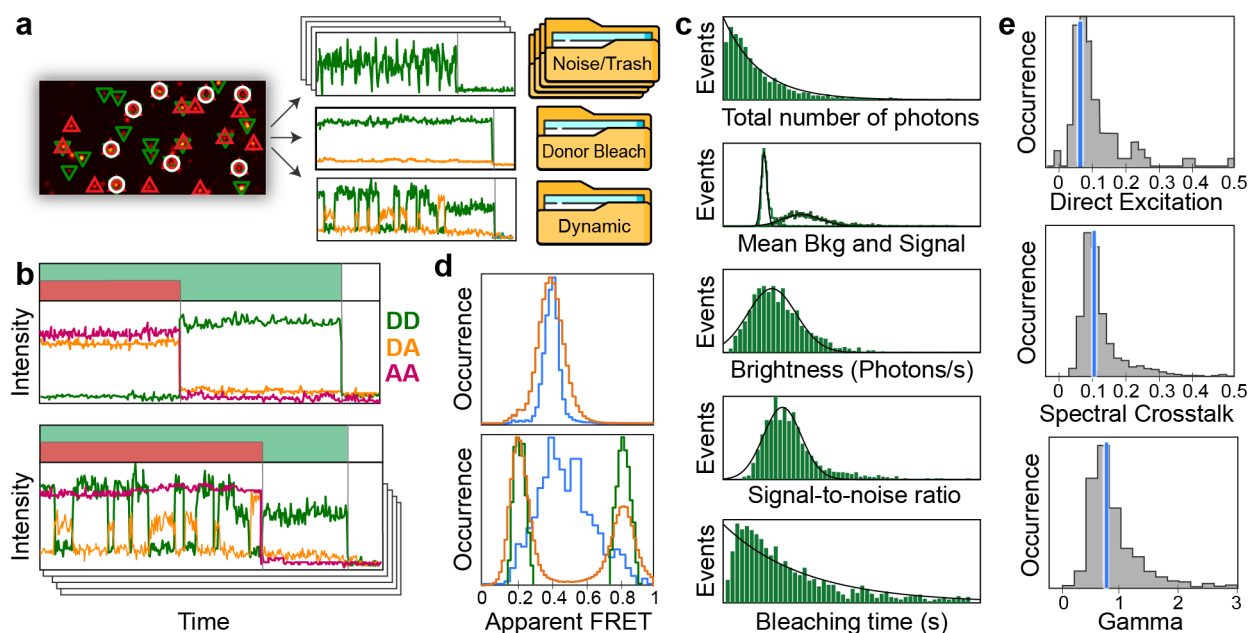


Figure 4. Single-molecule trace analysis of 2c FRET data with the Deep-LASI software. (a) The analysis of extracted traces begins with categorization based on the (1) underlying single-molecule assay, (2) numbers of dyes, (3) trace quality, (4) photochemistry involved, and (5) dynamic information available. (b) Two exemplary intensity traces are shown for two-color smFRET measurements using the ALEX scheme. (Top) A molecule exhibiting fluorescence only from the donor fluorophore and (bottom) a dual-labeled dynamic molecule. The donor signal (green), sensitized acceptor emission (orange) and direct acceptor excitation (red) are shown. Shaded regions in green and red mark the selected regions in each detection channel to be considered in further analyses. (c) Deep-LASI offers an overview of information from selected regions of the single-molecule traces. In addition to FRET, it displays molecule-wise histograms of signal (Sig) and background intensities (Bkg), brightness (Hz), signal-to-noise ratio and photobleaching time (s) with the corresponding fits. The histograms shown are for green detection after green excitation. (d) SmFRET histograms of (top) a static sample and (bottom) a dynamic sample interconverting between two states characterized by low and high FRET efficiencies. The apparent FRET efficiency of the sample is depicted frame-wise (orange), molecule-wise (blue) and state-wise (bottom panel, dark green). (e) Distributions of the FRET correction factors are shown, which are used to correct for direct excitation, spectral crosstalk and differences in detection sensitivity (gamma) for a simulated two-color smFRET sample. The median values of each histogram are indicated as a blue line.

In the second step of the characterization procedure, it is necessary to mark the regions of the useful traces to be included in further analyses (Figure 4b). Selection of the desired regions is possible with an activated cursor on the intensity trace panels. The selection can be general to define the regions in all channels to be included in the final histograms or kinetic analyses, or can be specific to each detection channel (shaded in the corresponding color) for determining individual photobleaching steps and regions to be used for the calculation of FRET correction parameters later on. Once correction factors have been estimated, users can choose to visualize the data at the level of apparent or accurate FRET. Correction factors are used to account for donor leakage into the acceptor channel, direct excitation of the acceptor, and differences in detection sensitivity of the donor and acceptor molecules. In addition, the individual intensity traces can be displayed with or without background correction.

Statistical overview of selected traces

The Deep-LASI software offers the possibility to evaluate and visualize the characteristics of selected frames, traces and, ultimately, the analytical results. From the first interface, the fluorescence properties of the different fluorophores can be assessed (Figure 4c). For EMCCD cameras, the characteristics of the amplification can be included to convert the camera counts to approximate photon numbers (otherwise, the signal in camera counts will be plotted). These distributions showcase the total signal until photobleaching (number of photons), the total signal and mean background per molecule (in kHz), the background-corrected brightness for the corresponding channels (in Hz), the individual signal-to-noise (SNR) ratios and the time until photobleaching of the respective fluorophores (in s). The histograms for each channel are automatically fitted to mono-exponential or Gaussian functions. The fit results are given in the respective panels.

Single Molecule FRET Analysis

In our research group, we focus on the single-molecule FRET experiments and evaluation. Hence, parts of the software are specialized for smFRET analysis from experiments carried out on immobilized molecules. With FRET, it is possible to investigate structural properties or dynamics due to FRET's strong dependency on the distance between fluorophores [8]. From the selected regions of the corresponding molecules, it is possible to calculate the apparent FRET efficiency histograms, that is the FRET efficiency determined from background corrected intensities without any further corrections. These can be plotted for each frame and molecule (frame-wise) or averaged value determined individually for each molecule (molecule-wise) Figure 4d. Frame-wise FRET histograms contain all FRET values obtained across different molecules and frames, giving a comprehensive projection of accessible FRET states in the sample from all selected molecules (Figure 4d, orange line). Alternatively, the molecule-wise (or trace-wise) histogram reports an average FRET value for each single molecule over the selected frame range (Figure 4d, blue line). Notably, for static samples, molecule-wise and frame-wise histograms will coincide whereas, for dynamic molecules, they will not. To overcome this, it is possible to plot histograms state-wise when analyzing dynamic traces (see dynamic analysis below; Figure 4d, lower panel, green line).

To capitalize on the ability of FRET to measure distances accurately on the sub nanometer regime, it is necessary to correct the apparent FRET efficiency for direct excitation of the acceptor, spectral crosstalk of the donor fluorophore into the acceptor channel and variations in detection sensitivity to the various fluorophores. Depending on which molecule photobleaches first, it is possible to determine a subset of the correction factors directly from the individual traces. In the case where the donor undergoes photobleaching before the acceptor, the software calculates the direct excitation correction factor using the residual emission

of the acceptor directly excited by the donor laser excitation (Figure 4e, top panel). Conversely, if the acceptor photobleaches before the donor molecule, the spectral crosstalk correction factor is determined as the residual donor emission detected in the acceptor channel (Figure 4e, middle panel). After correcting the trace for direct excitation and spectral crosstalk, the same trace can be used for determination of the detection efficiency correction factor from the ratio of the changes in acceptor and donor intensity after the acceptor's photobleaching step (Figure 4e, bottom panel). Once all individual traces are assessed for possible correction factors, the distribution is plotted and the software computes the average, median and mode of the distribution for each correction factor and dye pair. For accurate estimate of the various correction factors, a minimum number of continuous frames after photobleaching step should be included (we use a minimum of 20 frames). To kick out spurious values from the distributions, a maximum tolerable value for all correction factors can be entered. Values above the maximum will not be included in the calculation of the average, median and mean. The correction factors that cannot be determined directly from the traces are taken from the distribution (referred to as global correction factors,). We typically use the median of the distribution as it was found to be most robust given typical statistics, but the average or mode can also be selected (Figure 4e). The user also has the option to use the global correction factors for all traces or to enter the values individually for each trace. Once the correction factors are determined, accurate FRET values as well as distances can also be displayed. These together with additional parameters such as stoichiometry and FRET efficiencies (both, accurate and apparent FRET) can be viewed in a second interface. All histograms can be normalized and/or fit to a wide variety of functions.

Machine-learning analysis of dynamic trajectories

To analyze dynamic samples, additional functionalities are available in the Deep-LASI software. One can choose from two HMM analyses (Kevin Murphy [37] or Pomegranate [36]) or automatically via deep neural networks (DNN) [21] (Figure 5a). The results provide an estimation of the underlying states and kinetics within the individual trajectories. Hence, one generates in the end a "digitalized" version of the state pathway, which allows determination of the transition rates via the calculated dwell-time distributions.

Hidden Markov Modeling can be performed on 1-color, 2-color and 3-color data. For each data type, the FRET efficiencies or the intensities can be used as input. The number of states, mean values, standard deviations and the transition matrix can be initialized either using prior knowledge of the user, random uniform distributions or estimations based on k-means clustering. Other adjustable model parameters include the convergence threshold, the maximum number of iterations and the choice between local or global HMM. Local HMM creates a new model for each trace, whereas a global HMM utilizes one model (rates and states) for all traces of a selected category. Like other analysis tabs, this analysis can also be exploited on any desired category(s). The states and kinetics of 1-color and 2-color data are straightforward to model, as they inherently

represent distances in only one dimension. In 3-color FRET experiments, distances can be extracted in three dimensions by combining the FRET efficiencies of all fluorophore pairs. However, 3-color FRET is complicated by the strong interdependence of the FRET efficiencies and the numerous correction terms that are necessary to convert the apparent FRET efficiencies to actual distances. These corrections introduce significant uncertainties, making it difficult to properly model the system using HMM and identifying state transition. Therefore, The software focuses on using apparent FRET efficiencies and uncorrected intensities to accurately analyze 3-color FRET kinetics, treating the states of a given molecule as unique combinations of FRET efficiencies or intensities [38]. This is achieved by employing multivariate HMM, where each trace is 2-dimensional and each observation is a multivariate vector. In case of 3-color FRET, the multivariate vectors can contain either the three FRET efficiencies or the five intensity channels that are relevant for determining the kinetic information. Direct excitation of the last acceptor is excluded as it provides no kinetic information. Regardless of the number of colors, the software provides the option to use HMM on traces that were manually selected or classified by a deep neural network.

After running the HMM, multiple corresponding panels with the number of states, state values and transition probability matrices are updated. These will depend upon the executed mode, i.e. local or global. [Figure 5a](#) shows an example of a dynamic, two- state system with independent transitions between these states. One the trace panel of the HMM tab, the individual traces and their corresponding Viterbi path are shown ([Figure 5b](#), top panel). One can click through all traces present in the selected category to check the accuracy of the predicted states and transitions sequences.

DNNs can also be used for the kinetic analysis after the state classification step has been performed (.e.g [Figure 5b](#), bottom panel). Here, there is the option to run a "number of states" classifier to determine the predicted number of states observed in each trace ([Figure 5c](#), left panel). The user can then run a particular state-classifier model (i.e. for 2 states, 3 states or 4 states) on the selected data or use the output of the "number of states" classifier to automatically use the corresponding model for state assignment. The software plots the average confidence level for the state assignment over the individual traces ([Figure 5c](#), right panel). For the example shown here, the software is very confident regarding the existence of two states and their corresponding state assignments.

After running the machine learning approach of choice (i.e. HMM or DNN), a digitalized state pathway is generated for each trace. This allows one to calculate a state-wise distribution from the state trajectories ([Figure 4d](#), bottom panel, green line). The state-wise trajectories can be plotted normalized to the number of transitions or weighted by the number of frames contributing to each state. To analyze the underlying dynamics, the Deep-LASI software utilizes transition density plots (TDPs) to visualize the detected transitions between initial and final states within the data ([Figure 5d](#), left panel). From the TDP, the number of states, their corresponding values (e.g. FRET efficiencies), their connectivity and the number of transitions between

different pairs of states. Transition rates can be obtained directly from the output of a global HMM analysis. Alternatively, they can be calculated by selecting individual populations in the TDP and then fitting the corresponding dwell-time distribution (Figure 5d, right panel).

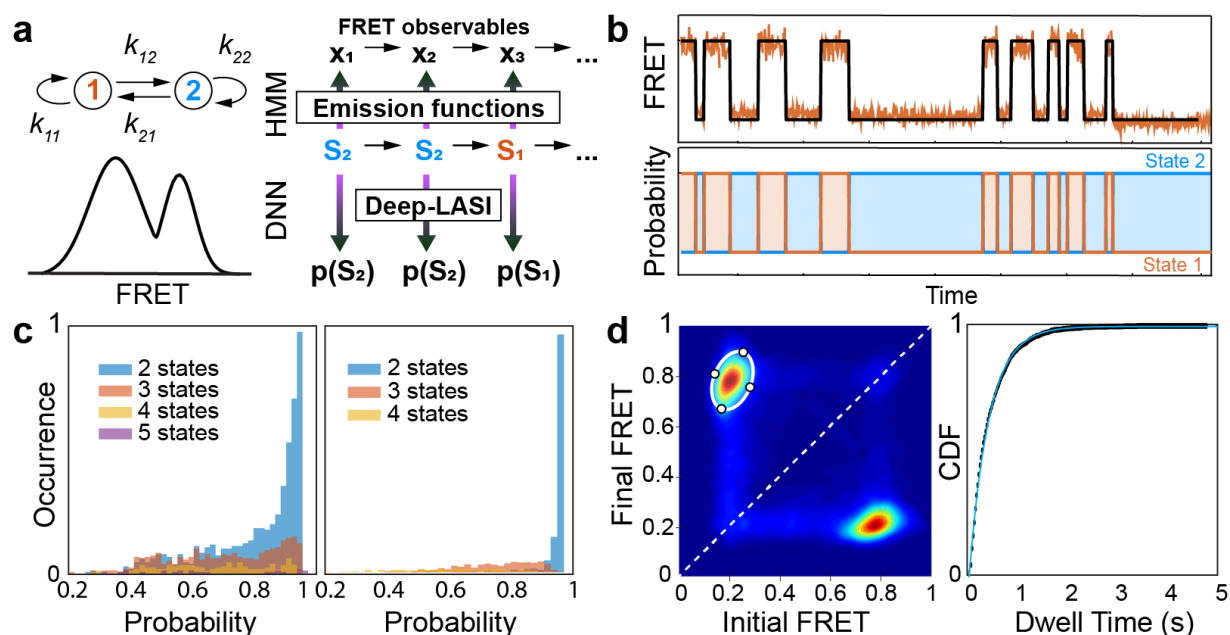


Figure 5. Kinetic analysis with the Deep-LASI software. (a) Left: Schematic representation of a two-state dynamic FRET system with the transition rates (k_{ij}) between the two states. Right: The software extracts the dynamic interconversion rates between the states through an HMM analysis or via state probabilities derived using a deep-learning algorithm based either on the measured FRET values or on the intensities directly. (b) Results of a kinetic analysis for an exemplary two-state smFRET trace. Top: The most probable state path (black) generated through an HMM analysis (i.e. the Viterbi path) or via the neural network and (bottom) the corresponding state probabilities. (c) For an automated kinetic analysis, the neural network identifies the most likely number of detected states in each individual trace (left panel). The state path is then determined based on the number of detected states and a histogram of the average confidence of the assigned states in each trace is given (right panel). (d) Left: To extract the underlying dwell times and kinetic rates, TDP are generated. By selected individual clusters in the TDP, which summarize transitions of dynamic molecules from an initial to a final FRET state, the dwell time distribution for the cluster is calculated. Right: The corresponding cumulative Dwell-Time Distribution function for the selected population shown on the left.

Automatic Analysis

The analysis of single-molecule data, especially for molecules immobilized on a surface, usually takes days or weeks, even for a single day of measurement and is prone to bias from the person analyzing the data. In addition, due to the time necessary for obtaining a reasonable amount of statistics, the parameter space that can be analyzed via such experiments is limited. To overcome such issues, automatic analysis tools using trained deep neural networks are available in the Deep-LASI software. Hence, each step of the analysis workflow described for manual evaluation above can be performed automatically. DNNs are available for trace

classification (with region selection), number of state determination and state trajectory analyses, and can be applied individually. In addition, the entire workflow from sorting and categorizing time traces, determining the photobleaching steps, calculating method specific correction factors, and state pathway determination can also be done automatically with the click of a single button [21]. After running the automated analysis, the workflow continues with evaluation of the TDPs by selecting clusters and fitting their dwell time distributions. On typically smFRET measurements with 4000 frames, the automated analysis is performed within 20-100 ms per intensity trace and has been implemented for one-, two- and three-color data.

Currently, we have tested the Deep-LASI software on DNA origami structures as well as protein systems [21]. However, the number of possible sorting categories and traces characteristics that the DNN has been trained on is not exhaustive. Therefore, for advanced users, the Deep-LASI software has the option to simulate one-color single molecule data as well as two- and three-color smFRET traces. In addition, the simulated data can also be used to train new neural networks, if desired.

2. Additional attributes of Deep-LASI

The motivation for designing and publishing Deep-LASI was manifold. We wanted to develop a software package for the community that (1) is easy to use for everyone independent of their scientific maturity or disciplinary background, (2) contains a high degree of automatization with respect to data extraction and analysis to save time and remove user bias, (3) is compatible with other single-molecule methods and setups with temporal resolution and (4) provides state-of-the-art tools for analyzing single-molecule trajectories. In addition, the program should provide (5) advanced tools for analyzing experimental data from multi-color FRET experiment up to three-colors (and potentially up to four in the future). These include the ability to simulate single-molecule data and train new machine learning approaches (HMM and neural networks) that are extendable for future single-molecule assays. Here, we summarize these general aspects of the developed software package Deep-LASI.

Ease-of use / User friendliness

To make the software easily accessible to a broad range of users and establish a universal analysis environment, Deep-LASI was designed to be easy to learn with a clear workflow. Each step in the workflow is accompanied with its own graphical user interface (GUI) guiding the from reading-in the raw data over data extraction and analysis to visualization of the results. After each step of data processing and analysis, it is possible to save the current status of the project, giving the user complete freedom to stop the analysis and resume at a later point in time. The resulting parameters and plots can be exported to external programs for presentation or publication purposes.

Automation

Gathering sufficient statistics in single molecule experiments requires collecting data from thousands of single-molecules. Hence, the Deep-LASI software can read-in hundreds of consecutive movies and extract the single molecule traces automatically after the conditions have been determined for the first movie. As data extraction and evaluation can be performed without supervision, new analyses and experiments are now realizable. For example, it is now possible to utilize more of the collected information such as also analyzing partially labeled molecules or performing a series of measurements as a function of experimental conditions that would be otherwise unthinkable when performing a manual analysis.

Compatibility

Although written for the analysis of smFRET data, Deep-LASI is applicable to all single-molecule experiments that end up measuring time traces independent of the setup and raw data file format. The program is compatible with ALEX excitation schemes as well as a non-ALEX excitation for two-color experiments and can read-in diverse file formats such as tif, ptu, hdf5 and npz. It is also possible to import other file formats by converting them in to a pregiven txt format [13, 29]. Notably, for measurements with significantly different noise characteristics than those encountered in fluorescence-based methods, the currently incorporated DNNs may need to be retrained.

Adaptability

The development of single molecule methodologies is a quickly advancing field and the analysis needs are exceedingly dynamic and often specific for each single project. Accordingly, it should be possible to easily modify and adapt the analysis approaches. Therefore, the source code for Deep-LASI is freely available on the GitLab platform where active feedback and comments can be given (e.g., reporting bugs and suggesting improvements), and user can contribute new functionalities. This is not only possible, but also appreciated.

Unique Methods

Deep-LASI combines an easy-to-use manual trace analysis software with state-of-the-art deep neural networks for automated data processing of one-, two- and three-color data. Many researchers are exploiting the advantages of smFRET but often avoid three-color assays due to the intrinsic complexity of the experiments and time-consuming analysis. Thereby, they also miss out on additional information that can be extracted by adding another fluorophore to the system. In addition, the only other software we are aware of for manually analyzing multi-color smFRET traces is SMACKS [18, 39]. Unique features of Deep-LASI are the automated

analysis of state pathways in kinetic data, and its ability to automatically analyze three-color smFRET data and extract absolute distances.

3. Application of Deep-LASI software on Experimental Data

Having introduced the Deep-LASI software, we now showcase its application to real three-color experimental data. For this, we show the results from experiments on a previously published L-shaped DNA origami structure [21]. The L-shaped origami structure contains a flexible tether that can bind to three distinct states (Figure 6a). The tether is labeled with Cy3B at the 3'-end and can bind to protruding strands placed at positions referred to as 6, 9 and 12 o'clock. The binding sites consist of complementary ssDNA strands of 7 nt length at 6 and 12 o'clock and 7.5-nt length (i.e., a strand with 8 nt and 1 bp mismatch) at 9 o'clock. Binding of the tether occurs by spontaneous base-pairing with the single-stranded protruding strands. To monitor the movement, we introduced two additional fluorophores Atto488 and Atto647N on the structure close to the complementary strands at 6 and 12 o'clock positions, respectively.

Data were collected using a three-color ALEX scheme of blue, green and red excitation and collected on three separate EMCCD cameras. Approximately 8,000 traces containing all three fluorophores were extracted from 100 movies. Using the automated neural network analysis, a trace classification was performed. The distribution of classes is shown in Figure 6b. Of the 7,990 traces extracted from the data, a dynamic classification was most probable for 740 (or 9%) of the trace, which were then utilized for further analyses. Figure 6c shows a representative single-molecule dynamic trace alongside with the apparent FRET efficiencies and kinetic analysis. From the selected traces, the software provides an overview of various parameters. Here, we show the total number of photons, the signal intensity and background, the brightness, the signal-to-noise ratio for the BB, GG and RR channels as well as the time until the photobleaching of the corresponding fluorophore (Figure 6d). When the measurement is of sufficient quality, one can proceed with the analysis. Next, we analyzed here the regions of the traces selected for determination of the FRET correction factors for spectral crosstalk, direct excitation as well as differences in detection sensitivity [5, 6, 8] (exemplified in Figure 4e for one dye pair). These are necessary for determining accurate FRET efficiencies. As neural networks also classify the active fluorophores in each frame, it is possible to extract the maximum number of frames in the data that can be utilized for the individual factors. In addition, for three-color samples it is also possible to utilize information from the dual-color labeled complexes. Hence, the automated analysis often more accurate than the corresponding manually analyzed traces. Traces that exhibit dynamic can then be further analyzed using either HMM or deep neural networks. Figure 6c(iv-v) depicts the Viterbi path of FRET efficiencies and state-probabilities returned from the deep-neural network for the representative smFRET trace. As discussed above, the neural network also generates confidence level histograms (Figure 6e). The left histogram indicates

a large number of traces containing dynamics between two of the states and a significant fraction of traces displaying all three expected states. The right histogram depicts the average confidence of the neural network in assigning the corresponding frames to the two and three states within each trace.

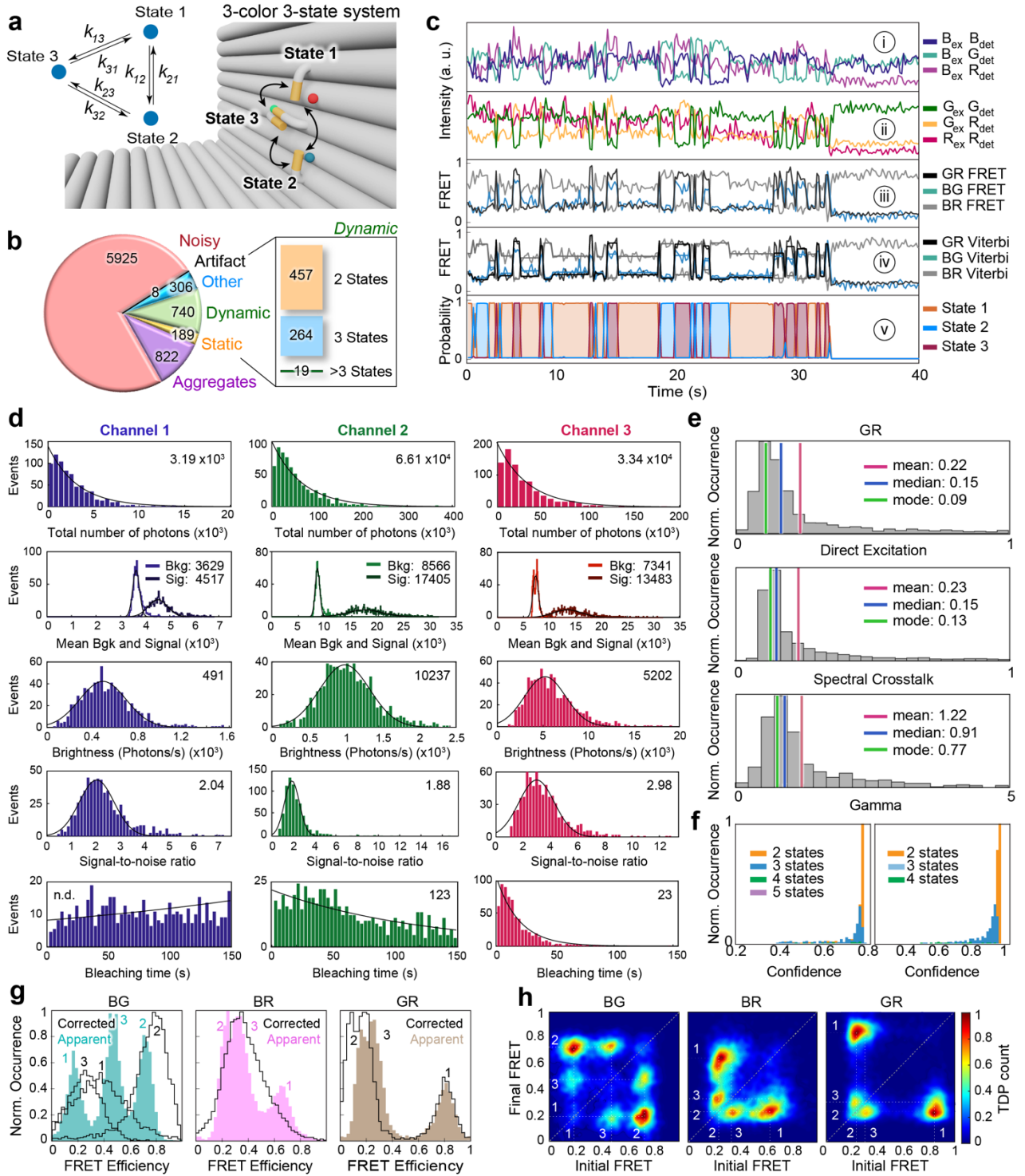


Figure 6. Application of the Deep-LASI software for analyzing three-color, three-state single-molecule FRET data. (a) Schematic overview of an L-shaped DNA origami structure with three binding positions, which are defined as 6, 9 and 12

o'clock. To probe the movement of the tether (labeled with Cy3B), Atto488 and Atto647N are placed near two of the binding sites (6 o'clock and 12 o'clock respectively). The inset shows the corresponding kinetic scheme. **(b)** A pie chart showing the distribution of the 7990 traces that have been sorted into the given categories (Static, Dynamic, Noisy, Aggregate, Artifact and Other). The 740 dynamic traces have been further categorized based on the number of states observed on each dynamic trace. **(c)** An exemplary single-molecule trace showing the six intensity and three apparent FRET traces for the 3-state system. Panel i shows the intensity traces detected in the blue, yellow and red channels after blue excitation. Panel ii depicts the intensity traces after yellow and red excitation. Panel iii shows the corresponding FRET efficiencies for each of the three dye pairs followed by the Viterbi path (panel iv) and state probabilities (panel v) derived by three-color HMM and a deep-learning assisted kinetic analysis respectively. **(d)** Photostatistics of the three individual detection channels. The histograms are automatically fit to a Gaussian or exponential functions and the fit parameter reported. Histograms represent the total number of detected photons before photobleaching, the total signal and background (normalized to the number of pixels in the particle mask), the background corrected molecular brightness, the signal-to-noise ratio and the time until photobleaching of the respective fluorophore. **(e)** Distributions of the FRET correction factors extracted from the data for direct excitation, spectral crosstalk and differences in detection sensitivity (γ) for the GR FRET pair. The mean, median and mode values for each histogram are indicated in red, blue and green, respectively. **(f)** DNN confidence output for the number of state classifier (left) and the state assignment classifier (right). **(g)** Apparent frame-wise FRET efficiency histograms are shown as shadowed plots for each dye pair with the three efficiency populations for the BG, BR and GR pairs. The frame-wise weighted, state-wise corrected FRET histograms are outlined in black. **(h)** TDPs are shown illustrating the transitions detected for all three dye pairs using apparent state-wise FRET efficiencies. Panel A is reproduced under the terms of the Creative Common CC-BY 4.0 License[21]. Copyright 2023, The Authors. Published by Springer Nature.

After completing the analysis, the results can be summarized using various tools. For the smFRET data shown here, we compare the frame-wise apparent FRET efficiency histograms (colored) and state-wise corrected FRET efficiency histograms (black lines) for each dye pair (Figure 6f). For the BG dye pair, three FRET states are well resolved with apparent efficiency values of 0.18, 0.73, and 0.48, corresponding to the states 1, 2 and 3 respectively. As expected from the design and associated Förster radius values of 53 and 65 Å [40-42], the 9 and 12 o'clock positions are more difficult to discern for the BR and GR FRET pairs. Nevertheless, the GR shows three populations with 0.83, 0.22, and 0.30 in apparent FRET efficiency for the same states 1-3. Although applying FRET correction factors in 3cFRET experiments usually results in broadened FRET histograms, the neural network correctly extracts the designed state-wise averaged accurate FRET histograms. Having the additional dimensions available in 3cFRET, it is possible to distinguishing the three states. Interestingly, as expected, the three FRET populations in BR converge into a single FRET state at a FRET efficiency of 0.35 upon correction.

Lastly, the Deep-LASI software enables the creation of TDPs in separate windows for each dye pair (Figure 6g). The user can then manually select populations in the plot and fit the resulting dwell-time distributions to determine the underlying transition rates between the corresponding states. By fitting the dwell time curves (Figure 6h), we identified state residency times of 0.65, 0.69, and 1.40 s for the states 1, 2 and 3, respectively. As the same state trajectories exist for each dye-pair, the state transitions can be selected from the TDPs the provide the best contrast between the two corresponding states.

Conclusions

In conclusion, Deep-LASI is a software package that allows for a straightforward and rapid analysis of single-molecule time trajectories. The software supports multiple formats from a variety of methods such as wide-field and confocal measurements. It allows for the convenient analysis single-molecule data starting with multiple-channel registration, trace sorting and categorization, determination of the photobleaching steps, calculation of FRET correction factors and kinetic analyses based on HMM or DNNs. Each step can be performed manually or automatically. It offers advanced functionalities for handling and interpreting single-molecule data in one-, two and three-colors such as the quantitative analysis of three-color smFRET data. By introducing Deep-LASI, we encourage researchers to exploit the capacities of single-molecule techniques without being concerned about the software environment or complicated, time-consuming analysis steps. As the field develops, the analysis requirements will change. Hence, the software is open source, inviting programming experts to extend the capabilities of Deep-LASI to address the expanding analysis needs of a rapidly growing research field.

Availability

The software is publicly available as source code, requiring MATLAB or a precompiled, standalone distribution for Windows or MacOS are hosted in a Gitlab repository under <https://gitlab.com/simon71/deeplasi>. A detailed manual is found at <https://deep-lasi-tutorial.readthedocs.io/en/latest/documentation.html>. The experimental data are provided in the Zenodo database (<https://zenodo.org/record/7561162>).

Author Contributions

S.W. developed and implemented the deep-learning algorithm, Deep-LASI in the manual analysis software written by C.B.S. P.A. collected the single-molecule TIRF data. S.W. and P.A. performed the Deep-learning assisted analyses. P.A. wrote the first draft of the manuscript and created the figures. All authors contributed to revising the manuscript. E.P. and D.C.L. supervised the project.

Declarations of Interests

All authors declare no conflict of interest.

Acknowledgments

We thank Philip Messer for valuable input and support regarding implementation of wavelets in the software for particle detection. We thankfully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 201269156 – SFB 1032 Project B03 (to D.C.L.), individual grants to PL696/4-1 (to E.P.). D.C.L. gratefully acknowledges funding from the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder through the ONE MUNICH Project Munich Multiscale Biofabrication. D.C.L. and E.P. gratefully acknowledge the financial support of the Ludwig-Maximilians-Universität München via the Department of Chemistry, the Center for NanoScience (CeNS) and the LMUinnovativ program BioImaging Network (BIN).

References

1. Ha, T.E., T.; Ogletree, DF.; Chemla, DS.; Selvin, PR.; Weiss, S., *Probing the interaction between two single molecules: fluorescence resonance energy transfer between a single donor and a single acceptor*. Proc. Natl. Acad. Sci. U S A, 1996: p. 6264-8.
2. Petrosyan, R., A. Narayan, and M.T. Woodside, *Single-Molecule Force Spectroscopy of Protein Folding*. J Mol Biol, 2021. **433**(20): p. 167207.
3. Colson, L., et al., *Trends in Single-Molecule Total Internal Reflection Fluorescence Imaging and Their Biological Applications with Lab-on-a-Chip Technology*. Sensors (Basel), 2023. **23**(18).
4. Roy, R., S. Hohng, and T. Ha, *A practical guide to single-molecule FRET*. Nat Methods, 2008. **5**(6): p. 507-16.
5. Hellenkamp, B., et al., *Precision and accuracy of single-molecule FRET measurements—a multi-laboratory benchmark study*. Nat Methods, 2018. **15**(9): p. 669-676.
6. Agam, G., et al., *Reliability and accuracy of single-molecule FRET studies for characterization of structural dynamics and distances in proteins*. Nat Methods, 2023. **20**(4): p. 523-535.
7. Ha, T.T., A.; Liang, J.; Caldwell, W.; Deniz, A.; Chemla, D.; Schultz, P.; Weiss, S., *Single-molecule fluorescence spectroscopy of enzyme conformational dynamics and cleavage mechanism*. Proc. Natl. Acad. Sci. USA, 1999. **96**: p. 893–898.
8. Lerner, E., et al., *FRET-based dynamic structural biology: Challenges, perspectives and an appeal for open-science practices*. Elife, 2021. **10**(60416).
9. Kapanidis, A.N.L., N.K.; Laurence, T.A.; Doose, S.; Margeat, E.; Weiss, S., *Fluorescence-aided molecule sorting: Analysis of structure and interactions by alternating-laser excitation of single molecules*. PNAS, 2004. **101**(24): p. 8936-8941.
10. Hildebrandt, L.L., S. Preus, and V. Birkedal, *Quantitative single molecule FRET efficiencies using TIRF microscopy*. Faraday Discuss, 2015. **184**: p. 131-42.
11. Salem, C.-B., E. Ploetz, and D.C. Lamb, *Probing dynamics in single molecules*, in *Spectroscopy and Dynamics of Single Molecules* 2019. p. 71-115.
12. Kudryavtsev, V., et al., *Combining MFD and PIE for accurate single-pair Forster resonance energy transfer measurements*. Chemphyschem, 2012. **13**(4): p. 1060-78.
13. Gotz, M., et al., *A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories*. Nat Commun, 2022. **13**(1): p. 5402.
14. de Lannoy, C.V., et al., *FRETboard: Semisupervised classification of FRET traces*. Biophys J, 2021. **120**(16): p. 3253-3260.
15. Huisjes, N.M., et al., *Mars, a molecule archive suite for reproducible analysis and reporting of single-molecule properties from bioimages*. eLife, 2022. **11**: p. e75899.
16. Kinz-Thompson, C.D. and R.L. Gonzalez, Jr., *Increasing the Time Resolution of Single-Molecule Experiments with Bayesian Inference*. Biophys J, 2018. **114**(2): p. 289-300.
17. Preus, S., et al., *iSMS: single-molecule FRET microscopy software*. Nat Methods, 2015. **12**(7): p. 593-4.
18. Schmid, S., M. Gotz, and T. Hugel, *Single-Molecule Analysis beyond Dwell Times: Demonstration and Assessment in and out of Equilibrium*. Biophys J, 2016. **111**(7): p. 1375-1384.
19. Verma, A.R., et al., *Increasing the accuracy of single-molecule data analysis using tMAVEN*. bioRxiv, 2023.
20. Heiss, G., et al., *Conformational changes and catalytic inefficiency associated with Mot1-mediated TBP–DNA dissociation*. Nucleic Acids Research, 2019. **47**(6): p. 2793-2806.
21. Wanninger, S., et al., *Deep-LASI: deep-learning assisted, single-molecule imaging analysis of multi-color DNA origami structures*. Nat Commun, 2023. **14**(1): p. 6564.
22. Thomsen, J., et al., *DeepFRET, a software for rapid and automated single-molecule FRET data classification using deep learning*. Elife, 2020. **9**.
23. Li, J., et al., *Automatic classification and segmentation of single-molecule fluorescence time traces with deep learning*. Nat Commun, 2020. **11**(1): p. 5833.

24. Liao, J., et al., *Deep-learning-based methods for super-resolution fluorescence microscopy*. Journal of Innovative Optical Health Sciences, 2022. **16**(03).
25. Yao, R., et al., *Net-FLICS: fast quantitative wide-field fluorescence lifetime imaging with compressed sensing - a deep learning approach*. Light Sci Appl, 2019. **8**: p. 26.
26. Huang, B., et al., *Enhancing image resolution of confocal fluorescence microscopy with deep learning*. PhotonIX, 2023. **4**(1).
27. Ragone, M., et al., *Deep learning modeling in microscopy imaging: A review of materials science applications*. Progress in Materials Science, 2023. **138**: p. 101165.
28. Fablab. *Deep-LASI Documentation*. 2023; Available from: <https://deep-lasi-tutorial.readthedocs.io/en/latest/index.html>.
29. Deep-LASI. *Deep-LASI Software*. 2023; Available from: <https://gitlab.com/simon71/deeplasi>.
30. Reddy, B.S. and B.N. Chatterji, *An FFT-based technique for translation, rotation, and scale-invariant image registration*. IEEE Transactions on Image Processing, 1996. **5**(8): p. 1266-1271.
31. Mallat, S.G., *A theory for multiresolution signal decomposition: the wavelet representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989. **11**(7): p. 674-693.
32. Kapanidis, A.N., et al., *Alternating Laser Excitation of Single Molecules*. Acc Chem Res, 2005. **38**(7): p. 523-33.
33. Daubechies, I., *Ten Lectures on Wavelets*1992, Philadelphia: SIAM.
34. Mallat, S.G., *A wavelet tour of signal processing : the sparse way*. 3rd ed2009, Amsterdam ; Boston: Elsevier/Academic Press. xx, 805 p.
35. Starck, J.L. and F. Murtagh, *Astronomical Image and Data Analysis*2007: Springer Berlin Heidelberg.
36. Schreiber, J., *Pomegranate: fast and flexible probabilistic modeling in python*. Journal of Machine Learning Research, 2018. **18**(164): p. 1-6.
37. Murphy, K. *Hidden markov model (HMM) toolbox for Matlab*. 1998; Available from: <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>.
38. Barth, A., L. Voith von Voithenberg, and D.C. Lamb, *Quantitative Single-Molecule Three-Color Förster Resonance Energy Transfer by Photon Distribution Analysis*. J Phys Chem B, 2019.
39. Gotz, M., et al., *A Multicolor Single-Molecule FRET Approach to Study Protein Dynamics and Interactions Simultaneously*. Methods Enzymol, 2016. **581**: p. 487-516.
40. Ploetz, E., et al., *Förster resonance energy transfer and protein-induced fluorescence enhancement as synergetic multi-scale molecular rulers*. Scientific Reports, 2016. **6**: p. 33257.
41. Atto-Tec. *Förster Radius of Atto dyes*. 2023; Available from: https://www.atto-tec.com/images/ATTO/Katalog_Preisliste/Katalog_2019_2020.pdf.
42. Ross, J., et al., *Multicolor Single-Molecule Spectroscopy with Alternating Laser Excitation for the Investigation of Interactions and Dynamics*. The Journal of Physical Chemistry B, 2007. **111**(2): p. 321-326.

A.3 Paper 3: A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories⁷⁸

Götz, M., Barth, A., Bohr, S.S.R., Börner, R., Chen, J., Cordes, T., Erie, D., Gebhardt, C., Hadzic, M., Hamilton, G., Hatzakis, N., Hugel, T., Kisley, L., Lamb, D. C., Lannoy, C., Mahn, C., Dunukara, D., Ridder, D., Sanabria, H., Schimpf, J., Seidel, C. A. M., Sigel, R. K. O., Sletfjording, M. B., Thomsen, J., Vollmar, L., Wanninger, S., Weninger, K. R., Xu, P., Schmid, S. *Nature Communications*, **13**, 5402 (2022).

DOI: <https://doi.org/10.1038/s41467-022-33023-3>



A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories

Received: 20 March 2022

Accepted: 30 August 2022

Published online: 14 September 2022

Check for updates

Markus Götz^{1,20} , Anders Barth^{2,21}, Søren S.-R. Bohr^{3,4}, Richard Börner^{5,22}, Jixin Chen⁶, Thorben Cordes⁷, Dorothy A. Erie^{8,9}, Christian Gebhardt⁷, Mélodie C. A. S. Hadzic⁵, George L. Hamilton^{10,23}, Nikos S. Hatzakis^{3,4}, Thorsten Hugel^{11,12}, Lydia Kisley^{13,14}, Don C. Lamb¹⁵, Carlos de Lannoy¹⁶, Chelsea Mahn¹⁷, Dushani Dunukara¹³, Dick de Ridder¹⁶, Hugo Sanabria¹⁰, Julia Schimpf^{11,18}, Claus A. M. Seidel², Roland K. O. Sigel⁵, Magnus Berg Sletfjerding^{3,4}, Johannes Thomsen^{3,4}, Leonie Vollmar^{11,18}, Simon Wanninger¹⁵, Keith R. Weninger¹⁷, Pengning Xu¹⁷ & Sonja Schmid¹⁹

Single-molecule FRET (smFRET) is a versatile technique to study the dynamics and function of biomolecules since it makes nanoscale movements detectable as fluorescence signals. The powerful ability to infer quantitative kinetic information from smFRET data is, however, complicated by experimental limitations. Diverse analysis tools have been developed to overcome these hurdles but a systematic comparison is lacking. Here, we report the results of a blind benchmark study assessing eleven analysis tools used to infer kinetic rate constants from smFRET trajectories. We test them against simulated and experimental data containing the most prominent difficulties encountered in analyzing smFRET experiments: different noise levels, varied model complexity, non-equilibrium dynamics, and kinetic heterogeneity. Our results highlight the current strengths and limitations in inferring kinetic information from smFRET trajectories. In addition, we formulate concrete recommendations and identify key targets for future developments, aimed to advance our understanding of biomolecular dynamics through quantitative experiment-derived models.

How does biomolecular function arise from structural dynamics? This largely unsolved question is central for the understanding of life at the molecular scale. However, the transitions between various conformational states have remained challenging to detect, quantify, and interpret. Over the past two decades, single-molecule Förster resonance energy transfer (smFRET) detection has emerged as a powerful technique to study the dynamics of single biomolecules under physiological conditions using fluorescence as a readout¹. A unique aspect of smFRET is its ability to link space and time, i.e., to connect structural with kinetic information under both equilibrium and non-equilibrium

conditions, which is often unachievable using ensemble methods. By measuring the distance-dependent energy transfer from a donor to an acceptor fluorophore, distances in the range of 4 to 12 nm can be measured with sub-nanometer precision and accuracy². Various experimental implementations exist that allow one to measure smFRET on diverse timescales from picoseconds to hours. All of this makes smFRET an ideal tool in the growing field of dynamic structural biology³.

To study conformational dynamics of one single molecule for an extended time (seconds to minutes), dye-labeled biomolecules are

A full list of affiliations appears at the end of the paper. e-mail: goetz@picoquant.com; schmid@nanodynlab.org

most commonly immobilized on passivated glass slides and imaged using camera-based brightfield detection, or confocal detection using avalanche photodiodes (APDs)². The resulting fluorescence time traces have a time resolution of about 10 ms – 100 ms for the most common camera-based studies, and picoseconds for single-photon counting APDs. The observation time per single molecule is limited by photobleaching, leading to an average bandwidth of less than three orders of magnitude in time^{4–6}. Conformational transitions of the biomolecule change the inter-dye distance leading to discrete steps in the fluorescence signal and the FRET efficiency (Fig. 1). This desired time-resolved distance information is convoluted with largely Gaussian noise in the experiment (from autofluorescence background, detector noise, laser fluctuations, etc.). Moreover, noise and photobleaching are intrinsically coupled: increasing the laser power for a better signal-to-noise ratio causes faster photobleaching, which reduces the temporal bandwidth of the experiment. As a result, signal interpretation in terms of biomolecular states and specific transitions between them is not trivial.

A multitude of analytical approaches have been developed to infer the number of functional states and quantify kinetic rate constants from noisy experimental data. Frequently, hidden Markov models (HMMs)⁷ are used to infer an idealized state sequence from which dwell-time distributions are compiled, which are then fit (with exponentials) to obtain kinetic rate constants^{8,9}. Alternatively, the transition matrix that is part of every HMM can directly be converted to kinetic rate constants. The HMM formalism is based on a discrete memoryless Markov process that infers a set of parameters (probabilities of states, transitions, and observations) to describe the observed sequence of FRET efficiencies. Many extensions of the HMM formalism have been developed^{10–15} including Bayesian approaches^{16–19}, and very fast kinetics (low energy barrier crossings) can be inferred from single-photon arrival times^{20–22}.

Often, multiple input models are compared based on a scarcity criterion to avoid bias in the selection of the optimal model size (i.e., the number of states and rate constants), and hence the number of free parameters^{8,23–25}. Other analysis approaches, such as correlation analysis^{26–30} and discretization methods based on cluster analysis^{31–34}, may treat the raw data in a model-free way while the extraction of individual rate constants (rather than residence times only) still relies on an initial guess of a model. The growing number of analytical methods renders it increasingly difficult to decide on the optimal tool

for a specific application and to judge whether the described improvements justify the time cost of implementation. Hence, it was identified during a round table discussion of the smFRET community (Fluorescence subgroup, Biophysical Society Meeting 2019, Baltimore, US) that a critical assessment of the available tools is needed.

Here, we present the results of a comparative multi-laboratory study that provides a systematic evaluation of eleven analysis tools (summarized in Table 1) using simulated as well as experimental data of varied complexity. Three of the analysis tools were utilized under different conditions, leading to a comparison of 14 different analyses. While clearly not all existing analysis tools could be covered (new tools are released continuously), this blind study (illustrated in Fig. 1) allows us to directly assess the performance of the different analysis approaches for the inference of kinetic information from single-molecule FRET trajectories and to identify their strengths and weaknesses. Specifically, we assess the accuracy of the inferred kinetic model (i.e., the kinetic rate constants and their connectivity) plus the associated uncertainties, and this for kinetic models of varied size, from the simple case of a two-state system (Fig. 2) to the more complex case of a non-equilibrium three-state system (Fig. 3), and finally to degenerate multi-state systems (Figs. 4, 5). All analyses were performed by the expert labs of each tool to ensure optimal implementation (see Methods for details).

Results

The archetypal 2-state system

We first consider the simplest case of a kinetic 2-state system, which could represent alternation between two conformations of a biomolecule in dynamic equilibrium, or transient biomolecular interactions. The kinetics of this system are described by two rate constants (Fig. 2a). In a blind study, we analysed simulated and experimental smFRET data using the diverse set of analysis tools summarized in Table 1 and detailed in the Supplementary Methods. Simulated test data (described in Methods) has the advantage that the underlying ground truth (GT, i.e., the simulation input) is known, which facilitates the evaluation of the inferred results, while, for experimental data, the GT is naturally not known. Figure 2b depicts an example of the simulated traces. We note that it closely resembles the experimental trace in Fig. 2e. Based on a dataset of such simulated traces ($n=75$), all laboratories inferred FRET efficiencies (Fig. 2c) and rate constants (Fig. 2d), which agree very well: the FRET efficiencies deviate by less

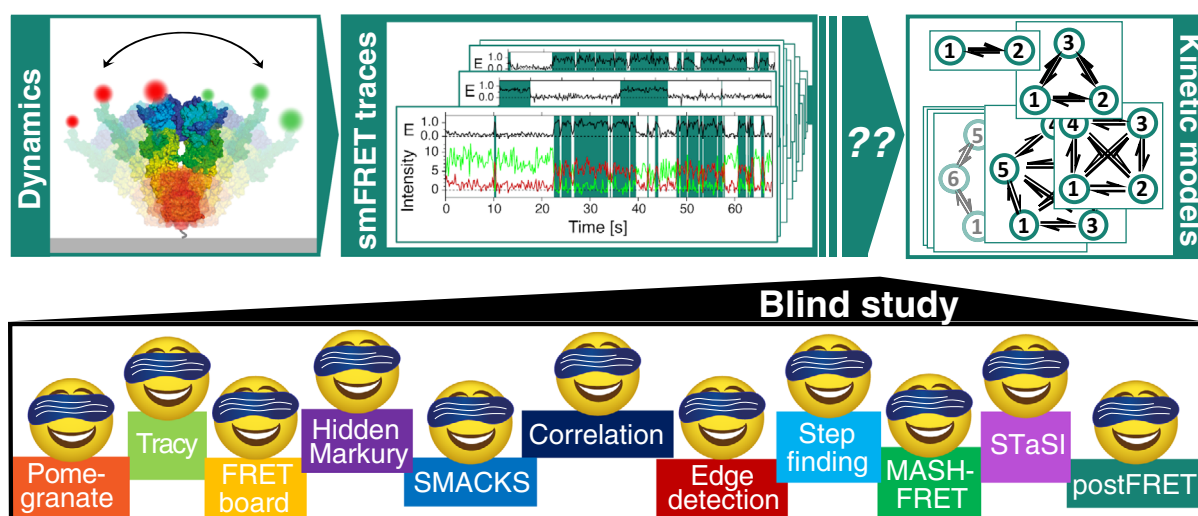


Fig. 1 | This blind study reports on the performance of diverse analysis tools to describe single-molecule dynamics with quantitative kinetic rate constants. Biomolecular dynamics of proteins and nucleic acids can be detected by smFRET and other single-molecule techniques. Extracting testable kinetic rate models from

the experimental time traces is complicated by experimental shortcomings. Multiple labs joined forces to directly test the performance of diverse analytical approaches to infer kinetic rate constants in a blind study.

Table 1 | Overview of the kinetic analysis approaches and software tools considered herein, grouped based on similarity

#	Tool name	Keywords	Description	Model selection	Uncertainty
1	Pomegranate	HMM	The python package Pomegranate is used for efficient and iterative modeling, fitting and evaluation of state numbers using the BIC. Dwell time analysis is subsequently performed after defining all transitions using a multivariate Gaussian fitting scheme and unbinned maximum likelihood fitting.	BIC	95% CI
2	Tracy	HMM	Global HMM analysis was performed while setting the FRET efficiency and sigma as parameters to be learned. The state transitions and the state dwell times are selected by the user in a transition density plot and fitted with an exponential to obtain the rate constants.	Manual inspection	95% CI
3	FRETboard ^{33,35}	Semi-supervision, remotely served	A semi-supervised classification tool served remotely through a browser window. Users supervise the training of a classification model of choice, by manually correcting classification of example traces until the quality of automated classification is satisfactory.	Path probability	95% CI
4	Hidden-Markury	1D/2D- HMM	Hidden-Markury is a trace analysis software based on an interactive Jupyter notebook script, supporting global 1D FRET efficiency traces or 2D donor & acceptor photon streams, optionally treating degenerate states, forbidden transitions, fixed model parameters.	BIC	Sub-sampling
5	SMACKS ³³	1-3D- HMM	Semi-ensemble HMM is used to extract one kinetic model from many smFRET fluorescence traces without prior discretization in two steps: (1) per trace HMM optimization (2) global per dataset optimization of the kinetic model, with pre-trained intensity parameters.	BIC	95% CI
6	SMACKS (SS)	1-3D- HMM	Test for user bias in semi-supervised inference: independent second analysis using SMACKS by S. Schmid.	BIC	95% CI
7	Correlation	Discretized correlation	An unbiased, model-independent approach to obtain quantitative relaxation times from the negative amplitude of the cross-correlation function ^{26,27} . To enable a quantitative analysis of multi-state systems, a filtered correlation analysis ⁵² is performed based on the state sequence obtained with a step-finding algorithm ³⁶ .	BIC	95% CI
8	Edge finding (CK)	CK filter	The Chung Kennedy non-linear filter is applied to the time records of donor, acceptor and/or FRET efficiency to identify state transition points as sudden increases in the standard deviation of points in forward/backward predictor windows. Transition edges are confirmed by a two-sample student's t-test on the forward/backward windows.	Manual inspection	Not assessed
9	Edge finding(k-means)	k-means clusters	All data points in either the donor and acceptor or the FRET efficiency time traces are assigned to distinct clusters. The mean value of each cluster is calculated and the points are reassigned to clusters to iteratively minimize the differences between the point values. Transition edges are identified as cluster assignment changes.	Manual inspection	Not assessed
10	Step finding	Line fitting	The entire dataset is iteratively fit with an increasing number of line segments. The addition of line segments is accepted if the overall fit quality is improved significantly. Rate constants are derived from dwell time analysis of line segments, which are assigned to a FRET state based on their mean FRET efficiencies. (SEM: standard error of the mean.)	Manual inspection	SEM / 68% CI
11	STaSI	Student's t-test	Detects step transition using the Student's t-test. Segments are grouped into states by hierarchical clustering. The optimum number of states is established using a minimum description length equation that sums the goodness of fit measured using the L1 norm to consider the sparseness of the states and transitions. (MDL: minimum description length.)	MDL	95% CI
12	MASH-FRET (bootstrap) ³⁴	STaSI, vbFRET, bootstrap	AMATLAB-based GUI for the simulation and analysis of smFRET videos and fluorescence time traces ⁴⁵ . Initial FRET states are obtained using STaSI and a BIC selection on 2D-Gaussian mixtures that model the global transition density plot. Refined FRET states, transition rate constants and uncertainties are then obtained using vbFRET and single exponential fit on bootstrapped dwell time histograms.	BIC	Standard deviation (2σ)
13	MASH-FRET (prob.) ³⁵	STaSI, vbFRET, DPH	The degeneracies of FRET states are estimated from ensemble dwell time histograms by performing a BIC selection on phase-type distributions. The fix-sized transition rate matrix is finally optimized using the Baum-Welch algorithm on hard-assigned FRET state trajectories.	BIC	95% CI
14	postFRET	Monte Carlo	Simple thresholding is used for an initial assessment of the rate constants. A computationally-intensive Monte Carlo simulation is then used to find simulated trajectories that contain the same rate and error pattern as the experimental ones to guess a possible truth. Compare the two and adjust the guess for the next iteration. Noisy data is binned for the initial thresholding. (LAD: least absolute deviations).	LAD	68% CI

All tools are detailed in the Supplementary Methods.
BIC Bayesian information criterion, CI confidence interval.

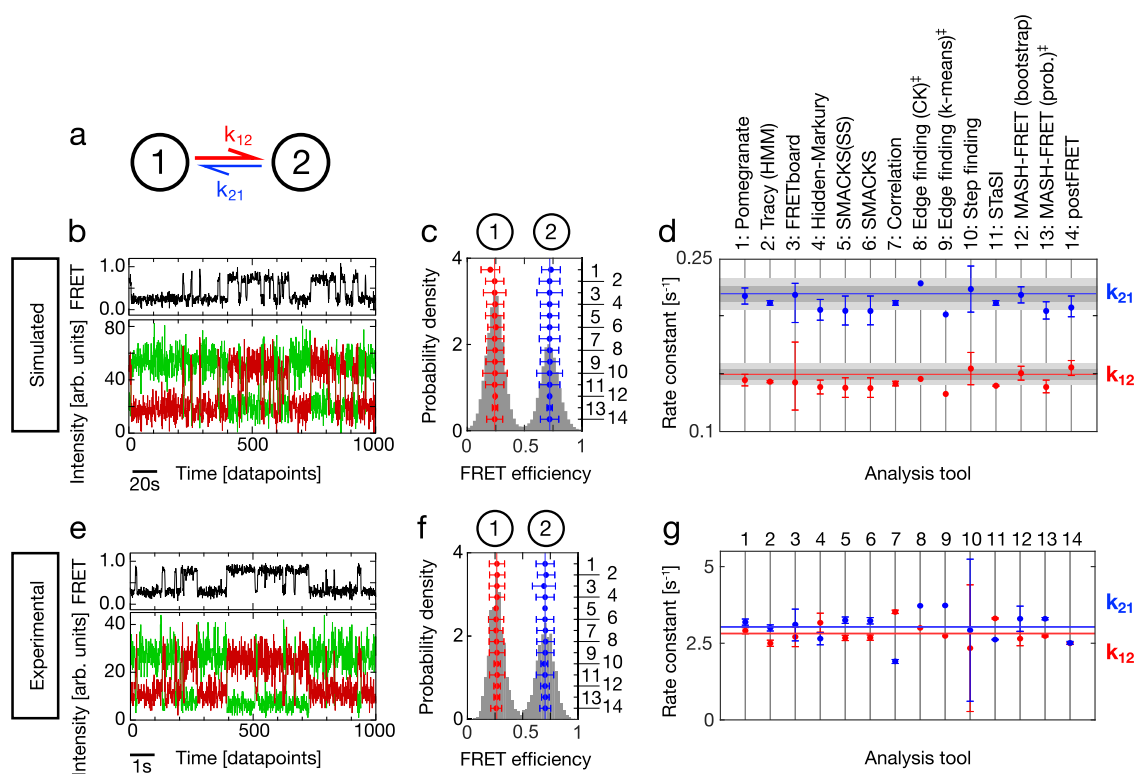


Fig. 2 | Quantification of simulated and experimental kinetics between two states. **a** Illustration of the kinetic model with two states (circles) connected by forward and backward rate constants: k_{12} and k_{21} . **b** A simulated FRET trace showing the donor and acceptor fluorescence intensity (green, red) and the FRET efficiency (FRET, black), representative for the dataset used in **(c, d)**: $n(\text{traces}) = 75$, $n(\text{datapoints}) = 59,486$, sampling rate = 5 Hz, time per datapoint = 200 ms. **c** FRET efficiency histogram (gray) with assigned states on top and inferred FRET efficiencies in red and blue. Numbers on the right axis refer to the analysis tools specified in **(d)**. Vertical lines indicate the mean over all tools. Sample size as in **(b)**. The error bars represent standard deviations. **d** Rate constants and uncertainties inferred from the dataset in **c** by different labs using the respective analysis tools. The ground truth (GT) is indicated by horizontal red and blue lines, the intrinsic uncertainty of the dataset (see text) is represented by dark gray (1σ) and light gray (2σ) intervals. Sample size as in **(b)**. Uncertainty measures (CI, SD) as listed in Table 1. **e** An

experimental time trace with colors as in **(b)**, representative for the dataset used in **(f, g)** with $n(\text{traces}) = 19$, $n(\text{datapoints}) = 226,100$, using 10 ms time bins resulting in 100 Hz sampling, kindly provided by B. Schuler. **f** FRET histogram with color code and axis labels as in **(c)**. Sample size as in **(e)**. The error bars represent standard deviations. No uncertainties were submitted for tool #5. **g** Inferred rate constants from the experimental dataset in **(f)**. Color code as in **(d)**. Horizontal red and blue lines indicate the mean of the inferred rate constants. Sample size as in **e**. Uncertainty measures (CI, SD) as listed in Table 1. Supplementary Fig. 2 shows the experimental data and analysis with ten times higher time resolution. ‡ denotes results that were submitted after the GT was known. The model size was restricted to two states. *FRETboard* and *Step finding* found erroneously large uncertainty intervals, which has been corrected in their latest software versions. See Supplementary Datafiles. Source data are provided as a Source Data file for panels **(c, d, f, and g)**.

than 17% from the GT (1% average deviation), and the inferred rate constants deviate with a maximum of 12% from the GT (5% average deviation), with a slight systematic underestimation in most cases, i.e., the determined rate constants were slower. *Pomegranate*, *FRETboard*, and *Step finding* infer the most accurate rate constants under the tested conditions (Table 3). The equilibrium constants $K = k_{21}/k_{12}$ vary generally less since systematic deviations balance each other in this case (Supplementary Fig. 1a). In contrast, the reported uncertainty measures vary greatly, independent of the analysis type (0.4% to 21% relative to the inferred rate constant). For comparison, we estimated the minimal uncertainty given the finite size of the dataset, by quantifying the standard deviation of the rate constants obtained from one million simulated samples (see Methods). This standard deviation is $\geq 3\%$ of the rate constants for the provided dataset (gray and light gray bars in Fig. 2b shown for 1σ and 2σ , respectively). Thus, most analysis tools reported reasonable uncertainty estimates, while some tools reported uncertainties that are smaller than this lower limit (*Tracy*, *Correlation*, *STaSI*) or provided no uncertainty measures (*Edge finding*). *FRETboard* version 0.0.2 reported consistently very large uncertainties, which was solved in their latest software version 0.0.3 (ref. 35, cf. Supplementary Datafiles). *Step finding* version 0.0.1 initially found erroneously large uncertainties that have been corrected in the latest software version 0.0.2 (cf. Supplementary Datafiles). We note that

various methods are currently in use for estimating uncertainties which complicates the direct comparison.

Next, we consider experimental data (see Methods), which naturally contains all typical noise sources and experimental artefacts (Fig. 2e–g). As there is no GT for experimental data, we assessed the consistency of the inferred FRET efficiencies and rate constants using the coefficient of variation (CV, i.e., the standard deviation divided by the mean). We found excellent agreement for all inferred FRET efficiencies (CV $\leq 2\%$). The rate constants vary by 12% and 16% (CV for k_{12} and k_{21} , respectively), consistent with the variation found for simulated data (Fig. 2d). Again, no correlation of the rate constants with respect to the analysis approach is evident, but the tendency of a given tool for large or small uncertainties is conserved (Fig. 2d, g), with *FRETboard* and *Step finding* reporting the largest uncertainties, and *STaSI*, *MASH-FRET (prob.)*, *postFRET*, and *Correlation* the smallest uncertainties. In most cases, the equilibrium constants (Supplementary Fig. 1b) agree well with each other and with the equilibrium populations of the FRET histogram, while some results are inconsistent with the latter (*Hidden-Markury*, *Correlation*, *STaSI*, and *postFRET*).

One important factor in dynamic smFRET data is the signal-to-noise ratio (SNR), which depends on the acquired signal per data point and can be controlled by the integration time (also known as exposure time). We explicitly tested the effect of a ten-fold shorter integration

time. On the one hand, this offers better sampling of fast kinetics due to the increased time resolution (1 kHz instead of 0.1 kHz sampling), but, on the other hand, it results in a lower signal-to-noise ratio which is more challenging for state identification. In addition, at 1 kHz sampling, the data shows single-photon discretization and non-Gaussian noise (Supplementary Fig. 2a, b), thus deviating from the basic assumptions underlying most of the considered analysis tools. Indeed, the overall agreement of the rate constants at this lower SNR was reduced: CV = 33% and 45% for k_{12} and k_{21} , respectively (Supplementary Fig. 2c), indicating that the benefit of the increased time resolution is minor in this case. Nevertheless, the equilibrium constants agree very well again (CV = 2%, when excluding the two clear outliers in Supplementary Fig. 2d) due to the cancelation of systematic shifts for both rate constants (Supplementary Fig. 2e). Comparing the rate constants inferred at 1 kHz and 0.1 kHz sampling, *pomegranate*, *Tracy*, *Correlation*, *MASH-FRET*, and *Step finding* reported similar values (Supplementary Fig. 2e), while *STaSI* inferred slower rate constants for faster sampling. Conversely, *FRETboard*, and *SMACKS* inferred faster rate constants for faster sampling, either due to fitting noise or due to short events that are missed at lower time resolution. The latter is less plausible, given that the inferred rate constants are 20-fold smaller than the 0.1 kHz sampling rate. Thus, a comparison between 0.1 kHz and 1 kHz sampling can serve to estimate the robustness of the analysis tools towards non-Gaussian noise. Taken together, fundamentally different analysis approaches inferred consistent rate constants and FRET efficiencies from a simple, two-state system both for simulated data and experimental data with varied SNR.

Directional sequences in a non-equilibrium steady-state system

Many biomolecular systems involve more than just two functionally relevant states, leading to more intricate kinetic models with more rate

constants and, hence, more degrees of freedom. Such systems with three or more states can show a conceptually unique thermodynamic phenomenon: the non-equilibrium steady-state, in which a biomolecule, such as a motor protein or a molecular machine such as F_0F_1 -ATP synthase, is driven by continuous external energy input, e.g. in the form of a chemical gradient³⁶, light^{37,38}, or ATP. As a result, conformational states may appear in a preferred sequence order, causing a non-zero net flow, e.g. for the 3-state system depicted in Fig. 3a:

$$\Delta G_{1 \rightarrow 2 \rightarrow 3 \rightarrow 1} = -k_B T^* \ln \left(\frac{k_{21} \cdot k_{32} \cdot k_{13}}{k_{12} \cdot k_{23} \cdot k_{31}} \right) \neq 0 \quad (1)$$

The unique ability to directly observe the non-equilibrium steady-state is a prime example of the merits of single-molecule studies. Hence, we investigated it explicitly, using smFRET data simulated with a kinetic 3-state model and a non-zero counter-clockwise flow: $\Delta G_{1 \rightarrow 2 \rightarrow 3 \rightarrow 1} < 0$ (Fig. 3a, b). As an additional challenge, this data contained fluorescence intensity variation between individual dye molecules, as observed in experimental data due to varied local dye environment and orientation, inhomogeneities in excitation intensity and polarisation, and also variations in detection efficiency³⁹.

All analysis tools found the three clearly separated FRET efficiency populations (Fig. 3c), while the inferred rate constants varied more than for the 2-state systems above (Fig. 3d). Most tools systematically underestimated k_{13} and k_{31} and overestimated all other rate constants. This may be attributed to the inevitable effect of time discretization and related intensity averaging: time-weighted averaging (e.g. camera blurring) of the FRET efficiencies can lead to mid-FRET observations that are indistinguishable from those caused by a bona fide biomolecular conformation. While, at the single datapoint level this discretization artefact cannot be prevented, the inference accuracy may

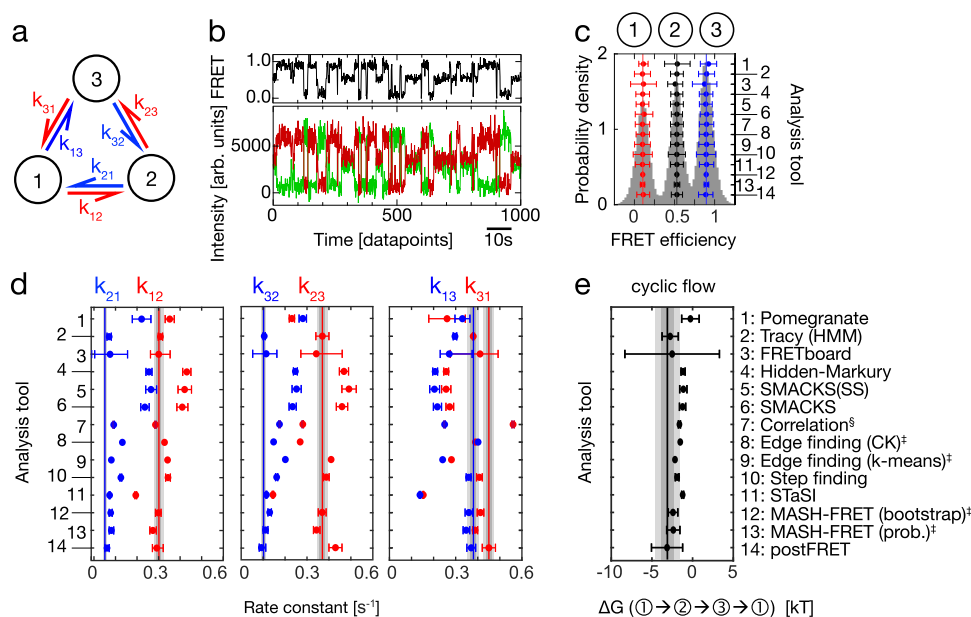


Fig. 3 | Quantitative analysis of a non-equilibrium steady-state system. **a** An illustration of the simulated three-state model with a counter-clockwise net flow. States (circles) are connected by forward and reverse rate constants as specified. **b** A simulated smFRET trace with donor and acceptor fluorescence intensity (green, red) and FRET efficiency (FRET, black), representative for the dataset used in **(c, d, e)**: $n(\text{traces}) = 150$, $n(\text{datapoints}) = 82,594$, sampling rate = 10 Hz, time per datapoint = 100 ms. **c** SmFRET histogram overlaid with the inferred FRET efficiencies (right axis, numbers as in **e**) and assigned states on top. Sample size as in **(b)**. The error bars represent standard deviations. **d** Inferred rate constants are shown in red and blue as specified. Vertical lines indicate the GT. The intrinsic uncertainty of the dataset is represented by dark gray (1σ) and light gray (2σ)

intervals. Sample size as in **(b)**. Uncertainty measures (CI, SD) as listed in Table 1. Analysis tools are numbered as in **(e)**. **e** The inferred cyclic flow in the counter-clockwise direction determined by calculating ΔG from Eq. (1) and compared with the GT value (solid vertical line). The uncertainty intervals (dark and light gray) are plotted as in **(d)**. Sample size as in **(b)**. Uncertainty measures (CI, SD) as listed in Table 1. Additional simulations to validate the dataset are shown in Supplementary Fig. 3. ‡ denotes results that were submitted after the GT was known. Edge finding did not report uncertainties. § denotes that the misassignment of start and end states was corrected after the GT was known. Source data are provided as a Source Data file for panels **(c, d, and e)**.

be improved by treating discretization-induced averaging explicitly in the analysis^{33,40}; or using pulsed illumination to reduce blurring^{41,42}. Overall, *postFRET* and *Tracy* inferred the most accurate rate constants with average GT deviations of 9% and 14%, respectively. As shown in Fig. 3e, qualitatively, the net flow was correctly resolved (most accurately by *postFRET*, *Tracy*, and *FRETboard*), while quantitatively it was mostly underestimated, which we attribute to the aforementioned systematic misallocation of transitions between states 1 and 3. For this simulated dataset, the theoretical lower limit of the uncertainty (as introduced above for the 2-state system) is smaller because the dataset is larger. About half of the tools reported uncertainties that are in line with this lower limit (grey intervals), while the other half reported none or too small uncertainties. We would like to stress that such a quantification of net flow is only meaningful when no detailed balance constraints are imposed during the rate inference, which was the case for the tools considered here. Altogether, the rate constants of the non-equilibrium 3-state system with intensity variation were less accurate than those of the 2-state system, and also the uncertainty estimation was challenging in this case. Nevertheless, the steady-state flow was qualitatively well resolved by most tools.

States with overlapping FRET efficiencies

Many biological systems show multi-exponential dwell-time distributions with long and short dwell times for the same apparent FRET state^{6,43–45}. This can, for example, arise when the one-dimensional

reaction coordinate spanned by the FRET pair is not sufficient to uniquely identify structural states in 3D space. Such kinetic heterogeneity is difficult to interpret because transitions between states with identical or overlapping FRET efficiencies cannot be directly observed in the recorded time traces, while they can often be inferred kinetically. To investigate this case, we simulated kinetic heterogeneity based on a four-state model (Fig. 4a) where states 1 and 2 have the same low-FRET efficiencies, and states 3 and 4 have the same high-FRET efficiencies. Again, the fluorescence traces included intensity variations between FRET pairs as observed in the experiment (introduced in the previous section), and also donor and acceptor blinking was included, as an additional imperfection of the data. Figure 4b shows example traces from the simulation and Fig. 4c shows the FRET efficiency histogram with two peaks. Without a priori knowledge of the model size, most tools identified the correct number of two apparent FRET states, while *FRETboard* used three FRET states to describe the data. *Edge finding* was not developed to deal with such kinetic heterogeneity, and *Pomegranate*, *Correlation*, *STaSI* and *MASH-FRET (bootstrap)* reported FRET efficiencies but no kinetic models. In the following, we use cumulative dwell-time distributions derived from each inferred model (Fig. 4d, detailed in Methods) to compare models with the correct number of FRET states but differences in the kinetic model, such as the connectivity of states or the number of hidden states (rate constants of all inferred models are reported in the Supplementary Table 1, and in the Supplementary Datafiles). Out of the seven independently inferred

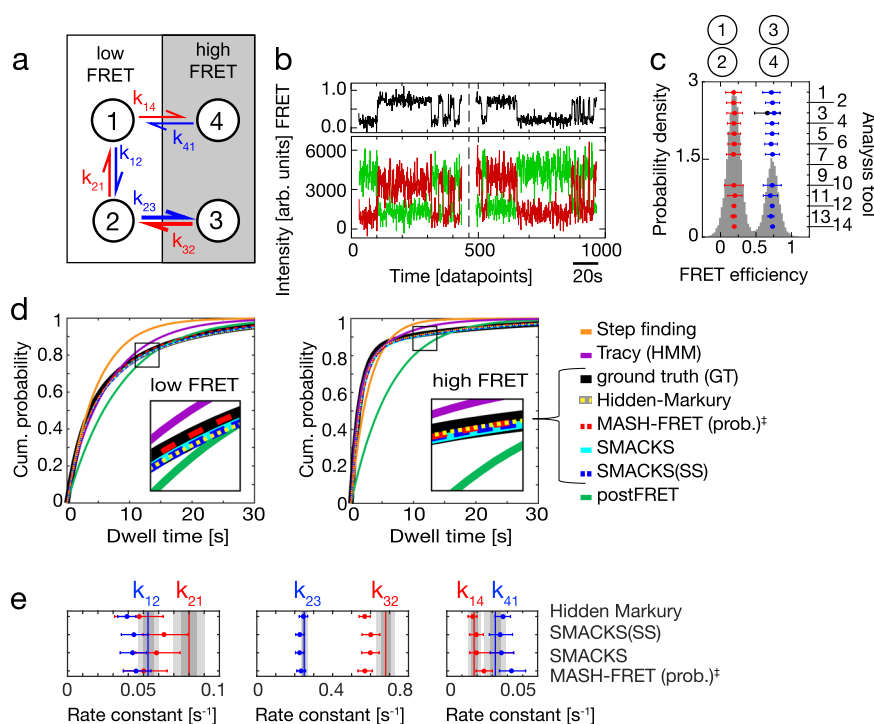


Fig. 4 | Resolving kinetic heterogeneity: states with indistinguishable FRET efficiencies but different kinetics. **a** An illustration of the simulated GT model with states (circles) connected by forward and reverse rate constants. States 1 and 4 as well as states 2 and 3 have indistinguishable FRET efficiencies, causing kinetic heterogeneity. **b** Two simulated FRET traces offset in time with donor and acceptor fluorescence intensity (green, red) and FRET efficiency (FRET, black) are shown, representative for the dataset used in **(c)**: $n(\text{traces}) = 250$, $n(\text{datapoints}) = 56,794$, sampling rate = 5 Hz, time per datapoint = 200 ms. **c** FRET histogram with inferred FRET efficiencies overlaid (right axis; legend as in Table 1 and in all Figures). Sample size as in **(b)**. The error bars represent standard deviations. **d** Comparison of cumulative dwell time distributions derived from the kinetic models with two FRET states (detailed in Methods). The GT histogram is shown as a bold black line. Insets show zoomed-in views of the data indicated by the squares. **e** Quantitative

comparison of the four most accurately inferred kinetic models: the GT values are represented as red and blue vertical lines. Sample size as in **(b)**. Uncertainty measures (CI, SD) as listed in Table 1. The intrinsic uncertainty of the dataset is shown as dark gray (1 σ) and light gray (2 σ) intervals. Beyond the six displayed rate constants, these additional rate constants were inferred: for Hidden Markury $k_{31} = 0.045$ and $k_{34} = 0.003$, for SMACKS $k_{13} = 0.0001$, $k_{31} = 0.0055$, $k_{34} = 0.0034$, for MASH-FRET (prob.) \dagger $k_{31} = 0.033$. All inferred values of all models are reported in the Supplementary Tables 1 and in the Supplementary Datafiles. \dagger denotes results that were submitted after the GT was known. No results were reported by *Edge finding*. Participants were informed that kinetic heterogeneity may be involved, but not in which configuration. Source data are provided as a Source Data file for panels **(c)**, **(d)**, and **(e)**.

kinetic models, the two models without kinetic heterogeneity (by *Step finding* and *postFRET*) show the largest deviations from the GT, as these models cannot reproduce the multi-exponential nature of the dwell-time distribution. On the other hand, the four models inferred by the HMM-based *Hidden-Markury*, *SMACKS*, *SMACKS(SS)*, as well as *MASH-FRET (prob.)* show good agreement with the GT and overlay the GT in the low- and high-FRET case (compare Fig. 4d). A quantitative comparison of these four models and their uncertainties with the GT is provided in Fig. 4e. It shows accurate rates and some collective underestimation of rates k_{12} , k_{21} , k_{32} , likely due to missed fast events. In addition, some rates were inferred that are not present in the GT (see Fig. 4e caption). Taken together, several tools inferred the correct model size (number of states) and accurate cumulative dwell-time distributions, but model selection – and in particular the selection of the correct connectivity of states – remains a main challenge in inferring kinetic information from smFRET trajectories. It is, however, encouraging that several analysis tools can already deduce kinetic models that closely reproduce the GT even under difficult conditions involving kinetic heterogeneity.

Full complexity of a black-box experiment

Encouraged by the previous results, we tested all tools vis-à-vis the full experimental complexity to see if they perform similarly as in the simulated case (Fig. 4). Three experimental datasets of the same

biological system (protein binding to a fluorescently labelled DNA, see Methods), under different experimental conditions and thus different kinetic behaviour, served as a test case. However, the analysts had no prior information on the molecular system causing the dynamics. This means that all the effects discussed so far could potentially be present in these experimental datasets: multiple FRET states, diverse noise sources, fluorophore blinking, directional steady-state flow and kinetic heterogeneity. In addition, the fluorescence intensity variation between single molecules was particularly high in these datasets (see Fig. 5a, d, g), which complicated the inference of the number of states and rates involved (subsequently referred to as model selection). Under these complex conditions, the inferred number of FRET states (Fig. 5b, e, h) varied more than in the simulated case (discussed in Fig. 4). Most tools found two FRET states (Fig. 5b, e, h, some of them including kinetic heterogeneity), but also three, four, or more different FRET states were reported (Supplementary Fig. 4), and the kinetic rate constants varied accordingly. Given the inherent lack of GT information in experimental data, we cannot quantitatively assess the accuracy in this comparison. To balance this fact, we qualitatively compare the inferred results for all three datasets. The 6–7 models with two FRET states (and possibly more hidden states) are compared in (Fig. 5c, f, i). Other models with three, four, or more FRET states are compared in Supplementary Figs. 4–6. (All inferred rate constants are given in Supplementary Tables 2–4 and Supplementary Datafiles). Again, we

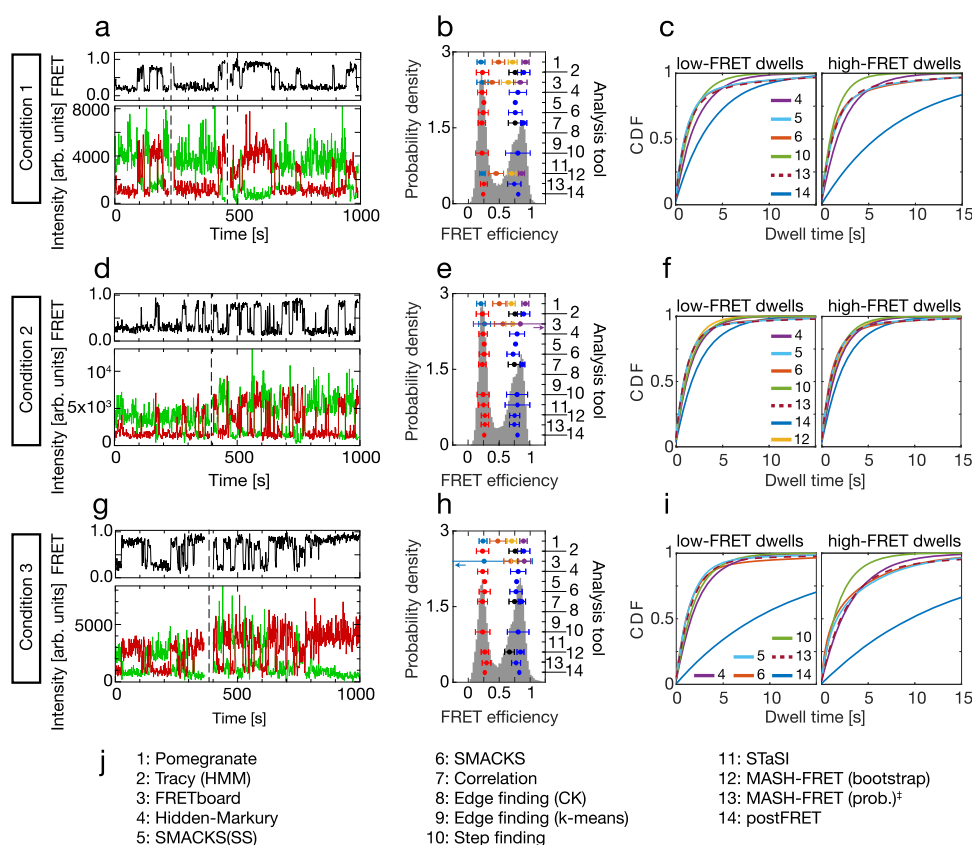


Fig. 5 | Increased experimental complexity. Results inferred from three experimental datasets where naturally no GT exists. **a, d, g** Experimental traces, offset in time and separated by dashed vertical lines, with donor and acceptor fluorescence intensity (green, red) and FRET efficiency (FRET, black), representative for the datasets used in **(b, c)**, **(e, f)**, **(h, i)**, respectively, with n (traces): 134, 163, 118; and n (datapoints): 36,604, 37,067, 43,512; sampling rate = 33 Hz, time per datapoint = 30 ms. All three datasets were kindly provided by M. Schlierf. **b, e, h** FRET efficiency histograms and FRET efficiencies inferred by the analysis tools numbered as in **(j)**. Sample sizes as in **(a, d, g)**, respectively. The error bars represent standard deviations. For clarity, only the smallest reported model is shown for each analysis tool,

up to a maximum of four FRET states. All inferred FRET efficiencies are shown in Supplementary Fig. 4, and all inferred results are provided in the Supplementary Tables 2–4 and in the Supplementary Datafiles. Purple arrow in **(e)**: the error bar extends to 1.61. Teal arrow in **(h)**: the error bar extends to -0.53 . **c, f, i** Cumulative distribution functions (CDF) of the dwell-times simulated using the inferred kinetic models with two FRET states, obtained with the tools numbered as in **(j)**. **j** Legend with all analysis tools. No results were reported by *Edge finding*. [‡] denotes results that were submitted after all other results were known. Source data are provided as a Source Data file for panels **(b, c, e, f, h, and i)**.

use cumulative dwell-time distributions (cf. last section) derived from each inferred model (Fig. 5c, f, i) to facilitate the comparison of models with the same number of FRET states but possibly different state connectivity. The distributions are thus single- or double-exponential depending on the reported kinetic model. The five tools that inferred two FRET states and qualitatively similar kinetic models under all three conditions despite different analysis approaches, are the HMM-based *Hidden-Markury* and *SMACKS*, as well as *Step finding*, *postFRET* and *MASH-FRET (prob.)*. While *postFRET* consistently inferred slower rate constants, the qualitative agreement among the other five tools is surprisingly good ($CV \leq 25\%$ for the average residence time) despite the complexity of the input data, the missing prior knowledge about the system, and the different analysis approaches used.

Altogether, we conclude that model selection and state allocation are currently the key challenges in the analysis of kinetic data. In this study, we focused only on the analysis of fluorescence intensity and FRET efficiency data. The addition of complementary information from simulations or experiments (e.g., static molecular structures and other observables, such as fluorescence lifetimes, anisotropy, and more) may help to elucidate complicated or otherwise under-determined systems^{30,46,47}.

Discussion

In this blind study, we compared eleven kinetic analysis tools for the inference of quantitative kinetic rate constants based on single-molecule FRET trajectories. We explicitly considered the major (kinetic) challenges that the single-molecule experimentalists are typically confronted with: determining the best model to describe the data, especially with multiple FRET states, a varying signal-to-noise ratio, directional non-equilibrium steady-state flow, and kinetic heterogeneity (i.e., states with indistinguishable FRET efficiency but distinct kinetics). We assessed the inferred FRET efficiencies, rate constants, and the reported uncertainties, based on three simulated datasets and four experimental datasets from two biological systems measured using two different setups in different laboratories. The simulated data allowed us to directly assess the accuracy of the inferred rate constants using the known ground truth model and to judge the plausibility of the reported uncertainty measures, while the experimental data shows the relevance and validity of this study.

We found that the number of states was correctly inferred by all tools, as long as their FRET efficiencies were clearly separated (Figs. 2 and 3). In the presence of kinetic heterogeneity with overlapping FRET states, model selection was more challenging (Fig. 4). In this case, three tools successfully inferred models that accurately reproduce the dwell-

time distribution of the GT despite overlapping FRET states (*Hidden-Markury*, *MASH-FRET*, *SMACKS*). In general, the accuracy of the rate constants inferred by all tools decreased with increasing model size and complexity, where time discretization artefacts and inter-trace intensity variation become increasingly challenging. The equilibrium constants and steady-state flow were more accurately inferred than individual rate constants due to the cancellation of systematic errors (Supplementary Figs. 1 and 2d, e, Fig. 3). Caution is advised with the uncertainties of rate constants since different uncertainty measures are reported by different approaches. Even for small models (Figs. 2 and 3), we found that some uncertainty estimates were smaller than the uncertainties caused by the finite dataset size, while interestingly, more plausible uncertainties were reported for the more complex model in Fig. 4 (Supplementary Fig. 4). In general, the comparison of uncertainties is complicated by the fact that no common standard exists and the mathematical interpretation of the reported uncertainty intervals differs from tool to tool.

When comparing various analysis frameworks, model-free approaches are generally considered advantageous for an unbiased data analysis. However, HMM-based tools (that compare several input models based on scarcity criteria) were found to be more robust towards data heterogeneity (Figs. 4 and 5, Supplementary Fig. 2). Nevertheless, we did not observe a clear overall clustering of the inferred rate constants with the underlying analysis framework, likely due to differences in the data handling beyond the used algorithms (e.g. supervised, semi-supervised, or unsupervised inference). The total analysis durations (processing and computation) ranged from a few minutes to several hours depending on the analysis tool and the model size, with *StaSI* and *Step finding* ranking among the fastest, and *SMACKS* among the slower tools. In the course of this study, multiple conceptual oversights could be found and solved in a number of tools, which is a direct constructive result of this collaborative comparison study that led to the general recommendations stated in Table 2. Additionally, a simple shareable smFRET data format was introduced (Supplementary Note 1) and utilized by all twelve labs working in diverse software environments. We anticipate that this data format will facilitate future collaborations and significantly lower the barrier for an experimentalist to adopt a newly developed analysis tool if it supports the accepted format.

Looking ahead, a particularly promising outlook is the possibility to characterize individual states with individual noise patterns more accurately, using machine learning. Recently, deep learning approaches have been developed for the unbiased selection of single molecule traces for further kinetic analysis^{48,49}. Similar approaches could be

Table 2 | General recommendations for users and developers of kinetic inference tools^a

(i)	As a general consistency test, the inferred kinetic model (connectivity and rate constants) can be simulated and the output of the simulation compared to the original input data. For example, the simulator used herein is publicly available as a simple and powerful (MATLAB) tool to test whether the proposed model can generate data analogous to the original input, e.g. regarding FRET histogram, smFRET traces, etc.
(ii)	Potential biases in the analysis (e.g. regarding model size, state occupation, etc.) can be revealed by subjecting the re-simulated data (with known ground truth) to the same analysis approach as the experimental data.
(iii)	Where possible, kinetic models with a specific number and connectivity of states are preferred over mean residence times, since the latter leave the individual transition rate constants undetermined for more than 2 states.
(iv)	Uncertainty measures are necessary indicators of significance, and a unified standard would greatly improve their comparability. The 95% confidence interval was the most frequently used uncertainty measure in this study, and we encourage its use as a common standard for the future.
(v)	Benchmarking new analysis tools using datasets of varied complexity – including models with more than 2 states – can reveal systematic errors, e.g. regarding the weighting of multiple rate constants that depopulate a given state, an issue encountered in this study.
(vi)	Benchmarking new software with established test data helps the potential users to judge the added benefits of newly introduced analysis tools. The diverse datasets used herein are publicly available and can serve to assess a tool's performance under varied experimental conditions.
(vii)	Supporting broadly accepted file formats for newly developed analysis tools facilitates fast dissemination in the field. We offer the simple format described in Supplementary Note 1, which proved to be very useful for this study.

^aIn the course of this study, several difficulties with the analysis of kinetic data have become apparent. Out of this experience, we have compiled a list of recommendations for those developing and using kinetic analysis tool.

Table 3 | Summary of the test conditions for the individual datasets, including the prior information on ground truth (GT) and number (N) states, as well as three data characteristics: kinetic heterogeneity, photo-physics, and signal-to-noise ratio (SNR)

	GT known? ^a	N states predefined?	Kinetic heterogeneity	Photo-physics ^b	SNR ^c
Fig. 2 (sim.)	No	Yes, 2.	No	Clean	4
Fig. 2 (exp.)	No	Yes, 2.	Not observed	Mainly clean	4
Fig. 3 (sim.)	No	No	No	Intensity variation	3
Fig. 4 (sim.)	No	No	Yes	Intensity variation & blinking	4
Fig. 5 (exp.)	No	No	Yes observed	Intensity variation & blinking	3

^aExceptions are labelled with a dagger in all figures.

^bSee simulation parameters in Supplementary Table 5.

^cThe SNR was obtained from the FRET efficiency histogram using Gaussian fits and $\text{SNR} = |\mu_1 - \mu_2| / \sqrt{\sigma_1^2 + \sigma_2^2}$.

envisioned for a model-free kinetic analysis, which bears the potential to improve model selection significantly^{18,50}. Demonstrating such new tools using public training datasets and supporting the simple file format introduced in this study, will accelerate the dissemination of the newest theoretical developments within the community of single-molecule experimentalists. Additional important aspects for future benchmark studies include the ‘data greediness’ of a given tool, e.g. the amount of data and the number of transitions-per-trace (given by the relation of biomolecular dynamics versus bleach rate) that are required for accurate rate inference.

In conclusion, this blind study on kinetic inference from smFRET data further validates the use of smFRET in deciphering biomolecular rates. It unequivocally reveals the current strengths and weaknesses of the various analysis approaches when tested against frequently encountered phenomena in smFRET experiments, and provides a reference standard for the continuous developments in this active field. We anticipate that this study will serve the community as a guide for data interpretation, spark future developments in kinetic inference, and therefore help to advance our understanding of biomolecular dynamics leading to function.

Methods

Procedure of this benchmark study

The need for a comparison of analysis tools for smFRET trajectories has grown with the increasing number of smFRET users and published tools. This was addressed at the Biophysical Society Meeting 2019 (Baltimore, US) by initiating a kinetic software challenge, short kinSoftChallenge. In line with more efforts to assess, promote, and potentially standardize experimental and analytical smFRET procedures (Refs. 2,3, 51 in preparation), the kinSoftChallenge represents an important step aimed to improve the reliability and accuracy of kinetic inference from smFRET trajectories. In a first round of the study (July 2019 to November 2019), the participants received three simulated datasets (shown in Figs. 2, 3, and 4). In the second round (December 2019 to February 2020), the participants analyzed the experimental dataset shown in Fig. 5. Experimental data with high and low SNR was compared in a third round (November to December 2020, shown in Fig. 2, and Supplementary Fig. 2). The individual test conditions are described in the text and summarized in Table 3. All challenge rounds were conducted as blind studies, i.e., the participants did not have ground truth information during data analysis (exceptions are labeled with a dagger in all Figures).

Simulation of smFRET trajectories

In short, simulated smFRET datasets were generated to mimic fluorescence traces obtained by TIRF-based experiments. State trajectories were modeled with a continuous-time approach and later discretized. Similar to experiments, this allows state transitions to occur during the integration time window (time bin of the detector). Noise was added to the fluorescence intensity traces using experiment-derived parameters to generate realistic data.

In more detail, for each molecule a continuous-time state trajectory was simulated based on the kinetic model, as specified by a transition rate matrix. A summary of the specific simulation parameters is given in the Supplementary Table 5 and all configuration files with all parameters are provided as Supplementary Datafiles. First, the trace length was determined from an exponential distribution described by the rate of photobleaching. The trace length was rejected if it was shorter than a minimal trace length and truncated to a maximal trace length (see Supplementary Table 5). Then, a random initial state was chosen based on the probability of being in a particular state given the transition rate matrix. Starting from this state, dwell times for all possible transitions to the other states were drawn randomly from exponential distributions defined by the transition rates, and the shortest dwell time determined the transition and the new state of the system. This process was repeated until the full trace length was reached. This state trajectory was then converted into discrete-time fluorescence intensity traces using a specified sampling rate. For each time bin (i.e., camera frame), the donor and acceptor intensities upon donor excitation and the intensity of the acceptor upon acceptor excitation were drawn from state-specific Gaussian distributions (specified by the means μ_i and covariance matrices given in the configuration file). The intensity in each channel during a time bin is given by the weighted average of all states visited during this specific time bin.

Typically, single-molecule fluorescence traces show variations in the fluorescence level between individual molecules, due to, amongst others, local variations in excitation power and local dye environment¹³. To take these variations into account, two additional sources of per-trace intensity variations were considered for the simulated data shown in Figs. 3 and 4. First, for each molecule, individual intensity levels for each state were chosen. To do so, the intensity level was drawn from an empirically determined state-specific Gaussian distribution (with mean μ_i and standard deviation $5 \cdot \sqrt{\mu_i}$). Second, for each molecule, an individual brightness factor was determined by 1.20^r where r was randomly chosen from the interval $[-1, 1]$. Thus, this factor is distributed in the interval $[0.83, 1.20]$ and all channels were multiplied by the same factor. For the simulated data shown in Fig. 4, independent blinking of the donor and acceptor dye was modeled by a simple 2-state system (“bright”, “dark”). In the case of an acceptor dark state, the FRET efficiency was set to zero. Details are given in Supplementary Table 5.

Five hundred additional datasets from the same parameter set were created and compared, to validate that the dwell time distribution of the dataset used in this study shows the expected behaviour (see Supplementary Fig. 3). Configuration files with all simulation parameters (including the ground truth for the kinetic models) for the synthetic data in Figs. 2, 3, and 4 can be found in the Supplementary Datafiles. The MATLAB scripts used for the simulation are publicly available at: www.kinSoftChallenge.com and <https://doi.org/10.5281/zenodo.5701310>. A Supplementary Table with the simulation

parameters and a Supplementary Note on the file format used herein are provided in the Supplementary Information file.

Estimated minimal uncertainty of rate constants inferred from simulations

Because of the finite number of traces per datasets, only a limited random sample of dwell times is observed for each given transition, resulting in a variation of the rate constants inferred from different datasets with identical ground truth. In order to estimate this lower bound of the uncertainty for the inference of rate constants from a finite dataset, we randomly drew the same number of dwell times as provided in the simulated challenge dataset from an exponential distribution with time constant $\tau = 1/k$. The maximum likelihood estimator (MLE) for the rate constant that produced this set of dwell times Δt is given by $1/\Delta t$. This calculation of the MLE was repeated one million times. The standard deviation of these 1 million MLEs is a function of the number of dwell times present in the challenge data set – the more dwell times are observed, the narrower the MLE distribution – and hence, it depends on the transition rate constants and the total observation time. We used this standard deviation as an estimate of the lower bound for the uncertainty of inferred rate constants from the simulated datasets.

Simulation of cumulative dwell-time distributions from inferred kinetic models

In order to compare submissions with the same number of FRET states but different underlying kinetic models (i.e., number of hidden states and connectivity), we simulated dwell times from the submitted kinetic models for the three datasets shown in Figs. 4 and 5. This yields cumulative dwell-time distributions that are characteristic for the kinetic model. Dwell times were accumulated from simulations of continuous time state trajectories (Supplementary Note 1) that included roughly 200x (Fig. 4d) or 400x (Fig. 5c, f, i) more time points than the original datasets.

Origin of the experimental datasets

The experimental data shown in Fig. 2 and Supplementary Figs. 1, 2 was kindly provided by Benjamin Schuler. It shows the interaction between the nuclear-coactivator binding domain of CBP/p300 (NCBD) and the intrinsically disordered activation domain of the steroid receptor coactivator 3 (ACTR), measured using confocal single-photon detection⁵. The experimental data shown in Fig. 5 and Supplementary Fig. 4 was kindly provided by Michael Schlierf. It shows binding of single-strand binding proteins (SSB) to a fluorescently labelled DNA hairpin, measured in prism-type total-internal reflection fluorescence (TIRF) mode using camera-based detection (EMCCD)⁴.

Procedures of the kinetic analyses

Detailed descriptions of all analysis tools are provided in the Supplementary Methods in the Supplementary Information file. All inferred results are provided as Supplementary Datafiles.

Reporting summary

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

Data availability

The simulated and experimental smFRET data used in this study are available at www.kinsoftchallenge.com and <https://doi.org/10.5281/zenodo.5701310>. All inferred results are provided in the Supplementary Data files. Supplementary figures, notes, and methods are provided in the Supplementary Information file. Source data are provided with this paper.

Code availability

The simulation code and parameters to generate the simulated datasets are available at <https://doi.org/10.5281/zenodo.5701310>. All software tools are available: Pomegranate v0.0.1 at <https://github.com/hatzakislab/DeepFRET-GUI>; Tracy v4.4.8 upon request as it is being replaced by a new program for multi-color analysis (contact: Don C. Lamb [d.lamb@lmu.de], requests will be addressed as soon as possible, typically within 1 week); FRETboard v0.0.3 at <https://github.com/cvdelannoy/FRETboard>; Hidden-Markury v0.0.1 at <https://github.com/ChristianGebhardt/Hidden-Markury>; SMACKS v1.4 at <https://github.com/sciSonja/SMACKS>; Correlation v0.1b at <https://doi.org/10.5281/zenodo.5512005>; Edge finding (CK and k-means) v0.0.1 at <https://www.physics.ncsu.edu/weninger/KinSoft.html>; Step finding v0.0.2 at <https://github.com/SMB-Lab/PyStepFinder>; StaSI v0.0.1 at <https://github.com/LandesLab/StaSI>; MASH-FRET v1.3.2 (bootstrap and probabilistic) at <https://github.com/RNA-FRETtools/MASH-FRET>; post-FRET v4.0 at <https://github.com/nkchenjx/postFRET>.

References

- Lerner, E. et al. Toward dynamic structural biology: two decades of single-molecule Förster resonance energy transfer. *Science* **359**, (2018).
- Hellenkamp, B. et al. Precision and accuracy of single-molecule FRET measurements—a multi-laboratory benchmark study. *Nat. Methods* **15**, 669–676 (2018).
- Lerner, E. et al. FRET-based dynamic structural biology: challenges, perspectives and an appeal for open-science practices. *eLife* **10**, e60416 (2021).
- Grieb, M. S. et al. Dynamic stepwise opening of integron attC DNA hairpins by SSB prevents toxicity and ensures functionality. *Nucleic Acids Res.* **45**, 10555–10563 (2017).
- Zosel, F., Soranno, A., Buholzer, K. J., Nettels, D. & Schuler, B. Depletion interactions modulate the binding between disordered proteins in crowded environments. *Proc. Natl Acad. Sci.* **117**, 13480–13489 (2020).
- Schmid, S. & Hugel, T. Controlling protein function by fine-tuning conformational flexibility. *eLife* **9**, e57180 (2020).
- Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**, 257–286 (1989).
- McKinney, S. A., Joo, C. & Ha, T. Analysis of single-molecule FRET Trajectories using Hidden Markov Modeling. *Biophys. J.* **91**, 1941–1951 (2006).
- Blanco, M. & Walter, N. G. Chapter 9 - Analysis of Complex Single-Molecule FRET Time Trajectories. in *Methods in Enzymology* (ed. Walter, N. G.) vol. 472 153–178 (Academic Press, 2010).
- König, S. L. B. et al. BOBA FRET: Bootstrap-based analysis of single-molecule FRET data. *PLOS ONE* **8**, e84157 (2013).
- Keller, B. G., Kobitski, A., Jäschke, A., Nienhaus, G. U. & Noé, F. Complex RNA folding kinetics revealed by single-molecule FRET and Hidden Markov Models. *J. Am. Chem. Soc.* **136**, 4534–4543 (2014).
- Pirchi, M. et al. Photon-by-Photon Hidden Markov Model analysis for microsecond single-molecule FRET Kinetics. *J. Phys. Chem. B* **120**, 13065–13075 (2016).
- Schmid, S., Götz, M. & Hugel, T. Single-molecule analysis beyond Dwell Times: demonstration and assessment in and out of Equilibrium. *Biophys. J.* **111**, 1375–1384 (2016).
- Zarrabi, N., Schluesche, P., Meisterernst, M., Börsch, M. & Lamb, D. C. Analyzing the Dynamics of single TBP-DNA-NC2 complexes using Hidden Markov Models. *Biophys. J.* **0**, (2018).
- Hon, J. & Gonzalez, R. L. Bayesian-estimated hierarchical HMMs enable Robust analysis of single-molecule kinetic Heterogeneity. *Biophys. J.* <https://doi.org/10.1016/j.bpj.2019.02.031> (2019).
- Bronson, J. E., Fei, J., Hofman, J. M., Gonzalez, R. L. & Wiggins, C. H. Learning rates and states from Biophysical Time Series: a Bayesian

- approach to model selection and single-molecule FRET data. *Bio-phys. J.* **97**, 3196–3205 (2009).
17. van de Meent, J.-W., Bronson, J. E., Wiggins, C. H. & Gonzalez, R. L. Empirical Bayes methods enable advanced population-level analyses of single-molecule FRET experiments. *Biophys. J.* **106**, 1327–1337 (2014).
 18. Sgouralis, I. et al. A Bayesian nonparametric approach to single molecule Förster Resonance Energy Transfer. *J. Phys. Chem. B* **123**, 675–688 (2019).
 19. Kinz-Thompson, C. D., Ray, K. K. & Gonzalez, R. L. Bayesian inference: The Comprehensive Approach to analyzing single-molecule experiments. *Annu. Rev. Biophys.* **50**, 191–208 (2021).
 20. Chung, H. S. et al. Extracting rate coefficients from single-molecule Photon Trajectories and FRET efficiency Histograms for a fast-folding protein. *J. Phys. Chem. A* **115**, 3642–3656 (2011).
 21. Gopich, I. V. & Szabo, A. Decoding the pattern of Photon colors in single-molecule FRET. *J. Phys. Chem. B* **113**, 10965–10973 (2009).
 22. Harris, P. D. et al. Multi-parameter photon-by-photon hidden Markov modeling. *Nat. Commun.* **13**, 1000 (2022).
 23. Liu, Y., Park, J., Dahmen, K. A., Chemla, Y. R. & Ha, T. A comparative study of multivariate and univariate Hidden Markov Modelings in Time-Binned single-molecule FRET data analysis. *J. Phys. Chem. B* **114**, 5386–5403 (2010).
 24. Greenfeld, M., Pavlichin, D. S., Mabuchi, H. & Herschlag, D. Single molecule analysis research tool (SMART): an integrated approach for analyzing single molecule data. *PLOS ONE* **7**, e30024 (2012).
 25. Kelly, D., Dillingham, M., Hudson, A. & Wiesner, K. A new method for inferring Hidden Markov Models from noisy time sequences. *PLOS ONE* **7**, e29703 (2012).
 26. Gurunathan, K. & Levitus, M. FRET fluctuation spectroscopy of diffusing biopolymers: contributions of conformational dynamics and translational diffusion. *J. Phys. Chem. B* **114**, 980–986 (2010).
 27. Opanasyuk, O. et al. Unraveling multi-state molecular dynamics in single-molecule FRET experiments. II. Quantitative analysis of multi-state kinetic networks. *J. Chem. Phys.* **157**, 031501 (2022).
 28. Schuler, B. Perspective: chain dynamics of unfolded and intrinsically disordered proteins from nanosecond fluorescence correlation spectroscopy combined with single-molecule FRET. *J. Chem. Phys.* **149**, 010901 (2018).
 29. Chung, H. S., Cellmer, T., Louis, J. M. & Eaton, W. A. Measuring ultrafast protein folding rates from photon-by-photon analysis of single molecule fluorescence trajectories. *Chem. Phys.* **422**, 229–237 (2013).
 30. Kilic, S. et al. Single-molecule FRET reveals multiscale chromatin dynamics modulated by HP1 α . *Nat. Commun.* **9**, 235 (2018).
 31. Watkins, L. P. & Yang, H. Detection of intensity change points in time-resolved single-molecule measurements. *J. Phys. Chem. B* **109**, 617–628 (2005).
 32. Shuang, B. et al. Fast step transition and state identification (STaSI) for discrete single-molecule data analysis. *J. Phys. Chem. Lett.* **5**, 3157–3161 (2014).
 33. Chen, J., Pyle, J. R., Sy Piecco, K. W., Kolomeisky, A. B. & Landes, C. F. A two-step method for smFRET data analysis. *J. Phys. Chem. B* **120**, 7128–7132 (2016).
 34. Gauer, J. W. et al. Chapter Ten - single-molecule FRET to measure conformational dynamics of DNA mismatch repair proteins. in *Methods in Enzymology* (eds. Spies, M. & Chemla, Y. R.) vol. 581 285–315 (Academic Press, 2016).
 35. de Lannoy, C. V., Filius, M., Kim, S. H., Joo, C. & de Ridder, D. FRETboard: Semisupervised classification of FRET traces. *Biophys. J.* **120**, 3253–3260 (2021).
 36. Diez, M. et al. Proton-powered subunit rotation in single membrane-bound F₀F₁-ATP synthase. *Nat. Struct. Mol. Biol.* **11**, 135–141 (2004).
 37. Croce, R. & van Amerongen, H. Light harvesting in oxygenic photosynthesis: Structural biology meets spectroscopy. *Science* **369**, eaay2058 (2020).
 38. Kühlbrandt, W. Bacteriorhodopsin — the movie. *Nature* **406**, 569–570 (2000).
 39. Edel, J. B., Eid, J. S. & Meller, A. Accurate single molecule FRET efficiency determination for surface immobilized DNA using maximum likelihood calculated lifetimes. *J. Phys. Chem. B* **111**, 2986–2990 (2007).
 40. Kilic, Z., Sgouralis, I. & Pressé, S. Generalizing HMMs to continuous time for fast kinetics: Hidden Markov Jump Processes. *Biophys. J.* **120**, 409–423 (2021).
 41. Farooq, S. & Hohlbein, J. Camera-based single-molecule FRET detection with improved time resolution. *Phys. Chem. Chem. Phys.* **17**, 27862–27872 (2015).
 42. Nicholson, D. A. & Nesbitt, D. J. Pushing camera-based single-molecule kinetic measurements to the Frame Acquisition Limit with Stroboscopic smFRET. *J. Phys. Chem. B* **125**, 6080–6089 (2021).
 43. Frauenfelder, H., Sligar, S. G. & Wolynes, P. G. The energy landscapes and motions of proteins. *Science* **254**, 1598–1603 (1991).
 44. Sunney Xie, X. Single-molecule approach to dispersed kinetics and dynamic disorder: Probing conformational fluctuation and enzymatic dynamics. *J. Chem. Phys.* **117**, 11024–11032 (2002).
 45. Henzler-Wildman, K. & Kern, D. Dynamic personalities of proteins. *Nature* **450**, 964–972 (2007).
 46. Feng, X. A., Poyton, M. F. & Ha, T. Multicolor single-molecule FRET for DNA and RNA processes. *Curr. Opin. Struct. Biol.* **70**, 26–33 (2021).
 47. Schuler, B. et al. Binding without folding – the biomolecular function of disordered polyelectrolyte complexes. *Curr. Opin. Struct. Biol.* **60**, 66–76 (2020).
 48. Thomsen, J. et al. DeepFRET, a software for rapid and automated single-molecule FRET data classification using deep learning. *eLife* **9**, e60404 (2020).
 49. Li, J., Zhang, L., Johnson-Buck, A. & Walter, N. G. Automatic classification and segmentation of single-molecule fluorescence time traces with deep learning. *Nat. Commun.* **11**, 5833 (2020).
 50. Sgouralis, I. & Pressé, S. An introduction to infinite HMMs for single-molecule data analysis. *Biophys. J.* **112**, 2021–2029 (2017).
 51. Agam, G. et al. Reliability and accuracy of single-molecule FRET studies for characterization of structural dynamics and distances in proteins. *bioRxiv* <https://doi.org/10.1101/2022.08.03.502619> (2022).
 52. Felekyan, S., Kalinin, S., Sanabria, H., Valeri, A. & Seidel, C. A. M. Filtered FCS: species auto- and cross-correlation functions highlight binding and dynamics in biomolecules. *ChemPhysChem* **13**, 1036–1053 (2012).
 53. Aggarwal, T., Materassi, D., Davison, R., Hays, T. & Salapaka, M. Detection of steps in single molecule data. *Cell. Mol. Bioeng.* **5**, 14–31 (2012).
 54. Hadzic, M. C. A. S., Börner, R., König, S. L. B., Kowanko, D. & Sigel, R. K. O. Reliable state identification and state transition detection in fluorescence intensity-based single-molecule Förster resonance energy-transfer data. *J. Phys. Chem. B* **122**, 6134–6147 (2018).
 55. Börner, R. et al. Simulations of camera-based single-molecule fluorescence experiments. *PLOS ONE* **13**, e0195277 (2018).
 56. Hadzic, M. C. A. S., Sigel, R. K. O. & Börner, R. Single-molecule kinetic studies of nucleic acids by Förster resonance energy transfer. *Methods Mol. Biol.* **2439**, 173–190 (2022).

Acknowledgements

We thank Benjamin Schuler and Michael Schlierf for providing experimental smFRET data. We thank the 2019 chair of the Biological fluorescence subgroup of the Biophysical Society (BPS), Paul Wiseman, and the co-chairs of the FRET in Biophysics Discussion Forum, Claus Seidel and

Hugo Sanabria, for providing a platform to initiate the kinSoftChallenge. M.G. was funded by the Deutsche Forschungsgemeinschaft (German Research Foundation) – Project no. 431471305. R.K.O.S. thanks the SNF (200020_165868 and 200020_192153) and UZH for financial support. R.B. thanks the University of Applied Sciences Mittweida for financial support. L.V., J.S., T.H. were supported by the Collaborative Research Centre SFB1381 funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 403222702 – SFB 1381. S.S. was supported by the Postdoc.Mobility fellowship no. P400PB_180889 by the Swiss National Science Foundation. J.C. thanks US National Human Genome Research Institute (NHGRI) Project-ID 1R15HG009972. L.K. and D.D.M. thank the Case Western Reserve University College of Arts and Sciences for support. D.A.E. supported by NIH grant R35 GM127151. K.R.W. supported by NIH grants R01 GM132263 and R01 GM118508. C.D.L. and D.D.R. were supported by grant 16SMPS05 from the Institutes Organization of the Dutch Research Council (NWO-I, formerly FOM). N.S.H. acknowledges support from Carlsberg foundation Distinguished associate professor program (CF16-0797) Vellux foundation center of excellence BIONEC (grant no 18333) and the NovoNordisk foundation (NNF14CC00001 and NNF16OC0021948), G.H. and H.S. acknowledge support by NIH 1P20GM130451 and 2R01MH0 81923-11A1, and NSF 1749778. A.B. and C.A.M.S. acknowledge support by the European Research Council through the Advanced Grant 2014 hybridFRET (number 671208).

Author contributions

All authors analysed data and/or discussed the results and contributed to writing the manuscript. M.G. and S.S. compiled all results, prepared figures, and wrote the article draft together with A.B. M.G. performed and analysed simulations. S.S. designed and initiated the study, and organized the collaboration in consultation with all authors.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-022-33023-3>.

Correspondence and requests for materials should be addressed to Markus Götz or Sonja Schmid.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

Reprints and permission information is available at <http://www.nature.com/reprints>

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022

¹Centre de Biologie Structurale, CNRS UMR 5048, INSERM U1054, Univ Montpellier, 60 rue de Navacelles, 34090 Montpellier, France. ²Institut für Physikalische Chemie, Lehrstuhl für Molekulare Physikalische Chemie, Heinrich-Heine-Universität, Universitätsstr. 1, 40225 Düsseldorf, Germany. ³Department of Chemistry & Nano-science Center, University of Copenhagen, 2100 Copenhagen, Denmark. ⁴Novo Nordisk Foundation Centre for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen, 2100 Copenhagen, Denmark. ⁵Department of Chemistry, University of Zurich, 8057 Zurich, Switzerland. ⁶Department of Chemistry and Biochemistry, Ohio University, Athens, OH, USA. ⁷Physical and Synthetic Biology, Faculty of Biology, Ludwig-Maximilians-Universität München, Großhadernerstr. 2-4, 82152 Planegg-Martinsried, Germany. ⁸Department of Chemistry, University of North Carolina, Chapel Hill, NC 27599, USA. ⁹Lineberger Comprehensive Cancer Center, University of North Carolina, Chapel Hill, NC 27599, USA. ¹⁰Department of Physics and Astronomy, Clemson University, Clemson, SC 29634, USA. ¹¹Institute of Physical Chemistry, University of Freiburg, Freiburg, Germany. ¹²Signalling Research Centers BIOSS and CIBSS, University of Freiburg, Freiburg, Germany. ¹³Department of Physics, Case Western Reserve University, Cleveland, OH, USA. ¹⁴Department of Chemistry, Case Western Reserve University, Cleveland, OH, USA. ¹⁵Department of Chemistry and Center for Nano Science (CeNS), Ludwig Maximilians-Universität München, Butenandtstraße 5-13, 81377 München, Germany. ¹⁶Bioinformatics Group, Wageningen University, Droevendaalsesteeg 1, 6708PB Wageningen, The Netherlands. ¹⁷Department of Physics, North Carolina State University, Raleigh, NC 27695, USA. ¹⁸Spemann Graduate School of Biology and Medicine (SGBM), University of Freiburg, Freiburg, Germany. ¹⁹NanoDynamicsLab, Laboratory of Biophysics, Wageningen University, Stippeneng 4, 6708WE Wageningen, The Netherlands. ²⁰Present address: PicoQuant GmbH, Rudower Chaussee 29, 12489 Berlin, Germany. ²¹Present address: Department of Bionanoscience, Kavli Institute of Nanoscience Delft, Delft University of Technology, Van der Maasweg 9, 2629 HZ Delft, The Netherlands. ²²Present address: Laserinstitut Hochschule Mittweida, University of Applied Sciences Mittweida, 09648 Mittweida, Germany. ²³Present address: Department of Biochemistry and Molecular Pharmacology, New York University School of Medicine, New York, NY 10016, USA.

✉ e-mail: goetz@picoquant.com; schmid@nanodynlab.org

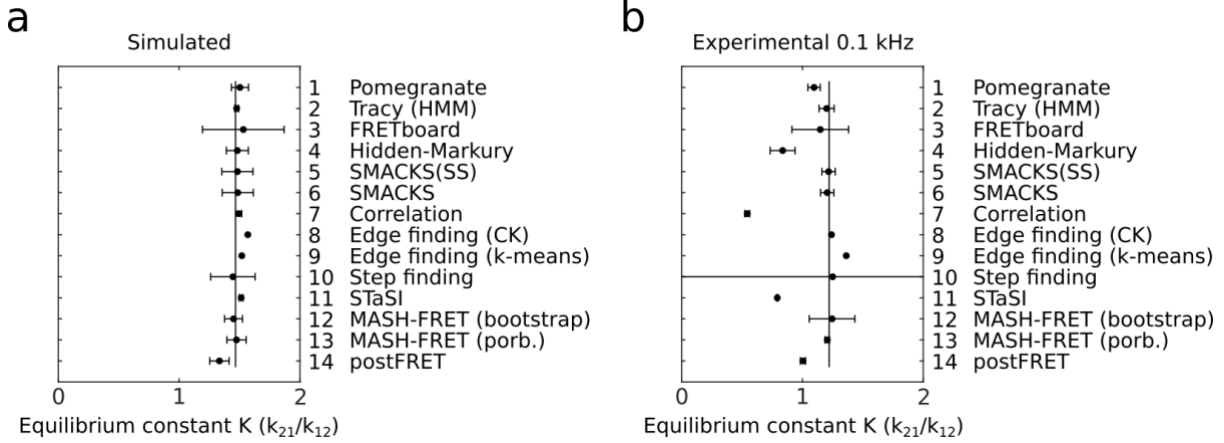
Supplementary Information:

A blind benchmark of analysis tools to infer kinetic rate constants from single-molecule FRET trajectories

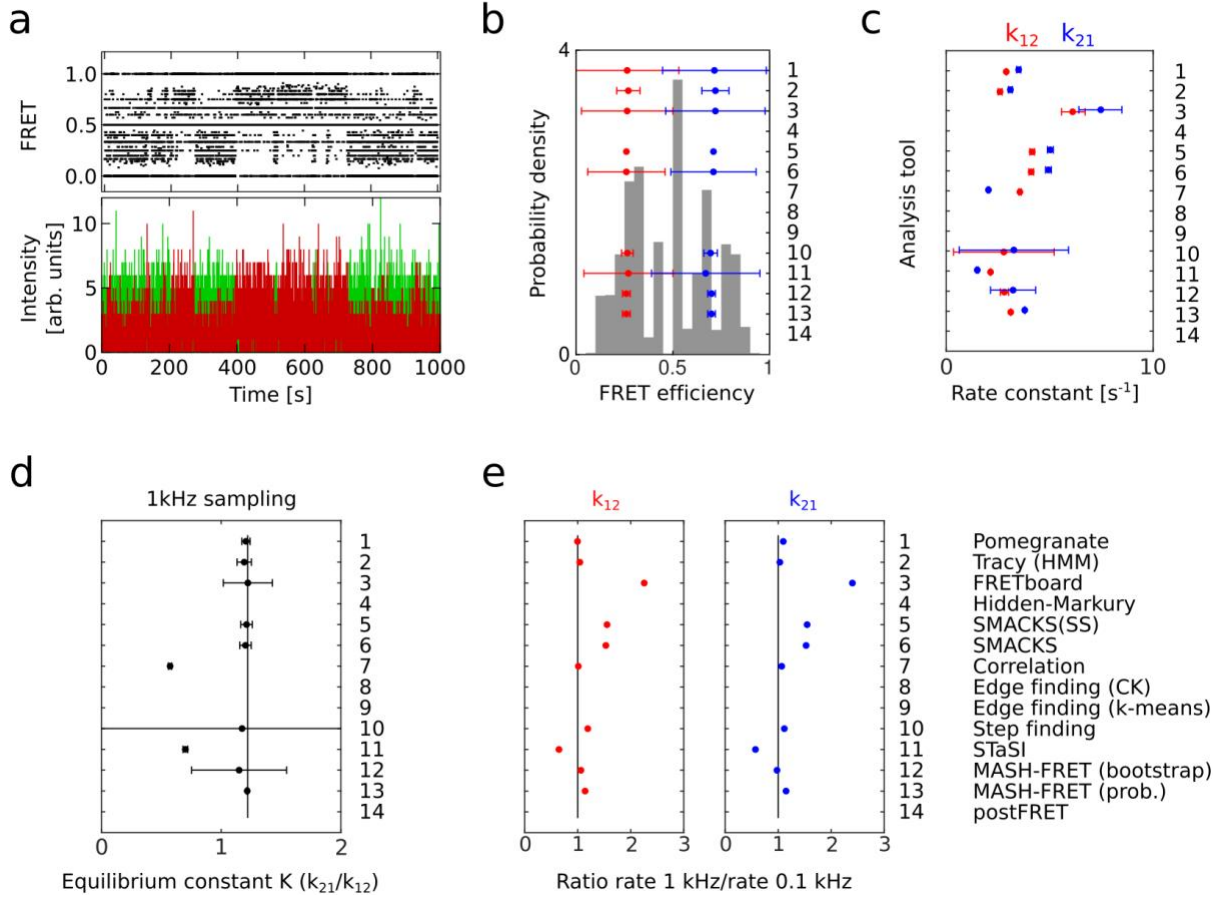
Götz et al.

1 Supplementary Figures	2
Supplementary Figure 1: Equilibrium constants of the kinetics between two states shown in Figure 2.	2
Supplementary Figure 2: Supplementary results from experimental data with high sampling rate and low SNR.	3
Supplementary Figure 3: Validation of the simulated datasets.	4
Supplementary Figure 4: FRET efficiency histograms and all inferred FRET states for the experimental datasets shown in Figure 5.	5
Supplementary Figure 5: Comparison of the kinetic models with three FRET states inferred for the datasets shown in Figure 5.	6
Supplementary Figure 6: Comparison of the kinetic models with four FRET states inferred for the datasets shown in Figure 5	7
2 Supplementary Notes	8
Supplementary Note 1: A simple file format for smFRET trajectories	8
3 Supplementary Methods (numbered as in the main text)	9
Supplementary Method 1: Pomegranate	9
Supplementary Method 2: Tracy	11
Supplementary Method 3: FRETboard	13
Supplementary Method 4: Hidden-Markury	15
Supplementary Method 5 & 6: SMACKS	17
Supplementary Method 7: Correlation	19
Supplementary Method 8: Edge finding (CK)	27
Supplementary Method 9: Edge finding (k-means)	28
Supplementary Method 10: Step finding	30
Supplementary Method 11: STaSI	32
Supplementary Method 12 & 13: MASH-FRET (bootstrap & probabilistic)	34
Supplementary Method 14: postFRET	38
4 Supplementary Tables	43
Supplementary Tables 1: Inferred kinetic models for the data shown in Fig. 4.	43
Supplementary Tables 2: Inferred kinetic models for the data shown in Fig. 5a-c.	46
Supplementary Tables 3: Inferred kinetic models for the data shown in Fig. 5d-f.	49
Supplementary Tables 4: Inferred kinetic models for the data shown in Fig. 5g-i.	52
Supplementary Table 5: Parameters for the simulation of smFRET traces.	55
5 Supplementary References	56

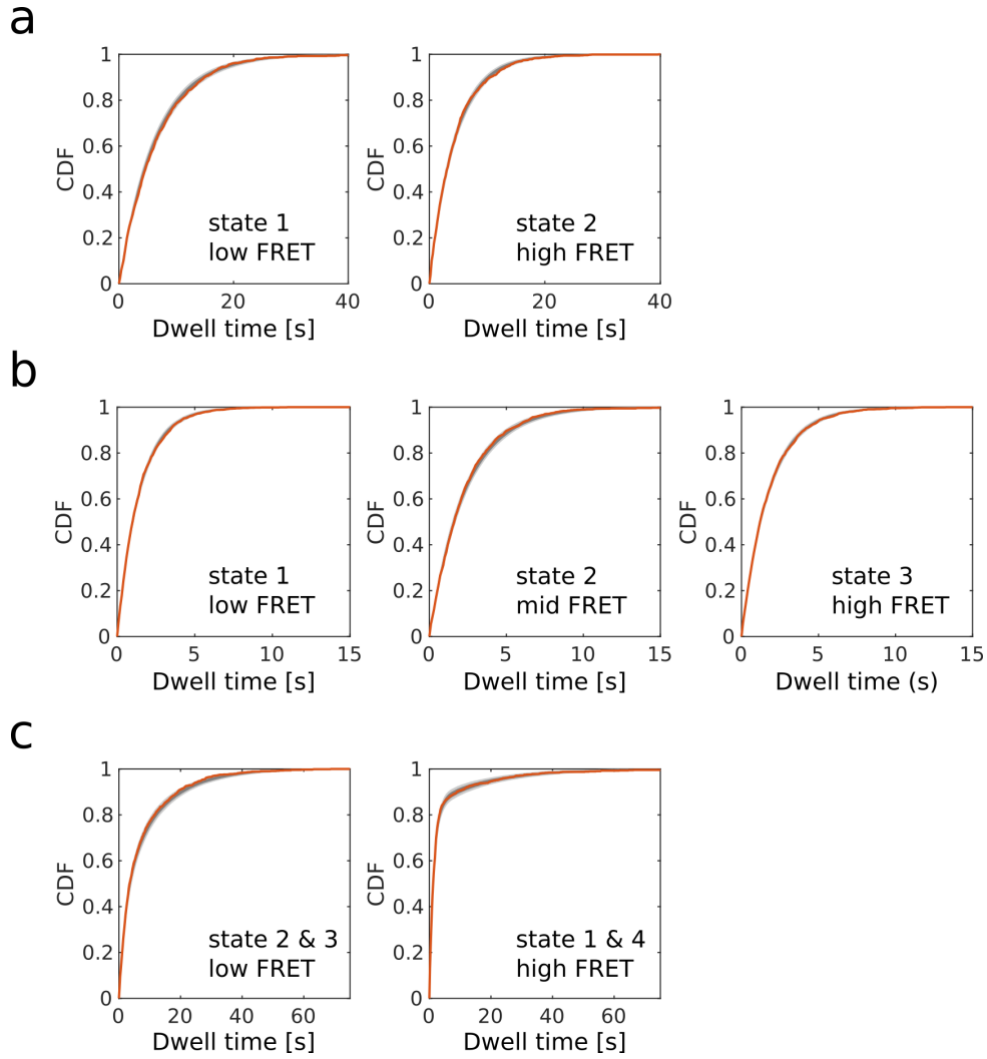
1 Supplementary Figures



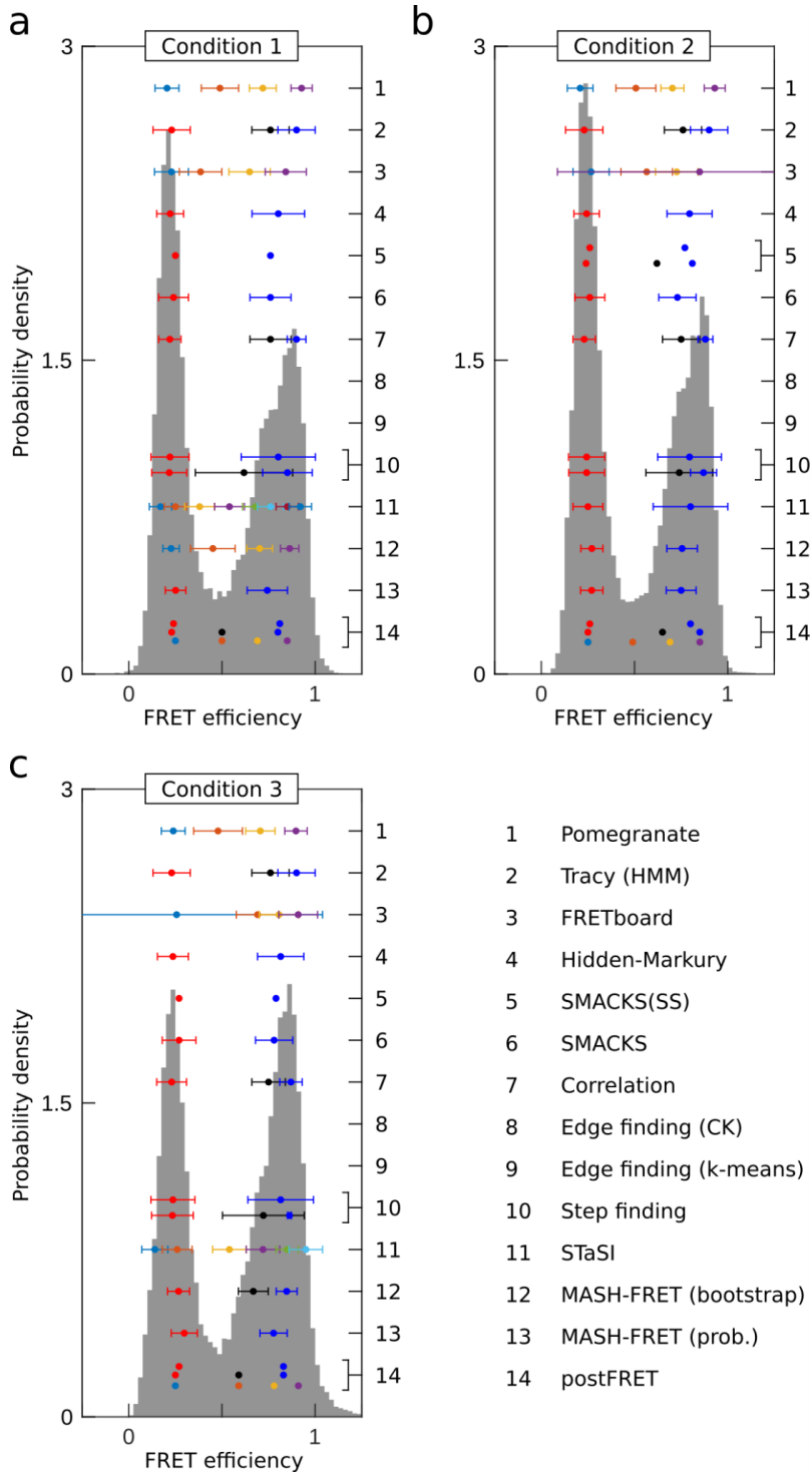
Supplementary Figure 1: Equilibrium constants of the kinetics between two states shown in Figure 2. **a** The inferred equilibrium constant for the simulated dataset. The vertical line indicates the ground truth value. Uncertainty measures (CI, SD) as listed in Table 1. **b** The inferred equilibrium constant for the experimental data with 0.1 kHz sampling rate. The vertical black line indicates the ratio of the two well-separated FRET efficiency populations, as estimated by dividing the number of datapoints with FRET $E < 0.5$ by those with FRET $E > 0.5$. Uncertainty measures (CI, SD) as listed in Table 1.



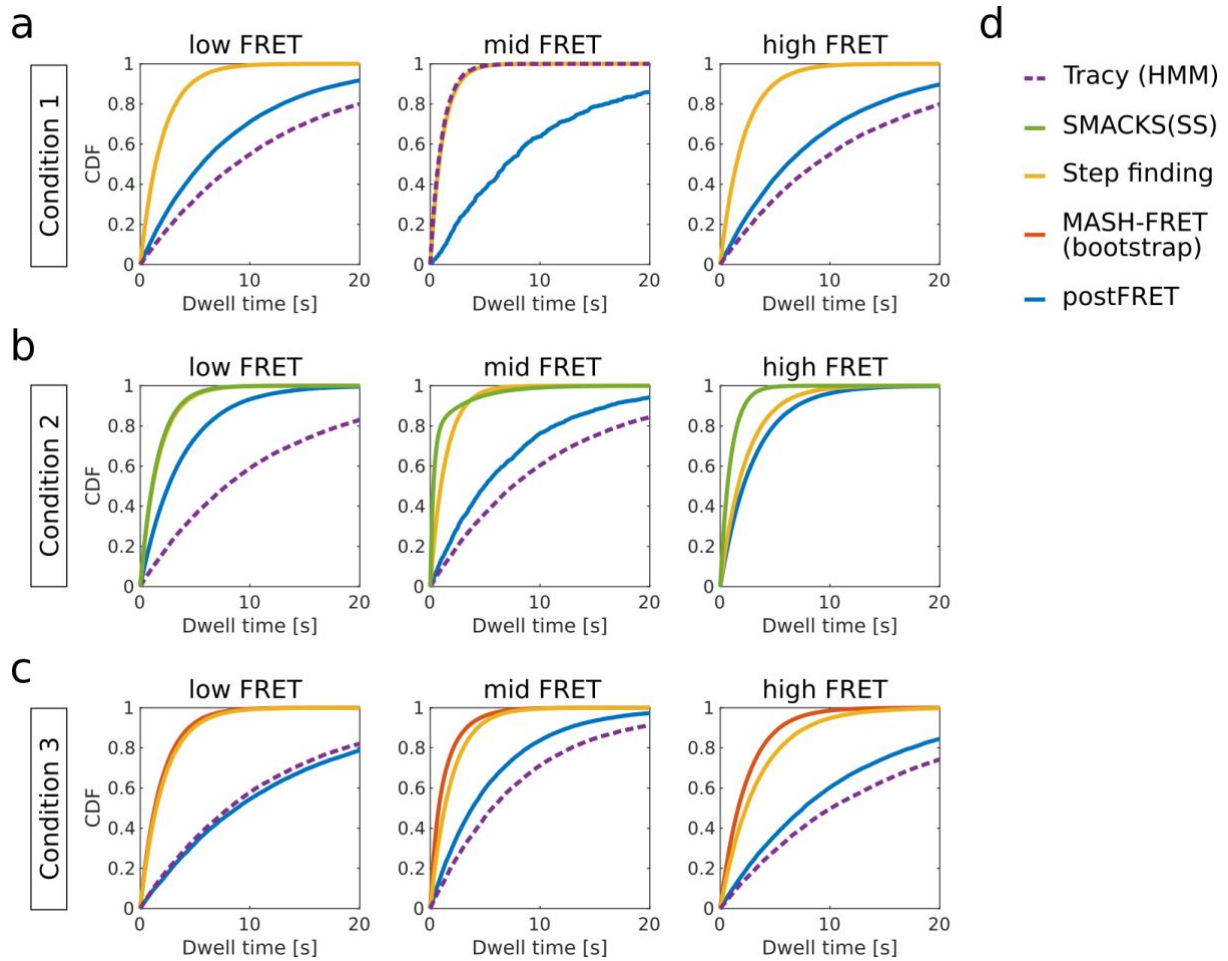
Supplementary Figure 2: Supplementary results from experimental data with high sampling rate and low SNR. **a** Donor (green) and acceptor (red) fluorescence and FRET efficiency (FRET, black) trace for the molecule shown in Fig. 2e, representative for the dataset used in b, c, d, e: $n(\text{traces})=19$, $n(\text{datapoints})=2,261,000$, using 1 ms time bins resulting 1 kHz sampling rate. **b** Corresponding FRET efficiency histogram (gray) and inferred FRET efficiencies in red and blue. The error bars represent standard deviations. **c** Corresponding inferred rate constants from experimental data, and uncertainty measures (CI, SD) as listed in Table 1. **d** Equilibrium constant for the experimental datasets with 1 kHz sampling. The vertical black line indicates the population ratio as estimated from the FRET efficiency histogram at 0.1 kHz sampling (Fig. 2f) by dividing the number of observations with FRET $E < 0.5$ by those with FRET $E > 0.5$. Uncertainty measures (CI, SD) as listed in Table 1. **e** Ratio of the rate constant inferred from data with 1 kHz vs 0.1 kHz sampling for rate k_{12} (red) and k_{21} (blue). The black line indicates a ratio of one, i.e., rate constants inferred for both sampling rates are equal. A ratio above one means that the rate constant inferred from the 1 kHz dataset is larger than the one inferred from the 0.1 kHz dataset.



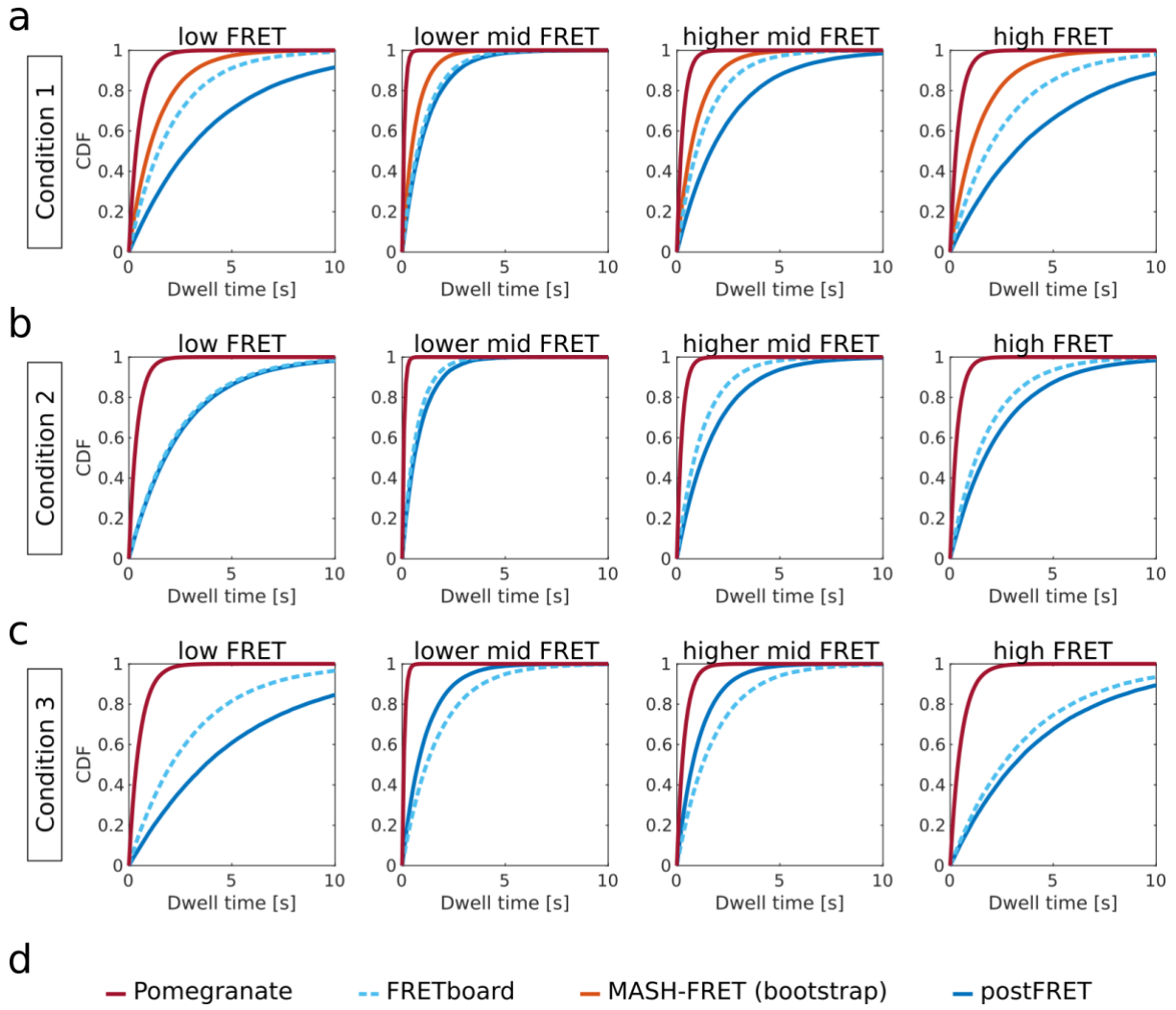
Supplementary Figure 3: Validation of the simulated datasets using cumulative distribution functions (CDF) of the dwell times. The orange line represents the simulated data used in this study. The spread between 500 datasets obtained from simulations with identical parameters is shown in dark and light grey intervals representing one and two standard deviations around the mean, respectively. **a** For the simulated data shown in Fig. 2. **b** For the simulated data shown in Fig. 3. **c** For the simulated data shown in Fig. 4.



Supplementary Figure 4: FRET efficiency histograms and all inferred FRET states for the experimental datasets shown in Figure 5. a-c The number of states and corresponding FRET efficiencies returned by the different analysis tools are shown for the experimental dataset shown in Fig. 5b,e,h, respectively. The error bars represent standard deviations. Brackets next to the right-hand tick markers indicate where multiple models with a different number of FRET states were submitted for a particular tool. The legend in c is valid for the entire figure (and throughout the paper).



Supplementary Figure 5: Comparison of the kinetic models with three FRET states inferred for the dataset shown in Figure 5. a-c Cumulative distribution function (CDF) of the dwell times simulated using the inferred kinetic models (described in Methods) are shown for conditions 1, 2, and 3, respectively. Please note the variation among the inferred FRET efficiencies of the low, mid and high FRET states (Supplementary Figure 4), hindering a direct kinetic comparison in this case. The model inferred by *Tracy* does not populate the high FRET state under condition 2, hence *Tracy* data is absent in the right panel in **b**.



Supplementary Figure 6: Comparison of the kinetic models with four FRET states inferred for the dataset shown in Figure 5. a-c Cumulative density function (CDF) of the dwell times simulated using the inferred kinetic models (described in Methods) are shown for conditions 1, 2, and 3, respectively. Please note the variation among the inferred FRET efficiencies of the four FRET states (Supplementary Figure 4), hindering a direct kinetic comparison in this case.

2 Supplementary Notes

Supplementary Note 1: A simple file format for smFRET trajectories

So far, no standardized and widely accepted file format for storing and exchanging smFRET trajectories is in use. A “single-molecule dataset” (SMD) file format has been proposed², based on JavaScript Object Notation (JSON), but has not been broadly adopted in the community. In this study, we opted for a simple tab-delimited text file format that is sufficient for the encountered time trace datafile sizes, circumvents intricate parsing, and was readily utilized by all participants of this study.

Each molecule is represented by a separate, tab-delimited text file. Each file contains a column with the time information and columns with the fluorescence intensities of the donor and acceptor after donor excitation ($I_{Dem|Dex}$, $I_{Aem|Dex}$). Additionally, columns with the acceptor intensity after acceptor excitation ($I_{Aem|Aex}$) and the apparent FRET efficiency $E_{app} = I_{Aem|Dex} / (I_{Aem|Dex} + I_{Dem|Dex})$ are present for the simulated dataset. The file format can be easily extended with additional columns for additional detection channels, e.g., more spectral and/or polarization channels. In addition to the time series data itself, further metadata, describing experimental conditions, acquisition parameters, and settings used to extract the intensities from the recorded raw data, can be included, either in the header for each tab-delimited file or in a separate file, as outlined in a recent position paper of the FRET community³. The broad acceptance attained in this study forms a promising starting point for the urgently needed dissemination of a common shareable file format for smFRET trajectories.

3 Supplementary Methods

All analysis tools are detailed here in the order of the numbering in the main text.

Supplementary Method 1: Pomegranate

A) OVERVIEW

The workstream in pomegranate utilizes the fast and flexible probabilistic models built into the python package Pomegranate for efficient and iterative model formulation, fitting and evaluation using the Bayesian Information Criterion (BIC). The presented version of the workstream requires data preprocessing, where smFRET trajectories are sorted and only valid trajectories (based on expert valuation) are passed on through the analysis. Dwell time analysis is subsequently performed after defining all transitions using a Multivariate Gaussian fitting scheme and unbinned maximum likelihood fitting. All parts of the process can be evaluated and customized using user inputs based on expert evaluation and iteratively improved.

A solution to tedious manual data sorting has since this analysis been implemented in an “end-to-end” GUI software⁴, that allows both sorting of smFRET trajectories based on deep learning and simultaneously applies the full analysis workstream presented and used here. The code used here as well as the compiled DeepFRET programme can be freely downloaded at www.hatzakislabs.com/#software. The Deep Neural network sorting step is not necessary here as the data are presorted.

B) WORKFLOW

Step 1: Model formulation

Accepted smFRET data is loaded into the Python workstream, where the FRET efficiency is calculated from donor and acceptor intensities and subsequently analyzed using Hidden Markov modelling. Initially the number of underlying FRET distributions should be determined to optimize the model fit. To extract this information, the software allows the user to fit all data to a Hidden Markov model containing between 1-n gaussian distributed state populations (shown to be a robust approximation of FRET distributions⁵) and corresponding transitions between them using and the Baum-Welch forward-backward algorithm. The models can then be compared and the best selected for further analysis using Bayesian Information Criterion (BIC) to penalize overfitting as shown earlier^{6,7}. Once a model is accepted and finalized, it can be saved for further use and evaluation.

Step 2: Trace by trace prediction

A finalized model can be readily applied to analyze smFRET trajectories. The software will for each individual trajectory use the Viterbi algorithm to calculate the most likely state for each observation based on the provided model. To further validate model predictions, a subset of trajectories can be visualized with the corresponding idealized state (model prediction) for user evaluation and validation. Based on expert knowledge, users may choose to reiterate Steps 1 and 2 for optimal fitting of smFRET data. For each data point, the most likely state is found and all data is saved to allow any further analysis or visualizations.

Step 3: Parameter extraction and evaluation

Using the model predictions, each transition in each trajectory ($E \rightarrow E+1$) is plotted in a Transition Density Plot (TDP) and separated using a Gaussian Mixture Model (GMM) fitting scheme. The number of clusters was determined using a combination of user inspection of unfitted data and BIC evaluation. For each fitted cluster, only data points within a 99 % confidence interval of the cluster center would be included for kinetic rate extraction. This ensures tighter clusters and fewer single “off-cluster” points. To extract kinetic transition rates, the dwell times of each cluster were fitted using a single exponential decay and maximum likelihood fitting, subsequent comparison to a two-component exponential decay using BIC was used to check for degenerative states.

C) MISCELLANEOUS

The software is based on publicly available packages for python 3.8.x and is thus cross-platform accessible – a compiled version can be found in DeepFRET as mentioned above. All analysis presented within this survey were performed on a desktop computer (MacBook Pro) with 3,1 GHz Dual-Core Intel Core i5 and 8 GB 2133 MHz LPDDR3. Total computation time varied between ~15s to a few minutes.

Supplementary Method 2: Tracy

A) OVERVIEW

The *Trace Intensity Analysis* toolbox, referred to as TRACY, was programmed in MATLAB and has been updated to run on MATLAB R2020a. The program is used for analyzing two-color single molecule FRET Traces. The program, written by Gregor Heiss⁸, performs the following tasks:

1. Extraction of single-molecule intensity trajectories for one or two channels with only donor excitation or with Alternating Laser Excitation from image stacks.
2. Allows framewise manual selection via mouse cursor. Frames providing useful fluorescence intensity data and bleaching steps for each dye can be selected.
3. Allows correction of the fluorescence intensity for determination of FRET values.
4. Allows user-defined categorization of smFRET traces and analysis of molecular subpopulations.
5. Analysis features includes a Hidden Markov Model analysis using the HMM toolbox in MATLAB written by Kevin Murphy (<https://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>). The HMM analysis can be run individually on each trace or globally on an entire dataset. The number of states can be estimated using a BIC analysis.
6. From the HMM analysis, a Viterbi path can be calculated for all manually selected traces and the kinetic rates can be obtained by fitting dwell time distributions using the MATLAB Curve Fitting Toolbox™.

TRACY is available upon request. As TRACY is being replaced by a new program for multiple color analysis, it will be provided "as is" with only a limited amount of support possible.

B) WORKFLOW

For the analysis performed in this study, the following workflow was followed.

Step 1: Data import and categorization

Data were loaded into TRACY and categorized manually by framewise selection of valid smFRET regions. Unselected regions in each trace were treated as bleached frames and were not used for further analysis.

Step 2: HMM analysis

A global HMM analysis was performed on each data set. For the HMM analyses, only the FRET efficiency data were used and not the donor and acceptor intensities. The mean FRET efficiency and sigma were set as learning parameters. A Viterbi path was then calculated from the given HMM parameters for the individual traces.

Step 3: Transition Density Plot (TDP)

Using the determined transitions, a transition density plot (TDP) between the learned states was calculated. The number of states were estimated visually by summation of all transitions as two-dimensional Gaussians. From the TDP, individual states were manually selected and the corresponding dwell-time histograms were fitted using either a single exponential function or a double-exponential function to obtain the transition rates.

In the end, the decision regarding which model to apply and analysis of the returned dwell-time distributions were determined manually based on visual inspection of the data and results from the initial analysis. The time involved varied from 20 to 60 min depending on the size and complexity of the data set.

C) Miscellaneous Parameter Settings

Except for the number of states of the HMM model, all parameters were kept the same for all data sets. A global HMM analysis was performed using an initial sigma of 0.1, 10,000 maximum iteration steps and a convergence threshold of 1E-06.

Technical Specifications

All datasets were analyzed using a MacBook Pro 2018 (Intel Core i7-8850H CPU 6 x 2.6 – 4.3 GHz, AMD Radeon Pro 560x, 16 GB DDR4-2400, macOS 10.14).

Supplementary Method 3: FRETboard

A) OVERVIEW

FRETboard is a semi-supervised FRET trace classification tool that is served remotely through a web browser. Using a simple click-and-point interface, the user may ‘teach’ a classification model to recognize certain patterns in the traces, by iteratively performing manual curation on an automatically classified example trace and then retraining the model using the corrected traces. This lends FRETboard the flexibility to easily adapt to different labeling schemes. To further expand this flexibility, the user may also experiment with different combinations of nine features derived from the original channels, and the application of different model structures. These properties and further details on FRETboard usage as applied in this challenge are described below.

FRETboard is available for usage on the Wageningen bioinformatics servers at www.bioinformatics.nl/FRETboard or can be installed and run locally from <https://github.com/cvdelannoy/FRETboard> .

B) WORKFLOW

The procedure is initialized by fitting an unsupervised model on all loaded traces in a traditional manner, using randomly generated initial parameters that are then fitted using an implementation of expectation maximization. The only user-provided guidance at this point is the number of states that should be recognized. Traces are classified, and the trace classified with least certainty, i.e. for which the state path probability normalized over sequence length was lowest, is presented to the user for manual correction of the classification. The probability of assigned states for a given trace may be poor in this trace because of the presence of noise. If this is the case, a user may choose to assign noisy measurements to a state they deem appropriate. However, if it is more appropriate to remove the noise, as is the case in bleaching and blinking events, the user may filter these measurements out by assigning them to a separate state reserved for such events. Such a state will then be discarded before FRET distribution and transition rate analysis. Alternatively, the model fit may suffer if the trace contains more or fewer states than those included in the current model. In that case, the number of states and classification must be adjusted appropriately.

After applying manual corrections, the first semi-supervised training round on all loaded traces is started. State distributions and transition rates can now be deduced from the corrected trace and be used as initial parameters, after which the model is refitted on supervised and unsupervised traces simultaneously using semi-supervised expectation maximization. After refitting, traces are reclassified, and the trace now marked by the lowest state path probability is presented to the user. The procedure is repeated until the user finds that presented traces are correctly classified.

For this challenge, we chose to manually classify five traces and retrain the model after each manual classification. Analysis typically took 15 minutes per data set and was performed using the remote server version running at www.bioinformatics.nl/FRETboard.

C) MISCELLANEOUS

Model structures

While essentially any supervise-able classification model type may be trained through FRETboard, three flavors of hidden Markov models (HMMs) are currently included by default. The “vanilla” structure produces a straightforward fully connected HMM sporting no further modifications. The “boundary-aware” structure adds additional “edge states” between states, which are trained on measurements around a detected transition. Transitions between states may only occur through these edge states. If state transitions are marked by a signature distribution in a certain feature, this distribution is captured by the edge state, which allows for more accurate detection of state transitions. The “GMM-HMM” structure also implements edge states, and in addition models emissions using a Gaussian mixture model (GMM), which adds the flexibility to classify noisier distributions as a single state using multiple Gaussians. The number of Gaussians per GMM is determined per state using a Bayesian information

criterion selection procedure. Users may also write a custom model structure implementation, using the provided template.

In this challenge, the vanilla structure was used for the two-state simulated data, while the GMM-HMM structure was used for the other data sets, to account for added noise and other complications.

Features

As different patterns may be better discernable using different features, users may activate and deactivate each of the nine included features as they see fit. These features include the original acceptor and donor channels, and the acceptor and donor channels during direct acceptor excitation if alternating laser excitation (ALEX) is employed. The proximity ratio E_{PR} is included as an approximation of FRET efficiency and is defined as:

$$E_{PR} = \frac{F_{Aem}^{Dex}}{F_{Aem}^{Dex} + F_{Dem}^{Dex}}$$

Here, F_{Aem}^{Dex} and F_{Dem}^{Dex} are the original donor and acceptor emission. The summed intensity F_{sum} is also included:

$$F_{sum} = F_{Aem}^{Dex} + F_{Dem}^{Dex}$$

Furthermore, we used two time-aggregated features that capture the variability of features over a sliding window of five measurements: the Pearson correlation coefficient between F_{Aem}^{Dex} and F_{Dem}^{Dex} and standard deviation of F_{sum} . These features may aid models in capturing feature distributions characteristic for state transitions. For this challenge, training was started using the default combination of features (E_{PR} , I_{sum} , standard deviation of I_{sum} and correlation coefficient). However, if training accuracy failed to reach levels above 95%, the best functioning combination of features was picked, by gauging the effect of toggling features on training accuracy.

Parameter extraction

FRET distributions are extracted by separating values from classified traces by state and calculating mean and standard deviation for each state. To obtain transition rates (F), a transition matrix (A) is derived from the classified data, which must then be converted from discrete to continuous rates and corrected for framerate (f_s):

$$F = I + f_s \times \log A$$

Here I is the identity matrix and \log denotes the natural matrix logarithm operation. 95% confidence intervals of transition rates are estimated by bootstrapping the classified traces and calculating F 100 times.

Supplementary Method 4: Hidden-Markury

A) OVERVIEW

Hidden-Markury is a trace analysis software based on a global optimization of one global kinetic model. It supports the global analysis of 1D FRET efficiency traces and 2D donor & acceptor photon streams with multiple model optimization options, such as degenerate states, forbidden transitions, and fixed model parameters. The core of the Hidden-Markov model and its optimization is based on the open python library hmmlearn (<https://hmmlearn.readthedocs.io/en/latest/index.html>)

The code of this software tool and its description is provided as interactive Jupyter notebooks and can be found in the GitHub repository (<https://github.com/ChristianGebhardt/Hidden-Markury>). Hereby, the notebook combines the high flexibility of individual code adaption and the user-friendly nature of interactive GUI elements with informative descriptions.

B) WORKFLOW

Step 0: Installation and Getting Started

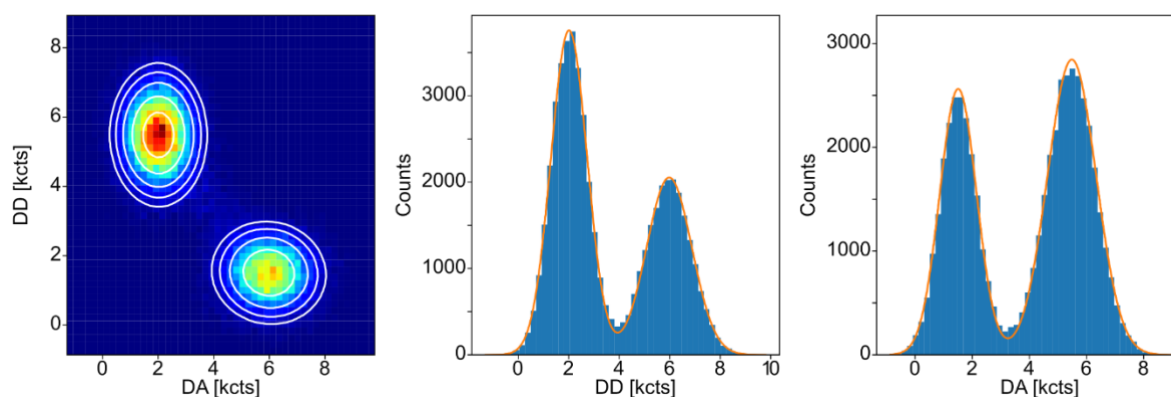
The usage of the Hidden-Markury software requires the cloning of the GitHub repository (<https://github.com/ChristianGebhardt/Hidden-Markury>) and the installation of the python packages (hmmlearn, numpy, pandas, scipy, matplotlib) as described in the README file in more detail. The repository provides exemplary data sets to get started with.

Step 1: Data Import

The data can be imported from various delimited data formats (.csv, .tsv, etc.) where the user only needs to specify the columns/rows of the different data sources from donor (DD) / acceptor channel (DA) or the FRET efficiency (E). The time information is automatically extracted from the first entry. The imported traces are visualized for manual inspection.

Step 2: Model Specification and Initialization

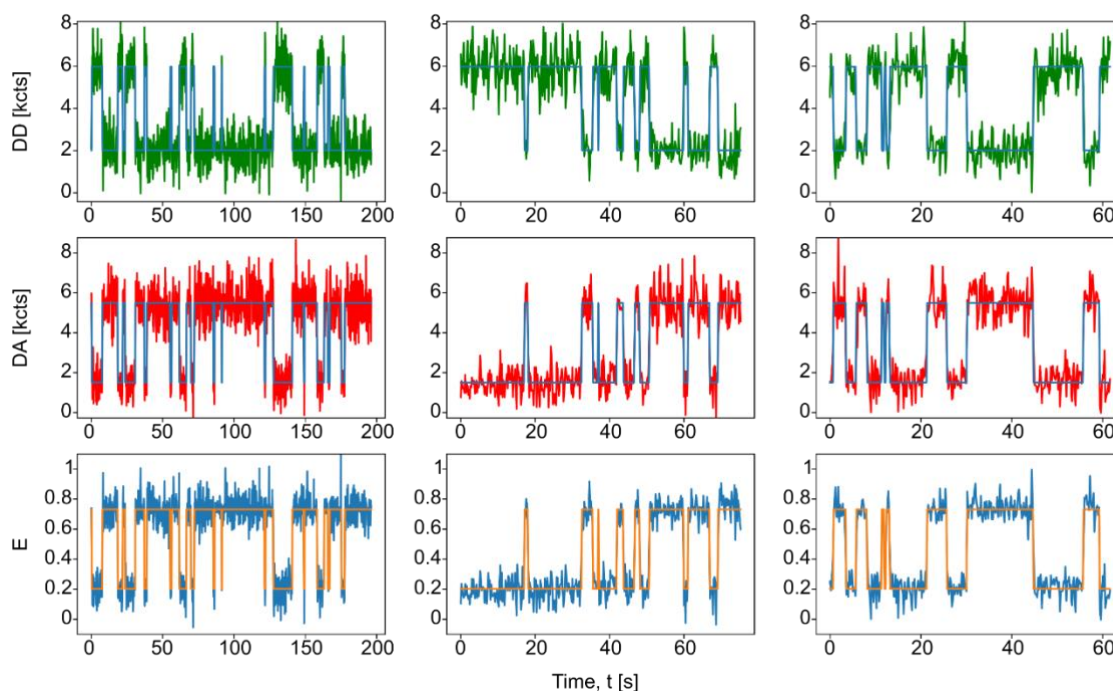
The Hidden-Markury notebook provides the options for 2D-trace analysis of donor (DD) and acceptor (DA) photon streams or 1D-trace analysis of the FRET efficiency traces. In both selected cases the states are initialised by a multi-Gaussian fit in the 2D/1D histogram, where the user needs to select the number of (non-degenerated) states (see Supplementary Method Figure 4.1).



Supplementary Method Figure 4.1: 2D-histogram of donor (DD) and acceptor photon counts (DA) for all time traces fitted with two 2D-Gaussian distributions (left). 1D-projections of the 2D fit results and the 1D-histograms of DD (center) and DA (right).

Step 3: Model Fitting and Prediction

The notebook allows to manually adapt the model fitting procedure by fixing values such as the Gaussian emission functions, the transition matrix or individual forbidden transitions during the model optimization. For degenerated states, the initial fit values from step 2 are required to be fixed in the model fitting step. The actual model optimization uses the expectation-maximization (EM) algorithm⁹ for a global optimization of all traces. In this study all values (Gaussian emission distributions, transition matrix, initial distribution) were optimized during this step. For the degenerated states, only the transition matrix and initial distribution were optimized.



Supplementary Method Figure 4.2: Three exemplary traces with the donor (DD) and acceptor photon counts (DA) in the top and center row, respectively, and the calculated FRET efficiency trace in the bottom row together with the predicted photon counts (blue) and FRET efficiencies (orange).

The Viterbi-algorithm is used for the prediction of the states in all traces based on the optimized model (see Supplementary Method Figure 4.2).

C) MISCELLANEOUS

The python-based software package is cross-platform, as it is built on publicly available python packages.

The analysis was performed on a desktop computer (64-bit operating system, Windows 10) with Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz (4 cores) and 8.00 GB RAM. The computation time varied on the timescale of 30 s to 2 min.

Supplementary Method 5: SMACKS

A) OVERVIEW:

Single molecule time traces hold valuable information about a protein's thermodynamics and kinetics. However, their analysis is far from trivial – especially, when their interpretation goes beyond apparent observations. *Single Molecule Analysis of Complex Kinetic Sequences* (SMACKS) uses mathematical models for pattern recognition to not only resolve statistically relevant rates from such traces but also their uncertainties¹. Thereby, SMACKS does not rely on dwell-times but takes every experimental data point into account to optimise one global kinetic model. Consequently, it allows for experimental variation between individual trajectories. At the core of SMACKS' analysis lie hidden Markov Models that establish a mathematical relation between experimental observations and their subsequent interpretation, which in turn is limited by a predefined number of states. The general procedure of how SMACKS was used for data analysis within this study will be described in the following. A more detailed description on how to use SMACKS can be found in the associated manual (<https://www.singlemolecule.uni-freiburg.de/software/smacks>). The source code can be downloaded from <https://www.singlemolecule.uni-freiburg.de/software/smacks>.

B) WORKFLOW:

Step 1: Software Installation and Data Import

SMACKS is implemented in Igor Pro (Wavemetrics). The program was started using the *startSMACKS* shortcut. To load the kinSoftChallenge data files, the dataID in the startSMACKS.ipf script editor was changed to StrConstant dataID = "time;g_g;r_g;r_r;fret;". They were imported by using the ascii importer in the SMACKS menu (-->Import ascii). Afterwards, as SMACKS only accepts files with up to three tab- or space-separated data columns, the dataID was changed back to StrConstant dataID = "g_g;r_g;r_r" for further analysis.

Step 2: Trace-by-Trace HMM (TbT)

The TbT workflow was started in the SMACKS menu (-->Init TbT). In the settings, the number of apparent states was assigned by eye according to the user's observation. The option FRET restraint was not selected. Adjustments as such were confirmed by clicking **Initialize**. To assign each trace a Viterbi path (meaning certain states according to the Viterbi algorithm), the procedure was applied to the whole dataset by choosing -->TbT Batch Converge in the SMACKS menu. Next, all associated Viterbi paths were individually checked and approved by browsing through the traces (using << and >>) and deleting parameters causing inappropriate Viterbi paths (using **Del**). For traces that were assigned a Viterbi path despite not reaching all states, the parameters were deleted as well. Eventually, by using **TbT Apply Means**, the mean of all saved (i.e. correct) parameters was applied to the remaining traces.

Step 3: Semi-Ensemble HMM (ENS)

Afterwards, the ENS workflow was started in the SMACKS menu (-->Init ENS). The detailed balance condition was not selected. Here, different state models were analysed based on the apparent states and optimised parameters found in the TbT workflow. Depending on the possibility of hidden states (i.e. states displaying the same FRET efficiencies while differing kinetically), different state configurations were tested. Different states were indicated by different numbers (0, 1, 2, etc.), whereas associated hidden states were denoted by doubling them (e.g., 001, 011, 0011, etc). Using the add-on feature -->Compare BICs, the state model that best represented the data (i.e. the one with the lowest BIC) was chosen. Confidence intervals for the chosen state model were calculated using the feature -->Confidence Intervals in the SMACKS menu.

Step 4: Calculating Kinetic Rates and FRET Efficiencies

From the -->TbT Apply Means procedure after the TbT workflow, the mean intensity values over all donor (D) and acceptor (A) traces were obtained. They were used to calculate the FRET efficiencies E according to:

$$\text{FRET } E = \frac{A}{A+D} \quad (5.1)$$

Corresponding uncertainties were calculated according to:

$$\sigma = \sqrt{\left(\frac{D}{(A+D)^2}\right)^2 \cdot \sigma_A^2 + \left(\frac{-A}{(A+D)^2}\right)^2 \cdot \sigma_D^2} \quad (5.2)$$

In the ENS workflow, SMACKS calculated a transition probability matrix and a covariance matrix of the Gaussian probability densities for all states. This calculation is performed in user-supplied time units. Therefore, giving specific time information is not necessary: the transition probabilities a_{ij} from state i to j are specified for this given time interval. Converting those probabilities from the transition probability matrix to rate constants k_{ij} (Hz) was done by multiplying them with with the sampling rate (frames/second) according to:

$$\text{kinetic rate } k_{ij} = \text{transition probability } a_{ij} \cdot \text{sampling rate} \quad (5.3)$$

Confidence Intervals were converted to Hz accordingly.

C) MISCELLANEOUS:

Parameter Settings

In steps 2 and 3, specific parameters can be set and varied. The initial parameters in the TbT workflow, however, were not adjusted. Instead, the given initial values were used. To find out which state model was most likely, the default number of iterations (500) given in the ENS workflow was halved to save computation time. Then, for a more precise calculation of the transition probabilities of the chosen model, the default values for the number of *Max. Iterations* (500) as well as for the *Convergence Threshold* (1E-15) were kept. However, this affected the values in the transition matrices only within the margins of error.

Technical Specifications

The synthetic datasets were analysed on a MiFcom desktop computer (8.00 GB RAM, Intel(R) Core(TM) i5-4460 CPU@3.20 GHz, 64-bit operating system, Windows 10). The experimental sets were analysed on a MiF desktop computer (32 GB RAM, Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30 GHz 2.30 GHz (2 prozessors), 64-bit operating system, Windows 10). In both cases, Igor Pro 6.37 was used.

Computation Time

The computation time varied depending on the complexity and size of the analysed set and the computer used for the analysis. The first synthetic data set containing a two-state system without hidden states was analysed within 30 minutes. The other two sets, in contrast, contained hidden states and – in case of the third set – more than two apparent states. Therefore, the second set required 4 hours, whereas the third took a little less than 13 hours to be analysed completely. For the experimental sets, computation time depended on time resolution. Under 10 kHz conditions, data analysis took 35 hours, whereas the 0.1 kHz data set was analysed within 1.5 hours.

Supplementary Method 7: Correlation

A) OVERVIEW

We recently presented a quantitative model for fluorescence correlation curves of complex multi-state kinetic networks obtained in single-molecule FRET experiments using MFD¹¹. Here, we extended this methodology for use with fluorescence traces of immobilized molecules. In principle, this simplifies the analysis by removing the diffusion term of the correlation function^{12–14}, but modifications are required to obtain accurate correlation functions from variable length traces. In MFD experiments, we are further able to compute filtered correlation functions by utilizing e.g. the lifetime information^{14–16}, but no such information is available in the given case. To this end, we applied a step-finding algorithm to convert the observed FRET efficiency traces into digitized state trajectories, which were further used to compute filtered correlation functions. In the ideal case, this allows us to resolve the kinetics even in multi-state networks of three or more interconverting states. While an accurate estimation of the microscopic rate constants for the more complicated cases involving degenerate FRET states was not possible, correlation analysis still offers a minimally biased approach to estimate the kinetic relaxation times of the corresponding transition rate matrix, which allows to validate kinetic models inferred by other methodologies.

All analysis was done in MATLAB. The code and all analysis files are available at <https://doi.org/10.5281/zenodo.5512005>.

B) WORKFLOW

Computation of correlation functions

The calculation of unbiased correlation functions from the variable-length fluorescence time traces required three problems to be addressed:

1. Correlation functions must be calculated for the total duration of the traces as dynamics and trace length may occur on a similar timescale. Edge-effects to the lower sampling of long lag times thus need to be accounted for.
2. Correlation functions obtained from traces of different lengths must be correctly averaged.
3. Errors arising due to trace-by-trace variability must be estimated.

The correlation function between two signals $S_A(t)$ and $S_B(t)$ is defined by:

$$G_{AB}(\tau) = \frac{\langle S_A(t)S_B(t+\tau) \rangle}{\langle S_A(t) \rangle \langle S_B(t+\tau) \rangle}, \quad (7.1)$$

where τ is the lag time and $\langle \dots \rangle$ denotes the time average for a long measurement. Generally, the correlation function is calculated only up to a time lag that is a fraction of the total measurement time ($\tau \ll T$, typically up to a maximum of 1/10 of the measurement time). To compute the correlation function over the whole length of the trace, it was calculated for every trace k as described in references^{17,18} by:

$$G_{k,AB}(\tau) = \frac{(T_k - \tau) \sum_{t=0}^{T_k - \tau} S_A(t)S_B(t+\tau)}{\sum_{t=0}^{T_k - \tau} S_A(t) \sum_{t=\tau}^{T_k} S_B(t+\tau)}, \quad (7.2)$$

where T_k is the length of the trace and the sums only extend over the valid ranges of the time trace, i.e. $0 \leq t \leq T_k - \tau$ for $S_A(t)$ and $\tau \leq t \leq T_k$ for $S_B(t)$.

Due to the variable length of the traces, the average correlation function over multiple traces is not simply equivalent to the average of all trace-wise correlation functions. To compute the average correlation function over all traces, the terms in the above equation were computed for every trace k and the final correlation function over all traces was computed as:

$$G_{AB}(\tau) = \frac{\sum_k (T_k - \tau) [\sum_k \sum_{t=0}^{T_k - \tau} S_A(t)S_B(t+\tau)]}{[\sum_k \sum_{t=0}^{T_k - \tau} S_A(t)] [\sum_k \sum_{t=\tau}^{T_k} S_B(t+\tau)]}. \quad (7.3)$$

Correlation functions were calculated for lag times on a multiple-tau scheme over stretches of linear time lags with exponentially increasing spacing, i.e. $\tau = 0, 1, 2, \dots, 19, 20, 22, 24, \dots, 38, 40, 44, 48 \dots$ etc (see reference¹⁹ for details).

To estimate the error due to trace-by-trace variability, we performed bootstrapping. For a set of N traces, we randomly selected N traces with replacement (that is, duplicate selections are allowed) 50 times, computed the average correlation function G_{AB} according to eq. 7.3, and determined the standard error of the mean from the set of correlation functions.

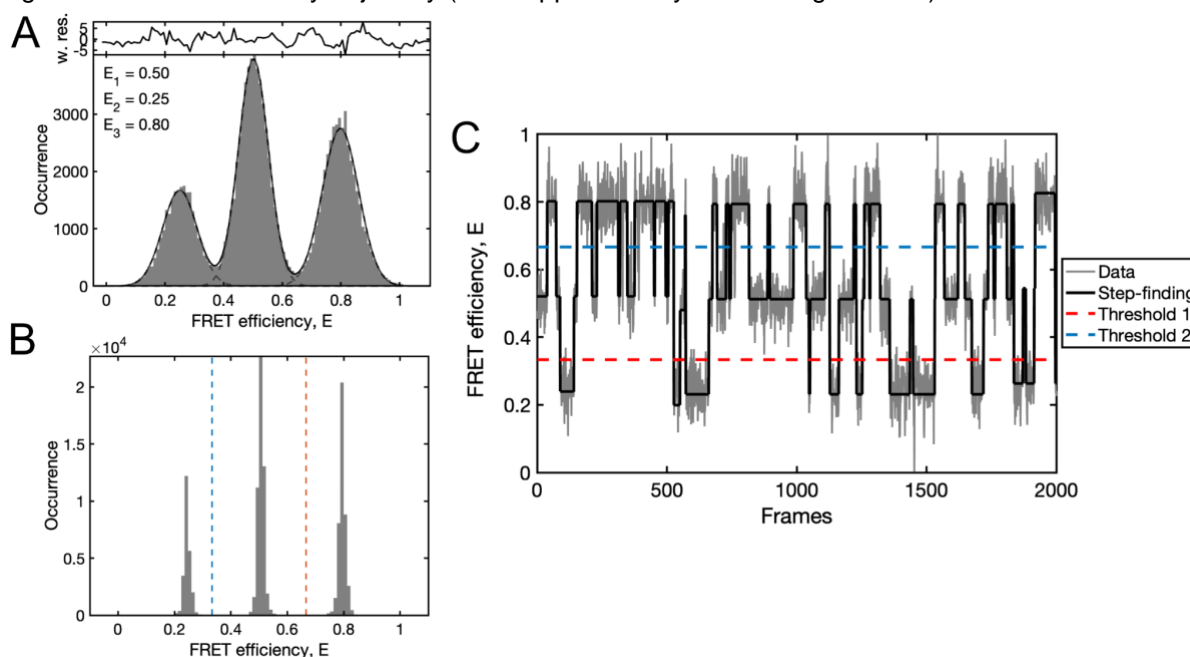
FRET-FCS correlation functions were calculated from the background-corrected intensities in the donor and FRET channels to compute the auto- and cross-correlation functions of the donor (D) and acceptor (A) signals (G_{DD} , G_{AA} , G_{DA} and G_{AD}). Filtered-FCS curves were computed from the digitized state trajectories determined from the step-finding analysis (see below). For the analysis, all possible auto- and cross-correlation functions between the identified states were used (i.e., 4 in the case of two FRET states and 9 in the case of three FRET states).

Estimation of FRET efficiencies of the different states

The minimal number of states and their FRET efficiencies are determined by Gaussian fitting of frame-wise FRET efficiency histograms. We used a Gaussian mixture model as implemented in the MATLAB function *fitgmdist*, based on an iterative Expectation-Maximization algorithm of the likelihood function. An exemplary fit is shown in Supplementary Method Figure 7.1A.

Step-finding algorithm

We apply the algorithm developed by Aggarwal et al. to identify steps in the FRET efficiency traces²⁰. The algorithm does not assume any particular kinetic scheme but estimates the optimal number of steps based on the noise of the signal. Overfitting is avoided by introducing a penalty for each transition. For the analysis, we set an estimated noise of based on the distribution width obtained from the Gaussian fitting analysis ($\sigma_E = 0.05$ - 0.1). No restraints are imposed on the FRET efficiencies of the steps. An exemplary result of the step-finding is shown in Supplementary Method Figure 7.1C. After the step finding, the stepwise FRET efficiency histograms were manually examined to identify thresholds to digitize the FRET efficiency trajectory (see Supplementary Method Figure 7.1B).



Supplementary Method Figure 7.1: Exemplary workflow for the correlation analysis of single-molecule time traces. A-B) FRET efficiency histograms of test dataset 2 before (A) and after (B) applying the step finding. Dashed lines in B indicate the used thresholds to define the state trajectory. C) Example of the step-finding algorithm. The idealized signal trajectory (black) is estimated from the noisy data (gray). Thresholds for the digitization of the FRET efficiency trajectory into states are shown as dashed lines.

FCS model functions

Depending on the complexity of the datasets, three different analyses approaches were used throughout this study, depending on whether the FCS curves were calculated from the fluorescence intensities (FRET-FCS) or the digitized state trajectories (filtered-FCS). Further, for cases where insufficient information could be inferred to decide on the number of kinetic states *a priori*, an empirical model was applied to estimate the kinetic relaxation times without consideration of the correlation amplitudes.

FRET-FCS

For two-state dynamics, analytical functions are known for the color-FCS curves:⁴

$$G_{DD}(\tau) = \frac{k_{12}k_{21}\Delta E^2}{(k_{12}(1-E_1) + k_{21}(1-E_2))^2} e^{-(k_{12}+k_{21})\tau} \quad (7.4)$$

$$G_{AA}(\tau) = \frac{k_{12}k_{21}\Delta E^2}{(k_{12}E_1 + k_{21}E_2)^2} e^{-(k_{12}+k_{21})\tau} \quad (7.5)$$

$$G_{DA}(\tau) = G_{AD}(\tau) = -\frac{k_{12}k_{21}\Delta E^2}{(k_{12}(1-E_1) + k_{21}(1-E_2))(k_{12}E_1 + k_{21}E_2)} e^{-(k_{12}+k_{21})\tau} \quad (7.6)$$

Here, E_1 and E_2 are the FRET efficiencies of the two states and k_{12} and k_{21} are the rates of going from state 1 to 2 and backwards, respectively. For the FRET-FCS analysis, the FRET efficiencies of the states were fixed to the values determined by the analysis of the FRET efficiency histograms.

Filtered-FCS

In filtered-FCS, the correlation functions represent the pure correlation functions between the different states. For the two-state case, the analytical correlation functions between states 1 and 2 are then given by:

$$G_{11}(\tau) = \frac{k_{12}}{k_{21}} e^{-(k_{12}+k_{21})\tau} + c \quad (7.7)$$

$$G_{22}(\tau) = \frac{k_{21}}{k_{12}} e^{-(k_{12}+k_{21})\tau} + c \quad (7.8)$$

$$G_{12}(\tau) = G_{21}(\tau) = -e^{-(k_{12}+k_{21})\tau} + c, \quad (7.9)$$

where c is a constant offset. The correlation functions are calculated from the matrix exponential of the transition rate matrix, $e^{K\tau}$, which can be obtained from the eigen-value decomposition of K :

$$K = \sum_{i=0}^{n-1} \Gamma_i \lambda_i \Rightarrow e^{K\tau} = \Gamma_0 + \sum_{i=1}^{n-1} \Gamma_i e^{\lambda_i \tau} \quad (7.10)$$

where λ_i are the eigen-values and Γ_i the eigen-matrices of K . A transition rate matrix of dimension n has $n - 1$ non-zero eigen-values, and the zero-th eigenvalue, $\lambda_0 = 0$, can be neglected in this case. The full correlation function is then obtained as:

$$G_{ab}(\tau) = \frac{\sum_{i=1}^{n-1} S_a^T \Gamma_i x_d S_b e^{\lambda_i \tau}}{\langle S_a \rangle \langle S_b \rangle} \quad (7.11)$$

where S_a and S_b are the state vectors, i.e. $\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$, x_d is the vector of equilibrium fractions of

the states and $\langle S_a \rangle$ and $\langle S_b \rangle$ are the average occupancies of the states, $\langle S_a \rangle = x_a$. Using this formalism, we can directly fit all rate constants of the transition rate matrix to the obtained correlation functions. A detailed derivation of the correlation functions is given in reference¹¹. The filtered-FCS curves were analyzed globally with respect to the transition rate matrix.

Empirical model functions for degenerate cases

For the advanced cases where degeneracy of FRET states is involved (i.e., states with identical FRET efficiencies but different kinetic properties), our method cannot resolve the kinetics accurately. However, it is still possible to determine the relaxation times of the kinetic process from the correlation functions using simplified multi-exponential model functions with n components.

$$G_{11}(\tau) = \sum_{i=1}^n A_{11,i} e^{-\frac{\tau}{\tau_i}} + c \quad (7.12)$$

$$G_{22}(\tau) = \sum_{i=1}^n A_{22,i} e^{-\frac{\tau}{\tau_i}} + c \quad (7.13)$$

$$G_{12}(\tau) = G_{21}(\tau) = - \sum_{i=1}^n A_{12,i} e^{-\frac{\tau}{\tau_i}} + c, \quad \text{where } \sum_{i=1}^n A_{12,i} = 1. \quad (7.14)$$

Here, c is a constant offset. Here, to reduce the number of free fit parameters, we took advantage of the fact that the cross-correlation functions contain the same information and that the amplitudes should sum to 1 for the filtered-FCS curves.

The obtained relaxation times τ_i correspond to the inverse of the negated eigenvalues of the transition rate matrix, $\lambda = \text{eig}(K)$, and can thus be compared to the input values:

$$\tau_i = -\lambda_i^{-1}. \quad (7.15)$$

No further interpretation of the obtained amplitudes is performed in this case. All relaxation times were globally optimized over all correlation functions.

Curve fitting

Curve fitting was performed based on the reduced chi-square χ_r^2 defined as:

$$\chi_r^2 = \frac{1}{N - k} \sum_i \frac{(G_{data,i} - G_{model,i})^2}{\sigma_i^2}, \quad (7.16)$$

where N is the number of data points, k is the number of parameters of the model and σ_i is the estimated error. Optimization was performed using the *fminsearch* function of MATLAB using the Nelder-Mead method²¹. For model selection, we applied the Bayesian information criterion²² given by:

$$\text{BIC} = k \ln(N) + \chi^2, \quad (7.17)$$

whereby the model with the lowest value for the BIC was chosen. Errors of the fitted parameters were estimated based on Metropolis-Hastings sampling of the posterior probability distribution of the model parameters using a flat prior^{23,24}. The Markov chain Monte Carlo sampler as implemented in the *mhsample* function of MATLAB was run for 10.000 iterations using a symmetric proposal distribution. The width of the proposal distribution was estimated based on the errors of the fit parameters determined from the non-linear least squares fitting using the Hessian matrix at the solution, and set to one-tenth of this value. Every hundredth sample of the Markov chain was kept and used to calculate the confidence intervals. All given errors are specified as 95% confidence intervals. An example of the posterior distribution obtained for the elements of the transition rate matrix of the synthetic three-state system is shown in Supplementary Method Figure 7.2B.

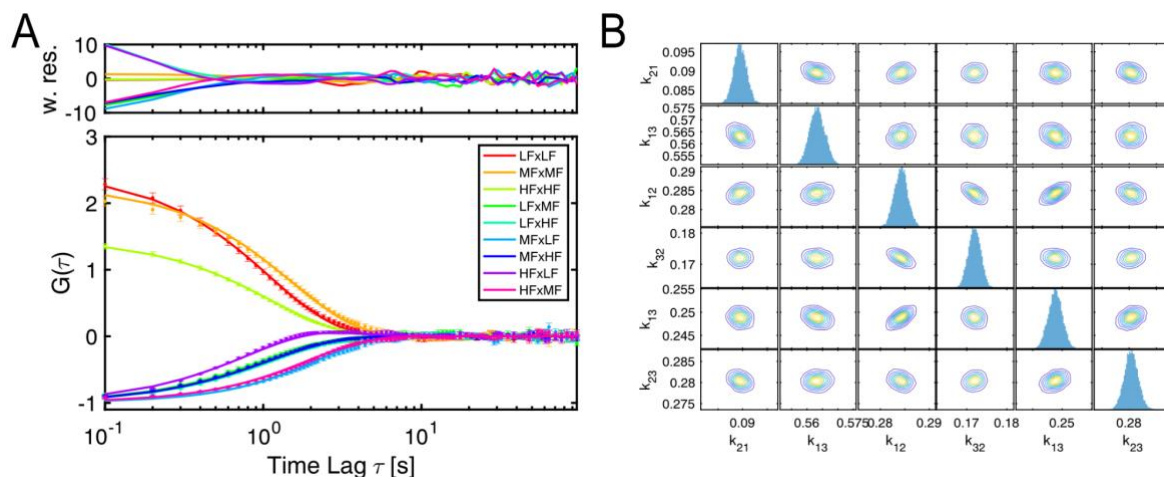
Analysis of the different datasets

Correlation functions for all synthetic datasets were obtained by the filtered-FCS workflow. The correlation functions for the synthetic datasets of level 1 and 2 were analyzed using the analytical model functions for a two- and three-state system, respectively. Due to the degeneracy of the FRET states for level 3, the empirical analysis was performed with a three-component model. The number of components was hereby estimated by comparing the BIC of the models with two to four components. Due to the high number of FRET states in the first experimental dataset, the empirical analysis with a two-component model was performed. For the second experimental dataset, the filtered-FCS workflow was applied with a two-state kinetic model for the 10-ms dataset. In addition, the correlation functions revealed a slower process on the minute timescale. To estimate the timescales of these dynamics, the empirical analysis with a two-component model was additionally applied to the filtered-FCS curves. For the 1-ms dataset, only FRET-FCS analysis could be performed as the step-finding algorithm could not be applied to the noisy traces.

Deviations of the correlation approach for the three-state system (Synthetic Data, Level 2)

The large deviations of the correlation approach for the synthetic three-state system were a partly surprising result as the method had worked better in previous benchmarks. Based on the ground-truth transition rate matrix of the three-state system (see Supplementary Datafiles), the two relaxation times of the simulated system are $\tau_1 = 1.07$ s and $\tau_2 = 1.41$. The accurate estimation of the microscopic rate constants (i.e., the elements of the transition rate matrix) crucially depend on an accurate estimation of

the amplitudes of the different exponential terms in the correlation curves. Given the low contrast between the two relaxation times (~30% difference), it is thus likely that the deviations of the inferred rate constants originate from inaccuracies of the estimated amplitudes due to the large overlap of the two exponential components. The fit of the correlation curves and the error estimation by the Markov chain Monte Carlo method are shown in Supplementary Method Figure 7.2.



Supplementary Method Figure 7.2: Correlation analysis of the synthetic dataset of level 2 (three-state system). Same data and sample size as in Figure 3. **A)** Filtered-FCS correlation functions (scatter plots) and fits (solid lines). Error bars represent the standard error of the mean of the trace-by-trace correlation functions obtained by bootstrapping. **B)** Posterior distributions of the rate constants of the transition rate matrix. All rates are given in s^{-1} .

Correlation analysis of the degenerate systems (Synthetic Data, Level 3)

While the FRET efficiency histogram analysis suggested a two-state system for this dataset, we clearly detected multiple relaxation times in the correlation curves that indicated a more complex kinetic network of three or more states. To infer the number of states, we applied the empiric model function with two, three or four relaxation times and compared the BIC values (Supplementary Method Table 7.1 and Supplementary Method Figure 7.3). We found a clear minimum of the BIC for three relaxation times, indicating that the kinetic system involved four states. Based on the existence of two distinguishable FRET states, we speculated that each of these would show a two-fold degeneracy. While it was not possible to extract the microscopic rate constants reliably, we could compare the relaxation times extracted by the empirical model to the ground truth values. From the filtered-FCS analysis, we estimated relaxation times of $\tau_1 = 1.3$ s (1.1-1.5), $\tau_2 = 10.8$ s (6.6-15.2) and $\tau_3 = 41$ s (30-62) (95% confidence intervals are given in brackets, see Supplementary Method Table 7.1 and Supplementary Method Figure 7.3), which correspond well to the relaxation times determined for the ground truth transition rate matrix of $\tau_1 = 1.05$ s, $\tau_2 = 8.23$ s and $\tau_3 = 27.00$ s. This indicates that correlation analysis can still be used to quantitatively assess the relaxation times corresponding to the transition rate matrix even for complex cases.

C) MISCELLANEOUS

Performance considerations

On a standard desktop computer, the calculation of FRET-FCS correlation curves for the datasets used in this study took less than a minute. For the filtered-FCS analysis, the limiting step is the step-finding algorithm, resulting in typical computation times of 10-15 minutes. Curve fitting generally took less than one minute. However, depending on how many models were tested, the total analysis procedure including human intervention could take up to 1 h.

Artifacts in correlation analysis of experimental data

For experimental data, several artifacts can potentially affect the correlation analysis. Here, we briefly review the most common problems, their effect on the correlation function and how they can be avoided.

1. Model selection and user bias

A major advantage of the correlation analysis is that the computation of the correlation function from the signal intensities is free of user bias. However, the quantitative analysis of the obtained correlation functions crucially requires estimating the number of kinetic states (and their FRET efficiencies) to select the correct model function. If the FRET efficiencies of the different kinetic states are sufficiently different, the number of states can be inferred from the framewise FRET efficiency histogram. Importantly, also the number of relaxation times found in the correlation curves (e.g., by use of the empirical model function in conjunction with the BIC) informs on the number of states, where N states will result in $N-1$ relaxation times. This information was used for the analysis of the synthetic dataset of level 3 with degenerate FRET states. While here the FRET efficiency histogram suggested a two-state system, the correlation curves indicated three relaxation times, in agreement with the four kinetic states of the simulated system. For the filtered-FCS analysis described here, the choice of the number of FRET states is also relevant for the computation of the filtered correlation curves, specifically for the thresholding step for the digitization of the FRET efficiency trajectory. This approach is also inapplicable to degenerate systems containing different kinetic states with identical FRET efficiencies.

2. Signal spikes due to impurities

Some of the experimental datasets used in this study showed clear signal spikes both in the donor and acceptor channels that exceed the variation expected for Poissonian noise and do not show the characteristic anti-correlated behavior expected for FRET dynamics (see Fig. 5a,d,g of the main text). Such signal spikes could be of photophysical nature or originate from diffusing impurities (such as unreacted fluorophores) that transiently enter the observation volume. The expected effect is the appearance of an additional fast component in the autocorrelation functions of the donor and acceptor channels for the FRET-FCS analysis. Notably, such signal fluctuations should not affect the cross-correlation function as they should be uncorrelated between the donor and acceptor channels. The effect on the filtered-FCS analysis is more difficult to assess as it depends on whether the step-finding algorithm will erroneously detect the fast changes of the FRET efficiency due to the signal spikes as a transition.

3. Trace-by-trace variability

It is generally assumed that all molecules that are considered for the analysis behave identically, however some trace-by-trace variation might occur due to incomplete filtering of dysfunctional molecules, fluorescent impurities, or biologically relevant functional heterogeneity (e.g., originating from a variation of post-translational modifications or allosteric control). Such heterogeneity will skew the average correlation function away from that of the pure species of interest and result in large variations of the shape of the correlation functions. In turn, the experimental uncertainty from the bootstrap procedure will be overestimated, and consequently the reduced chi-squared estimator will be systematically underestimated.

4. Slow intensity fluctuations

Slow intensity fluctuations in measurements of immobilized molecules might originate from instabilities of the excitation laser, resulting in slow power fluctuation on the minute to hour timescale, or focal drift due to z-drift of the sample, reducing the detectable signal as the molecule moves outside of the focal plane. As the FRET-FCS analysis is applied directly to the fluorescence intensities, such slow fluctuations will be reflected in the resulting FCS curves. On the other hand, for the filtered-FCS curves the digitized state trajectories are determined based on the FRET efficiency trace which remains unaffected by the signal fluctuations, assuming that the donor and acceptor channels are equally affected and neglecting a change of the noise of the FRET efficiency trace due to the intensity modulation. Slow intensity fluctuation should thus have a minor effect on the filtered-FCS curves.

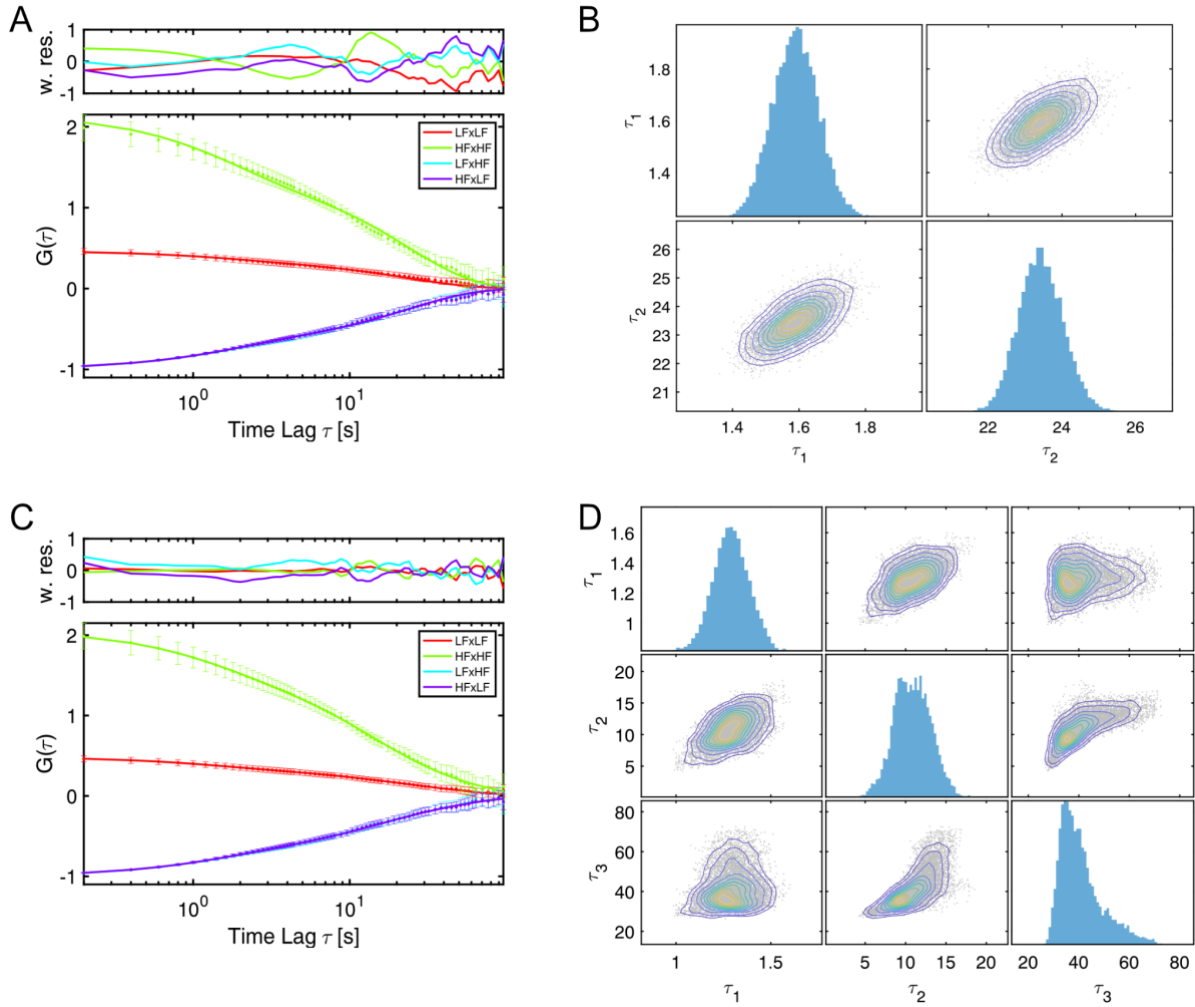
5. Photophysical artifacts

Correlation analysis is sensitive to all effects that modulate the fluorescence intensity. Common unwanted photophysical effects are photoblinking, e.g., due to the population of long-lived dark states such as radical ions²⁵, and intensity changes due to spectral shifts²⁶. Special care should be taken in correlation analysis to avoid that photoblinking of the acceptor fluorophore is interpreted as FRET dynamics, which will feature as a prominent term in the cross-correlation function. Photoblinking of the donor fluorophore is readily detected from a drop of the intensity to the background level. Photoblinking of the acceptor can be identified by the use of alternating laser excitation schemes that include the

intermittent probing of the acceptor fluorophore^{27,28}, allowing to exclude those time intervals where the acceptor was inactive. Generally, the formation of long-lived dark states can be reduced by the application of reducing-oxidizing systems, e.g. by the addition of TROLOX²⁵.

Supplementary Method Table 7.1: Analysis results of the correlation analysis. Rate constants are reported as fitted value \pm 95% confidence interval. All rate constants are given in s^{-1} and relaxation times in s. States are ordered from low to high FRET efficiency, i.e. 1=low-FRET, 2=mid-FRET, 3=high-FRET. For asymmetric confidence intervals (dataset 3, empirical three-exp. model), the lower and upper bounds of the 95% confidence intervals are given in brackets.

Dataset	Method	k_{12} [s^{-1}]	k_{21} [s^{-1}]	k_{13} [s^{-1}]	k_{31} [s^{-1}]	k_{23} [s^{-1}]	k_{32} [s^{-1}]	χ_r^2
1	FRET-FCS	0.142 \pm 0.002	0.210 \pm 0.002	-	-	-	-	1.07
	fFCS	0.142 \pm 0.002	0.212 \pm 0.002	-	-	-	-	0.998
2	fFCS	0.091 \pm 0.003	0.284 \pm 0.003	0.560 \pm 0.005	0.249 \pm 0.003	0.173 \pm 0.003	0.280 \pm 0.003	1.83
		τ_1 [s]	τ_2 [s]	τ_3 [s]	τ_4 [s]		BIC	χ_r^2
3	empirical, 2 exp	1.6 \pm 0.2	23.4 \pm 1.2	-	-	-	71.38	0.13
	empirical, 3 exp	1.3 (1.1-1.5)	10.8 (6.6-15.2)	41 (30-62)	-	-	68.89	0.03
	empirical, 4 exp	1.3 (1.1-1.5)	8.2 (6.3-16.1)	11.2 (6.2-16.8)	40 (30-55)	-	91.19	0.03



Supplementary Method Figure 7.3: Empirical analysis of the correlation functions for synthetic dataset of level 3 involving degenerate FRET states. Same data and sample size as in Figure 4. The data were analyzed using two exponential terms (A-B) or three exponential terms (C-D). **A,C**) Filtered-FCS correlation functions (scatter plots) and fits (solid lines). **B,D**) Posterior distributions of the relaxation times. All times are given in seconds. The χ_r^2 changes from 0.13 to 0.03 and the BIC decreases from 71.38 to 68.89, indicating that the three-exponential model is to be preferred (see Supplementary Method Table 7.1). Inclusion of a fourth component results in an increase of the BIC to 91.19 and shows no further decrease of the χ_r^2 . Error bars in A and C represent the standard error of the mean of the trace-by-trace correlation functions obtained by bootstrapping.

Supplementary Method 8: Edge finding (CK)

A) OVERVIEW

Our method is based on the Chung-Kennedy filtering approach previously used in ion channel experiments²⁹ and subsequently by Haran in single molecule FRET experiments³⁰. Applied to a time-series of points representing donor intensity, acceptor intensity or FRET ratio, we associated each data point with a 'forward' window containing a fixed number of later points and a 'backward' window a number of earlier points. The standard deviation (or variance) of the points in such a window will be largest when an edge occurs near the center of the window. By monitoring the standard deviation of the forward and backward windows when scanning through the data time trace, the location of transition edges can be identified by the maxima of these standard deviations (or variance)³¹.

Additional description of this Chung-Kennedy edge detection method is published elsewhere³¹. The source code is available for download at: <https://www.physics.ncsu.edu/weninger/KinSoft.html> .

B) WORKFLOW

In our implementation for the data in this study, we considered only the FRET efficiency time record and constructed only a single window of points. Any window whose standard deviation exceeded a threshold and was also a local maximum was identified to contain a transition edge at its center data point. The window size and critical standard deviation value were determined empirically from the training sets provided with the challenge and verified by inspection by an experienced user. Next, the forward and backward windows were constructed around these potential transition edge locations and the two sample t-test was applied. Transition edges were accepted if the significance level (alpha) of the t-test comparing these data windows examining the FRET states before and after the potential transition edge was above an empirically determined value set by examining the training sets. For the 2 level systems, parameters were: Local window size = 5 data points; threshold for FRET efficiency variance in the window = 0.03; t-test alpha = 0.6. For the 3 level system: Local window size = 5 data points; threshold for FRET efficiency variance in the window = 0.018. t-test alpha = 0.4. Bleaching and blinking events can be removed in a pre-processing step by identifying segments where the sum of donor and acceptor intensities are below a threshold. Removal of blinking and bleaching was only performed on the experimental data but not on the simulated data where the effect was absent. Events immediately before or after removed intervals were not used for kinetic analyses.

Once transition edges were identified, FRET states were categorized according to Gaussian fitting of all FRET data points. FRET = 0.55 was the dividing line between states for the 2 state systems, and FRET = 0.35 and FRET = 0.7 were the dividing lines between states for the 3 state systems as determined by locating local minima in the histograms of all FRET values. The dwell time in each state between edges was calculated and histograms of dwell times were assembled for each FRET state. Fitting exponential decay functions to the dwell time histograms yielded estimates of the apparent rates of transitions out of the states. Multiplying apparent rates by the fraction of transitions to a specific state divided by the total number of transitions out of a state ('branching ratio') gives the true transition rate in the 3 state system^{32,33}.

C) MISCELLANEOUS

Although not required here, it is notable that for more challenging data that Haran has demonstrated increased sensitivity in transition edge detection is possible by exploiting the expected anti-correlation of donor and acceptor intensities associated with transitions of FRET efficiency³⁰. This method was implemented by examining donor and acceptor intensities separately, forming local 'forward' and 'backward' windows for each, and identifying maxima in the sum of the donor and acceptor variances in these windows to reveal transition edges.

Supplementary Method 9: Edge finding (k-means)

A) OVERVIEW

The k-means edge finding method is based on an iterative clustering algorithm that assigns data into groups based on the similarity of each point to group properties³⁴⁻³⁷.

Our application can apply the k-means edge finding algorithm to time records of donor intensity, acceptor intensity, or FRET efficiency, or a 2-dimensional representation of donor and acceptor intensities. In this project, we only considered the FRET efficiency data. The clustering algorithm groups the FRET efficiency data points without consideration of the time aspect of the record. The number of clusters to be formed is selected based on the apparent number of states in a histogram of all FRET efficiency points for all time traces. There are more formal approaches in k-means theory to determine the number of target clusters³⁸⁻⁴⁰, but those were unnecessary with these data.

The goal is to group the FRET efficiency points into clusters representing the distinct states present in the data. The algorithm proceeds generally by choosing initial values as 'centers' for each group or cluster. All the data points are then assigned to the group which they are closest to by some distance measure without regard to their temporal position in the time trace. The centers of each group are then recalculated by averaging the value of all points in that group. The algorithm then proceeds in an iterative manner whereby all data points are reassigned to the groups minimizing their distance from the new center values. This process of recalculating the center and then reassigning points to the nearest group repeats until points no longer change groups.

The source code is available for download at: <https://www.physics.ncsu.edu/weninger/KinSoft.html>.

B) WORKFLOW

In the specific application for the data in this project, we used FRET efficiency values as the input for the k-means algorithm. Before performing clustering, the data is preprocessed to remove bleaching and blinking. For this project, the simulated data did not require preprocessing, whereas bleaching and blinking was removed from the experimental data by excluding data points where the sum of donor and acceptor intensities was below a threshold determined by inspection of traces (threshold 27 after 22 point smoothing). The events immediately before or after removed intervals were not used for kinetic analyses.

We determined the number of clusters (or number of 'centers') to use by examining time traces and the FRET efficiency histograms assembled from all the data. To choose initial values for the centers of the clusters, all data point values were ordered by FRET efficiency value and then divided into equally sized groups matching the desired number of clusters. A data point was randomly selected from each cluster to serve as the initial "center" for the cluster. Next, each data point was assigned to the cluster for which the 'distance', which is defined as $(\text{data value} - \text{cluster 'center' value})^2$, was minimized. Once all data points were assigned to clusters, the centers for each group were recalculated by averaging the values of all of the data points in the group. Finally, the process was repeated with data points being reassigned based on minimizing the distance to the new centers. This process was repeated until the data points no longer changed clusters. The final value of each "center" was interpreted at the FRET efficiency of a distinct state. All of the data points in a cluster were assigned that FRET state. The data points were then interpreted in the temporal order of the time trace yielding a sequence of FRET state vs. time. Data points where the FRET efficiency state changed were determined as transition edges.

The k-means algorithm assigns points to clusters based on their value without reference to their temporal sequence. For this reason, rare assignments of points to incorrect clusters typically result in time records having transitions to different states that last only one time bin and then return to the previous state. If the data acquisition rate is substantially faster than the rates characterizing the kinetic system, such one time bin events are expected to be rare. For example, if $k_{\text{data acquisition}} = 10^* K_{\text{characteristic kinetics}}$ then ~10% of events are expected to be one time bin ; if $k_{\text{data acquisition}} = 100^* K_{\text{characteristic kinetics}}$ then

~1% of events are expected to be one time bin . If an erroneous one time bin event breaks up a longer dwell into two shorter dwells, it can have a negative impact on determining the kinetic rates of the system. Therefore, we designed a protocol to remove one time bin events that met certain criteria.

For isolated transitions to other states that last one time bin where the clusters on either side are different from each other, we examined the donor and acceptor values in the clusters to decide how to reassign the one frame. We calculated the squared differences of the donor value on the one time bin event from the averaged donor values of all points assigned to the FRET state of each adjacent cluster and a similar squared differences for the acceptor values:

$$\text{ClosenessToXCluster} = \sqrt{(\text{donor value} - \text{average donor value for cluster X})^2} \\ + \sqrt{(\text{acceptor value} - \text{average acceptor value for cluster X})^2}$$

where X indicates which cluster. The one time bin event was reassigned to the cluster for which the *ClosenessToXCluster* was smallest. If the clusters on either side of a one time bin transition were the same, we used an average of the donor and acceptor values of the points in that cluster to calculate the closeness to the current cluster *ClosenessToCurrentCluster* and an average of the points in the sections immediately adjacent to the one bin event to calculate closeness to the adjacent cluster *ClosenessToAdjacentCluster*. We then calculated:

$$\text{DifferenceInDistances} = \text{abs}(\text{ClosenessToCurrentCluster} - \text{ClosenessToAdjacentCluster})$$

and the standard deviation of points in the adjacent cluster:

$$\text{stdAdjacentCluster} = \text{standard Deviation of all donor (or acceptor) values of adjacent cluster.}$$

If the *ClosenessToAdjacentCluster* < *ClosenessToCurrentCluster* and the

DifferenceInDistances < *stdAdjacentCluster* for both donor and acceptor signals, then we assigned the point to the state of the adjacent cluster. Otherwise, we left it in the original cluster, maintaining an event lasting only one time bin.

When multiple one time bin events occurred sequentially, we averaged the FRET efficiency in the 3 or more one time bin events and assigned all of them to the same cluster that was closest.

Once all points were finalized into clusters, the dwell time of the behavior in a state was measured as the time between edges and assembled into histograms for all events in a given state (first and last events in every trace as well as events preceding and following blinking were discarded). The histograms were fit with exponential decay functions to determine the lifetimes. For the 3 state system, the true transition rates were determined from the apparent rates by multiplying by the fraction of transitions to a specific state divided by the total number of transitions out of a state ('branching ratio')^{32,33}.

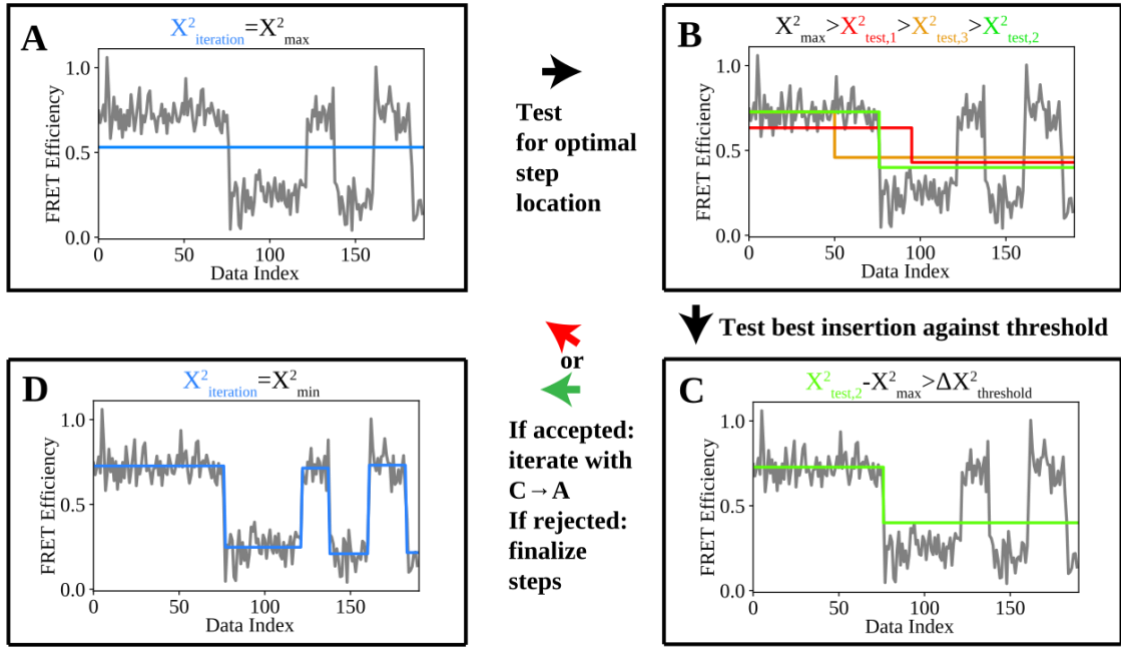
Supplementary Method 10: Step finding

A) Overview

Analysis with step finding was performed using an in-house implementation of a step identification algorithm written in Python (<https://github.com/SMB-Lab/PyStepFinder>) that sorts one-dimensional data into piecewise line segments without enforcing a number of states or kinetic network. Sorting is accomplished through iterative forward addition of line segments to a piece-wise linear fit of each trace. At each iteration, starting with one line for the whole trace, all current line segments are tested to find the optimal position at which to split the segments into two segments. The sum of squared residuals ($X^2 = (x - \underline{x})$, with x the value of the test statistic and \underline{x} the mean of the test statistic for the segment) is used as the quality of fit statistic for the fit of the two new segments. This value is compared to the previous segment and the addition of a segment is accepted if the improvement in the quality of fit is significant. This process is summarized visually in (Supplementary Method Figure 10.1). Several options can be specified in performing step finding with this implementation, including choice of the statistic of interest for fitting (mean, rms, variance, slope), a sliding window size for calculating running averages of observed statistics of interest for comparison to the segment means, a minimum length of each line segment (in number of data points), and a threshold for improvement in quality of fit per segment addition as well as a thresholding mode (ratiometric, flat improvement, statistical tests, etc.). For this study, all analysis was performed using the segment means as the statistic of interest with no running average, a minimum segment length of two data points, and the quality of fit threshold set to the variance of each trace. For other test statistics, represents a running average value of the test statistic, in which case the minimum segment length must be at least the size of the averaging window. PyStepFinder is appropriate for other series data as well. We have used a similar tool available in MATLAB previously for analysis of time-series force data from optical tweezer experiments⁴¹. Currently, PyStepFinder is unable to identify states with degenerate mean signals unless those states are distinguishable by other parameters, such as signal variance. Thus, for this study the step finding algorithm was only used to resolve the non-degenerate FRET efficiencies in degenerate datasets.

B) Workflow

Step finding was performed for each FRET efficiency trace to identify. When FRET efficiency was not provided directly, it was calculated using $\text{Efficiency} = I_A / (I_A + I_D)$, where I_A and I_D are the donor and acceptor signal intensities, respectively. Following identification of each transition step in the dataset, each line segment was classified as belonging to one of the non-degenerate FRET states. The number of states was determined by user input following histogramming of FRET efficiency data. The means and widths of these states were determined using Gaussian fits. Line segments then were classified into one of the states based on their mean FRET efficiencies. Dwell times for each line segment were determined by multiplying the number of points in each by the binning resolution of the input dataset. Kinetic rate constants were determined per state from $K_i = 1/\underline{\tau}$, with $\underline{\tau}$ the average dwell time within a state. Rate constants for individual kinetic pathways (i.e., state i to j , k_{ij}) were determined by multiplying K_i by the ratio of transitions from state i to state j the total number of transitions from state i , $k_{ij} = K_i \cdot N_{ij}/N_i$. Uncertainty estimates for k_{ij} were determined by calculating the standard error of the mean (SEM) associated with the mean dwell time ($\text{SEM}_{\tau,i} = \sigma_{\tau,i}/\sqrt{N_i}$) with $\sigma_{\tau,i}$ the standard deviation of the dwell times. The SEM was then propagated through to each kinetic rate using the standard uncertainty propagation formula, $\Delta k_{ij} = \sqrt{\text{SEM}_{\tau,i}^2 \cdot (\partial k_{ij} / \partial \tau_i)^2}$. The first and last line segment for each trace were ignored for this analysis to avoid artificially reduced dwell times associated with each state.



Supplementary Method Figure 10.1: Forward addition of steps in PyStepFinder. A) Data is initially fit with a horizontal line at the mean of the data test statistic and initial X^2 is calculated. B) Candidate locations are tested for insertion of a step between two segments to replace the original segment. X^2 is calculated for each potential replacement and the one with the lowest X^2 is taken as the best candidate. C) The change in the X^2 calculated for the insertion candidate relative to the original iteration X^2 is compared to a threshold value. The criterion for the flat improvement is shown, but other thresholding modes can be defined. If the improvement in X^2 satisfies the threshold, iteration continues from A for all existing segments. If the criterion is not met, the segment is accepted as final until no further insertions are accepted for any segment. D) The finalized trace for which insertion of any additional segments would not significantly improve the fit.

Supplementary Method 11: STaSI

A) OVERVIEW

The Step Transition and State Identification (STaSI) method was introduced in 2014 to determine the number of states and step transitions between states for piecewise constant data with a minimum description length (MDL) as the objective function⁴². The step transitions are detected using the Student's t- test and the segments are grouped into states by hierarchical clustering. The optimum number of states is then established using a minimum description length equation that sums the goodness of fit measured using L1 norm and the complexity of the fitting model derived to consider the sparseness of the states and transitions among states. More details on this can be found in the original report⁴².

Benefits of STaSI include not requiring time-tagged photon counting or photon counting in general. STaSI provides a better resolution to interpret noisy data with fast dynamics to avoid the need for binning. Binning can introduce artifact states in between real states and limits the temporal resolution of single-molecule FRET. STaSI also is objective, requiring no assumptions about the model and no user inputs other than the FRET efficiency trace in determining the number of states and transitions. Finally, the algorithm is written for smFRET data, but could be used for any piecewise constant signal. Here, we apply STaSI to the kinSoftChallenge data. While STaSI was developed only for state and step transition identification, we analyze the kinetics of the transitions identified by STaSI with a simple exponential fitting method to maintain its ease-of-use. Overall, STaSI is a user-friendly, objective method developed to analyze a variety of piecewise constant data conditions, including data with low signal-to-noise ratios, but will require future development to handle data with degenerate kinetics in different states.

CITATION: Shuang, B.; Cooper, D.; Taylor, J. N.; Kisley, L.; Chen, J.; Wang, W.; Li, C. B.; Komatsuzaki, T.; Landes, C. F. Fast Step Transition and State Identification (STaSI) for Discrete Single-Molecule Data Analysis. *J. Phys. Chem. Letters*. **2014**, 5, 3157 – 3161.

<https://pubs.acs.org/doi/10.1021/jz501435p>.

DOWNLOAD LINK: <https://github.com/LandesLab/STaSI>

B) WORKFLOW

The provided donor intensity (I_D) and acceptor intensity (I_A) values were converted to a FRET efficiency (E) using:

$$\text{Eq. 1} \quad E = \frac{I_A}{I_D + I_A}.$$

The STaSI GUI was then executed on the FRET E where all traces were analyzed together.

The grouped states and step transitions for the global minimum MDL was saved as a FRET E vector as the output of STaSI. Non-physical states identified with FRET E > 1 or < 0 were not considered in the final number of states or any of the kinetic analysis. The "FRET E" was the assigned state levels of the STaSI output. The "sigma FRET"/ σ (FRET E) was calculated from the standard deviation of the raw trace_i.txt FRET E data assigned to each state level in STaSI.

The "total duration of inference" for the STaSI analysis was recorded using the tic and toc functions in MATLAB R2018a executed after the user input. All the computations were done using a Dell desktop computer with an Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz processor, 16.0 GB RAM and 64-bit Windows 10 Enterprise (2018) operating system.

The kinetics were calculated by an exponential fit of the cumulative distribution of dwell times spent in each state before transitioning to another state. Any false transitions caused by concatenating the individual traces together in the STaSI analysis were removed (*i.e.* connecting end of trace i with the beginning of trace $i+1$). The cumulative probability distributions of dwell times for each state were fit to a simple exponential decay model,

$$\text{Eq. 2} \quad P(t > \tau) = \sum_{i=1}^n A_i e^{-k_i \tau},$$

where $P(t > \tau)$ is the cumulative distribution of observing a dwell time of $\geq \tau$ for a given time t , A is the amplitude, k is the rate, and n being the number of components⁴⁴. MATLAB's built in 'fit' function using the Trust-Region-Reflective Least Squares was used with the inverse of the mean value of dwell time

as an initial guess for k and 1 for A . Data for Experiment 1 and Experiment 3 were fit with $n = 1$. Data from Experiment 2 was fit with $n = 2$.

Since none of the experimental data resulted in a simple two-state transition without degeneracies we further analyzed the resulting values from Equation 2 by^{43,45}:

$$\text{Eq. 3} \quad k_{ab} = A_{ab} / [\sum_{i=1}^n \left(\frac{A_{ai}}{k_{ai}} \right)].$$

Here, a is the starting state and b is the ending state for the kinetic rate transition of interest and n represents the total number of states. A_{ab} can either be extracted from the fit from non-normalized cumulative distributions, or, as we used here, the total number of transitions observed from state a to state b . For example, Equation 3 states that the rate of state 1 to state 2 transition ($a=1$ and $b=2$) is calculated as the ratio between the number of transitions from state 1 to state 2 divided by the total dwell time in state 1 spent before transitioning any other state (2, 3, ... n). This method is used because, $k_{ab} \neq 1/\tau_{ab}$ for non unimolecular reactions⁴⁵.

Supplementary Method 12 & 13: MASH-FRET

A) OVERVIEW

MASH-FRET is a MATLAB-based software package freely available on github (<https://github.com/RNA-FRETTools/MASH-FRET>) demanding the following toolboxes:

- Symbolic Math Toolbox
- Image Processing Toolbox
- Statistics and Machine Learning Toolbox
- Curve Fitting Toolbox

MASH-FRET was developed on MATLAB2016a but has recently passed to MATLAB2020b. Computation times were measured on a computer equipped with an Intel Core i7-3632QM CPU (2.2GHz) and 8GB RAM.

The steps to reproduce the results obtained are the following⁵²:

1. Install and start MASH-FRET v.1.3.2 as described in the online documentation (https://rna-frettools.github.io/MASH-FRET/Getting_started.html)
2. Go to MASH-FRET's menu *Routines > Standard analysis > All steps* and select the set of files to analyze
3. A first message box pops up: enter the number of FRET states if known or press "No" otherwise
4. A second message box pops up: choose the proper noise distribution according to your data set
5. Once the analysis routine is completed, you can find the analysis summary in file *[data file name]_results_[J]states.txt* at the same location as your data files.

B) WORKFLOW

a. Determination of the FRET state configuration

The procedure used to determine the FRET state configuration with the MASH-FRET (bootstrap) and (prob.) methods is based on the conclusions of a preliminary comparative study of several algorithms⁴⁶.

The provided acceptor $I_A(t)$ and donor $I_D(t)$ intensity signals were not further processed as they were provided free of background and dye photobleaching. Additionally, the necessary data to correct the differences between donor and acceptor quantum yields, *i.e.*, the control acceptor signal upon acceptor direct excitation, was not part of the provided data sets. FRET-time traces $FRET(t)$ were therefore directly calculated according to:

$$FRET(t) = \frac{I_A(t)}{I_A(t) + I_D(t)}. \quad (12.1)$$

Aberrant FRET values below -0.2 and above 1.2 FRET units were ignored in the following analysis.

Individual FRET-time traces were discretized into FRET state sequences using the algorithm STaSI⁴². Because this algorithm does not make any assumptions about the kinetics of state transitions, and thus prevents the detection of false transitions towards noise-induced artefactual states, it has proven to be the most suitable to identify the genuine FRET states⁴⁶. The maximum number of states to be found in each FRET trajectory was arbitrarily set to a large number, *e. g.* 10.

To group the FRET states of all sequences into one global state configuration, we chose to sort them drawbacks of a one-dimensional distribution, *i.e.*, the merging of state populations having similar FRET values, by splitting the population along an additional axis: the FRET state forwarding the transition in the trajectory. After smoothing with a Gaussian filter, the TDP was modelled with a mixture of isotropic

2D-Gaussians, which centers were locked on a V -by- V grid, with V the number of global FRET states. In addition, Gaussian clusters on the TDP diagonal were used to group, and then exclude from the analysis, the artefactual and noise-induced low-amplitude state transitions. To determine the most sufficient model size V_{opt} , Gaussian matrices with increasing dimension, *i.e.* $V=2$ to 10, were fitted to the TDP using an expectation-maximization (EM) approach, and the model rendering the lowest Bayesian information criterion (BIC), calculated as

$$\text{BIC}(V) = (2V^2 + V - 1)\log(M) - 2\log[l(V)], \quad (12.2)$$

where M is the total number of states in the trajectories and l the likelihood of the model, was selected.

With the number of observable FRET states at hand, FRET-time traces were re-discretized into more accurate state sequences with the Bayesian-based algorithm vbFRET. Indeed, although a model-free algorithm provides a better global view of the state configuration, it is not suitable for detecting short-lived states. However, vbFRET is designed for Gaussian-distributed trajectory noise and fails to properly characterize *e. g.* low-photon-count trajectories that generate Poisson noise. In such cases, the sequences generated by STaSI were used. The vbFRET algorithm was constrained to found V_{opt} states at maximum and its well-known propensity to detect artefactual blur states⁴⁷ was post-corrected by ignoring all one-data-point states found in trajectories.

To accurately determine the global FRET states, a mixture of multivariate Gaussians, which centers were locked on a V_{opt} -by- V_{opt} grid, was fitted to the new TDP using the EM approach mentioned above. Global FRET values were derived from the Gaussian means and the associated errors, δ_{FRET} , from the average Gaussian standard deviations in the x -direction σ_x , such as:

$$\delta_{\text{FRET},v} = \frac{1}{V-1} \sum \sigma_{x,vi}. \quad (12.3)$$

b. Estimation of the transition rate constants

Transition rate constants were determined in two different ways: the *bootstrap* and the *probabilistic* (*prob.*) approaches. The *bootstrap* approach is only suitable for non-degenerate state systems, *i.e.*, for states with distinct FRET values, whereas the *prob.* approach suits all types of systems.

The "bootstrap" approach

Dwell times Δt associated to each global FRET state were collected from accurate FRET state sequences and normalized cumulative distributions F were built. The complementary distribution was subsequently fit with a single exponential function such as

$$1 - F(\Delta t_v) \sim \exp\left(\frac{-\Delta t_v}{\tau_v}\right), \quad (12.4)$$

where τ_v is the state lifetime. As time-binned data suffer from the absence of very short dwell times, the normalized complementary cumulative histogram $1 - F(\Delta t_v)$ of dwell times Δt_v is used instead of raw counts. This minimizes the impact of the first histogram bins while preserving the overall shape.

Transition rate constants k were derived from the state lifetimes and the numbers of transitions w using the relation

$$k_{v,v'} = \frac{w_{v,v'}}{\tau_v \sum w_{v,k}}. \quad (12.5)$$

Please note that for a two-state system, the transition rates are the direct inverse of the lifetimes, *i.e.*, $k'_{v,v'} = \frac{1}{\tau_v}$.

The outcome of such analysis are single estimates of the rate constants. To estimate the error δ_k on rate constants k , the variability of state lifetimes across the trajectory sample is evaluated using the bootstrap-based analysis called BOBA-FRET⁴⁸. BOBA-FRET infers the bootstrap means and bootstrap standard deviations of all fitting parameters for the given sample, including τ . The variability can then be propagated to k such as:

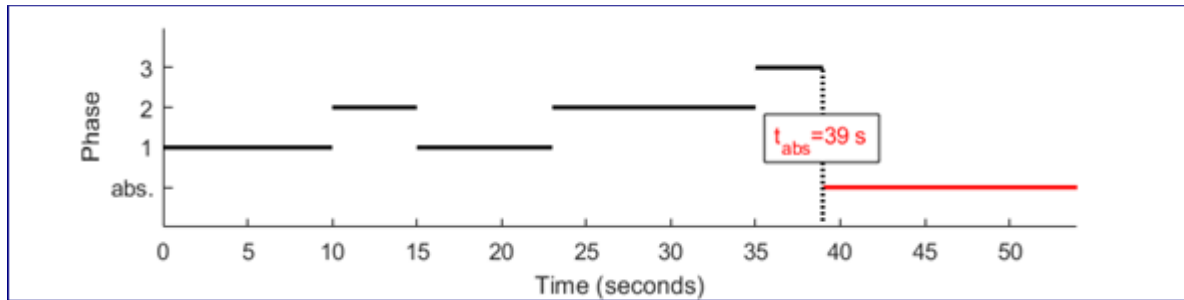
$$\delta_{k,v,v'} = \frac{\sigma_{\tau,v}}{\bar{\tau}_v} \bar{k}_{v,v'} \quad (12.6)$$

where $\bar{\tau}$ and σ_τ are respectively the bootstrap mean and standard deviation of parameter τ and \bar{k} is the rate constant derived from $\bar{\tau}$ using Eq. 12.5. Intervals with 95% confidence were given as $\bar{k} \pm 2\delta_k$.

The "probabilistic" approach

The presence of degenerate states usually breaks the single exponential shape of the dwell time distribution, resulting in sums and convolutions of multiple distributions. The *probabilistic* approach first solves the state degeneracies, *i.e.* the numbers of degenerate states hidden behind the same FRET values, from the shapes of ensemble dwell time distributions, and second, optimizes the transition probability matrix for the set of FRET state sequences.

Phase-type distributions (PH) are used, *e. g.* in queuing and insurance risk theory, to estimate the time, t_{abs} , a Markov jump process takes to reach an absorbing state, depending on the number of phases D it can go through. Such a jump process involving 3 phases is illustrated below (Supplementary Method Figure 13.1)



Supplementary Method Figure 13.1. Illustration of a Markovian jump process. going through a number of phases $D = 1 \dots 3$, *i.e.* the number of degenerated states, before reaching the absorbing state, thus, the observable state transition, *i.e.* between two observed FRET states.

In comparison to our problem, the phases labeled 1 to D are the degenerate states behind a same FRET value, the Markov jump process characterizes the transitions between these degenerate states, the absorbing state is any state having a different FRET value, and the absorbing times t_{abs} are the dwell times Δt measured in the state sequences. Therefore, PH distributions make perfect candidates to model the dwell time histograms compiled for a degenerate state system. As the data provided for analysis were time-binned trajectories, discrete PH distributions (DPH) were used instead. The DPH probability density function f depends on transition probabilities between degenerate states and to the absorbing state (state 0), p , as well as on starting probabilities, π . All in all, it is expressed as:

$$f(\Delta t_v) = (\pi_1, \pi_2, \dots, \pi_D) \times \begin{pmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,D} \\ p_{2,1} & p_{2,2} & \dots & p_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ p_{D,1} & p_{D,2} & \dots & p_{D,D} \end{pmatrix}^{\Delta t_v - 1} \times \begin{pmatrix} p_{1,0} \\ p_{2,0} \\ \vdots \\ p_{D,0} \end{pmatrix} = \pi T^{\Delta t_v - 1} \mu \quad (13.1)$$

Where π is called the initial distribution of phases, T the sub-intensity matrix and μ the exit rate vector.

After re-binning the dwell times using a bin size 10-time larger than the resolution time in order to minimize the impact of the lack of very short dwell times in time-binned data while preserving the overall shape, dwell time histograms were modelled with a DPH involving D degenerate states. To determine the most sufficient model size D_{opt} for each histogram, DPHs with increasing dimensions, *i.e.* $D = 1$ to 4, were fitted using an EM approach described previously⁴⁹ and the combined model rendering the lowest BIC was selected. In our particular case, the BIC of the combined model was calculated as the sum of the BIC values obtained for individual dwell time histograms, such as:

$$\text{BIC} = \sum \text{BIC}(D_v) = \sum np(D_v) \times \log(M_v) - 2 \sum l(D_v), \quad (13.2)$$

where M_v is the number of dwell times in the histogram, l the likelihood of the model, and where the number of free parameters np is calculated as:

$$np(D) = D^2 - 1 \quad (13.3)$$

With the final model size at hand, we determined the corresponding transition rate constants by applying the Baum-Welch⁵⁰ algorithm to state trajectories, *i.e.*, to noiseless trajectories, in which the state assignment is inflexible. Therefore, the algorithm only optimizes the transition probability matrix by iterating expectation and maximization of state probabilities at each time bin of each state trajectory. It eventually converges to a maximum likelihood estimator of transition probabilities that are then converted into rate constants, using the relation

$$k_{j,j'} = \frac{p_{j,j'}}{t_{\text{exp}}} \quad (13.4)$$

where $k_{j,j'}$ is the rate constant that governs transitions from state j to state j' (in seconds⁻¹) and t_{exp} is the bin time in trajectories (in seconds).

The negative and positive errors δ_k^- and δ_k^+ on rate coefficients were estimated via a 95% confidence likelihood ratio test described elsewhere⁵¹, giving an estimated range delimited by the lower bound $k - \delta_k^-$ and the upper bound $k + \delta_k^+$.

To ensure the validity of the inferred model, a set of synthetic state trajectories is produced using the kinetic model parameters and the experimental mensuration (sample size, trajectory length), which is then compared to the experimental data set⁴⁵. Special attention is given to the shape of each dwell time histogram, the populations of observed states and the number of transitions between observed states.

C) MISCELLANEOUS

MASH-FRET delivers transition rates restricted to 2-state systems up to version v.1.2.1 and below. This has been corrected. The development of MASH-FRET (prob.) has been initiated by the kinsoftchallenge to solve degenerate FRET-state systems, *i.e.*, FRET states comprising kinetic heterogeneity. Therefore, the analyses of round 1 and 2 have been repeated with the new software version of MASH-FRET (prob.) and are labelled as "post-ground truth submission" where necessary. Further, the determination of the number of observable FRET states in MASH-FRET was modified meanwhile the submission process, which led to discrepancies between the bootstrap and the prob. method (compare Fig. 5 of the main text). The most recent version v.1.3.2 of the software yields two observable FRET states for all three data sets for both methods and as presented for MASH-FRET (prob.) in Fig. 5b,e and h of the main article.

Supplementary Method 14: postFRET

A) OVERVIEW

The concept of the postFRET analysis⁴⁵ is to fit the experimental single-molecule FRET (smFRET) data first then simulate similar data for comparison. Source codes (MATLAB) are available at <https://github.com/nkchenjx/postFRET>. A simple thresholding method is used, i.e. set a threshold (e.g. the FRET value in the middle of two states) to distinguish the two states. This kind of analyzed results contains two major errors: (1) state miss-assignment due to the noise, (2) state miss-assignment due to camera blurring. After assigning states with the threshold, >hundreds of virtual data are simulated in the hope that one can find one or more trajectories that look just like the experimental data using the same analysis method, e.g. the thresholding method. Because we know the ground truth of the simulated data, we assume that the hidden truth of the real experimental data is the same as the simulated data that looked the same (minimizing L1-norm, the absolute values of the percentage errors, as the judging standard in Ver 1.0 and 2.0). L1-norm is used instead of L2-norm (such as the least square root method). The former works better in many simulated conditions in postFRET.

The guessing algorithm of the simulated data used in the codes is a semi-exhaustive searching algorithm called JCFit (available on GitHub, <https://github.com/nkchenjx/JCFit>), a fitting algorithm that searches a parameter in an equation (model) within a defined boundary. The searching spacing is exponentially distributed away from the initial guess to the boundary. E.g. -10 to 10 are the boundaries and 1.0 is the initial guess, and 0.1 is the searching accuracy and $\ln(2)$ is the exponential factor, then the searching space is [1.0, 1.1, 1.3, 1.7, 2.5, 4.1, 7.3, 10] going up, and [1.0, 0.9, 0.7, 0.3, -0.5, -2.1, -5.3, -10] going down. The boundaries in version 2.0 are set mobile among searching iterations.

B) WORKFLOW

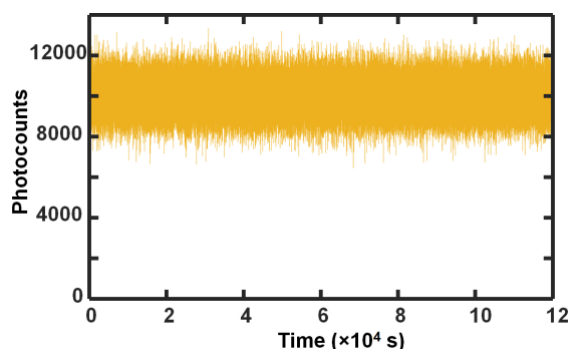
The MATLAB codes are divided into a few steps with the file name Sx_xxxx.m, where Sx represents the step order. Examples are given for the training data level 1.

Step 1. Load data. Load all trajectories into one single matrix and mark the end of each trajectory in a separate vector. The photobleaching information is analyzed and used later in the postFRET analysis.

Step 2. Load key. If the key of the rate constants is known (as in the training data), load the key (type in manually). This step is not needed for real data. Thus, if the key is not known, give a random guess based on the number of states observed.

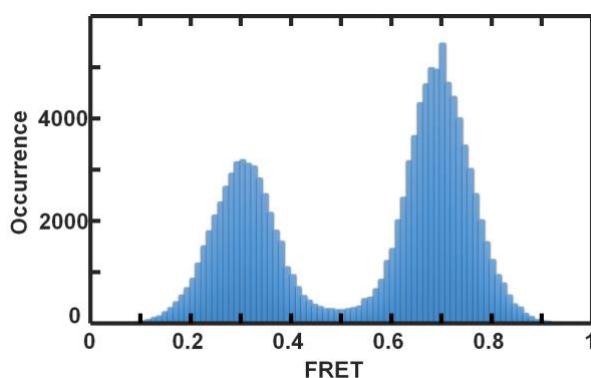
For the level 1 training data, the key is given: **[0, 0.666; 1, 0]**. Note the diagonal in postFRET is always 0. The direction is column to row, 90 degree rotation of the kinSoftChallenge format.

Step 3. Find the noise. Normalize the total counts of the acceptor and donor channel for each trace (**Supplementary Method Figure 14.1**). Then analyze the noise model in the sum of the two channels using the standard deviation of the normalized total signal. Then calculate the average noise of each channel. The latter is used to simulate the trajectory later. The noise model is pretended to be unknown and a Gaussian model is identified for the training data.



Supplementary Method Figure 14.1. The total photocounts of the 100 traces. This trajectory is used to analyze the noise level of the signal.

Step 4. Manually determine the number of states (**Supplementary Method Figure 14.2**), the state values and analyze the data using the simple thresholding method (**Supplementary Method Figure 14.3**). The thresholds are set in the middle of two adjacent states.

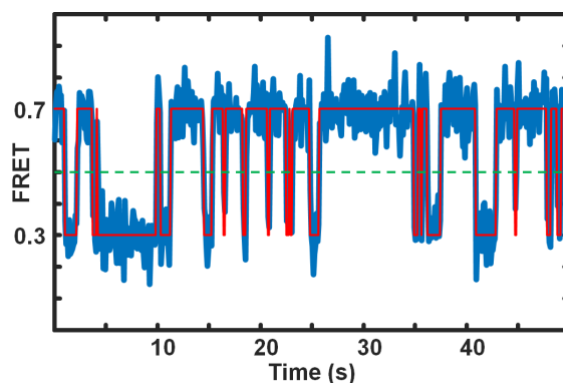


Supplementary Method Figure 14.2. Two states are identified with FRET values of 0.3 and 0.7. The uncertainty is the sigma of the two Gaussian peaks.

The thresholding analysis gives the transition rates: **[0,0.63; 0.92, 0]** (**Supplementary Method Figure 14.3**). The detailed procedure has been described in the cited paper and its supporting information. This value is only slightly biased to the truth by the noise because the signal-to-noise level is high in this set of data. It will be more biased with a higher noise level. Briefly⁴⁵

$$k_{if} = \frac{N_{if}}{\sum_{f=1}^N t_{if}} = \frac{N_{if}}{t_i}$$

Where k_{if} is the rate constant from state i to state f , N_{if} is the fitted total number of transitions from state i to state f , t_{if} is the sum of the dwell times of state i to state f , and t_i is the total dwell time in state i . If the dwell time is only one pixel, it is merged to the previous state as noise. The first and the last transition of a trace is also removed from the counting to avoid the edge effect.



Supplementary Method Figure 14.3. Thresholding analysis of the FRET trajectory (showing the first 50 s data). The dashed line indicates the threshold.

The total computational time of all the above steps is negligible on a regular desktop computer for this set of data (less than 1 minute). The thresholding state identification takes 1.3 s, which is linearly proportional to the length of the data.

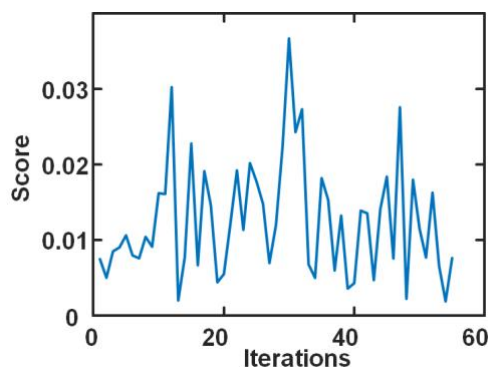
Step 5. postFRET analysis by simulating data with the same photoblinking value and the same noise to find similar trajectories to the raw data. The rate constant key is ignored here so one can just compare the rate constants between the raw data and the simulated data. The goal is to minimize the difference by searching the “real rates” of the simulations. The scoring equation is:

$$WL = \sum_i \left| \frac{R_{E,i} - R_{S,i}}{R_{E,i}} \right|$$

where R_E is the analyzed rate of the experimental data, and R_S is the rate of the simulated data, i is the i^{th} non-zero rate in the rate matrix.

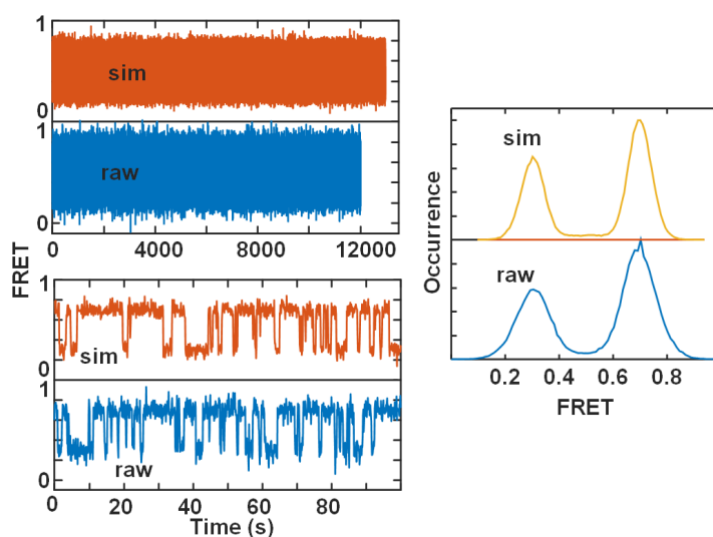
The searching space (defined by boundaries) is from $\frac{1}{2}$ to 2 times the initial guess that has been exponentially distributed from the guess value to the boundaries. Thus, the boundary is changing when the initial guess changes from iteration to iteration.

For the level 1 data, the algorithm finds a simulated trajectory very similar to the raw data in the first iteration and this is repeated a number of iterations showing the variation of the scores (**Supplementary Method Figure 14.4**). The best score is approaching the theoretical best 0% during some iterations. However, the value varied in a small region from 3% to 0%. Because no consistent score decay is observed, all values of all iterations are kept for error analysis.



Supplementary Method Figure 14.4. The score of the postFRET over the iterations. The computation time of each iteration is ~ 100 s (single CPU) on a regular desktop with a 3.4 GHz Intel i7 CPU. Parallel computing of multiple CPU and GPU has not been activated.

The mean value and standard deviation of the rate constants of the 55 iterations are $[0, 0.66 \pm 0.02; 0.95 \pm 0.02, 0]$. Comparing to the key $[0, 0.666; 1, 0]$, this is a better value than the results obtained from the simple thresholding method $[0, 0.63; 0.92, 0]$. An example trajectory of the 55 guesses is shown in **Supplementary Method Figure 14.5**. One can see that the simulated data is different from the raw data but carries similar state and kinetic information.



Supplementary Method Figure 14.5. An example simulated trajectory comparing to the raw data. The bleaching time is stochastic. The simulated data in this example is longer than the real data. Zoom in on the trajectory of 100 traces showing a similar pattern between the simulated (sim) data and the raw data (left). The distribution of the two states also shows a similar pattern (right). The raw data is wider in the distribution than the simulated data indicates that the noise level is slightly lower-estimated in the simulated data. No attempt is tested to increase the noise level of the simulation to match the raw data.

The error bars associated with the mean rate constants of these iterations are obtained from the standard deviation of the rates among iterations. Two times the error bars represent 95% confidence of the boundaries. The positive (upper) and the negative (lower) standard deviation are slightly different in this set of data which is also reported. Because the noise is relatively symmetric in the data, the difference is small.

C) MISCELLANEOUS

Running time. The total computational time of steps 1 to 4 is negligible on a regular desktop computer (CPU Intel i7 3.4 GHz) for this set of data (less than 1 minute). The thresholding state identification takes 1.3 s, which is linearly proportional to the length of the data. The running time of step 5 is ~100 s each iteration for the training data, which is linearly proportional to the length of the raw data, and linearly proportional to $n(n-1)$, where n is the number of states since the codes search each transition in an iteration. Parallel and GPU computation can significantly reduce the simulation time for postFRET.

The postFRET code works for two-state and multi-states smFRET analysis but is not coded to detect degenerated states. The code only analyzes FRET values and does not analyze the donor and acceptor channel separately. To make that change, the concept should still work but a significant amount of modification is required.

The postFRET code is expected to be more competitive in analyzing slightly noisier data and data with very fast transition rate constants approaching the time resolution of data collection. That kind of data has significant amounts of camera blurring events. However, it cannot analyze too noisy data when state mis-assignment becomes too large for the thresholding method. For those data, binning must be applied to increase the signal-to-noise ratio or other state-identification methods are needed to replace the thresholding method.

4 Supplementary Tables

We provide here all inferred values concerning the data discussed in Figs. 4 and 5 of the main text. This is an excerpt of the complete inferred results found in the Supplementary Datafiles (excel sheets). The inferred FRET efficiency levels and kinetic models are specified using the following nomenclature:

Nomenclature

FRET E	E ₁	E ₂	...
kinetic model	0	k ₁₂	...
	k ₂₁	0	...

Supplementary Tables 1

The rate constants of the GT and the inferred models shown in Fig. 4 of the main text. The rate constants are specified in s⁻¹. The full submission, including standard deviations of the FRET efficiencies and uncertainties of the rate constants, can be found in the Supplementary Datafiles. Please note: the order of states in the Supplementary Datafiles corresponds to the submission of the participants and may thus differ from the order given here.

0) Ground truth

FRET E	E ₁ = 0.18	E ₂ = 0.18	E ₃ = 0.73	E ₄ = 0.73
kinetic model	k ₁₁ = 0	k ₁₂ = 0.053	k ₁₃ = 0	k ₁₄ = 0.018
	k ₂₁ = 0.080	k ₂₂ = 0	k ₂₃ = 0.250	k ₂₄ = 0
	k ₃₁ = 0	k ₃₂ = 0.680	k ₃₃ = 0	k ₃₄ = 0
	k ₄₁ = 0.032	k ₄₂ = 0	k ₄₃ = 0	k ₄₄ = 0

1) Pomegranate

FRET E	E ₁ = 0.181	E ₂ = 0.714
kinetic model	n.a.	n.a.
	n.a.	n.a.

2) Tracy (HMM)

FRET E	E ₁ = 0.18	E ₂ = 0.18	E ₃ = 0.73	E ₄ = 0.73
kinetic model	k ₁₁ = 0	k ₁₂ = 0	k ₁₃ = 0.0937	k ₁₄ = 0.0291
	k ₂₁ = 0	k ₂₂ = 0	k ₂₃ = 0	k ₂₄ = 0.8846
	k ₃₁ = 0.953	k ₃₂ = 0	k ₃₃ = 0	k ₃₄ = 0
	k ₄₁ = 0.0971	k ₄₂ = 0.0484	k ₄₃ = 0	k ₄₄ = 0

3) FRETboard

FRET E	E ₁ = 0.196	E ₂ = 0.658	E ₃ = 0.752
kinetic model	k ₁₁ = 0	k ₁₂ = 0.125	k ₁₃ = 0.003
	k ₂₁ = 0.513	k ₂₂ = 0	k ₂₃ = 0
	k ₃₁ = 0.027	k ₃₂ = 0	k ₃₃ = 0

4) Hidden-Markury

FRET E	$E_1 = 0.186$	$E_2 = 0.186$	$E_3 = 0.725$	$E_4 = 0.725$
kinetic model	$k_{11} = 0$	$k_{12} = 0.039$	$k_{13} = 0$	$k_{14} = 0.017$
	$k_{21} = 0.047$	$k_{22} = 0$	$k_{23} = 0.246$	$k_{24} = 0$
	$k_{31} = 0.045$	$k_{32} = 0.569$	$k_{33} = 0$	$k_{34} = 0.003$
	$k_{41} = 0.037$	$k_{42} = 0$	$k_{43} = 0$	$k_{44} = 0$

5) SMACKS(SS)

FRET E	$E_1 = 0.19$	$E_2 = 0.19$	$E_3 = 0.72542$	$E_4 = 0.72542$
kinetic model	$k_{11} = 0$	$k_{12} = 0.04362695$	$k_{13} = 0$	$k_{14} = 0.01930565$
	$k_{21} = 0.063392$	$k_{22} = 0$	$k_{23} = 0.226308$	$k_{24} = 0$
	$k_{31} = 0$	$k_{32} = 0.601065$	$k_{33} = 0$	$k_{34} = 0$
	$k_{41} = 0.0351999$	$k_{42} = 0$	$k_{43} = 0$	$k_{44} = 0$

6) SMACKS

FRET E	$E_1 = 0.19$	$E_2 = 0.19$	$E_3 = 0.71$	$E_4 = 0.71$
kinetic model	$k_{11} = 0$	$k_{12} = 0.0428$	$k_{13} = 0.0001$	$k_{14} = 0.0195$
	$k_{21} = 0.0584$	$k_{22} = 0$	$k_{23} = 0.2254$	$k_{24} = 0$
	$k_{31} = 0.0055$	$k_{32} = 0.5985$	$k_{33} = 0$	$k_{34} = 0.0034$
	$k_{41} = 0.0359$	$k_{42} = 0$	$k_{43} = 0$	$k_{44} = 0$

7) Correlation

FRET E	$E_1 = 0.18$	$E_2 = 0.18$	$E_3 = 0.73$	$E_4 = 0.73$
kinetic model	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.

8) Edge finding (CK)

n.a.

9) Edge finding (k-means)

n.a.

10) Step finding

FRET E	$E_1 = 0.185$	$E_2 = 0.726$
kinetic model	$k_{11} = 0$	$k_{12} = 0.19$
	$k_{21} = 0.327$	$k_{22} = 0$

11) STaSI

FRET E	$E_1 = 0.2$	$E_2 = 0.7$
kinetic model	n.a.	n.a.
	n.a.	n.a.

12) MASH-FRET (bootstrap)

FRET E	$E_1 = 0.186$	$E_2 = 0.186$	$E_3 = 0.726$	$E_4 = 0.726$
kinetic model	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.

13) MASH-FRET (probabilistic)

FRET E	$E_1 = 0.181$	$E_2 = 0.181$	$E_3 = 0.708$	$E_4 = 0.708$
kinetic model	$k_{11} = 0$	$k_{12} = 0.045$	$k_{13} = 0$	$k_{14} = 0.024$
	$k_{21} = 0.050$	$k_{22} = 0$	$k_{23} = 0.233$	$k_{24} = 0$
	$k_{31} = 0.033$	$k_{32} = 0.569$	$k_{33} = 0$	$k_{34} = 0$
	$k_{41} = 0.043$	$k_{42} = 0$	$k_{43} = 0$	$k_{44} = 0$

14) postFRET

FRET E	$E_1 = 0.19$	$E_2 = 0.73$
kinetic model	$k_{11} = 0$	$k_{12} = 0.1$
	$k_{21} = 0.139$	$k_{22} = 0$

Supplementary Tables 2

Kinetic models for the data shown in Fig. 5a-c of the main text, inferred by the participating groups with the specified tools. Units of the rate constants are in s^{-1} . The full submission, including standard deviations of the FRET efficiencies and uncertainties of the rate constants, can be found in the Supplementary Datafiles.

1) Pomegranate

FRET E	$E_1 = 0.205$	$E_2 = 0.489$	$E_3 = 0.719$	$E_4 = 0.927$
kinetic model	$k_{11} = 0$	$k_{12} = 0.5428$	$k_{13} = 0.7998$	$k_{14} = 0.6016$
	$k_{21} = 2.6295$	$k_{22} = 0$	$k_{23} = 2.8355$	$k_{24} = 3.7314$
	$k_{31} = 1.2125$	$k_{32} = 0.9753$	$k_{33} = 0$	$k_{34} = 1.2084$
	$k_{41} = 0.9096$	$k_{42} = 0.8069$	$k_{43} = 0.7324$	$k_{44} = 0$

2) Tracy (HMM)

FRET E	$E_1 = 0.23$	$E_2 = 0.76$	$E_3 = 0.9$
kinetic model	$k_{11} = 0$	$k_{12} = 0$	$k_{13} = 0.03$
	$k_{21} = 0.9$	$k_{22} = 0$	$k_{23} = 0$
	$k_{31} = 0$	$k_{32} = 0.029$	$k_{33} = 0$

3) FRETboard

FRET E	$E_1 = 0.229$	$E_2 = 0.385$	$E_3 = 0.648$	$E_4 = 0.842$
kinetic model	$k_{11} = 0$	$k_{12} = 0.1341376$	$k_{13} = 0.1197335$	$k_{14} = 0.1764494$
	$k_{21} = 0.5357143$	$k_{22} = 0$	$k_{23} = 0.1897321$	$k_{24} = 0.1636905$
	$k_{31} = 0.2962113$	$k_{32} = 0.0849598$	$k_{33} = 0$	$k_{34} = 0.2870264$
	$k_{41} = 0.1867587$	$k_{42} = 0.0371747$	$k_{43} = 0.1079837$	$k_{44} = 0$

4) Hidden-Markury

FRET E	$E_1 = 0.222$	$E_2 = 0.802$
kinetic model	$k_{11} = 0$	$k_{12} = 0.383$
	$k_{21} = 0.36$	$k_{22} = 0$

5) SMACKS(SS)

FRET E	$E_1 = 0.25$	$E_2 = 0.25$	$E_3 = 0.76$	$E_4 = 0.76$
kinetic model	$k_{11} = 0$	$k_{12} = 0.0772712$	$k_{13} = 0$	$k_{14} = 0.724288$
	$k_{21} = 0.0799077$	$k_{22} = 0$	$k_{23} = 0$	$k_{24} = 0$
	$k_{31} = 0$	$k_{32} = 0$	$k_{33} = 0$	$k_{34} = 0.0818057$
	$k_{41} = 0.605016$	$k_{42} = 0$	$k_{43} = 0.0710662$	$k_{44} = 0$

6) SMACKS

FRET E	$E_1 = 0.24$	$E_2 = 0.24$	$E_3 = 0.76$	$E_4 = 0.76$
kinetic model	$k_{11} = 0$	$k_{12} = 0$	$k_{13} = 0.068$	$k_{14} = 0$
	$k_{21} = 0.007$	$k_{22} = 0$	$k_{23} = 0.622$	$k_{24} = 0.164$
	$k_{31} = 0.107$	$k_{32} = 0.749$	$k_{33} = 0$	$k_{34} = 0$
	$k_{41} = 0$	$k_{42} = 0.077$	$k_{43} = 0.017$	$k_{44} = 0$

7) Correlation

FRET E	$E_1 = 0.22$	$E_2 = 0.76$	$E_3 = 0.9$
kinetic model	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.

8) Edge finding (CK)

n.a.

9) Edge finding (k-means)

n.a.

10) Step finding (2 FRET states)

FRET E	$E_1 = 0.221$	$E_2 = 0.802$
kinetic model	$k_{11} = 0$	$k_{12} = 0.522$
	$k_{21} = 0.669$	$k_{22} = 0$

10b) Step finding (3 FRET states)

FRET E	$E_1 = 0.217$	$E_2 = 0.618$	$E_3 = 0.851$
kinetic model	$k_{11} = 0$	$k_{12} = 0.295$	$k_{13} = 0.153$
	$k_{21} = 0.57$	$k_{22} = 0$	$k_{23} = 0.256$
	$k_{31} = 0.238$	$k_{32} = 0.182$	$k_{33} = 0$

11) STaSI

FRET E	$E_1 = 0.17$	$E_2 = 0.25$	$E_3 = 0.38$	$E_4 = 0.54$	$E_5 = 0.68$	$E_6 = 0.76$	$E_7 = 0.85$	$E_8 = 0.92$
kinetic model	$k_{11} = 0$	$k_{12} = 0.17$	$k_{13} = 0.085$	$k_{14} = 0.132$	$k_{15} = 0.076$	$k_{16} = 0.054$	$k_{17} = 0.033$	$k_{18} = 0.033$
	$k_{21} = 0.078$	$k_{22} = 0$	$k_{23} = 0.08$	$k_{24} = 0.157$	$k_{25} = 0.094$	$k_{26} = 0.078$	$k_{27} = 0.09$	$k_{28} = 0.025$
	$k_{31} = 0.294$	$k_{32} = 0.631$	$k_{33} = 0$	$k_{34} = 0.381$	$k_{35} = 0.268$	$k_{36} = 0.251$	$k_{37} = 0.199$	$k_{38} = 0.078$
	$k_{41} = 0.356$	$k_{42} = 0.717$	$k_{43} = 0.169$	$k_{44} = 0$	$k_{45} = 0.327$	$k_{46} = 0.423$	$k_{47} = 0.305$	$k_{48} = 0.198$
	$k_{51} = 0.195$	$k_{52} = 0.33$	$k_{53} = 0.131$	$k_{54} = 0.207$	$k_{55} = 0$	$k_{56} = 0.178$	$k_{57} = 0.246$	$k_{58} = 0.119$
	$k_{61} = 0.061$	$k_{62} = 0.228$	$k_{63} = 0.102$	$k_{64} = 0.282$	$k_{65} = 0.126$	$k_{66} = 0$	$k_{67} = 0.105$	$k_{68} = 0.071$
	$k_{71} = 0.054$	$k_{72} = 0.124$	$k_{73} = 0.07$	$k_{74} = 0.172$	$k_{75} = 0.175$	$k_{76} = 0.073$	$k_{77} = 0$	$k_{78} = 0.059$
	$k_{81} = 0.051$	$k_{82} = 0.082$	$k_{83} = 0.048$	$k_{84} = 0.065$	$k_{85} = 0.106$	$k_{86} = 0.092$	$k_{87} = 0.065$	$k_{88} = 0$

12) MASH-FRET (bootstrap)

FRET E	$E_1 = 0.451$	$E_2 = 0.863$	$E_3 = 0.227$	$E_4 = 0.702$
kinetic model	$k_{11} = 0$	$k_{12} = 0.193$	$k_{13} = 0.843$	$k_{14} = 0.455$
	$k_{21} = 0.106$	$k_{22} = 0$	$k_{23} = 0.264$	$k_{24} = 0.273$
	$k_{31} = 0.176$	$k_{32} = 0.279$	$k_{33} = 0$	$k_{34} = 0.221$
	$k_{41} = 0.194$	$k_{42} = 0.466$	$k_{43} = 0.353$	$k_{44} = 0$

13) MASH-FRET (probabilistic)

FRET E	$E_1 = 0.251$	$E_2 = 0.251$	$E_3 = 0.743$	$E_4 = 0.743$
kinetic model	$k_{11} = 0$	$k_{12} = 0.015$	$k_{13} = 0.062$	$k_{14} = 0.677$
	$k_{21} = 0$	$k_{22} = 0$	$k_{23} = 0$	$k_{24} = 0.072$
	$k_{31} = 0.014$	$k_{32} = 0.003$	$k_{33} = 0$	$k_{34} = 0.028$
	$k_{41} = 0.569$	$k_{42} = 0.072$	$k_{43} = 0$	$k_{44} = 0$

14) postFRET (2 FRET states)

FRET E	$E_1 = 0.24$	$E_2 = 0.81$
kinetic model	$k_{11} = 0$	$k_{12} = 0.2070822$
	$k_{21} = 0.0711783$	$k_{22} = 0$

14b) postFRET (3 FRET states)

FRET E	$E_1 = 0.23$	$E_2 = 0.5$	$E_3 = 0.8$
kinetic model	$k_{11} = 0$	$k_{12} = 0.0009144$	$k_{13} = 0.0738384$
	$k_{21} = 0.0058854$	$k_{22} = 0$	$k_{23} = 0.0400443$
	$k_{31} = 0.0638665$	$k_{32} = 0.0002925$	$k_{33} = 0$

14c) postFRET (4 FRET states)

FRET E	$E_1 = 0.25$	$E_2 = 0.5$	$E_3 = 0.69$	$E_4 = 0.85$
kinetic model	$k_{11} = 0$	$k_{12} = 0.0011741$	$k_{13} = 0.08894954$	$k_{14} = 0.10552059$
	$k_{21} = 0.48463045$	$k_{22} = 0$	$k_{23} = 0.30084519$	$k_{24} = 0.01306089$
	$k_{31} = 0.03141345$	$k_{32} = 0.29437991$	$k_{33} = 0$	$k_{34} = 0.04156456$
	$k_{41} = 0.11838752$	$k_{42} = 0.00822046$	$k_{43} = 0.04257101$	$k_{44} = 0$

Supplementary Tables 3

Kinetic models for the data shown in Fig. 5d-f of the main text, inferred by the participating groups with the specified tools. Units of the rate constants are in s^{-1} . The full submission, including standard deviations of the FRET efficiencies and uncertainties of the rate constants, can be found in the Supplementary Datafiles.

1) Pomegranate

FRET E	$E_1 = 0.208$	$E_2 = 0.507$	$E_3 = 0.703$	$E_4 = 0.93$
kinetic model	$k_{11} = 0$	$k_{12} = 0.754$	$k_{13} = 0.916$	$k_{14} = 0.817$
	$k_{21} = 3.101$	$k_{22} = 0$	$k_{23} = 2.773$	$k_{24} = 5.424$
	$k_{31} = 1.144$	$k_{32} = 1.088$	$k_{33} = 0$	$k_{34} = 1.779$
	$k_{41} = 0.763$	$k_{42} = 1.133$	$k_{43} = 0.945$	$k_{44} = 0$

2) Tracy (HMM)

FRET E	$E_1 = 0.23$	$E_2 = 0.76$	$E_3 = 0.9$
kinetic model	$k_{11} = 0$	$k_{12} = 0.038$	$k_{13} = 0$
	$k_{21} = 0.042$	$k_{22} = 0$	$k_{23} = 0$
	$k_{31} = 0.23$	$k_{32} = 0.52$	$k_{33} = 0$

3) FRETboard

FRET E	$E_1 = 0.267$	$E_2 = 0.565$	$E_3 = 0.726$	$E_4 = 0.849$
kinetic model	$k_{11} = 0$	$k_{12} = 0.066317$	$k_{13} = 0.180713$	$k_{14} = 0.117159$
	$k_{21} = 0.754011$	$k_{22} = 0$	$k_{23} = 0.545455$	$k_{24} = 0.069519$
	$k_{31} = 0.448457$	$k_{32} = 0.121142$	$k_{33} = 0$	$k_{34} = 0.189866$
	$k_{41} = 0.183439$	$k_{42} = 0.049089$	$k_{43} = 0.258365$	$k_{44} = 0$

4) Hidden-Markury

FRET E	$E_1 = 0.243$	$E_2 = 0.795$
kinetic model	$k_{11} = 0$	$k_{12} = 0.523$
	$k_{21} = 0.492$	$k_{22} = 0$

5) SMACKS(SS) (2 FRET states)

FRET E	$E_1 = 0.26$	$E_2 = 0.26$	$E_3 = 0.77$	$E_4 = 0.77$
kinetic model	$k_{11} = 0$	$k_{12} = 0.0464469$	$k_{13} = 0$	$k_{14} = 0.75673$
	$k_{21} = 0.0777646$	$k_{22} = 0$	$k_{23} = 0$	$k_{24} = 0$
	$k_{31} = 0$	$k_{32} = 0$	$k_{33} = 0$	$k_{34} = 0.0767983$
	$k_{41} = 0.676816$	$k_{42} = 0$	$k_{43} = 0.0375133$	$k_{44} = 0$

5b) SMACKS(SS) (3 FRET states)

FRET E	$E_1 = 0.24$	$E_2 = 0.62$	$E_3 = 0.62$	$E_4 = 0.81$
kinetic model	$k_{11} = 0$	$k_{12} = 0.308986$	$k_{13} = 0.119066$	$k_{14} = 0.162909$
	$k_{21} = 1.77861$	$k_{22} = 0$	$k_{23} = 0$	$k_{24} = 1.61935$
	$k_{31} = 0.200454$	$k_{32} = 0$	$k_{33} = 0$	$k_{34} = 0.0708802$
	$k_{41} = 0.307538$	$k_{42} = 0.613002$	$k_{43} = 0.0852028$	$k_{44} = 0$

6) SMACKS

FRET E	$E_1 = 0.26$	$E_2 = 0.26$	$E_3 = 0.73$	$E_4 = 0.73$
kinetic model	$k_{11} = 0$	$k_{12} = 0$	$k_{13} = 0.069$	$k_{14} = 0$
	$k_{21} = 0.006$	$k_{22} = 0$	$k_{23} = 0.668$	$k_{24} = 0.102$
	$k_{31} = 0.049$	$k_{32} = 0.813$	$k_{33} = 0$	$k_{34} = 0$
	$k_{41} = 0$	$k_{42} = 0.081$	$k_{43} = 0.041$	$k_{44} = 0$

7) Correlation

FRET E	$E_1 = 0.23$	$E_2 = 0.75$	$E_3 = 0.88$
kinetic model	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.

8) Edge finding (CK)

n.a.

9) Edge finding (k-means)

n.a.

10) Step finding (2 FRET states)

FRET E	$E_1 = 0.243$	$E_2 = 0.795$
kinetic model	$k_{11} = 0$	$k_{12} = 0.565$
	$k_{21} = 0.588$	$k_{22} = 0$

10b) Step finding (3 FRET states)

FRET E	$E_1 = 0.243$	$E_2 = 0.74$	$E_3 = 0.87$
kinetic model	$k_{11} = 0$	$k_{12} = 0.436$	$k_{13} = 0.121$
	$k_{21} = 0.6$	$k_{22} = 0$	$k_{23} = 0.067$
	$k_{31} = 0.292$	$k_{32} = 0.09$	$k_{33} = 0$

11) STaSI

FRET E	$E_1 = 0.25$	$E_2 = 0.25$	$E_3 = 0.8$	$E_4 = 0.8$
kinetic model	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.	n.a.

12) MASH-FRET (bootstrap)

FRET E	$E_1 = 0.755$	$E_2 = 0.271$
--------	---------------	---------------

kinetic model	$k_{11} = 0$	$k_{12} = 0.53$
	$k_{21} = 0.66$	$k_{22} = 0$

13) MASH-FRET (probabilistic)

FRET E	$E_1 = 0.27$	$E_2 = 0.27$	$E_3 = 0.75$	$E_4 = 0.75$
kinetic model	$k_{11} = 0$	$k_{12} = 0$	$k_{13} = 0$	$k_{14} = 0.043$
	$k_{21} = 0.006$	$k_{22} = 0$	$k_{23} = 0.047$	$k_{24} = 0.728$
	$k_{31} = 0.003$	$k_{32} = 0.062$	$k_{33} = 0$	$k_{34} = 0$
	$k_{41} = 0.026$	$k_{42} = 0.691$	$k_{43} = 0$	$k_{44} = 0$

14) postFRET (2 FRET states)

FRET E	$E_1 = 0.26$	$E_2 = 0.8$
kinetic model	$k_{11} = 0$	$k_{12} = 0.338698$
	$k_{21} = 0.345414$	$k_{22} = 0$

14b) postFRET (3 FRET states)

FRET E	$E_1 = 0.25$	$E_2 = 0.65$	$E_3 = 0.85$
kinetic model	$k_{11} = 0$	$k_{12} = 0.00109842$	$k_{13} = 0.21834065$
	$k_{21} = 0.00301381$	$k_{22} = 0$	$k_{23} = 0.08283056$
	$k_{31} = 0.27579656$	$k_{32} = 0.00353179$	$k_{33} = 0$

14c) postFRET (4 FRET states)

FRET E	$E_1 = 0.25$	$E_2 = 0.49$	$E_3 = 0.69$	$E_4 = 0.85$
kinetic model	$k_{11} = 0$	$k_{12} = 0.0002070865$	$k_{13} = 0.1452989654$	$k_{14} = 0.2005696159$
	$k_{21} = 0.2068609902$	$k_{22} = 0$	$k_{23} = 0.8709210325$	$k_{24} = 0.0475529494$
	$k_{31} = 0.4412775024$	$k_{32} = 0.0248333492$	$k_{33} = 0$	$k_{34} = 0.0361363519$
	$k_{41} = 0.2763561503$	$k_{42} = 0.0829865525$	$k_{43} = 0.006026129$	$k_{44} = 0$

Supplementary Tables 4

Kinetic models for the data shown in Fig. 5g-i of the main text, inferred by the participating groups with the specified tools. Units of the rate constants are in s^{-1} . The full submission, including standard deviations of the FRET efficiencies and uncertainties of the rate constants, can be found in the Supplementary Datafiles.

1) Pomegranate

FRET E	$E_1 = 0.239$	$E_2 = 0.479$	$E_3 = 0.706$	$E_4 = 0.897$
rate model	$k_{11} = 0$	$k_{12} = 0.504$	$k_{13} = 0.588$	$k_{14} = 0.755$
	$k_{21} = 2.171$	$k_{22} = 0$	$k_{23} = 2.944$	$k_{24} = 4.19$
	$k_{31} = 0.946$	$k_{32} = 0.612$	$k_{33} = 0$	$k_{34} = 1.147$
	$k_{41} = 0.505$	$k_{42} = 0.597$	$k_{43} = 0.684$	$k_{44} = 0$

2) Tracy (HMM)

FRET E	$E_1 = 0.23$	$E_2 = 0.76$	$E_3 = 0.9$
rate model	$k_{11} = 0$	$k_{12} = 0.019$	$k_{13} = 0.017$
	$k_{21} = 0.048$	$k_{22} = 0$	$k_{23} = 0.027$
	$k_{31} = 0.011$	$k_{32} = 0.007$	$k_{33} = 0$

3) FRETboard

FRET E	$E_1 = 0.257$	$E_2 = 0.691$	$E_3 = 0.806$	$E_4 = 0.909$
rate model	$k_{11} = 0$	$k_{12} = 0.106142$	$k_{13} = 0.1358$	$k_{14} = 0.045267$
	$k_{21} = 0.173973$	$k_{22} = 0$	$k_{23} = 0.280822$	$k_{24} = 0.078082$
	$k_{31} = 0.243874$	$k_{32} = 0.231039$	$k_{33} = 0$	$k_{34} = 0.038506$
	$k_{41} = 0.063406$	$k_{42} = 0.083031$	$k_{43} = 0.076993$	$k_{44} = 0$

4) Hidden-Markury

FRET E	$E_1 = 0.237$	$E_2 = 0.815$
rate model	$k_{11} = 0$	$k_{12} = 0.376$
	$k_{21} = 0.256$	$k_{22} = 0$

5) SMACKS(SS)

FRET E	$E_1 = 0.27$	$E_2 = 0.27$	$E_3 = 0.79$	$E_4 = 0.79$
rate model	$k_{11} = 0$	$k_{12} = 0.0181472$	$k_{13} = 0$	$k_{14} = 0.593785$
	$k_{21} = 0.0505473$	$k_{22} = 0$	$k_{23} = 0$	$k_{24} = 0$
	$k_{31} = 0$	$k_{32} = 0$	$k_{33} = 0$	$k_{34} = 0.270328$
	$k_{41} = 0.578138$	$k_{42} = 0$	$k_{43} = 0.431789$	$k_{44} = 0$

6) SMACKS

FRET E	$E_1 = 0.27$	$E_2 = 0.27$	$E_3 = 0.78$	$E_4 = 0.78$
rate model	$k_{11} = 0$	$k_{12} = 0$	$k_{13} = 0.032$	$k_{14} = 0$
	$k_{21} = 0$	$k_{22} = 0$	$k_{23} = 0.442$	$k_{24} = 0.228$
	$k_{31} = 0.108$	$k_{32} = 0.926$	$k_{33} = 0$	$k_{34} = 0.393$
	$k_{41} = 0$	$k_{42} = 0.141$	$k_{43} = 0$	$k_{44} = 0$

7) Correlation

FRET E	$E_1 = 0.23$	$E_2 = 0.75$	$E_3 = 0.87$
rate model	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.
	n.a.	n.a.	n.a.

8) Edge finding (CK)

n.a.

9) Edge finding (k-means)

n.a.

10) Step finding (2 FRET states)

FRET E	$E_1 = 0.237$	$E_2 = 0.815$
rate model	$k_{11} = 0$	$k_{12} = 0.443$
	$k_{21} = 0.444$	$k_{22} = 0$

10b) Step finding (3 FRET states)

FRET E	$E_1 = 0.234$	$E_2 = 0.722$	$E_3 = 0.862$
rate model	$k_{11} = 0$	$k_{12} = 0.318$	$k_{13} = 0.106$
	$k_{21} = 0.374$	$k_{22} = 0$	$k_{23} = 0.107$
	$k_{31} = 0.126$	$k_{32} = 0.118$	$k_{33} = 0$

11) STaSI

FRET E	$E_1 = 0.14$	$E_2 = 0.26$	$E_3 = 0.54$	$E_4 = 0.72$	$E_5 = 0.85$	$E_6 = 0.95$
rate model	$k_{11} = 0$	$k_{12} = 0.179$	$k_{13} = 0.129$	$k_{14} = 0.082$	$k_{15} = 0.014$	$k_{16} = 0.014$
	$k_{21} = 0.039$	$k_{22} = 0$	$k_{23} = 0.188$	$k_{24} = 0.198$	$k_{25} = 0.089$	$k_{26} = 0.025$
	$k_{31} = 0.209$	$k_{32} = 0.921$	$k_{33} = 0$	$k_{34} = 0.585$	$k_{35} = 0.381$	$k_{36} = 0.095$
	$k_{41} = 0.028$	$k_{42} = 0.298$	$k_{43} = 0.174$	$k_{44} = 0$	$k_{45} = 0.241$	$k_{46} = 0.071$
	$k_{51} = 0.007$	$k_{52} = 0.133$	$k_{53} = 0.1$	$k_{54} = 0.186$	$k_{55} = 0$	$k_{56} = 0.028$
	$k_{61} = 0.01$	$k_{62} = 0.068$	$k_{63} = 0.06$	$k_{64} = 0.123$	$k_{65} = 0.053$	$k_{66} = 0$

12) MASH-FRET (bootstrap)

FRET E	$E_1 = 0.267$	$E_2 = 0.847$	$E_3 = 0.668$	$E_4 = 0.668$	$E_5 = 0.668$	$E_6 = 0.668$
rate model	$k_{11} = 0$	$k_{12} = 0.224$	$k_{13} = 0.007$	$k_{14} = 0.012$	$k_{15} = 0.084$	$k_{16} = 0.145$
	$k_{21} = 0.166$	$k_{22} = 0$	$k_{23} = 0.006$	$k_{24} = 0.01$	$k_{25} = 0.071$	$k_{26} = 0.123$
	$k_{31} = 1.749$	$k_{32} = 1.325$	$k_{33} = 0$	$k_{34} = 0$	$k_{35} = 0$	$k_{36} = 0$
	$k_{41} = 1.749$	$k_{42} = 0.271$	$k_{43} = 0$	$k_{44} = 0$	$k_{45} = 0$	$k_{46} = 0$
	$k_{51} = 0.221$	$k_{52} = 1.325$	$k_{53} = 0$	$k_{54} = 0$	$k_{55} = 0$	$k_{56} = 0$
	$k_{61} = 0.221$	$k_{62} = 0.271$	$k_{63} = 0$	$k_{64} = 0$	$k_{65} = 0$	$k_{66} = 0$

13) MASH-FRET (probabilistic)

FRET E	$E_1 = 0.298$	$E_2 = 0.298$	$E_3 = 0.777$	$E_4 = 0.777$
rate model	$k_{11} = 0$	$k_{12} = 0$	$k_{13} = 0.017$	$k_{14} = 0.53$
	$k_{21} = 0$	$k_{22} = 0$	$k_{23} = 0$	$k_{24} = 0.009$
	$k_{31} = 0.013$	$k_{32} = 0$	$k_{33} = 0$	$k_{34} = 0$
	$k_{41} = 0.321$	$k_{42} = 0.003$	$k_{43} = 0$	$k_{44} = 0$

14) postFRET (2 FRET states)

FRET E	$E_1 = 0.27$	$E_2 = 0.83$
rate model	$k_{11} = 0$	$k_{12} = 0.0304298$
	$k_{21} = 0.0217805$	$k_{22} = 0$

14b) postFRET (3 FRET states)

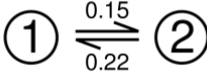
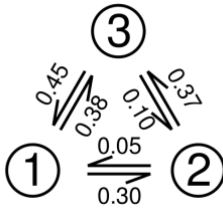
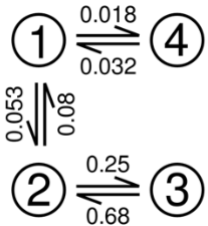
FRET E	$E_1 = 0.25$	$E_2 = 0.59$	$E_3 = 0.83$
rate model	$k_{11} = 0$	$k_{12} = 0.00032411$	$k_{13} = 0.02923996$
	$k_{21} = 0.05430609$	$k_{22} = 0$	$k_{23} = 0.07807773$
	$k_{31} = 0.00175319$	$k_{32} = 0.04096638$	$k_{33} = 0$

14c) postFRET (4 FRET states)

FRET E	$E_1 = 0.25$	$E_2 = 0.59$	$E_3 = 0.78$	$E_4 = 0.91$
rate model	$k_{11} = 0$	$k_{12} = 0.0537498259$	$k_{13} = 0.0468467928$	$k_{14} = 0.0352511436$
	$k_{21} = 0.2059474513$	$k_{22} = 0$	$k_{23} = 0.5658220374$	$k_{24} = 0.0844386522$
	$k_{31} = 0.1690830952$	$k_{32} = 0.4614106448$	$k_{33} = 0$	$k_{34} = 0.247915522$
	$k_{41} = 0.0291453273$	$k_{42} = 0.0784158076$	$k_{43} = 0.0674403618$	$k_{44} = 0$

Supplementary Table 5

Parameters for the simulation of smFRET traces.

	Fig. 2	Fig. 3	Fig. 4
kinetic model (rate constants next to arrows are in s ⁻¹)			
sampling rate (s ⁻¹)	5	10	5
bleach rate (s ⁻¹)	0.007	0.02	0.025
min. trace length (s)	50	10	8
max. trace length (s)	400	200	200
kinetic heterogeneity?	no	no	yes
assignment state -> FRET level	① -> low FRET ② -> high FRET	① -> low FRET ② -> mid FRET ③ -> high FRET	①, ② -> low FRET ③, ④ -> high FRET
blinking included?	no	no	yes $k_{\text{bright}} = 7 \text{ s}^{-1}$ $k_{\text{dark}} = 0.007 \text{ s}^{-1}$
per trace emission variability? ^[a]	no	yes	yes
per trace excitation variability? ^[a]	no	yes	yes
SNR (estimate) ^[b]	4	3	4
number of traces	75	150	250

^[a] The exact parameters can be found in the configuration files for the simulation in the Supplementary Datafiles.

^[b] The SNR estimate is based on the separation and width of the peaks in the FRET efficiency histogram. Peaks were fitted with Gaussian distributions and the two peaks with minimal separation were considered. The SNR was then calculated by $|\mu_1 - \mu_2| / \sqrt{\sigma_1^2 + \sigma_2^2}$, where μ and σ are the mean and standard deviation of the Gaussian functions, respectively. Exact parameters for the state-specific fluorescence intensities can be found in the configuration files for the simulation in the Supplementary Datafiles.

5 Supplementary References

1. Schmid, S., Götz, M. & Hugel, T. Single-Molecule Analysis beyond Dwell Times: Demonstration and Assessment in and out of Equilibrium. *Biophys. J.* **111**, 1375–1384 (2016).
2. Greenfeld, M. *et al.* Single-molecule dataset (SMD): a generalized storage format for raw and processed single-molecule data. *BMC Bioinformatics* **16**, 3 (2015).
3. Lerner, E. *et al.* FRET-based dynamic structural biology: Challenges, perspectives and an appeal for open-science practices. *eLife* **10**, e60416 (2021).
4. Thomsen, J. *et al.* DeepFRET, a software for rapid and automated single-molecule FRET data classification using deep learning. *eLife* **9**, e60404 (2020).
5. Manz, C. *et al.* Single-molecule FRET reveals the energy landscape of the full-length SAM-I riboswitch. *Nat. Chem. Biol.* **13**, 1172–1178 (2017).
6. Stella, S. *et al.* Conformational Activation Promotes CRISPR-Cas12a Catalysis and Resetting of the Endonuclease Activity. *Cell* **175**, 1856-1871.e21 (2018).
7. Bohr, S. S.-R. *et al.* Direct observation of *Thermomyces lanuginosus* lipase diffusional states by Single Particle Tracking and their remodeling by mutations and inhibition. *Sci. Rep.* **9**, 16169 (2019).
8. Heiss, G. Single-molecule microscopy study of nano-systems. (Ludwig-Maximilians-Universität München, 2011).
9. Bilmes, J. A. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *Int. Comput. Sci. Inst.* **4**, 126 (1998).
10. Schmid, S. Single Protein Dynamics at Steady State Quantified from FRET Time Traces. (Technische Universität München, 2017).
11. Opanasyuk, O. *et al.* Unraveling multi-state molecular dynamics in single-molecule FRET experiments -- Part II: Quantitative analysis of multi-state kinetic networks. *J. Chem Phys*, **157**, 031501 (2022).
12. Margittai, M. *et al.* Single-molecule fluorescence resonance energy transfer reveals a dynamic equilibrium between closed and open conformations of syntaxin 1. *Proc. Natl. Acad. Sci.* **100**, 15516–15521 (2003).
13. Torres, T. & Levitus, M. Measuring Conformational Dynamics: A New FCS-FRET Approach. *J. Phys. Chem. B* **111**, 7392–7400 (2007).
14. Felekyan, S., Sanabria, H., Kalinin, S., Kühnemuth, R. & Seidel, C. A. M. Chapter Two - Analyzing Förster Resonance Energy Transfer with Fluctuation Algorithms. in *Methods in Enzymology* (ed. Tetin, S. Y.) vol. 519 39–85 (Academic Press, 2013).
15. Felekyan, S., Kalinin, S., Sanabria, H., Valeri, A. & Seidel, C. A. M. Filtered FCS: Species Auto- and Cross-Correlation Functions Highlight Binding and Dynamics in Biomolecules. *ChemPhysChem* **13**, 1036–1053 (2012).
16. Kapusta, P., Wahl, M., Benda, A., Hof, M. & Enderlein, J. Fluorescence lifetime correlation spectroscopy. *J. Fluoresc.* **17**, 43–48 (2007).
17. Laurence, T. A. *et al.* Correlation Spectroscopy of Minor Fluorescent Species: Signal Purification and Distribution Analysis. *Biophys. J.* **92**, 2184–2198 (2007).
18. Kudryavtsev, V. *et al.* Monitoring dynamic systems with multiparameter fluorescence imaging. *Anal. Bioanal. Chem.* **387**, 71–82 (2007).
19. Felekyan, S. *et al.* Full correlation from picoseconds to seconds by time-resolved and time-correlated single photon detection. *Rev. Sci. Instrum.* **76**, 083104 (2005).
20. Aggarwal, T., Materassi, D., Davison, R., Hays, T. & Salapaka, M. Detection of Steps in Single Molecule Data. *Cell. Mol. Bioeng.* **5**, 14–31 (2012).
21. Nelder, J. A. & Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **7**, 308–313 (1965).
22. Schwarz, G. Estimating the Dimension of a Model. *Ann. Stat.* **6**, 461–464 (1978).
23. Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).
24. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of State

- Calculations by Fast Computing Machines. *J. Chem. Phys.* **21**, 1087–1092 (1953).
25. Cordes, T., Vogelsang, J. & Tinnefeld, P. On the Mechanism of Trolox as Antiflickering and Antibleaching Reagent. *J. Am. Chem. Soc.* **131**, 5018–5019 (2009).
 26. Chung, H. S., Louis, J. M. & Eaton, W. A. Distinguishing between Protein Dynamics and Dye Photophysics in Single-Molecule FRET Experiments. *Biophys. J.* **98**, 696–706 (2010).
 27. Kapanidis, A. N. *et al.* Alternating-Laser Excitation of Single Molecules. *Acc. Chem. Res.* **38**, 523–533 (2005).
 28. Müller, B. K., Zaychikov, E., Bräuchle, C. & Lamb, D. C. Pulsed Interleaved Excitation. *Biophys. J.* **89**, 3508–3522 (2005).
 29. Chung, S. H. & Kennedy, R. A. Forward-backward non-linear filtering technique for extracting small biological signals from noise. *J. Neurosci. Methods* **40**, 71–86 (1991).
 30. Haran, G. Noise reduction in single-molecule fluorescence trajectories of folding proteins. *Chem. Phys.* **307**, 137–145 (2004).
 31. Gauer, J. W. *et al.* Chapter Ten - Single-Molecule FRET to Measure Conformational Dynamics of DNA Mismatch Repair Proteins. in *Methods in Enzymology* (eds. Spies, M. & Chemla, Y. R.) vol. 581 285–315 (Academic Press, 2016).
 32. Sass, L. E., Lanyi, C., Weninger, K. & Erie, D. A. Single-Molecule FRET TACKLE Reveals Highly Dynamic Mismatched DNA–MutS Complexes. *Biochemistry* **49**, 3174–3190 (2010).
 33. Kinz-Thompson, C. D., Bailey, N. A. & Gonzalez, R. L. Precisely and Accurately Inferring Single-Molecule Rate Constants. *Methods Enzymol.* **581**, 187–225 (2016).
 34. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**, 129–137 (1982).
 35. Forgey, E. Cluster analysis of multivariate data: Efficiency vs. interpretability of classification. *Biometrics* **21**, 768–769 (1965).
 36. Dubes, R. & Jain, A. K. Clustering techniques: the user's dilemma. *Pattern Recognit.* **8**, 247–260 (1976).
 37. Bock, H.-H. Clustering Methods: A History of k-Means Algorithms. in *Selected Contributions in Data Analysis and Classification* (eds. Brito, P., Cucumel, G., Bertrand, P. & de Carvalho, F.) 161–172 (Springer, 2007). doi:10.1007/978-3-540-73560-1_15.
 38. Chiang, M. M.-T. & Mirkin, B. Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *J. Classif.* **27**, 3–40 (2010).
 39. Steinley, D. & Brusco, M. J. Choosing the number of clusters in K-means clustering. *Psychol. Methods* **16**, 285–297 (2011).
 40. Kodinariya, T. M. & Makwana, P. R. Review on determining number of Cluster in K-Means Clustering. *Int. J.* **1**, 90–95 (2013).
 41. Ma, J. *et al.* Ensemble switching unveils a kinetic rheostat mechanism of the eukaryotic thiamine pyrophosphate riboswitch. *RNA* **27**, 771–790 (2021).
 42. Shuang, B. *et al.* Fast Step Transition and State Identification (STaSI) for Discrete Single-Molecule Data Analysis. *J. Phys. Chem. Lett.* **5**, 3157–3161 (2014).
 43. Benítez, J. J. *et al.* Probing Transient Copper Chaperone–Wilson Disease Protein Interactions at the Single-Molecule Level with Nanovesicle Trapping. *J. Am. Chem. Soc.* **130**, 2446–2447 (2008).
 44. Walder, R., Kastantin, M. & Schwartz, D. K. High throughput single molecule tracking for analysis of rare populations and events. *Analyst* **137**, 2987–2996 (2012).
 45. Chen, J., Pyle, J. R., Sy Piecco, K. W., Kolomeisky, A. B. & Landes, C. F. A Two-Step Method for smFRET Data Analysis. *J. Phys. Chem. B* **120**, 7128–7132 (2016).
 46. Hadzic, M. C. A. S., Börner, R., König, S. L. B., Kowerko, D. & Sigel, R. K. O. Reliable State Identification and State Transition Detection in Fluorescence Intensity-Based Single-Molecule Förster Resonance Energy-Transfer Data. *J. Phys. Chem. B* **122**, 6134–6147 (2018).
 47. Bronson, J. E., Fei, J., Hofman, J. M., Gonzalez, R. L. & Wiggins, C. H. Learning Rates and States from Biophysical Time Series: A Bayesian Approach to Model Selection and Single-Molecule FRET Data. *Biophys. J.* **97**, 3196–3205 (2009).
 48. König, S. L. B. *et al.* BOBA FRET: Bootstrap-Based Analysis of Single-Molecule FRET Data. *PLOS ONE* **8**, e84157 (2013).
 49. Bladt, M. & Nielsen, B. F. Estimation of Phase-Type Distributions. in *Matrix-Exponential*

- Distributions in Applied Probability* (eds. Bladt, M. & Nielsen, B. F.) 671–701 (Springer US, 2017). doi:10.1007/978-1-4939-7049-0_13.
50. Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**, 257–286 (1989).
 51. Schmid, S. & Hugel, T. Efficient use of single molecule time traces to resolve kinetic rates, models and uncertainties. *J. Chem. Phys.* **148**, 123312 (2018).
 52. Hadzic, MCAS., Sigel, RKO., & Börner, R. Single-molecule kinetic studies of nucleic acids by Förster resonance energy transfer. *Methods in Molecular Biology* **2439**, 173-190 (2022).

A.4 Paper 4: Elucidating the conformational changes of bacterial adhesin SdrG via single-molecule FRET and all-atom MD simulations

Elucidating the conformational changes of bacterial adhesin SdrG via single-molecule FRET and all-atom MD simulations

ABSTRACT

Mechanostable complexes are a key feature of bacterial virulence. Similarly, nanoscale domains formed by the staphylococcal adhesin SdrG promote biofilm formation and the development of antibiotic resistance. Therefore, a vivid understanding of the molecular mechanisms is crucial for the de novo design of potent antibiotic therapeutics. In this study, we used smFRET along with MD simulations to quantitatively describe the conformational states of SdrG in the absence and presence of Fg β . Strategic placement of the FRET pairs enabled us to unleash the inter- and intra-domain dynamics of SdrG in a sub-nanometer precision. Comparison of our smFRET data with the simulations suggests that the locking strand lack binding sites and possess intrinsic dynamics in a sub-millisecond timescale, results also suggest that the locking strand prefers a specific orientation and does not spontaneously switch between its open and closed states. Thus, our data further confirms the DLL binding mechanism of SdrG, and highlight the central role of the locking strand, whilst providing insights into its conformational dynamics of the SdrG:Fg β complex.

INTRODUCTION

Staphylococcus epidermis is the causative agent of most nosocomial infections (Otto, 2009). Accumulation of the adhesin protein SD-repeat protein G (SdrG) in nanoscale domains targets the N-terminus of the human fibrinogen β chain (Fg β) and forms a highly mechanostable complex, which results in the formation of biofilms and the eventual development of antibiotic resistance (Milles et al., 2018). Thus, an extensive understanding of the binding mechanisms of *S. epidermis* is crucial for the development of antibiotic therapy.

The ligand binding site of SdrG is located in subdomains N2 and N3, while the N-terminal N1 domain is reputed to be cleaved proteolytically *in vivo*. X-ray crystallographic experiments were performed to unravel the binding mechanisms of SdrG demonstrate the C terminal extension of the N3 domain 'locking strand' as a prominent player in the transition from an open to a peptide-bound closed state. Hence 'A dock, lock, and latch (DLL) binding mechanism has been proposed, elaborating how the target peptide initially first docks into an open binding trench between the SdrG subdomains N2 and N3, which is fixed by the locking strand, finally resulting in the connection of locking stand to the N2 domain through β -strand completion thus forming a latch and resulting in a closed conformation (Ponnuraj et al., 2003). However, despite the clear evidence of the lack of a latch in the open state, as the C-terminal amino acids could not be resolved in the crystal structure exact conformation of the locking strand remains obscure. Further investigations aided by Förster Resonance Energy Transfer (FRET) in ensemble studies illustrate proximity of the locking strand to the N3 domain in the absence of Fg β , and a conformational change after peptide addition, which hence proclaims the DLL mechanism (Bowden et al., 2008). Nevertheless, as the intramolecular distances were probed on the ensemble level with only two FRET pairs, elucidation of the conformational change remained qualitative.

Therefore, in this study combination of single-molecule FRET (smFRET) and all-atom molecular dynamics (MD) simulation was used to quantitatively describe the conformational states of SdrG in the absence and presence of Fg β . As the efficiency of the energy transfer between a single pair of donor and acceptor fluorophores is highly sensitive to their proximity (Ha et al., 1996), smFRET has been proven a powerful tool to measure intra- and intermolecular distances with sub-nanometer precision (Hellenkamp et al., 2018), and to investigate dynamic processes (Dimura et al., 2016, Lerner et al., 2018). Prudent placement of nine FRET pairs on SdrG allowed us to investigate the motion of the locking strand with respect to subdomains N2 and N3. Moreover, as illustrated in Figure 2; A we were able to unleash the inter- and intra-domain dynamics of SdrG.

As the experimental and simulated data in the presence of Fg β harmonized well with the intramolecular distances extracted from the SdrG: Fg β crystal structure, further investigations were done to evaluate the distances in the absence of the target peptide, yet with the full length of the ligand. A comparison of the data with the simulations suggests that locking strand lack binding sites and possess intrinsic dynamics in a sub-millisecond timescale. It is evident that the locking strand prefers a specific orientation and does not spontaneously switch between its open and closed state. Thus, in contrast to an earlier hypothesis (Bowden et al., 2008), our data reveal that the presence of Fg β is a prerequisite for the transition from the open to the closed SdrG conformation.

RESULTS

AFM investigation of the SdrG:Fg β complex

AFM (Atomic Force Microscopy) control experiments were performed by our collaboration partner Lukas Milles to investigate the potential influence of the fluorescent labels on the binding of SdrG to Fg β in wild-type (wt) and fluorescently labeled double-cysteine SdrG (Figure 1A). The single-molecule force spectroscopy measurements confirmed that rupture forces for SdrG: Fg β complexes are almost identical for wt and labeled SdrG (green and grey histograms in Figure 1B, respectively). These findings assure the binding affinity of the fluorescently labeled SdrG into Fg β with the same mechano-stability in the absence of the dyes. However, in our experiments, not all molecules are 100 % labeled with both donor and acceptor fluorophores, as labeling efficiencies for the single dyes were typically in the range of 30% – 60%. While SdrG molecules that carry no, a single, or both dyes could not be distinguished in the AFM experiment, the absence of additional populations in the rupture force histograms indicates a similar behavior for the different molecules and thus no obvious influence of the dyes. However, modulation of the binding affinity could not be measured by AFM.

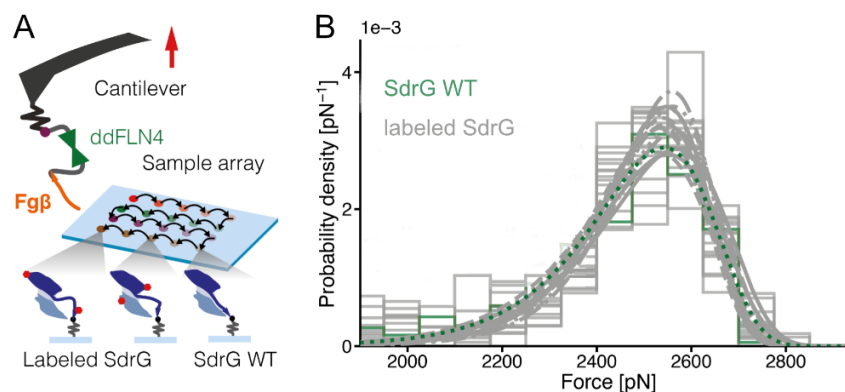


FIGURE 1: Fluorescently-labeled SdrG double-cysteine mutants form mechanostable complexes with the target peptide Fg β comparable to those with wt SdrG. A) A schematic of the experimental AFM setup including the ddFLN4 fingerprint domain (in green). Wt SdrG and fluorescently labeled double-cysteine SdrG mutants are covalently bound to the glass surface via PEG linkers and their ybbR-tag. Both SdrG and Fg β are force-loaded from their respective C-termini. The AFM cantilever is retracted at constant velocity until the SdrG:Fg β complex breaks. The glass surface is then moved to a new spot and the measurement repeated. B) Rupture forces for wt and labeled SdrG binding to Fg β (green and grey histograms, respectively). Shown are data acquired using SdrG double-cysteine mutants labeled with Alexa488 and Atto643, or with Atto532 and Atto643. No significant difference in rupture force and thus complex stability is detected for the various measurements.

smFRET investigation of the SdrG: Fg β complex

Thus upon the confirmation of the biological functionality of the double-labeled SdrG mutants, smFRET measurements were performed in order to evaluate the reproducibility of the closed conformation of the SdrG: Fg β crystal structure (Ponnuraj et al., 2003) in single molecule level. SmFRET experiments were performed in solution with the dye combination Atto532-Atto643 (Figure 2; A). To preserve the peptide bound closed state SdrG was measured in the presence of a high molar excess of Fg β (12.5 μ M, \sim 105 -fold molar excess), which is much higher than the dissociation constant (Kd) of \sim 400 nM (Ponnuraj et al., 2003). Resulting FRET efficiency histograms in Figure 2; B show a single population for most mutants, thus indicating a well-defined protein conformation in the presence of Fg β .

Intramolecular distances extracted from smFRET data were compared with the SdrG:Fg β crystal structure (PDB 1R17). Hence to obtain distances from the crystal structure, the sterically accessible volume (AV) of the FRET dyes was determined for each labeling position and the average FRET efficiency values were calculated (Kalinin et al., 2012). Distances from the experimental data were calculated using the photon distribution analysis (PDA) (Antonik et al., 2006). Many constructs demonstrated a single peak, therefore a dominant conformation.

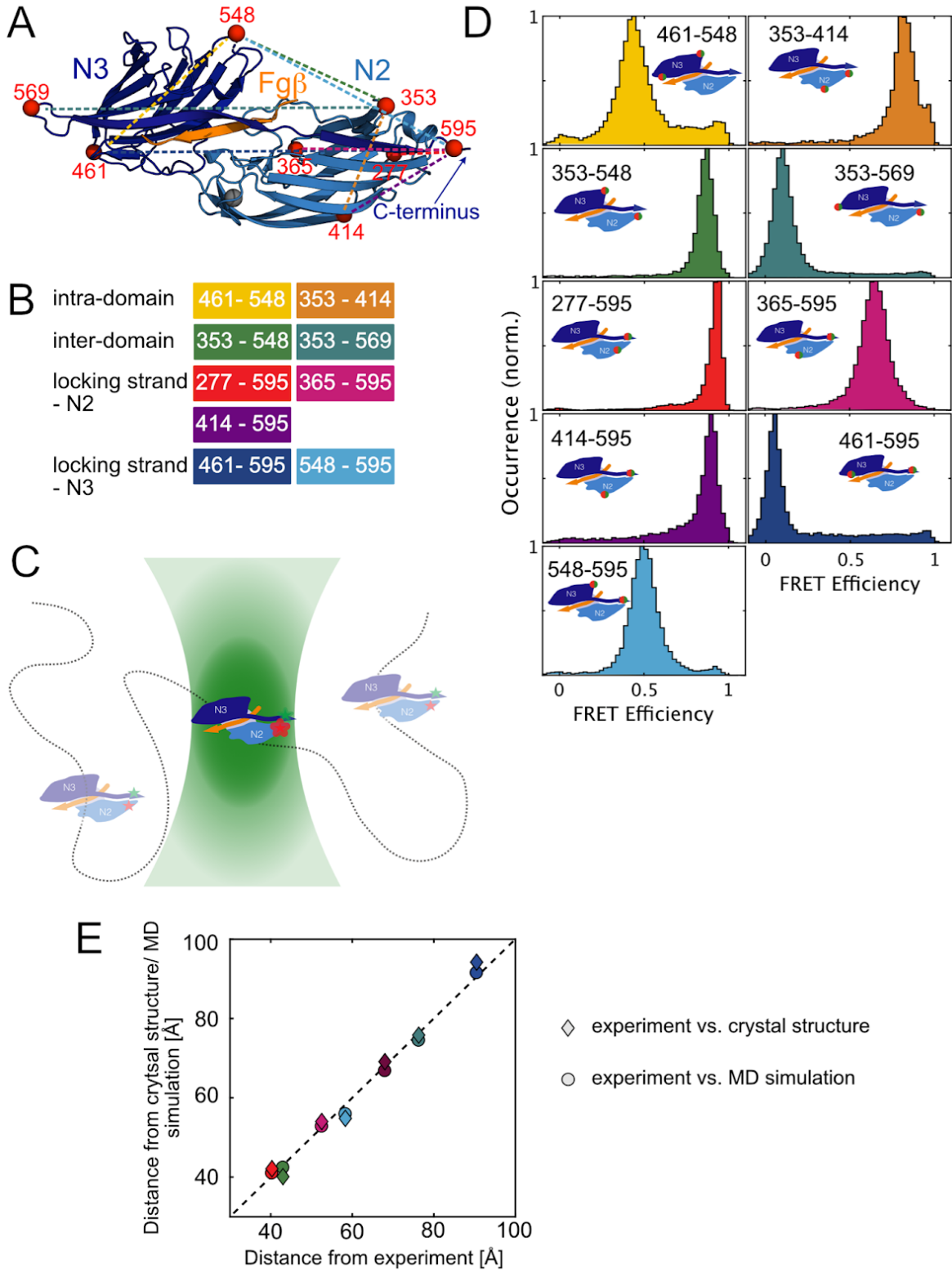


FIGURE 2: Conformation of the SdrG:Fg β complex studied by smFRET in solution. (A) Structure of Fg β (orange) bound to SdrG (blue). The C-terminal locking strand of SdrG locks the peptide in the binding pocket between subdomains N2 (dark blue) and N3 (light blue). Red spheres indicate Ca-atoms of the residues that were mutated to cysteines for stochastic labeling with Atto532 and Atto643. The labeling combinations investigated by smFRET are shown as dashed lines. (B) Overview of the labeling combinations to probe for intra- and inter-domain dynamics as well as for motions of the locking strand towards subdomains N2 and N3. (C) Diffusion of a double-labeled SdrG:Fg β complex through the confocal volume. The energy transfer efficiency between donor and acceptor dye is calculated for every single-molecule event. (D) Histograms of the molecule-wise smFRET efficiency for the investigated label combinations, shown as cartoons. The locations of the cysteine residues used for stochastic labeling are indicated by green-red-circles. (E) A comparison of the measured, simulated and theoretical distances for the different constructs extracted from the smFRET experiments, an all-atom MD simulation of the SdrG:Fg β complex and the SdrG:Fg β crystal structure (PDB 1R17), respectively. A Förster radius of 59 Å was used to convert experimental FRET efficiencies into distances. A linear correlation is observed between the experimental and theoretical values (indicated by circles) and between the experimental and simulated values (indicated by diamonds). The color code corresponds to the smFRET efficiency histograms in D.

Overall, the smFRET results of the peptide-bound conformation are in good agreement with the theoretical values with a root-mean-square deviation (RMSD) of 1.3 Å (Figure 2; E), which lies within the expected error range for smFRET experiments (Hellenkamp et al., 2018). As an additional control, we performed all-atom MD simulations of the SdrG:Fg β complex and extracted distances by AV calculations. The values from the simulation match with the experimentally determined distances with a RMSD of 2.4 Å (Figure 2; E) and are also in good agreement with the theoretical data from the crystal structure (RMSD of 1.9 Å). Thus, the distances derived from the closed conformation of the SdrG:Fg β crystal structure could be reproduced through our smFRET experiments and the all-atom MD simulation.

smFRET investigation of the apoprotein

Thus, upon the validation of the experimental approach, we next investigated the conformation of SdrG as an apoprotein. Until this stage, smFRET experiments were performed in the absence of Fg β . FRET efficiency histograms for the dye combination Atto532-Atto643 in Figure 3 depict a single main population for each construct (in color) and are compared to FRET efficiency histograms of SdrG in complex with Fg β (here shown in grey). The intra-domain sensors with both labels placed in either the N3 domain (residues 461-548, in yellow) or the N2 domain (residues 353-414, in orange) show

almost identical FRET efficiency histograms and thus similar intermolecular distances in the presence and absence of Fg β . Similarly, the inter-domain sensors with a single label positioned in the N2 and in the N3 domain (residues 353-548, in green, and residues 353-569, in teal) display only a minor change in FRET efficiency after peptide binding.

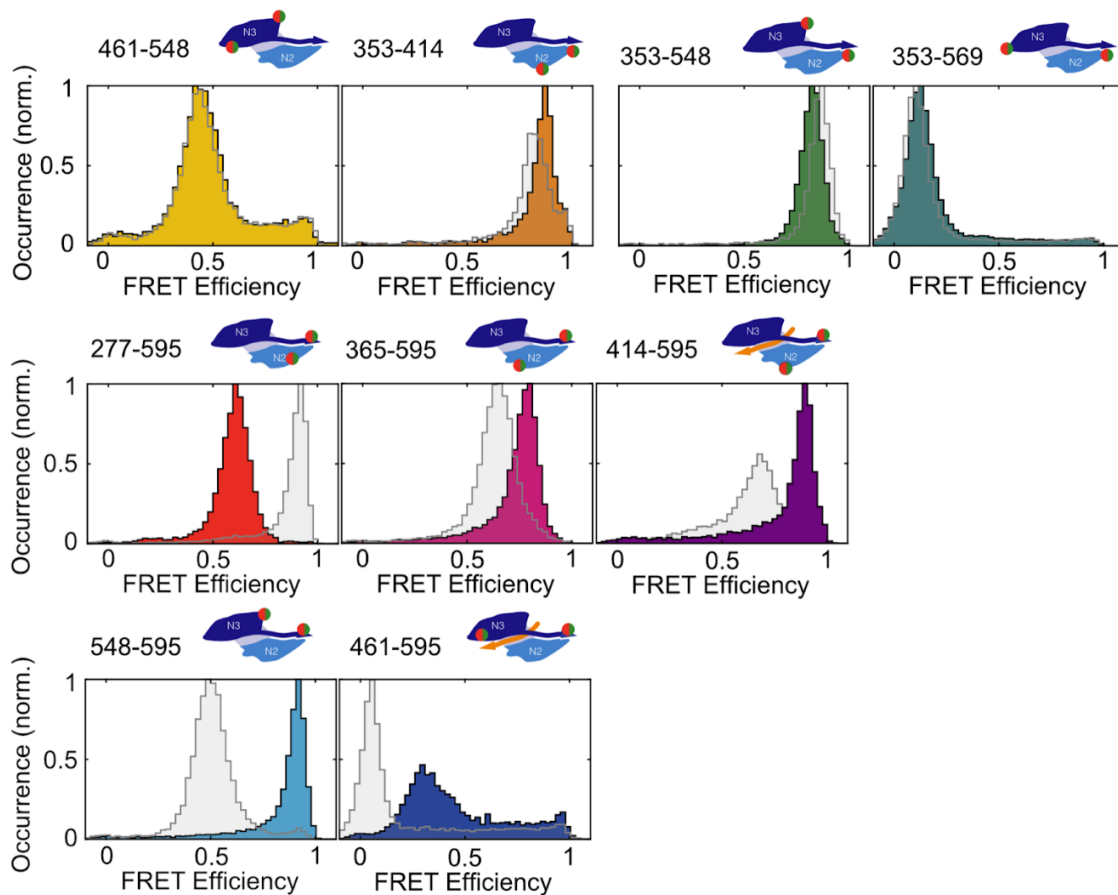


FIGURE 3: Conformation of the SdrG apoprotein studied by smFRET in solution. SmFRET efficiency histograms obtained in the absence of Fg β (in color) are compared to smFRET efficiency histograms of the SdrG:Fg β complex (in grey). Significant changes in FRET efficiency are detected for mutants with a label positioned on the locking strand at residue 595 and a label in the N2 or N3 domain (histograms in the second and third row in shades of red and blue, respectively).

In accordance with crystallographic data (Ponnuraj, 2003) and FRET experiments on the ensemble level (Bowden, 2008), our smFRET data confirm different conformational states of the locking strand in the presence and absence of Fg β , while the conformation of the two subdomains does not change significantly. Even though as suggested (Bowden,

2008), the mobility of the locking strand is a prerequisite for the DLL mechanism and defines the transition from an open to a closed state, we did not observe an equilibrium of the open and closed conformations for the SdrG apoprotein. However, dynamics on the sub-millisecond time scale indicate an open conformation. Histograms of FRET efficiency versus donor lifetime in supplementary figure 1 show a rightward shift from the static-FRET line for SdrG constructs with a label on the locking strand. As expected, no deviation from the static-FRET line is observed for the inter-domain sensor 353-569, as these label positions are insensitive to structural changes of the locking strand. The slight shift from the static-FRET line for the second inter-domain sensor 353-548 is most likely due to the close proximity of the FRET dyes in this construct. Our data thus suggest that the locking strand is flexible in the absence of Fg β , while adopting a conformation that clearly differs from its closed state in the SdrG:Fg β complex.

Proposed structural model for the open conformation of SdrG

To find a structural model for the open conformation, we compared FRET-derived distances to a MD simulation of the SdrG apoprotein. Although a crystal structure of SdrG in the absence of Fg β exists (PDB 1R19, (Ponnuraj et al., 2003)), we could not extract distances for comparison with our smFRET data, as almost all residues chosen as label positions are located in unresolved parts of the protein. However, having shown a good agreement of our all-atom MD simulation with the crystal structure of SdrG:Fg β , inspired the confidence to model the structure of the apoprotein based on our FRET data. To this end, we simulated an ensemble of structures in the open conformation. As a starting point, the crystal structure of SdrG:Fg β (PDB 1R17) was chosen, the peptide deleted and the C-terminus of the locking strand (residues 581-596) pulled away from its binding trench. For every frame of the simulation, intermolecular distances were extracted and compared to the FRET data. The structures that fit the experimental values best ($\chi_{red.}^2 < 1.5$), are presented in Figure 3; A. As the locking strand undergoes fast dynamic conformations in the absence of Fg β , these structures represent kinetically averaged positions. The locking strand, which is colored in red for better visibility, points away from the two domains. Its C-terminal part is unbound and shows a similar orientation for most selected structures. The absence of a specific binding site is supported by the fact that it has not been possible to resolve the C-terminal part of the locking strand in the crystal structure (Ponnuraj, 2003). The inset of Figure 3; B shows the previous model of the SdrG apoprotein with locking strand residues 581-596 missing. Thus, our model does not only confirm the extension of the locking strand into solution, but also indicates the preferred position of its C-terminus in the open conformation of SdrG and proves the absence of specific interactions.

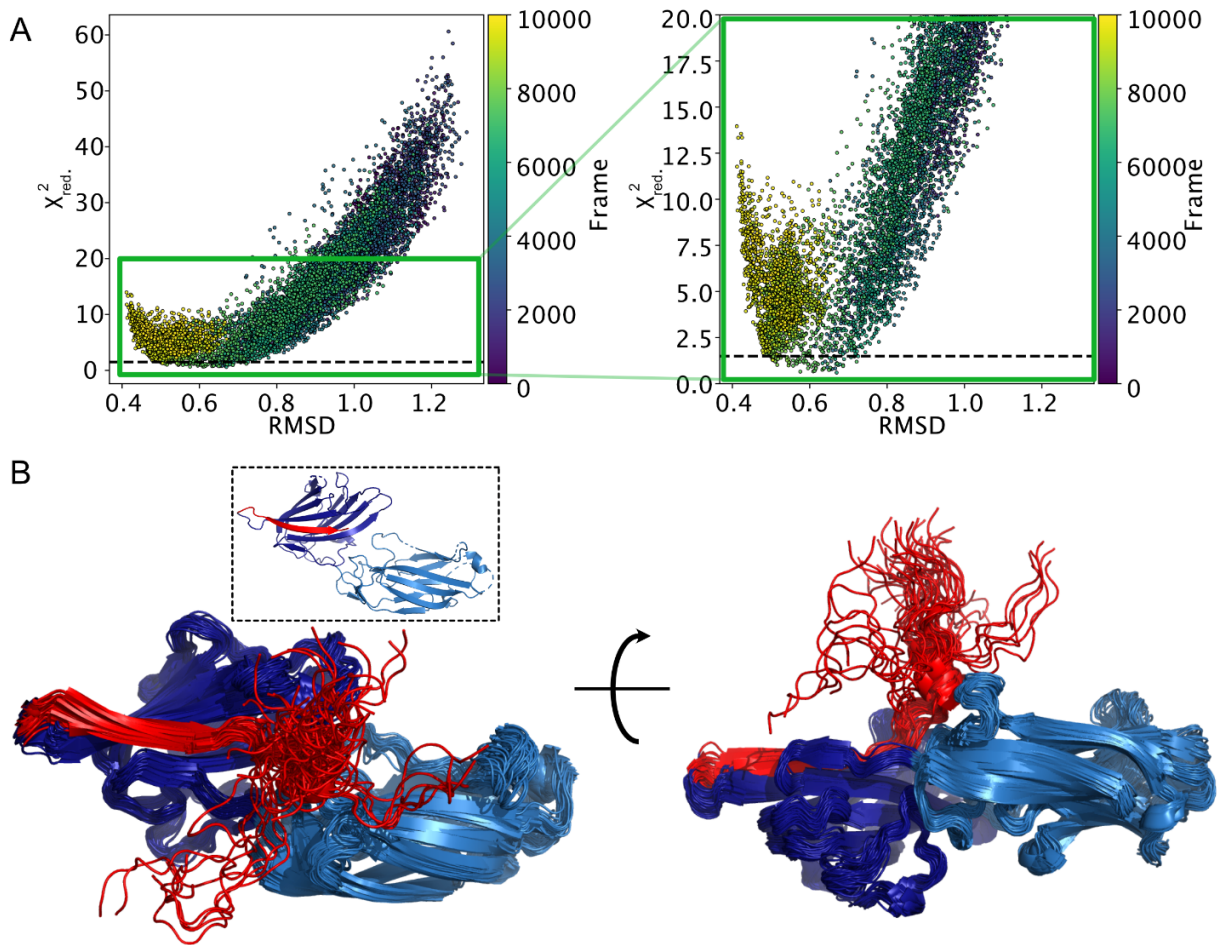


FIGURE 3: Structural model of the SdrG apoprotein based on smFRET data and all-atom MD simulations. A) Structural model resulting in $(x_{red}^2 < 1.5$ (indicated by the dashed black line) were selected to model the SdrG apoprotein. Plotted on the x-axis is the RMSD comparing the conformation of the SdrG apoprotein extracted from the MD simulations and the conformation of the SdrG:Fg β complex based on the crystal structure (PDB 1R17, Ponnuraj et al., 2003). B) Shown are two perspectives of 57 overlaid structures that fit the experimentally determined interdye distances ($(x_{red}^2 < 1.5$). The subdomains N2 and N3 are colored in light and dark blue, respectively, and the locking strand (residues 569-596) is shown in red. The inset shows the previous structural model based on X-ray crystallography, which could not resolve the C-terminal part of the locking strand (PDB 1R19, Ponnuraj et al., 2003).

Switching between an open and closed conformation of SdrG occurs on the seconds timescale in the presence of Fg β

While no interconversion between the open and closed conformations was indicated in the absence of Fg β , we observed a co-existence of two FRET efficiency populations at

low concentrations of Fg β (125 nM and 1.25 μ M). At higher peptide concentration, the equilibrium shifted towards the closed state for all investigated SdrG FRET sensors (Supplementary figure 2). The shifts are most distinct for constructs with a label positioned on the locking strand at residue 595 (histograms in shades of blue and red) with black and orange dashed lines highlighting the FRET efficiencies associated with the open and closed states, respectively.

No or little dynamic conversion of the two states was indicated on the time scale of diffusion, as exemplarily shown for SdrG constructs 548-595 and 277-595 (Figure 4; A and B). The FRET efficiency versus donor lifetime histograms in Figure 4; A clearly illustrate donor blinking, as the deviation from the static-FRET line follows a dynamic-FRET line connecting high- and no-FRET efficiencies (indicated by the blue dashed lines). Dynamic transitions between the FRET efficiency populations of the open and closed state would follow the red dashed dynamic-FRET lines, which does not apply here.

To investigate whether dynamic switching occurs on slower time scales, smTIRF experiments were performed with SdrG mutants 277-595 and 548-595. The immobilization to a biotinylated glass surface with streptavidin using the biotin-tag of SdrG allowed the observation of individual FRET sensors over a time range of seconds to minutes. The presence and photostability of both dyes was confirmed by ALEX (Kapanidis et al., 2005). The constructs were first measured in the absence of the peptide. As expected, a defined FRET efficiency population representing the open state was observed. Addition of Fg β to surface-immobilized SdrG resulted in the detection of a second FRET efficiency population, which we associated with the peptide-bound state. Increasing the concentration of Fg β resulted in an increase of the closed SdrG conformation. The respective framewise histograms are shown in Figure 4; C and D. Although the data quality and statistics for construct 277-595 were considerably better compared to experiments with construct 548-595, both SdrG mutants show similar trends. While most molecules were either detected in the open or the closed conformation in the presence of Fg β , we observed several molecules (~ 15 %) which dynamically switched between the two states. Exemplary fluorescence intensity time traces for constructs 548-595 and 277-595 acquired in the presence of 1 μ M Fg β are shown in Figure 4; E and F, respectively. In both cases, anti-correlated donor (GG) and FRET (GR) signals are observed, while the acceptor signal after acceptor excitation (RR) remains static. The FRET efficiency fluctuates on the timescale of seconds between two states associated with the open and closed conformation of SdrG, while no intermediate states are observed.

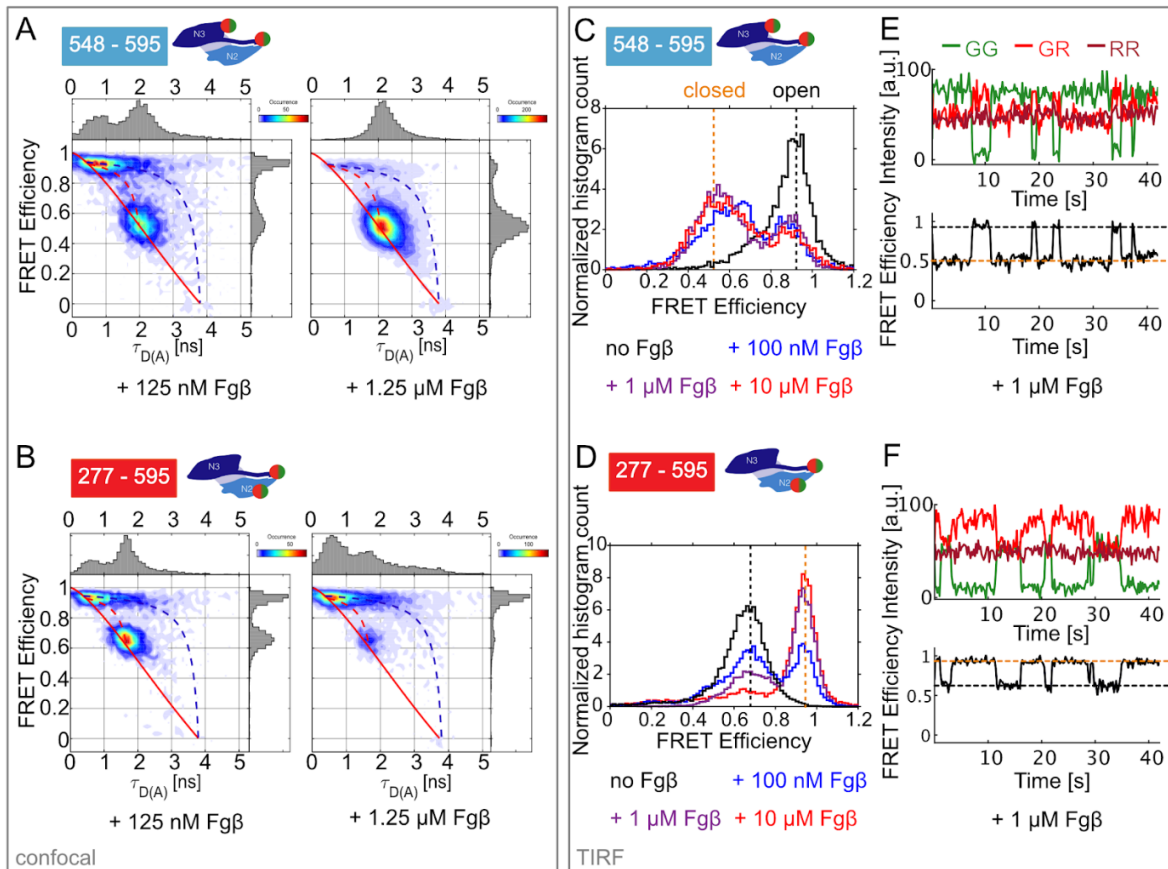


FIGURE 4: SdrG requires the presence of Fg β to switch between its open and closed conformation. The slow conformational dynamics are not resolvable for freely diffusing proteins on a confocal microscope (A-B) but can be observed for surface-immobilized constructs on a TIRF microscope (C-F). A-B) Two-dimensional histograms of the FRET efficiency versus the donor lifetime show a deviation from the static-FRET line (shown as a solid red line), which is caused by donor blinking (indicated by dashed blue lines). Conformational dynamics would follow the dashed red lines. C-D) Framewise histograms over all recorded FRET traces show an increase of molecules in the closed conformation (indicated by an orange dashed line) with increasing concentrations of Fg β . The numbers of analyzed molecules were 49, 46, 31 and 27 for SdrG construct 548-595 in the absence of Fg β and in the presence of 100 nM, 1 μ M and 10 μ M Fg β , respectively, and 87, 60, 73 and 63 for SdrG construct 277-595 in the absence of Fg β and in the presence of 100 nM, 1 μ M and 10 μ M Fg β , respectively. E-F) Exemplary fluorescence intensity time traces show the conformational switching between the open and closed conformation of SdrG in the presence of 1 μ M Fg β . The green signal after green excitation (GG) is shown in green, the red signal after green excitation (GR, i.e., FRET signal) in red, and the red signal after red excitation (RR) in dark red.

As the switching is much slower than the diffusion of the protein in solution (~ 3 ms), these dynamic transitions could not be resolved in confocal measurements.

DISCUSSION

In conclusion, the combination of smFRET experiments and all-atom MD simulations allowed us to propose a structural model for the dynamic SdrG apoprotein in the absence of its target peptide Fg β . We show that the previously unresolved C-terminus of the locking strand extends into solution and has no defined binding site. The absence of specific interactions suggests that the locking strand might act as a "fishing rod" to capture its target peptide, reeling it towards its binding site. Although intrinsically dynamic on the sub-millisecond timescale, the locking strand prefers a specific orientation and does not spontaneously switch between its open and closed states. By adding this missing piece to the structural information, we could not only confirm the DLL binding mechanism of SdrG (Ponnuraj, 2003) and highlight the central role of the locking strand but could also give insights into its conformational dynamics.

In contrast to an earlier hypothesis (Bowden et al., 2008), our data reveal that the presence of Fg β is a prerequisite for the transition from the open to the closed SdrG conformation, and a dynamic conversion was observed on the second timescale at sufficiently low peptide concentrations. As the locking strands only closes in the presence of Fg β , the binding cleft is kept consistently open.

Strategic placement of the FRET dyes into the subdomain N3 (at positions 461 and 548) and subdomain N2 (at positions 353 and 414, Figure 1; A) allowed us to gain insights into potential intra-domain dynamics.

In its native environment, bacterial SdrG must adhere tightly to human Fg β to withstand high mechanical forces and ensure a stable attachment of the microbe to the host tissue (Otto, 2009). The mechanical stress itself may significantly contribute to the formation of the tight bond. By combining smFRET with mechanical manipulation techniques such as AFM (He et al., 2012) - a method that has previously been used to characterize the extraordinarily mechanostability of the SdrG:Fg β complex (Milles et al., 2018) - or optical tweezers (Comstock et al., 2015), conformational changes can be investigated under force to get further insights into the critical process of bacterial adhesion. The combination with all atom-MD simulation allowed us to propose a structural model for the dynamic SdrG apoprotein including all parts of the flexible locking strand, providing information previously inaccessible to other methods. The simultaneous assessment of structure and dynamics thus rivals established structural biology techniques like X-ray crystallography, cryo-electron microscopy or NMR spectroscopy. As these methods require the fixation of biomolecules to characterize their three-dimensional structures, information on dynamics is not available. As shown in previous studies (Dimura et al., 2016), the combination of

quantitative smFRET and all-atom MD simulations is a valuable hybrid-approach to resolve the structure of dynamic biological systems and elucidate their function.

MATERIALS AND METHODS

Gene construction

The *Staphylococcus epidermidis* SdrG N2 N3 domain genes (UniProt: Q9KI13) had been synthesized codon-optimized for expression in *Escherichia coli* as linear DNA fragments (GeneArt – ThermoFisher Scientific, Regensburg, Germany) as described previously (Milles et al., 2018). These source plasmids for this work are available through Addgene (www.addgene.org): pET28a-SdrG_N2N3-HIS-ybbr, AddgeneID 101238 (Milles et al., 2018). Genes were inserted into pAC4 Vectors including a hexahistidine-, ybbr-tag, and AviTag via Gibson assembly 47 (New England Biolabs, MA, USA). All point mutations of amino acids to cysteines for dye attachment were created through polymerase chain reactions (Phusion Polymerase, New England Biolabs, MA, USA) with appropriate primers, followed by blunt-end ligation cloning using the T4 Ligase (Thermo Scientific, Massachusetts, USA) or Gibson assembly. The resulting open reading frames of all constructs were verified by DNA sequencing (Eurofins Genomics, Ebersberg, Germany).

Protein expression and purification

SdrG mutants were expressed in the competent *E. coli* strain CVB101 (Avidity Biotech) and purified on a NiNTA column (HisTrap FF 5mL on a Äkta Start system, both GE Healthcare, Massachusetts, USA) via their hexahistidine-tag. A detailed protocol has been previously described (Milles et al., 2018). Protein concentrations were determined by spectrophotometry at 280 nm with typical final concentrations of 20 - 100 μ M (NanoDrop 1000, Thermo Scientific, MA, USA).

Fluorescence labeling of double-cysteine SdrG mutants

A total of 9 double-cysteine mutants were generated and stochastically labeled with maleimide-functionalized dyes. The cysteine combinations were chosen to either probe the distance between domains N2 and N3 or to investigate the motion of the locking strand with respect to N2 and N3 (Figure 1B). As recombinant SdrG does not contain any cysteine residues, the introduced cysteines are the only available attachment points for the dyes. The protein was incubated with a 2-fold molar excess of the maleimide-functionalized dyes Atto532 and Atto643 (Atto-Tec, Siegen, Germany or Thermo Scientific, Massachusetts, USA) for three hours at room temperature in the presence of 1 mM TCEP. Excess dye was removed by ultrafiltration using molecular weight cut-off filters of 10 kDa (Merck Millipore, Burlington, Massachusetts, USA).

AFM single-molecule force spectroscopy

Glass surfaces and cantilevers were cleaned and coated with heterobifunctional α -Maleinimido-hexanoic-PEG-NHS (Rapp Polymere, Tübingen, Germany) and incubated with Coenzyme A to covalently pull down SdrG proteins via their ybbR-tag. AFM single-molecule force spectroscopy measurements were conducted at room temperature (approximately 25°C) and analyzed as described previously (Milles et al., 2018). Typically, 50.000-100.000 curves were recorded for each double labeled SdrG mutant.

smFRET measurements in solution

smFRET measurements of freely diffusing SdrG mutants were performed on custom-built confocal microscopes as described in (Nicoli et al., 2017, Kudryavtsev et al., 2012), which combine PIE (Pulsed Interleaved Excitation (Müller et al., 2005)) and MFD (Multiparameter Fluorescence Detection (Eggeling et al., 2001)). Excitation powers of 100 μ W were used for the green and red lasers when measuring Atto532/643-labeled proteins.

Measurements were performed in 8- or 4-well chamber slides (Nunc Lab-Tek, VWR), which were coated with Bovine Serum Albumin (BSA, New England Biolabs, Ipswich, Massachusetts, USA; 20 mg/mL diluted to 1 mg/mL in PBS). Labeled SdrG constructs were diluted to concentrations of 50-150 pM in imaging buffer (PBS and 1 mM Trolox to reduce photobleaching (Cordes et al., 2009)) and measured in the absence or presence of the unlabeled target peptide Fg β for up to 5 hours. The concentration of Fg β was varied between 125 nM and 12.5 μ M ($\sim 10^3$ -fold to 10^5 - fold molar excess).

Data were analyzed with the open-source software PAM (Pulsed Interleaved Excitation Analysis with MATLAB (Schimpf et al., 2018)). Bursts were selected using the all-photon burst search algorithm, which is based on a sliding time window approach (Nir et al., 2006). Parameters for the burst search were set to a minimum of 100 photons per burst, a time window of 500 μ s and 10 photons per time window. Incompletely labeled molecules were removed using typical thresholds < 10 for the ALEX-2CDE filter (Tomov et al., 2012). Correction factors for accurate FRET efficiencies were calculated according to standardized procedures (Hellenkamp et al., 2018). Correction factors were determined for every double-cysteine SdrG mutant individually to account for potential effects of the labeling positions on the dyes as well as for slight changes in setup alignment. Distances from the experimental data were calculated by photon distribution analysis (PDA) (Antonik et al., 2006).

smFRET measurements of surface-immobilized proteins

smFRET measurements of surface-immobilized SdrG constructs were performed on a custom-built objective-type TIRF microscope with lasers at 532 nm (Cobolt Samba, 100 mW) and 647 nm (Cobolt MLD, 120 mW), which were alternated on a time scale of 100 ms (+3 ms frame transfer). Fluorescence was collected by a 60x oil immersion objective (Apo TIRF 60x/1.49 Oil, Nikon) and separated by a 532/632 dichroic mirror (AHF Analysentechnik, Tübingen, Germany). Donor and acceptor emission were filtered using a HQ 580/75 and HQ 705/100 filter (AHF Analysentechnik, Tübingen, Germany), respectively.

The labeled SdrG constructs were measured in flow chambers, which were assembled as previously described (Bartnik et al., 2019). The flow chambers were first incubated with Streptavidin (Sigma-Aldrich, St. Louis, Missouri, USA; 0.2 mg/mL in PBS) for 15 min and washed with PBS. The protein samples were diluted to 100 pM in PBS and immobilized on the coverslip through biotin-streptavidin-biotin binding. Unbound proteins were washed away with PBS. For stabilization of the fluorophores, we flushed in a photo-cocktail containing PBS, 1 mM Trolox (Cordes et al., 2009) (UV activated for 5 min), 10% (v/v) glycerol, 1% (w/v) glucose and 10% (v/v) Glucoseoxidase-Catalase solution for oxygen removal (Stein, 2012). Varying concentrations of unlabeled target peptide Fg β (Stein et al., 2012).

Data were recorded with an alternating excitation sequence of green-red with typical laser powers of 15 mW for the green 532 nm laser and 8 mW for the red 647 nm laser at an exposure time of 100 ms for 1000 frames. TIRF data were analyzed with custom-written MATLAB programs (Mathworks, Massachusetts, USA). For analysis we exclusively chose double-labeled molecules showing fluorescence intensity after both donor and acceptor excitation and excluded acceptor- or donor-only species. Data were analyzed using the MATLAB and deep learning-based program *DeepLASI* (Wanninger et al., 2023).

TABLE 1: Correction factors α , β , γ and δ for the smFRET burst analysis of Atto532- and Atto643-labeled SdrG FRET sensors.

SdrG positions	label	α	β-factor	γ-factor	direct excitation δ
461-548		0.0224	0.8620	0.6436	0.0489
353-414		0.0216	0.9781	0.3818	0.0452
353-548		0.0202	0.9317	0.4533	0.0585
353-569		0.0212	1.1498	0.4982	0.0644
277-595		0.0222	0.8374	0.5350	0.0585
365-595		0.0225	0.9317	0.4405	0.0649

414-595	0.0193	1.1415	0.9265	0.0695
461-595	0.0186	0.906	0.6556	0.0687
548-595	0.0271	0.9196	0.6164	0.1530

All-atom molecular dynamics (MD) simulations

The structure of the *S. epidermidis* adhesin SdrG binding to fibrinogen β was solved by X-ray crystallography at 1.86 Å resolution and was obtained from the protein data bank (PDB: 1R17) (Ponnuraj et al., 2003). The apo state of the SdrG protein was constructed by first removing the fibrinogen β from the crystal structure. The locking strand was pulled away from the protein using OpenMM dynamics and its energy was locally minimized using the program ChimeraX (Goddard et al., 2018). Missing residues were added using DeepView - Swiss PdbViewer (Guex and Peitsch, 1997). The Gromacs 5.1.4 software package (Abraham et al., 2015) was used to simulate a total amount of approximately 120,000 atoms for both simulations. The a99SB-disp (Robustelli et al., 2018) force field with the TIP4P-D water model (Piana et al., 2015) was employed to describe the solvated protein.

Prior to the initialization of the MD simulations, an energy minimization procedure was conducted for 2000 steps using the steepest descent algorithm. The system was subsequently heated to 300 K and equilibrated for 1 ns. The MD simulations were performed assuming periodic boundary conditions in the NpT ensemble using Langevin dynamics for temperature and pressure coupling which were maintained at 300 K and 1 bar, respectively. A distance cut-off of 10.0 Å was applied to short-range non-bonded interactions using the Verlet neighbor search algorithm (Verlet, 1967). Long-range electrostatic interactions were treated using the particle-mesh Ewald (PME) method (Alston et al., 2021). For all MD simulations, Newton's equations of motion were integrated using the leap-frog algorithm at a time step of 2 fs. The holo state of the SdrG protein was simulated for 200 ns and the apo state for 1200 ns to obtain a broad distribution of strand locations.

AV calculations

In order to compare the experimental to theoretical distances extracted from the crystal structure of SdrG: Fg β (PDB: 1R17, Ponnuraj et al., 2003) or from the MD simulations, the accessible volumes for Atto532 and Atto643 attached to SdrG were calculated with the FRET positioning and screening software (Kalinin et al., 2012) using the C α -atoms of the respective cysteine residues as anchor points, the three radii AV3-model and the dye parameters summarized in Table 2. In the AV3-model, three spherical AVs with radii R1, R2, and R3 are calculated separately and the grid points in the final model are weighted according to the number of AVs. To compute FRET-averaged distances h_{RDAiE} and mean FRET efficiencies h_{Ei} , a Förster radius of 59 Å was used, which is the theoretical

value for the dye pair Atto532-Atto643 given by the manufacturer, calculated with the orientation factor $\kappa^2 = 2/3$ and a donor quantum yield of $\phi_D = 0.9$ (lifetime $\tau_D = 3.8$ ns). We chose this theoretical value, as donor lifetimes in our experiments were close to the theoretical value (we thus assume no influence on the photophysical properties of Atto532 in our system) and as fluorescence anisotropy measurements indicated no sticking of the dyes to SdrG.

Table 3: Dye parameters for accessible volume simulations with the AV3-model.

Dye	linker length [Å]	linker width [Å]	R1 [Å]	R2 [Å]	R3 [Å]
Atto532-maleimide	20.5	4.5	5.5	4.5	1.5
Atto643-maleimide	21	4.5	7.15	4.5	1.5

Structural modeling

To identify those structures in our all-atom MD simulations that are in good agreement with the measured smFRET data, the weighted data-model deviation x^2 (as defined in Kalinin et al., 2012) was calculated,

$$x^2 = \sum_{i=1}^N \frac{(R_{DA(i)} - R_{model(i)})^2}{\Delta R_{DA(i)}^2}$$

where N is the number of measured distances (i.e. the degrees of freedom), R_{DA} the distance calculated from the FRET measurement, R_{model} the distance obtained from the MD simulation and ΔR_{DA} the uncertainty of the measured data. ΔR_{DA} depends on systematic errors during the data analysis, the uncertainty of using the correct orientation factor κ^2 , as well as the uncertainty of using the correct dye model for AV calculations (Dimura et al., 2016). Here, we used $\Delta R_{DA} = 3$ Å for all measurements except for the high-FRET efficiency state measured for construct 548-595, for which we used $\Delta R_{DA} = 5$ Å.

The reduced x^2 value was calculated by dividing by N

$$x^2_{red.} = \frac{1}{N} x^2$$

REFERENCES

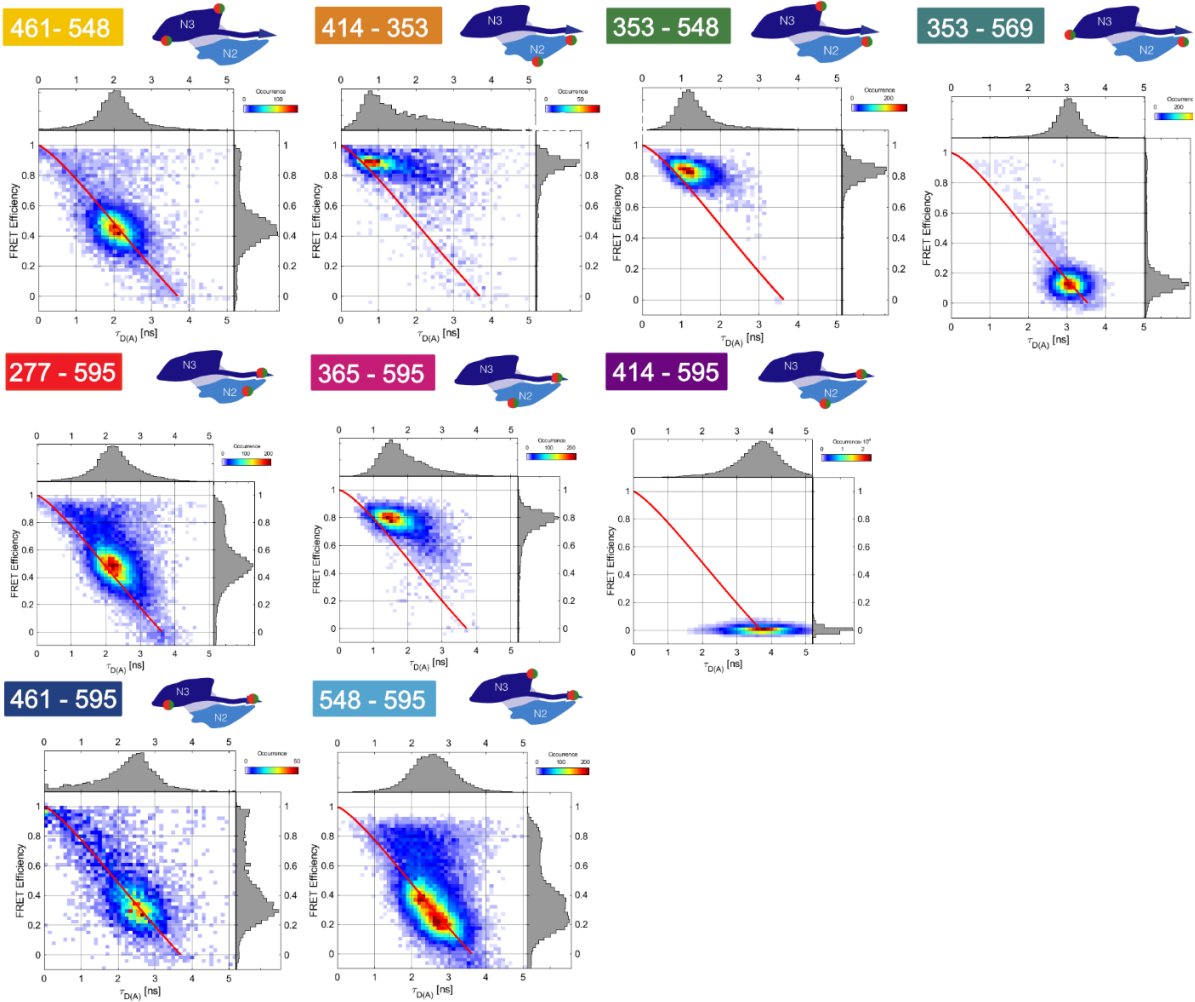
1. Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B. & Lindahl, E. 2015. Gromacs: High Performance Molecular Simulations Through Multi-Level Parallelism from Laptops to Supercomputers. *Software*, 1-2, 19-25.
2. Alston, J. J., Soranno, A. & Holehouse, A. S. 2021. Integrating Single-Molecule Spectroscopy and Simulations For The Study Of Intrinsically Disordered Proteins. *Methods*, 193, 116-135.
3. Antonik, M., Felekyan, S., Gaiduk, A. & Seidel, C. A. 2006. Separating Structural Heterogeneities from Stochastic Variations in Fluorescence Resonance Energy Transfer Distributions via Photon Distribution Analysis. *The Journal Of Physical Chemistry B*, 110, 6970-6978.
4. Bartnik, K., Barth, A., Pilo-Pais, M., Crevenna, A. H., Liedl, T. & Lamb, D. C. 2019. A Dna Origami Platform for Single-Pair Förster Resonance Energy Transfer Investigation of Dna–Dna Interactions and Ligation. *Journal Of The American Chemical Society*, 142, 815-825.
5. Bowden, M. G., Heuck, A. P., Ponnuraj, K., Kolosova, E., Choe, D., Gurusiddappa, S., Narayana, S. V., Johnson, A. E. & Hook, M. 2008. Evidence For The “Dock, Lock, And Latch” Ligand Binding Mechanism Of The Staphylococcal Microbial Surface Component Recognizing Adhesive Matrix Molecules (Mscramm) Sdrg. *Journal Of Biological Chemistry*, 283, 638-647.
6. Comstock, M. J., Whitley, K. D., Jia, H., Sokoloski, J., Lohman, T. M., Ha, T. & Chemla, Y. R. 2015. Direct Observation Of Structure-Function Relationship In A Nucleic Acid–Processing Enzyme. *Science*, 348, 352-354.
7. Cordes, T., Vogelsang, J. & Tinnefeld, P. 2009. On The Mechanism Of Trolox As Antiflickering And Antibleaching Reagent. *Journal Of The American Chemical Society*, 131, 5018-5019.
8. Dimura, M., Peulen, T. O., Hanke, C. A., Prakash, A., Gohlke, H. & Seidel, C. A. 2016. Quantitative FRET Studies And Integrative Modeling Unravel The Structure And Dynamics Of Biomolecular Systems. *Current Opinion In Structural Biology*, 40, 163-185.
9. Eggeling, C., Berger, S., Brand, L., Fries, J., Schaffer, J., Volkmer, A. & Seidel, C. 2001. Data Registration And Selective Single-Molecule Analysis Using Multi-Parameter Fluorescence Detection. *Journal Of Biotechnology*, 86, 163-180.
10. Goddard, T. D., Huang, C. C., Meng, E. C., Pettersen, E. F., Couch, G. S., Morris, J. H. & Ferrin, T. E. 2018. Ucsf ChimeraX: Meeting Modern Challenges In Visualization And Analysis. *Protein Science*, 27, 14-25.
11. Guex, N. & Peitsch, M. C. 1997. Swiss-Model And The Swiss-Pdbviewer: An Environment For Comparative Protein Modeling. *Electrophoresis*, 18, 2714-23.
12. Ha, T., Enderle, T., Ogle, D., Chemla, D. S., Selvin, P. R. & Weiss, S. 1996. Probing The Interaction Between Two Single Molecules: Fluorescence Resonance

- Energy Transfer Between A Single Donor And A Single Acceptor. *Proceedings Of The National Academy Of Sciences*, 93, 6264-6268.
13. Hellenkamp, B., Schmid, S., Doroshenko, O., Opanasyuk, O., Kühnemuth, R., Rezaei Adariani, S., Ambrose, B., Aznauryan, M., Barth, A., Birkedal, V., Bowen, M. E., Chen, H., Cordes, T., Eilert, T., Fijen, C., Gebhardt, C., Götz, M., Gouridis, G., Gratton, E., Ha, T., Hao, P., Hanke, C. A., Hartmann, A., Hendrix, J., Hildebrandt, L. L., Hirschfeld, V., Hohlbein, J., Hua, B., Hübner, C. G., Kallis, E., Kapanidis, A. N., Kim, J.-Y., Krainer, G., Lamb, D. C., Lee, N. K., Lemke, E. A., Levesque, B., Levitus, M., Mccann, J. J., Naredi-Rainer, N., Nettels, D., Ngo, T., Qiu, R., Robb, N. C., Röcker, C., Sanabria, H., Schlierf, M., Schröder, T., Schuler, B., Seidel, H., Streit, L., Thurn, J., Tinnefeld, P., Tyagi, S., Vandenberk, N., Vera, A. M., Weninger, K. R., Wünsch, B., Yanez-Orozco, I. S., Michaelis, J., Seidel, C. A. M., Craggs, T. D. & Hugel, T. 2018. Precision And Accuracy Of Single-Molecule Fret Measurements—A Multi-Laboratory Benchmark Study. *Nature Methods*, 15, 669-676.
 14. Kalinin, S., Peulen, T., Sindbert, S., Rothwell, P. J., Berger, S., Restle, T., Goody, R. S., Gohlke, H. & Seidel, C. A. 2012. A Toolkit And Benchmark Study for Fret-Restrained High-Precision Structural Modeling. *Nat Methods*, 9, 1218-25.
 15. Kapanidis, A. N., Laurence, T. A., Lee, N. K., Margeat, E., Kong, X. & Weiss, S. 2005. Alternating-Laser Excitation of Single Molecules. *Accounts Of Chemical Research*, 38, 523-533.
 16. Kudryavtsev, V., Sikor, M., Kalinin, S., Mokranjac, D., Seidel, C. A. & Lamb, D. C. 2012. Combining Mfd And Pie For Accurate Single-Pair Förster Resonance Energy Transfer Measurements. *Chemphyschem*, 13, 1060-1078.
 17. Lerner, E., Cordes, T., Ingargiola, A., Alhadid, Y., Chung, S., Michalet, X. & Weiss, S. 2018. Toward Dynamic Structural Biology: Two Decades Of Single-Molecule Förster Resonance Energy Transfer. *Science*, 359, Eaan1133.
 18. Milles, L. F., Schulten, K., Gaub, H. E. & Bernardi, R. C. 2018. Molecular Mechanism Of Extreme Mechanostability In A Pathogen Adhesin. *Science*, 359, 1527-1533.
 19. Müller, B. K., Zaychikov, E., Bräuchle, C. & Lamb, D. C. 2005. Pulsed Interleaved Excitation. *Biophysical Journal*, 89, 3508-3522.
 20. Nicoli, F., Barth, A., Bae, W., Neukirchinger, F., Crevenna, A. H., Lamb, D. C. & Liedl, T. 2017. Directional Photonic Wire Mediated By Homo-Förster Resonance Energy Transfer On A Dna Origami Platform. *Acs Nano*, 11, 11264-11272.
 21. Nir, E., Michalet, X., Hamadani, K. M., Laurence, T. A., Neuhauser, D., Kovchegov, Y. & Weiss, S. 2006. Shot-Noise Limited Single-Molecule Fret Histograms: Comparison Between Theory And Experiments. *The Journal Of Physical Chemistry B*, 110, 22103-22124.

22. Otto, M. 2009. Staphylococcus Epidermidis — The 'Accidental' Pathogen. *Nature Reviews Microbiology*, 7, 555-567.
23. Piana, S., Donchev, A. G., Robustelli, P. & Shaw, D. E. 2015. Water Dispersion Interactions Strongly Influence Simulated Structural Properties Of Disordered Protein States. *The Journal Of Physical Chemistry B*, 119, 5113-5123.
24. Ponnuraj, K., Bowden, M. G., Davis, S., Gurusiddappa, S., Moore, D., Choe, D., Xu, Y., Hook, M. & Narayana, S. V. 2003. A “Dock, Lock, And Latch” Structural Model For A Staphylococcal Adhesin Binding To Fibrinogen. *Cell*, 115, 217-228.
25. Robustelli, P., Piana, S. & Shaw, D. E. 2018. Developing A Molecular Dynamics Force Field For Both Folded And Disordered Protein States. *Proceedings Of The National Academy Of Sciences*, 115, E4758-E4766.
26. Schrimpf, W., Barth, A., Hendrix, J. & Lamb, D. C. 2018. Pam: A Framework For Integrated Analysis Of Imaging, Single-Molecule, And Ensemble Fluorescence Data. *Biophysical Journal*, 114, 1518-1528.
27. Stein, I. H., Capone, S., Smit, J. H., Baumann, F., Cordes, T. & Tinnefeld, P. 2012. Linking Single-Molecule Blinking To Chromophore Structure And Redox Potentials. *Chemphyschem*, 13, 931-937.
28. Tomov, T. E., Tsukanov, R., Masoud, R., Liber, M., Plavner, N. & Nir, E. 2012. Disentangling Subpopulations In Single-Molecule Fret And Alex Experiments With Photon Distribution Analysis. *Biophysical Journal*, 102, 1163-1173.
29. Verlet, L. 1967. Computer" Experiments" On Classical Fluids. I. Thermodynamical Properties Of Lennard-Jones Molecules. *Physical Review*, 159, 98.
30. Wanninger, S., Asadiatouei, P., Bohlen, J., Salem, C.-B., Tinnefeld, P., Ploetz, E. & Lamb, D. C. 2023. Deep-Learning Assisted, Single-Molecule Imaging Analysis (Deep-Lasi) Of Multi-Color Dna Origami Structures. *Biorxiv*, 2023.01.31.526220.

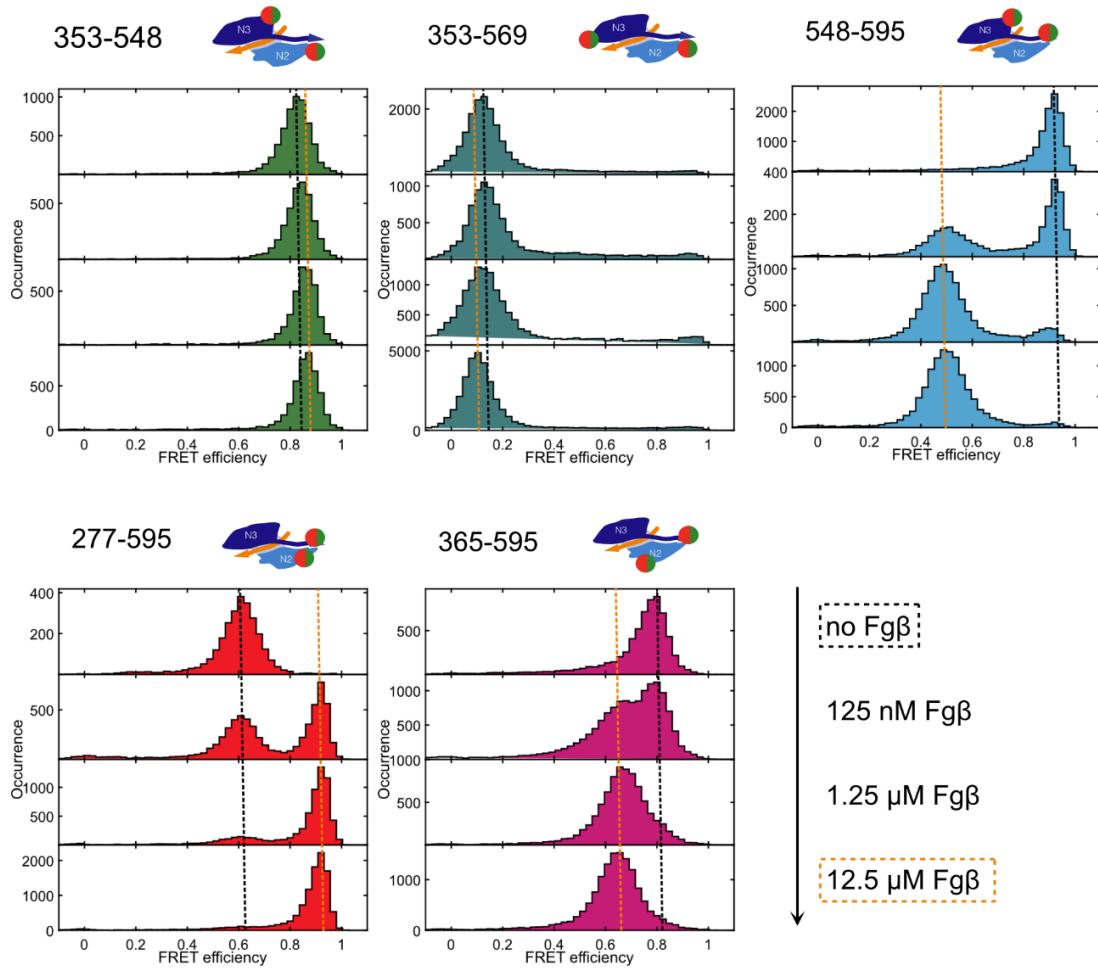
SUPPLEMENTARY INFORMATION

1. FRET efficiency versus donor lifetime for SdrG FRET sensors



SUPPLEMENTARY FIGURE 1: Conformational dynamics of the locking strand indicated in the absence of Fg β . Two-dimensional histograms of the FRET efficiency versus the donor lifetime show a clear deviation from the static-FRET line (shown as a solid red line) for SdrG mutants with a label on the locking strand (residue 595). Data are based on SdrG labeled with Atto532 and Atto643.

2. SmFRET efficiency histograms of SdrG in absence and different concentrations of peptide



SUPPLEMENTARY FIGURE 2: SmFRET efficiency histograms obtained in the absence and at different concentrations of Fg β for the dye combination Atto532-Atto643. Increasing concentrations of Fg β (from top to bottom) shift the equilibrium from the open conformation of SdrG (black dashed line) to the closed conformation (orange dashed line).