# Applications of deep learning in biology

**Inauguraldissertation**
zur Erlangung der Würde eines Doktors der Philosophie vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät der Universität Basel

von
Bastian Thaddäus Eichenberger

2023

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät auf Antrag von:

- Erstbetreuer: Dr. Jeffrey Alan Chao
- Zweitbetreuer: Prof. Dr. Patrick Matthias
- Externer Experte: Prof. Dr. Pierre Vandergheynst

Basel, den 23. Mai 2023

Prof. Dr. Marcel Mayor
Dekan

# Abstract

The rapid growth of biological data sets has created a pressing need for powerful computational tools to analyze and interpret this data. Deep learning, a subfield of artificial intelligence, has emerged as a promising approach for tackling this challenge. In this thesis, the applications of deep learning in biology, mainly focussing on image analysis and predicting protein-protein interaction are explored. I describe the basic components of deep learning models, including convolutional neural networks and modern attention-based networks, and discuss their potential for analyzing complex biological data sets. I also examine the challenges of integrating and applying deep learning in a biomedical context, including issues of data quality, model interpretability, and ethical considerations. Overall, this thesis highlights the exciting potential of deep learning to transform the field of biological research and provides a roadmap for future applications in this rapidly evolving field.

# Acknowledgments

I would like to take this opportunity to express my sincere gratitude to my supervisor, Jeff, for his guidance, support, and encouragement to pursue programming. Jeff's expertise and insights have been instrumental in shaping my research and helping me navigate the challenges along the way.

Next, a big thanks goes to Friedemann and Guillaume for providing me with an engaging project. Despite some waiting periods, it has been a truly enjoyable experience to work on. Our frequent discussions throughout the project have exemplified how science should always be approached – with open communication, collaboration, and a shared passion for discovery.

I would like to take a moment to express my sincere gratitude to my friends, especially Zhan, for their unwavering support throughout my academic journey. Whether they are currently in academia, have left, or were never a part of it, their friendship has been a constant source of inspiration, motivation, and positivity.

A few noteable people that should not remain nameless and have helped in answering questions, onboarding, collaboration, proof-reading, etc. are Daniel, Esther, Georg, Gregory, Jan, Jess, Luca, Markus, and Toby.

Finally, I would like to acknowledge the personal challenges I have faced during my PhD. It has not been easy, but I am proud of myself for persevering and seeking the help and support I needed. I hope that others who may be going through similar struggles are also able to seek help and support, and to know that they are not alone.

# Contents

# List of Figures

# List of Tables

# List of acronyms

| | | |
|---|---|---|
| **AF2** | AlphaFold 2 | 46 |
| **AI** | Artificial Intelligence | 1 |
| **ANN** | Artificial Neural Network | 2 |
| **AP-1** | Activator protein 1 | 48 |
| **API** | Application Programming Interface | 38 |
| **Adam** | Adaptive Moment Estimation | 14 |
| **BERT** | Bidirectional Encoder Representations from Transformers | 45 |
| **C/EBP** | CCAAT-enhancer-binding proteins | 48 |
| **CLI** | Command Line Interface | 21 |
| **CNN** | Convolutional Neural Network | 8 |
| **CREB** | cAMP response element-binding protein | 48 |
| **DHF** | Dihydrofolate | 47 |
| **DL** | Deep Learning | 2 |
| **DNA** | Deoxyribonucleic acid | 6 |
| **ELMO** | Embeddings from Language Models | 45 |
| **GPT** | Generative Pre-trained Transformer | 46 |
| **GPU** | Graphics Processing Unit | 23 |
| **IF** | Immunofluorescence | 39 |
| **ILSVRC** | ImageNet Large Scale Visual Recognition Challenge | 5 |
| **ISBI** | International Symposium on Biomedical Imaging | 22 |
| **KNIME** | Konstanz Information Miner | 38 |
| **LSTM** | Long-term short-term memory | 44 |
| **LoG** | Laplacian of Gaussian | 32 |
| **MAE** | Mean Absolute Error | 17 |
| **ML** | Machine Learning | 1 |
| **MSA** | Multiple Sequence Alignment | 47 |
| **MSE** | Mean Squared Error | 17 |
| **NAG** | Nesterov Accelerated Gradient | 14 |

# 1

# What is deep learning?

Artificial Intelligence (AI) is a field in computer science with the goal of developing self-reliant machines that can think and act like humans. These machines typically attempt to mimic human behavior by simulating natural intelligence and performing tasks by learning and problem-solving. Example applications of AI include self-driving vehicles such as Google's Waymo [1] and Apple's Siri voice assistant [2]. A subfield of AI that uses algorithms to build models capable of learning from data to improve performance at specific tasks is Machine Learning (ML). Specifically, ML uses large amounts of data to make predictions based on observed patterns. Algorithms belonging to ML can generally be classified into one of three categories:

- Supervised learning: the prediction of "labeled" data, meaning the target variable is known. In these algorithms, a function is learned to map feature vectors (inputs) to labels (output) based on examples (input-output pairs).
- Unsupervised learning: the discovery of patterns in unlabeled data. These algorithms can identify hidden features from data allowing patterns and similarities to become more evident.
- Reinforcement learning: the training of an agent to complete a task in a changing environment. The agent must decide on actions based on environmental stimuli and is rewarded based on how successful an action was relative to completing a given goal. Example applications include agents surpassing human performance in board games like Go (AlphaGo [3]) and video games like StarCraft II (AlphaStar [4]).

Representation learning techniques are a class of ML algorithms that aim to learn a useful representation of the input data for a given task. The main goal of these techniques is to automatically discover and extract meaningful and relevant features from the raw data, which can then be used as the input for other machine learning algorithms. The learned representation should ideally capture the underlying structure and patterns in the data, making it easier

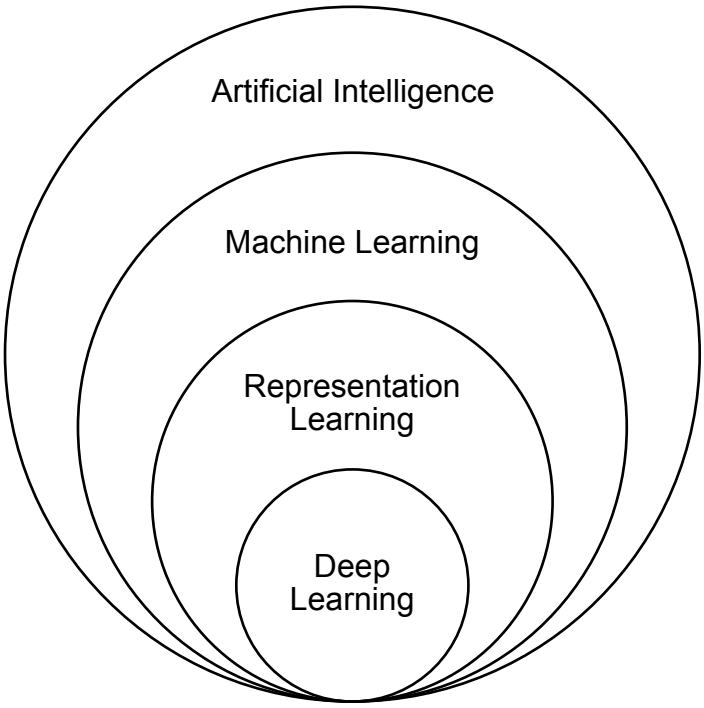



**Figure 1.1:** Venn diagram showing the relationship between artificial intelligence, machine learning, representation learning, and deep learning.

to perform tasks such as classification, clustering, or prediction. Deep Learning (DL) is an example of a representation learning technique (Figure 1.1) relying on neural networks with multiple layers to automatically learn representations of data. Deep learning draws inspiration from the structure and function of biological brains by stacking multiple layers of neurons. These stacks are termed Artificial Neural Networks (ANNs) and enable these algorithms to progressively extract higher-level features from the original input data. In the case of image processing for facial recognition, the first layers might identify edges and borders while later layers can identify concepts such as noses or eyes. This generally results in lower needs for human intervention and larger data requirements.

While traditional machine learning algorithms are undoubtedly useful to solve problems, they have been shown to struggle with more complex problems like Natural Language Processing (NLP) and image recognition. Similarly, due to their simple nature, ML models do not perform well with very large or unstructured datasets. Additionally, ML models rely on feature engineering: the process of extracting input features from raw data using domain knowledge in such a way as to improve model performance. Feature engineering needs to be performed by humans and can often be time-consuming and very difficult. For example, defining how a nose looks like in terms of pixel values. With deep learning, on the other hand, features are

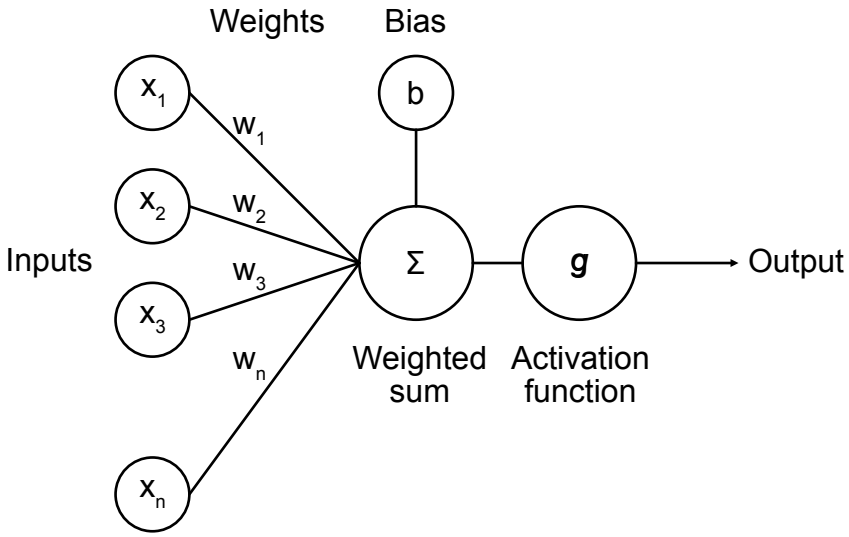



**Figure 1.2:** Overview of a neuron: the building block of artificial neural networks.

automatically extracted by the neural network without human intervention.

## 1.1 How do neural networks learn?

Artificial neural networks are composed of connected sets of individual neurons largely inspired by the communication of biological neurons in brains. These neurons (Figure 1.2) take inputs $x_{1\cdots n}$ and multiply them with input-specific weights $w_{1\cdots n}$. Additionally, a bias term $b$ is added independent of any inputs allowing for affine transformations of inputs [5]. All of these multiplications are combined by summation:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{1.1}$$

Since early work [6] showed that linear perceptrons are not sufficient to work as a universal classifier, a nonpolynomial activation function $g(z)$ is applied after this summation. This can be interpreted as the network deciding when to and when not to "fire". These activation functions introduce non-linearity into a network, allowing it to learn more complex relationships between the input and output. Neurons are then chained together in different architectures allowing for varied data representations. These deviate from biological connectionist models to improve efficiency, trainability, and understandability. For example, ANNs are usually static while neural pathways in brains are plastic [7].

So far, these neural networks are only useful in the "forward" direction (from input to output). To learn, however, there are three key steps: the computation of a loss term, "backpropagation" (from output to input), and updating gradients. Firstly, after an input is fed through the network, the resulting output is compared to the label with a loss function ($L$) which reflects the model's goodness of fit:

$$L(y, \hat{y}) \tag{1.2}$$

With $y$ being the ground truth and $\hat{y}$ the model's prediction. Next, the backpropagation algorithm computes the gradient of the loss function with respect to each weight:

$$\frac{\partial L}{\partial \theta_k} = \frac{\partial a_k}{\partial \theta_k} \frac{\partial L}{\partial a_k} \tag{1.3}$$

Where $a_k$ is the activation (output) from layer $k$ and $\theta_k$ are the parameters associated with that layer (i.e. weights and biases). Since it is inefficient to compute the gradients for each parameter from scratch, gradients are calculated for each layer moving back layer by layer. This is done by propagating the gradients backward through the network using the chain rule of calculus.

$$\begin{aligned} \frac{\partial L}{\partial \theta_{k-1}} &= \frac{\partial a_{k-1}}{\partial \theta_{k-1}} \frac{\partial a_{k-1}}{\partial a_k} \frac{\partial L}{\partial a_{k-1}} \\ &= \frac{\partial a_{k-1}}{\partial \theta_{k-1}} \frac{\partial L}{\partial \theta_k} \end{aligned} \tag{1.4}$$

This process allows us to compute the gradients of the loss function with respect to all of the model parameters, which can then be used to update the parameters using gradient descent. Taken together, backpropagation is a key algorithm in deep learning and has enabled the development of complex neural networks that can learn from large amounts of data. This computationally efficient method for computing gradients has enabled the training of deep networks with hundreds of layers [8].

Lastly, weights are updated using gradient descent. The goal of gradient descent is to find the values of the model parameters $\theta$ that minimize the loss function $L$. The algorithm starts with an initial set of parameters (see Chapter 2.1.4) and iteratively updates them in the direction of the negative gradient of the loss function. The size of each update step is directed by the learning rate $\alpha$. This process is repeated until the loss function is minimized or a stopping criterion, such as time or performance, is met:


**Figure 1.3:** Google search trends showing the relative interest over time in the term "deep learning". Numbers represent search interest relative to the highest point between 2003 and 2023.

$$\theta^{t+1} = \theta^t - \alpha * \frac{\partial L}{\partial \theta} \tag{1.5}$$

Taken together, one training step involves two components: the forward pass and the backward pass. During the forward pass, the input data is passed through the network and the output is computed. During the backward pass, the gradients of the loss function with respect to the output of each layer are computed and then propagated backward through the network to compute the gradients of the loss function with respect to the model parameters. These parameters are updated iteratively minimizing the loss function and increasing model performance.

## 1.2 The Deep Learning Revolution: From AlexNet to Chatbots

Deep learning has seen a steady increase in Google search traffic (Figure 1.3) since the release of AlexNet [9] in 2012. This neural network architecture competed in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [10] and greatly outperformed any competitors using traditional machine learning approaches. This showed that deep learning has come to be a practically feasible approach to solving challenging questions and sparked a deep learning "revolution" that has continued ever since [11].

There have been countless milestones in improving network architectures and model performance some of which will be discussed in later sections. Some example applications of

deep learning in daily life include speech recognition systems [12] employed in Apple's Siri or Amazon's Alexa, various image search systems [13] such as Google image search or recommender systems [14] heavily employed in social media or content platforms such as Netflix. Most recently, chatbots such as OpenAI's ChatGPT [15] have rapidly gained popularity and widespread adoption across the internet becoming the fastest-growing application to date even surpassing the like of Instagram [16].

## 1.3 Exemplary applications in biology

Although ANNs have been inspired by biology, their application has long been kept in the realm of computer science and commercial applications. Only recently have approaches and tools emerged that use various kinds of deep learning architectures for use in biological research. Biology is a prime field for leveraging deep learning due to the vast amounts of data generated in pre-existing experimental runs. Two key areas that are especially data-rich are genomics and image analysis.

One exemplary application in the field of genomics is the prediction of enhancer sequences as has been done in DeepEnhancer [17]. The authors have trained a convolutional neural network on large databases of experimentally derived sequences. By using such tools, experimentalists can focus more on validation and follow-up experiments rather than spending time on expensive high-throughput screens (e.g. using ChIP-seq [18]). Similarly, Zeng and Gifford [19] used CpGenie, a framework to learn Deoxyribonucleic acid (DNA) methylation sites to help in identifying regulatory non-coding variants.

In the adjacent field of image analysis, there have been various successful applications to improve a clinician's understanding of raw biomedical images [20, 21]. Since most methods can match or even surpass human performance, wrong decisions are greatly reduced and help in improving patient outcome [22]. In a laboratory setting, tools such as DeepLabCut [23, 24], used for pose estimation in video data, enable more efficient and accurate analyses of behavioral imaging data.

## 1.4 Problem statement

Deep learning holds great potential of revolutionizing all aspects of life. While there has been significant progress in leveraging deep learning in some fields, such as computer vision and natural language processing, its application in biology is still in its early stages. However,

the availability of a fast-growing array of open-source tools is making it easier than ever to get started with deep learning in biology. In light of these developments, this thesis aims to explore the potential of deep learning in biology through two case studies. First, how deep learning can be used in improving light-microscopy image analysis, and second in predicting the interaction of two proteins. Furthermore, shortcomings will be discussed that are currently hindering widespread adoption.

# 2

# Image analysis

A quarter of a cubic millimeter might sound minuscule but it can fit the entirety of a *Drosophila melanogaster* (common fruit fly) brain [25, 26]. This roughly equates to 200'000 cells [27]. Modern high aperture microscopes have enabled researchers to image this impressive structure at a resolution of $\sim$200nm. Since multiple brains are typically imaged per experimental condition, a single fluorescent color imaging session can quickly produce data in the terra-byte range – far too much for manual analysis. Due to rapid advances in microscope technology, researchers are often left with a large pile of data. However, where humans might struggle, computer algorithms can be used to identify features in an image.

## 2.1 Introduction to convolutional neural networks

In recent years, deep learning has seen a rapid rise in popularity, and it has become the go-to approach for various tasks. One of the areas where deep learning has shown remarkable success is computer vision. Convolutional Neural Networks (CNNs) are a class of deep neural networks that have revolutionized image analysis and are now the standard tool for various image-based tasks. The success of CNNs in image analysis has been largely due to their ability to automatically learn spatial hierarchies of features from image pixels, thus making them capable of handling large and complex datasets. The use of CNNs in image analysis will be the focus of subsequent discussions, including their structure, variations, and the various successful applications where they have been employed.

### 2.1.1 Basic components of a convolutional neural network

As described in Chapter 1, all ANNs rely on the neuron as building blocks. In the case of CNNs, while the general concept is the same, they differ in a few key ways. The convolution operation is a mathematical operation that applies a filter, known as a kernel, to an input signal

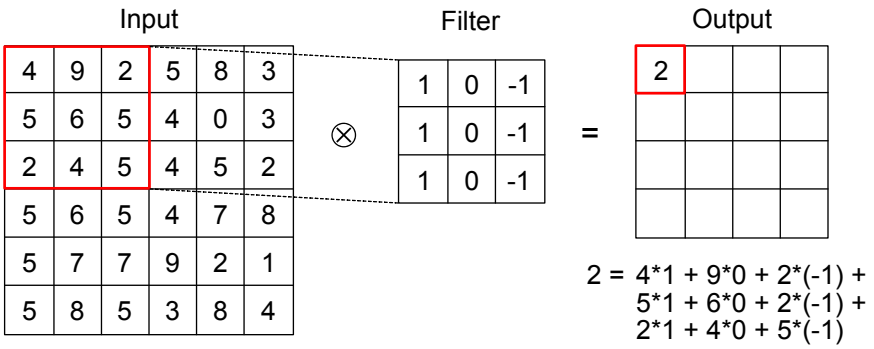



**Figure 2.1:** An example convolution operation with kernel size 3×3, stride 1, and no padding. Adapted from: indoml.com.

or image. The filter consists of a small matrix of weights that are learned by the network during training. The convolution operation works by sliding the filter over the input data, one small region at a time, and computing the dot product between the filter weights and the values in the corresponding region of the input data (Figure 2.1). This dot product produces a single output value, which represents the presence or absence of a certain feature in the input data. The output values are then collected into a new output matrix, called a feature map, which represents the locations of features in the input data. More specifically, the convolution operation works as follows:

- The input data is represented as a matrix, with the values in the matrix corresponding to the pixel intensities in the image. Most applications use 2D input images but networks can also be applied to 3D [28] stacks or videos [22].
- The filter is a smaller matrix of weights, typically with dimensions that are much smaller than the input data such as 1×1, 3×3, or 5×5. The filter is initialized at the beginning of training (see Chapter 2.1.4), and its weights are updated during training using backpropagation.
- The filter is then placed at the top left corner of the input matrix and its weights are multiplied element-wise with the values in the corresponding region of the input matrix (Figure 2.1).
- The resulting products are then summed up to produce a single output value, which is placed in the corresponding location in the output feature map.
- The filter is then shifted to the right by one or more pixels (known as the stride), and the same operation is performed on the new region of the input matrix, producing another output value in the feature map.

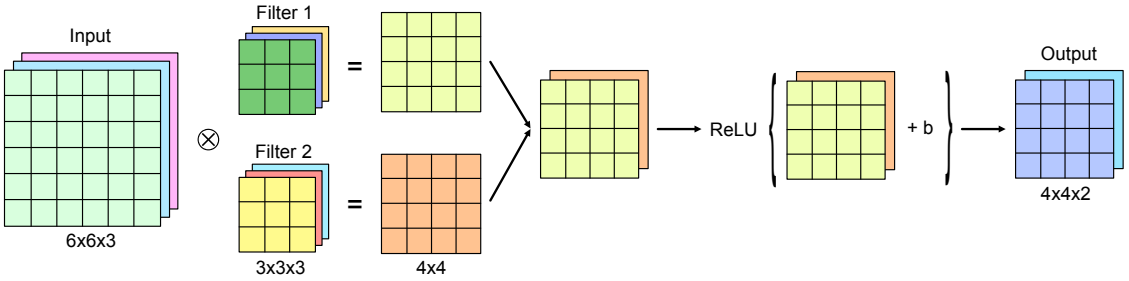



**Figure 2.2:** An overview of a convolutional layer ranging from input through a ReLU activation function to an output. Adapted from: indoml.com.

- This process is repeated for every possible position of the filter over the input matrix until the entire input matrix has been processed.
- The resulting feature map represents the presence or absence of a certain feature in the input data, and is passed on to the next layer in the CNNs for further processing.
- Depending on the filter size as well as other parameters such as padding or stride size, the feature map will typically have a different size compared to the input [29].

Overall, the convolution operation allows the CNNs to learn spatial features from input data, such as edges, corners, and shapes, that are important for most computer vision tasks.

Commonly used activation functions are Rectified Linear Units (ReLU) [30] and all of their derivates such as Gaussian Error Linear Units (GELUs), leaky ReLUs, and Exponential Linear Units (ELUs). ReLU follows the simple formula of:

$$f(z) = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

Most derivatives slightly differ in their negative term (e.g. leaky ReLU uses a small, non-zero, constant gradient $\beta$). One problem arising as networks become increasingly deep is the vanishing gradient problem. This refers to the phenomenon where the gradients of the loss function with respect to the weights become extremely small, making it difficult for the network to learn. This is a big issue in deep networks since the gradients are backpropagated through many layers. As the gradients are multiplied together (see Chapter 1.1), they can become exponentially small, especially when using activation functions such as sigmoid or tanh that saturate at the extremes of the input range. When the gradients become too small, the network stops learning, and its performance plateaus. ReLUs have become popular since

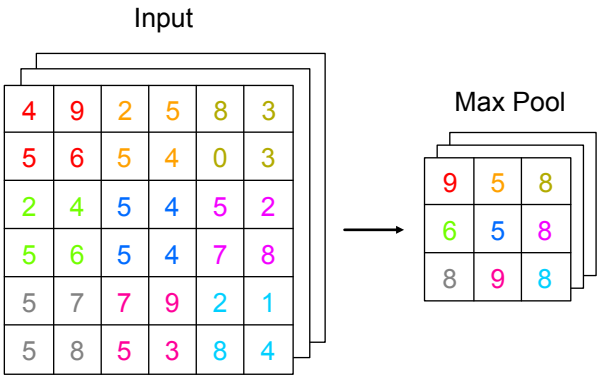



**Figure 2.3:** An example max pooling operation with size 2 and stride 2. Adapted from: indoml.com.

they help mitigate this problem. Additionally, since ReLU has a constant gradient of 1 for all positive input values it allows for faster and more stable training. Another benefit of ReLU is that it is computationally efficient to compute, both in terms of memory usage and processing time making it well-suited for large-scale applications such as video processing [31].

Each convolutional kernel is only able to capture local context. Since images typically have different (relative) context sizes, pooling layers are often added after convolutional layers to downsample feature maps and aggregate spatial information. This helps to reduce the dimensionality of the feature maps and makes the network more computationally efficient. Common pooling operations include max pooling and average pooling, which respectively select the maximum or average value within a local region of the feature map (Figure 2.3).

### 2.1.2 Improvements in regularization

Generalizability is a crucial factor in evaluating the effectiveness of a model. It refers to the model's ability to perform well on new, unseen data, beyond the data it was trained on. This is important because a model that only performs well on the training data, but fails to generalize to new data, is of little practical value. To help evaluate a model's generalizability, the typical way to split data is into three subsets:

- Training set: used to train the model's parameters and the largest subset.
- Validation set: used to tune the model's hyperparameters. Hyperparameters are parameters that are set by the user, rather than learned from the data. They affect the behavior and performance of the model such as the learning rate $\alpha$.

- Test set: to evaluate the model's performance after training and hyperparameter tuning.

Overfitting is another common problem that occurs when a model is too complex and memorizes the noise or patterns in the training set, rather than the underlying relationships in the data. This can result in poor performance on new data, and the model is said to be overfitted. To prevent overfitting, regularization techniques can be employed, which add penalties or constraints to the optimization objective to constrain the model's capacity.

Batch normalization is a technique that normalizes the input of each layer to have zero mean and unit variance over a batch of samples [32]. This helps to mitigate the effects of covariate shift, which can cause the distributions of activations to change during training and make the optimization process more difficult. By stabilizing the distributions of activations, batch normalization can improve the speed and stability of training and lead to better generalization.

L1 and L2 norm regularization add a penalty term to the loss function during training to encourage the weights to be small. L1 regularization adds the sum of the absolute values of the weights, while L2 regularization adds the sum of the squares of the weights. These penalties can help prevent overfitting by reducing the complexity of the model and encouraging it to learn simpler patterns.

Dropout randomly drops out a fraction of the activations in a layer during training [33]. This can help prevent overfitting by forcing the network to learn more robust features that are not overly dependent on any single activation. Dropout is effective in a variety of tasks and can improve the generalization performance of neural networks.

Other methods include data augmentation where new training data is generated from existing data by applying random transformations (e.g. mirroring, rotating, and shifting [34]) as well as early stopping where the training process is stopped once the performance on the validation set starts deteriorating past a set criterion.

### 2.1.3 Improvements in network architecture

Modern CNN architectures have come a long way since the groundbreaking AlexNet architecture was introduced in 2012 [9]. While AlexNet achieved unprecedented accuracy on the ImageNet dataset, subsequent architectures have further improved its performance by introducing several key innovations. One major trend in modern CNN architectures is the use of smaller convolutional filters, typically $3\times3$, rather than the $11\times11$ filters used in AlexNet. This has several advantages, including a reduction in the number of parameters in the model,

which helps to reduce overfitting and improve generalization, and the ability to capture more fine-grained features in the input data.

Another key innovation is the use of deeper architectures, with more layers than AlexNet. This has been made possible by the development of new techniques for training deep neural networks, such as batch normalization (see Chapter 2.1.2), which helps to mitigate the vanishing gradient problem that can occur in very deep networks (see Chapter 2.1.1). Deeper architectures have been shown to be better at capturing high-level, abstract features in the input data, and have achieved state-of-the-art performance on a variety of computer vision tasks [8]. Other innovations in modern CNN architectures include the use of residual (also termed skip) connections, which allow information to bypass certain layers in the network and can help to alleviate the vanishing gradient problem by allowing gradients to flow more easily through the network. Some modern CNN architectures such as ResNet [8] and DenseNet [35] have favored the usage of skip connections to preserve spatial information and have moved away from using pooling layers (see Chapter 2.1.1).

Modern CNN architectures have significantly improved the performance of the original AlexNet architecture by introducing a range of innovative techniques for capturing fine-grained features, training deep networks, and improving generalization. Two network architectures in particular have pushed the boundaries of performance:

- Inception [36, 37], introduced in 2014, uses multiple parallel convolutional layers with different filter sizes and pooling operations to capture features at different scales, which allows it to extract more diverse and informative features from the input data. Additionally, it uses $1 \times 1$ convolutions to reduce the dimensionality of the input and save computational resources, which improves the model's efficiency. This architecture also won the 2014 ILSVRC classification challenge.
- ResNet [8], introduced in 2015, uses residual connections to enable information to skip certain layers in the network, allowing it to learn from both the original input and the residual error of the network's prediction. This enables ResNet to train much deeper networks with up to hundreds of layers while avoiding the vanishing gradient problem. This architecture achieved state-of-the-art performance on a range of computer vision tasks and won the 2015 ILSVRC.

Recently, attention-based models (discussed in more detail in Chapter 4.2) have become popular for image classification [38]. These models use an attention mechanism to focus on important regions of the image. One example of this is the SENet [39], which uses a squeeze-and-excitation module to selectively amplify important features.

Overall, these architectures have significantly improved the performance of CNNs compared to AlexNet by using innovative techniques such as residual connections, parallel convolutional layers with multiple filters, and attention. These techniques help to address the issues of vanishing gradients, overfitting, and computational efficiency and have made CNNs even more powerful for image analysis and other applications.

### 2.1.4 Improvements in optimization

Weights in neural networks can be set (initialized) to any values before training. In the past, initializations were often done randomly or with fixed values, which often led to vanishing or exploding gradients, making them more prone to failure and training overall less reliable. Modern weight initializations such as Normal, Xavier [40], and He [41] provide more stable and efficient training. Normal weight initialization randomly samples weights from a standard normal Gaussian distribution. Xavier initialization, on the other hand, scales the weights based on the number of input and output neurons to ensure the variance of the activations is constant across layers. This helps to ensure that the gradients are neither too small nor too large. Finally, He initialization is similar to Xavier initialization but scales the weights differently to account for the ReLU activation function commonly used in modern networks.

Similarly, there have been various improvements in the choice of optimizers. Compared to the traditional Stochastic Gradient Descent (SGD), where weights are updated by subtracting the gradient of the loss with respect to each weight multiplied by a learning rate (see Chapter 1.1). Nesterov Accelerated Gradient (NAG) [42] is an extension of SGD that adds a momentum term to the update rule. Momentum allows one to better handle noisy gradients and converge faster by adding a fraction of the previous update to the current update. Adaptive Moment Estimation (Adam) [43] is a popular optimizer that uses a combination of momentum and adaptive learning rates. Adam adapts the learning rates for each weight based on their historical gradients, allowing for faster convergence and improved generalization. More recently, there have been strategies to manually [44] or automatically [45, 46] change the learning rate. Here, the learning rate starts at a high value allowing the model to explore parameter space effectively before decreasing gradually reducing to help converge toward the optimal solution. The choice of method and the rate of decay will vary depending on the specific problem and dataset characteristics.

### 2.1.5 Types of tasks and applications

CNNs have become a popular and effective tool for many image-based applications, including image classification, object detection, image segmentation, and more. These networks have demonstrated state-of-the-art performance on many benchmark datasets, as will be further discussed below.

**Image classification** Image classification attempts to assign one or more classes to an image (e.g. classifying cats and dogs). CNNs are the backbone of modern image classification methods, and their success is attributed to their ability to learn hierarchical representations of images. The previously described architectures Inception and ResNet, as well as countless other approaches (VGGNet [47], EfficientNet [48], CoCa [49]), primarily belong into this category. Image classification has demonstrated significant potential in improving the accuracy and speed of medical image analysis for important applications such as early detection of diseases like diabetic retinopathy [50] and COVID-19 [51]. By accurately identifying patterns in medical images, image classification models can assist healthcare professionals in making more informed and timely diagnoses, leading to better treatment options and ultimately, improved patient care.

**Object detection** Object detection is a critical task in computer vision that involves identifying and localizing objects within images. Modern object detection networks, such as R-CNN, Faster R-CNN, You only look once, and Single Shot Detector, have significantly advanced the field of computer vision by improving the accuracy and speed of object detection.

Region-Based Convolutional Neural Network (R-CNN) [52] was introduced in 2014 as one of the first object detection models that used a Region Proposal Network (RPN) to identify object regions before applying a CNN to classify the object within the region. The RPN predicts object proposals using a sliding window approach, generating proposals of various sizes and aspect ratios, which are then refined by a Region of Interest (RoI) pooling layer. The CNN then classifies the object within the RoI. Despite its high accuracy, R-CNN was slow and inefficient due to its multi-stage approach. Fast R-CNN [53] improves upon R-CNN by using the entire image as input to the CNN, and then using a RoI pooling layer to extract features for each proposed region from the CNN's output. This avoids the need to run the CNN separately for each region proposal. Faster R-CNN [54] builds upon Fast R-CNN by replacing the selective search algorithm with a RPN that learns to generate region proposals directly from the CNN's feature maps. This eliminates the need for a separate region proposal step and further improves speed and accuracy.

You only look once (YOLO) [55], introduced in 2016, was the first real-time object detection model that achieved high accuracy while being computationally efficient. YOLO divides the image into a grid and predicts the bounding box, class probabilities, and confidence score for each grid cell. YOLO uses a single CNN to predict the output for the entire image in one shot, making it extremely fast. Single Shot Detector (SSD) [56] is another real-time object detection model that uses a single CNN to predict the bounding boxes and class probabilities for multiple objects in an image. SSD applies convolutional filters of various sizes and aspect ratios to feature maps from different layers of the network, allowing it to detect objects at different scales and sizes.

Overall, these modern object detection networks have significantly improved the accuracy and speed of object detection in computer vision. They use innovative techniques such as region proposal networks, single CNNs, and grid-based predictions to improve efficiency while maintaining high accuracy. These techniques have made object detection practical for a wide range of applications, such as autonomous driving [57], surveillance [58], and robotics [59].

**Image segmentation** Segmentation models, such as U-Net and Mask R-CNN, are used for pixel-level prediction of object boundaries and are widely used in biomedical imaging, autonomous driving, and robotics. Segmentation models are usually differentiated between semantic and instance segmentation. In semantic segmentation, a binary segmentation map is returned on a class basis whereas instance segmentation also separates individual instances (e.g. cells or people) from each other. They build on the advancements in CNNs and object detection networks to achieve high accuracy and speed in semantic segmentation.

U-Net [60], introduced in 2015, is a fully convolutional network that combines a contracting path (encoder) with an expansive path (decoder) to create a U-shaped architecture. The contracting path consists of a series of convolutional layers followed by max-pooling layers that progressively downsample the input image. The expansive path then consists of a series of convolutional layers followed by up-sampling layers expanding the image back to its original size. The contracting path is generally used to capture the context of the image, while the expansive path captures the details of the objects. The output of the network is a binary, class-based, pixel-wise prediction of the object boundaries.

Mask R-CNN [61], introduced in 2017, builds on the architecture of Faster R-CNN [54] by adding a mask prediction branch to the output of the network. This allows for accurate pixel-level segmentation of objects in an image. The mask branch consists of a series of convolutional layers and upsampling layers that predict a binary mask for each object proposal.

**Other applications** There are many other applications of CNNs including object tracking [62], scene labelling [63], action recognition [64], and pose estimation [65]. Taken together, a multitude of advances have enabled CNNs to achieve state-of-the-art performance on a wide range of computer vision tasks, and have helped to establish deep learning as a powerful tool for image analysis and other applications.

### 2.1.6 Improvements in loss function

There are several common loss functions used for the different image-based tasks described above. The choice of loss function depends on the specific task and the desired output. For example, in image classification, the cross-entropy loss function is commonly used, which measures the difference between the predicted probability distribution over classes and the true class labels:

$$L(y, \hat{y}) = -\frac{1}{N} \sum y \log(\hat{y}) + (1 - y) \log(1 - p) \tag{2.2}$$

Here, $N$ is the number of samples in the dataset, $y$ is the true label for that class (either 0 or 1) and $\hat{y}$ is the predicted class probability. The first term computes the loss when the label is 1, and the second term when the label is 0. The negative sign ensures loss is minimized.

For more complex tasks such as object detection, the loss function typically includes both a classification loss and a localization loss, such as the RPN loss used in Faster R-CNN. In image segmentation, the Dice loss and binary cross-entropy loss are commonly used, which measure the similarity between the predicted segmentation and the ground truth segmentation at a pixel-level for each class. Other loss functions such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) may also be used for regression-based tasks:

$$L(y, \hat{y}) = \frac{1}{N} \sum (y - \hat{y})^2 \tag{2.3}$$

MSE squares the difference between prediction and ground truth thus amplifying positive and negative outliers equally leading to an overall regression to the mean.

## 2.2 Common tasks and problems in image analysis

Image analysis has been a fundamental area of research for as long as computer vision has been around. As the world becomes increasingly digital, the ability to extract meaningful information from digital images has become a crucial area of research. Most methods have been relying on mathematical operators to perform a wide range of tasks.

Image filtering is a common technique used for image enhancement, which aims to remove noise and improve image quality. Filtering techniques can be broadly classified as spatial and frequency domain-based methods. In the subfield of segmentation, the goal is to separate RoIs such as cells in an image from the background. This task is challenging, especially when objects have complex shapes, sizes, or similar intensities as the background. Several algorithms have been proposed for segmentation, including thresholding, edge-based methods, and clustering.

Object tracking is the process of following an object's position and motion in successive frames of a video sequence. Tracking algorithms can be classified into two categories: point-based and region-based methods. Point-based methods track individual points or features in successive frames, whereas region-based methods track the object's entire RoI (which are usually already segmented).

Image registration involves the alignment of two or more images that have been captured from different viewpoints, at different times, or using different methods. This task is crucial in many applications, including spatial transcriptomics to register images between readout steps [66]. Registration algorithms aim to find a spatial transformation (e.g. rotation, shifting) that aligns the images.

Spot detection is the process of locating individual fluorescent spots or particles in microscopy images. This task is commonly used in biological imaging to detect and quantify biomolecules [67] or protein expression levels [68]. Spot detection algorithms use various techniques primarily relying on user-defined thresholds [69, 70] to detect spots while reducing false positives.

## 2.3 Image analysis is enhanced with deep learning

Fluorescence microscopy is a powerful technique for visualizing biological samples at high resolution and has revolutionized our understanding of cellular processes. However, analysis of fluorescence microscopy data can be challenging, especially when dealing with large

datasets and complex structures. Deep learning models are capable of automatically learning spatial features from raw image data and can be trained to perform a variety of image analysis tasks, such as segmentation, classification, and object detection. One of the key advantages of deep learning for fluorescence microscopy analysis is its ability to handle large datasets with high variability. Some of the previously described traditional image analysis methods often require manual parameter tuning and feature engineering, which can be time-consuming and subjective. This makes analyses less reproducible and overall prone to errors. In contrast, deep learning models can automatically learn relevant features from the data and can be trained on large datasets to generalize to new samples. Another advantage of deep learning for fluorescence microscopy analysis is its ability to handle complex structures and patterns. For example, deep learning models can be trained to recognize and segment different cell types, organelles, or subcellular structures based on their unique fluorescence patterns. This can enable researchers to study cellular processes in more detail and to identify potential biomarkers or therapeutic targets. A selection of deep learning-based tools that have greatly improved image analysis are discussed below.

Cellular segmentation is a core task in nearly all biomedical image analyses. Two prominent deep learning-based examples, namely Cellpose [71] and Stardist [72, 73] have been trained on large datasets of fluorescence microscopy images and provide state-of-the-art segmentation performance on a wide range of biological samples. Both methods use slightly different approaches to separate cells and can be applied to both 2D and 3D images. Specifically, Cellpose uses a flow field that encodes vertical and horizontal gradients for each cell, so that cells near each other vertically can still be differentiated horizontally. Meanwhile, Stardist relies on Star-convex polygons parametrizing radial distances from the center of the nucleus.

Two further exemplary uses for deep learning in biomedical image analysis are in the field of denoising and lineage tracing. First, methods such as Noise2Void 1 [74] and 2 [75], as well as CARE [76], use artificially or biologically generated data to train a network to predict de-noised output images. Having less noisy images allows most downstream tasks to be more efficient such as cellular segmentation and lineage tracing. By annotating segmentation maps, neural networks such as DeepCell [77, 78, 79] can pick up features such as morphology, motion, and local neighborhood to create cell lineages. These methods have been able to outperform hidden Markov models on challenging datasets such as live imaging of cell division.

# 3

# deepBlink

Nucleic Acids Research. 2021 July 1st; Vol. 49, No. 13; 7292 - 7297


- **Bastian Th. Eichenberger**[*#]
- YinXiu Zhan[*#]
- Markus Rempfler
- Luca Giorgetti[#]
- Jeffrey A. Chao[#]

[*] These authors have contributed equally to this work

[#] Corresponding authors


Yinxiu Zhan and I coded the software and trained models. I coded the benchmarking routines and conceived the idea. Yinxiu Zhan and I directed the project and wrote the paper.

## 3.1 Abstract

Traditionally, mathematical operators designed for idealized spots are used to perform the detection of diffraction-limited spots in single-molecule microscopy images. This process necessitates manual adjustment of parameters which can be both time-consuming and unreliable. A novel neural network-based method, termed deepBlink, will subsequently be described enabling automatic detection and localization of spots. It has been demonstrated that deepBlink surpasses other state-of-the-art methods in accuracy when tested against six publicly available datasets containing synthetic and experimental data.

## 3.2 Introduction

In various imaging applications within biomedical research, the detection, counting and localization of sub-diffraction fluorescent signals (spots) is considered a crucial step. For instance, particle tracking in live cell imaging data and quantification of mRNAs in single molecule fluorescence in-situ hybridization (smFISH). Although developments in smFISH-based spatial transcriptomics methodologies have enabled the quantification of thousands of mRNAs in single cells, they have mainly focused on improving transcript barcoding strategies for multiplexing [80, 81] while relying on conventional threshold-based detection methods. Current spot detection methods face a challenge in accurately detecting and localizing spots in single-molecule fluorescent microscopy images with varying background brightness levels and spot qualities. The widely used TrackMate [70], for instance, is based on intensity thresholds and requires ad-hoc adjustments of parameters on a cell-by-cell and image-by-image basis, which are time-consuming and not always reproducible. Fully automated, user-friendly, accurate, and reproducible spot detection and localization methods are currently not available.

Deep learning methods use ANNs to automatically learn features and patterns from raw data. Despite the requirement of labeled data for deep learning, once a model is trained, it can be used without further modifications or parameter adjustments. State-of-the-art for computer vision applications is consolidated by the CNN through recent advances such as AlexNet [9]. Previously, CNNs were used for threshold-independent particle detection, but they were unable to localize particle positions with sub-pixel resolution [67, 82, 83]. This caused inaccurate positional measurements, which hindered their application in high-resolution microscopy. Furthermore, although the source code of some methods is available, none of them have been made easily accessible for non-experts.

deepBlink is a Command Line Interface (CLI)-based tool designed for automatic detec-

tion and localization of diffraction-limited spots. deepBlink utilizes neural networks to achieve sub-pixel localization of spots in microscopy images. A benchmarking study was conducted to compare deepBlink's performance in spot detection with state-of-the-art methods across multiple datasets having different signal-to-noise ratios (SNRs). It was demonstrated that deepBlink provides more accurate and precise detection of spots.

## 3.3 Methodology

### Training data

The creation of the datasets named "Microtubule", "Receptor", "Vesicle", "Particle", "smFISH", and "SunTag" were carried out using different methods. The first three datasets were derived from the International Symposium on Biomedical Imaging (ISBI) Particle Tracking Challenge 2012 [84]. The dataset consisted of 1200 images and was split into training, validation, and test sets, ensuring an equal distribution of SNRs and densities. The "Particle" dataset was generated using the "Synthetic Data Generator" Fiji plugin (smal.ws) described in Smal et al. [85]. A total of 576 images were generated with various SNRs and background intensities to resemble small diffraction-limited particles. The "smFISH" dataset was created from a pool of 643 manually annotated smFISH images published in Horvathova et al.[86]. Finally, the "SunTag" dataset was composed of 544 manually annotated live cell SunTag images published in Mateju et al. [87].

### Dataset labeling

The experimental smFISH and SunTag datasets were labeled with the help of TrackMate [70], where each image underwent manual thresholding and any incorrectly detected spots were corrected manually. The "All Spots statistics" tool was used to export the labels. Following the labeling of all images, the deepBlink command `deepBlink create` was utilized to convert the images and labels into a partitioned dataset file, which could be utilized for either training or evaluation.

### Network architecture

The U-Net architecture is the basis for deepBlink's default network [60]. To reduce the fully sized representation to a smaller one corresponding to the cell size, a second encoder was

used whose depth can be adjusted for the configured cell size. The encoding steps include a convolutional block, squeeze and excitation block, and spatial dropout (see Figure 3.1 and Supplemental Table S1). The decoding steps followed the same layout, but dropout was not employed. The architecture was implemented in Python version 3.7 using the TensorFlow framework version 2.2.

### Network training

The default settings provided by deepBlink config using the public CLI on version 0.1.0 were used to train all models. The training was performed with a cell-size of 4, a dropout rate of 0.3, a batch size of 2, a learning rate of 0.0001, and the AMSGrad stochastic optimizer. The training was conducted on a computer cluster with 64-bit CentOS Linux 7 (Core), 96 cores (3926 threads) Intel(R) Xeon(R) Platinum 8168 CPU 2.70 GHz, 1007 GB RAM, and 8 Nvidia GeForce RTX 2080 Ti Graphics Processing Units (GPUs) with 11 GB VRAM. On average, the training took around 5 h per model, and all models were trained for 200 epochs.

### Benchmarking of TrackMate

To determine the optimal parameters for TrackMate [70], a grid search was conducted on the training and validation images from each dataset. Since TrackMate relies on manual thresholding and is not a machine learning method, the grid search involved testing different values of "Estimated blob diameters" in the "LoG" filter option in Fiji. The resulting quality scores were normalized to the median of all images and divided into twenty quantiles. The F1 integral score was calculated for each image at each quantile. The parameters that yielded the highest mean F1 score on the combined training and validation dataset (i.e., blob diameter and absolute quantile value) were selected. Grid search is a hyperparameter tuning method where multiple combinations of parameter values are tested to identify the optimal set of parameters that produce the best performance.

### Benchmarking of DetNet

As a publicly available implementation of DetNet [83] was not available, a model following the training procedure described in the publication was implemented. It is worth noting that the number of parameters in this implementation was 10 times larger than the original model. Ground truth output masks were generated using the coordinate locations, with a single pixel

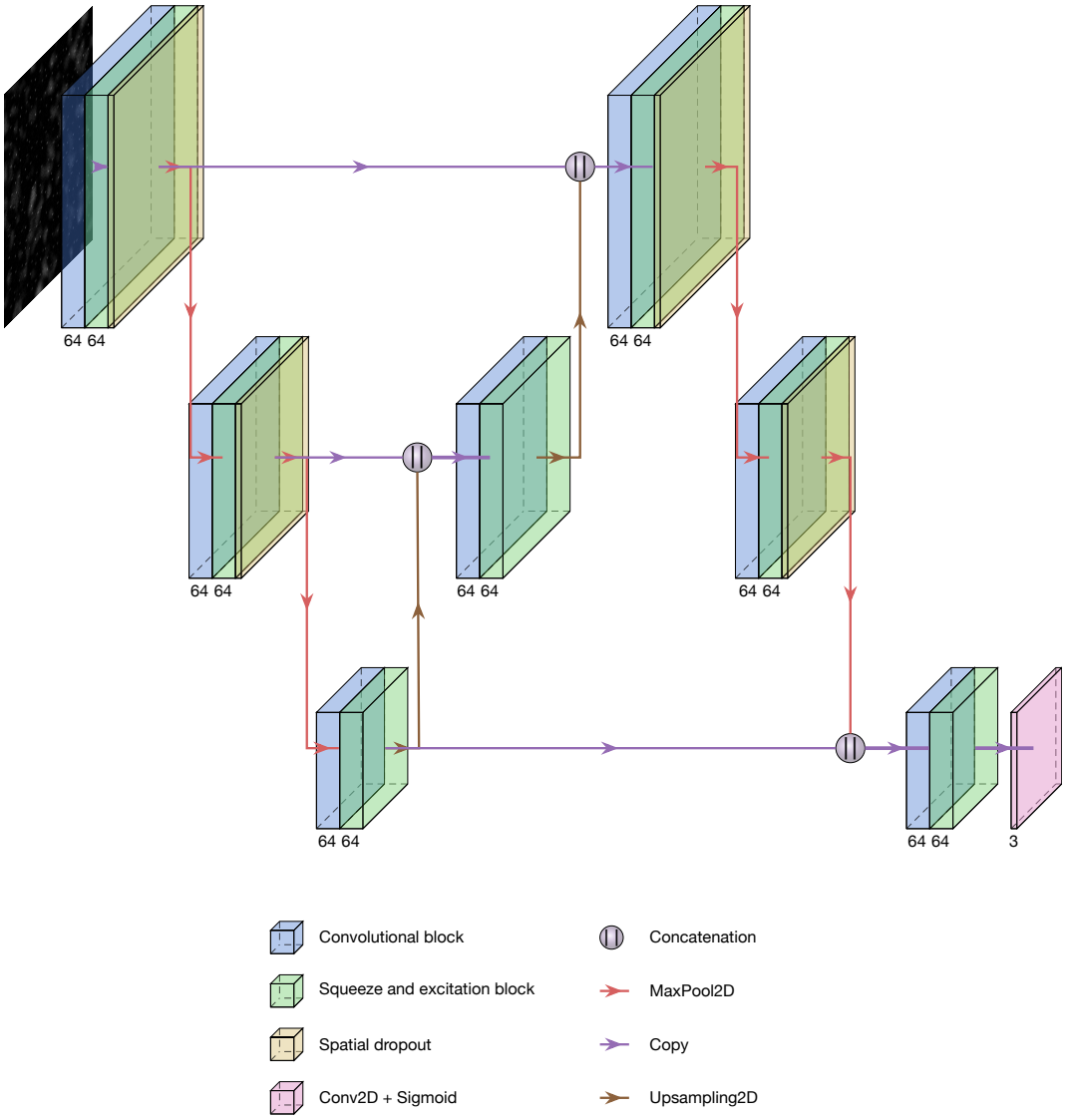



**Figure 3.1:** Overview of the default neural architecture for deepBlink (components are described in Supplemental Table S1). The figure was created using modified PlotNeuralNet code (https://github.com/HarisIqbal88/PlotNeuralNet).

of value one at the centroid position. To convert the output segmentation maps back to coordinates, they were binarized, and calculated the centroids of each fully connected component. To optimize the model, a hyperparameter search was conducted across twenty sigmoid shifts $\alpha$, which were linearly distributed. The $\alpha$ value that yielded the highest validation F1 score was selected for evaluating the model on the test set.

### Benchmarking of deepBlink

The training of deepBlink was performed using version 0.1.0 of the CLI. Each dataset was trained separately without optimizing hyperparameters specific to the dataset. Afterward, the models were evaluated on their corresponding test sets.

### Benchmarking of execution time

The time to predict coordinates from an in-memory image was measured on a set of 512 $\times$ 512 px images using Python's datetime module. To ensure consistency, dependencies were pre-loaded before benchmarking. The measurements were conducted on a MacBook Pro with a 2.4 GHz Quad-Core Intel Core i5 and 16GB 2133 MHz LPDDR3 Memory. A total of seven rounds of measurements were performed, with each round consisting of 100 images. Mean and standard deviation was calculated across all rounds. It is worth noting that all methods were benchmarked on the same set of images.

### F1 integral score

To evaluate the detection and localization accuracy of the models, the F1 integral score was selected as the metric. A linear distribution of 50 cutoff values ranging from 0 to 3 px was created. At each cutoff value, the Hungarian method [88] was used to match the ground truth and predicted coordinates with Euclidean distances below the given cutoff. The trapezoidal rule was then used to integrate the F1 score curve (see Figure 3.4). The integrated value was normalized to the maximum area, resulting in a minimum score of 0 and a maximum score of 1. The algorithmic overview is described in Supplementary Algorithm 1, and the implementation can be found in deepBlink's source code under `deepblink.metrics.f1_integral`.

### Root mean square error

The definition of Root Mean Squared Error (RMSE) was as follows: at a cutoff value of 3 px, calculate the mean Euclidean distance of all predicted coordinates that are assigned to a ground truth coordinate, which are considered true positive.

$$RMSE(\hat{y}, y) = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{3.1}$$

## 3.4 Results and discussion

Here, deepBlink is presented as a novel neural network-based tool for the detection and localization of diffraction-limited spots. Images with varying background brightness levels and spot qualities can be batch-processed using deepBlink without the need for manual threshold adjustments. Spot detection and localization can be performed on images using a pre-trained model through deepBlink's interface (Figure 3.2, steps 5-9). To facilitate new users, pre-trained models can be downloaded within deepBlink (Figure 3.2, step 4). Additionally, custom models can be trained on labeled data (i.e. images and their corresponding spot-coordinate annotations) through deepBlink (Figure 3.2, steps 1-3). The plug-and-play software architecture with a central configuration file allows experienced deep learning practitioners to easily adjust existing functionalities or add new ones to customize training. A video tutorial (https://youtu.be/vIXMg4k79LQ) is provided with instructions on how to install and use deepBlink.

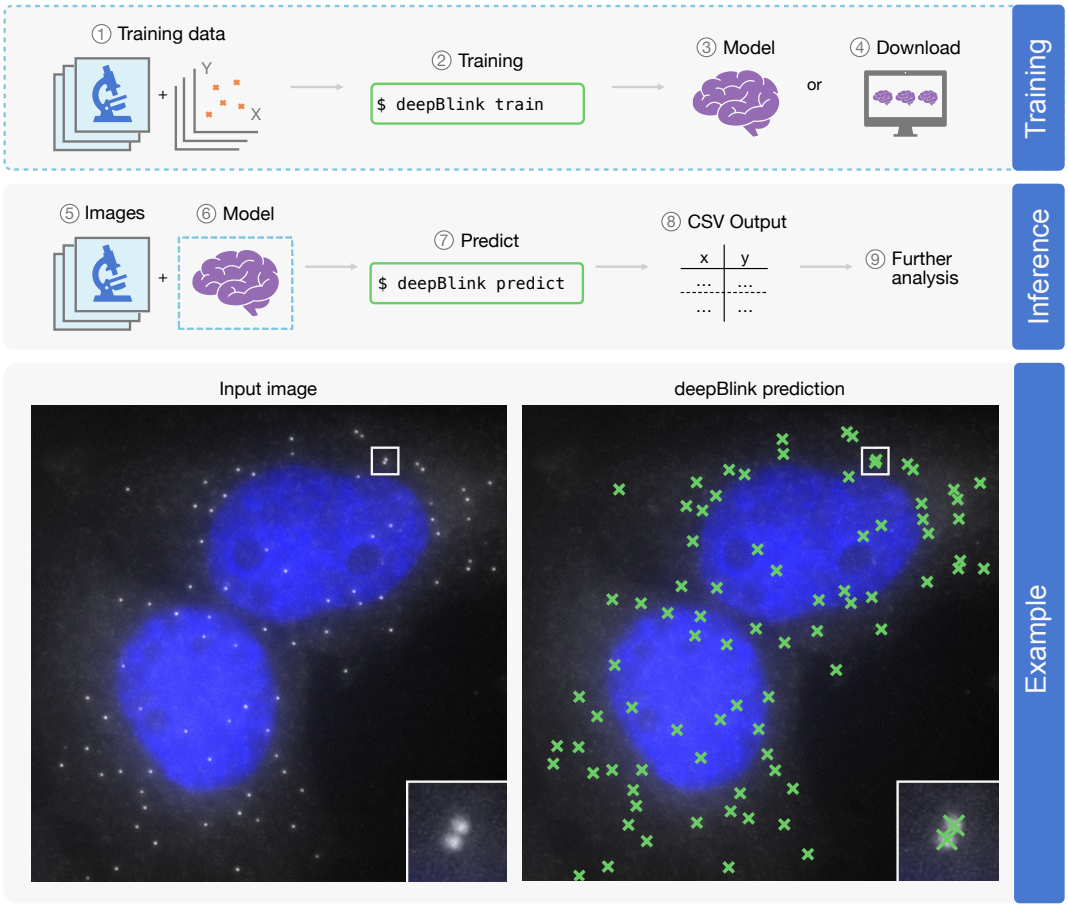



**Figure 3.2:** An overview of deepBlink's functionality. A pre-trained model is required for using deepBlink, which can be obtained by either training from scratch with custom images and coordinate labels (1-3) or by directly downloading it (4). To predict spot coordinates on new data, deepBlink takes in raw microscopy images (5) and the pre-trained model (6) and generates predictions (7), which are saved as a CSV file (8). The output CSV file can be easily integrated into further analysis workflows (9). As an example, a smFISH analysis is shown with blue indicating DAPI staining.

The U-Net-based network architecture [60] used by deepBlink maps raw microscopy data onto smaller regions called "grid-cells". This approach is similar to the region-based bounding-box predictions originally developed in the YOLO architecture [55]. The input xy coordinates are converted into a $n \times n \times 3$ grid, where the first layer indicates whether a grid-cell contains a spot or not, and the second and third layers provide the x and y positions, respectively, relative to the grid-cell. While the size of the grid cells can be varied depending on spot density, a size of $4 \times 4$ pixels in the input image yields good results across all tested datasets (see Figure 3.3). This size was found to strike a nice balance between too sparse labels leading to a class imbalance favoring the "no spots" case, and having multiple spots in one grid cell that

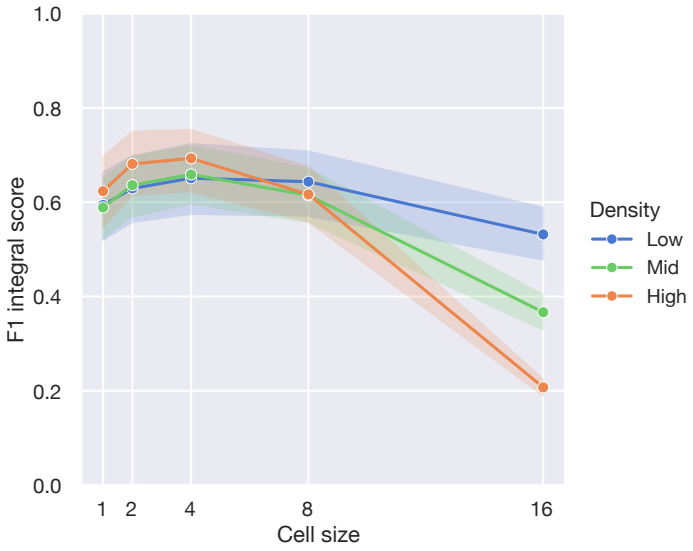



**Figure 3.3:** Relationship between cell size and F1 integral score at low, mid and high spot densities for the Receptor dataset.

cannot be regressed to [89]. Images with higher spot densities perform better even at smaller cell sizes, while larger cell sizes increase the risk of one grid-cell containing multiple spots. In the datasets tested, a cell size of 4 resulted in only around 0.0003% of grid-cells containing more than one spot (Figure 3.5). Each layer in the U-Net architecture contains standard 2D convolutional layers as well as a squeeze and excitation block [39]. This is a form of attention and will be discussed in the context of NLP.

For training, some key components are a small batch size of 2, the AMSGrad stochastic optimizer [90], a large number of 200 training epochs, and a custom loss function. The batch size determines the number of samples seen by the network per weight update. Here, a core limiting factor not warranting any increases is the typical GPU memory size with large image sizes ($512 \times 512$ px). However, smaller batch sizes have been shown to improve a model's generalizability [91]. The AMSGrad [90] optimizer was used which has been reported to fix a convergence issue with Adam-based optimizers. Specifically, AMSGrad uses the maximum of past squared gradients compared to Adam's exponential average. Although not the most significant impact, deepBlink still benefited from improved training stability and slightly enhanced performance. A large number of 200 epochs (i.e. iterations over all training data) could be chosen because of the heavy regularization employed which significantly reduced overfitting. This ensures that most users will not have to manually train for longer periods based on the dataset at hand. Lastly, a custom loss function was employed:

$$L_{\text{class}} = 1 - \frac{2|\hat{y}_{\text{class}} \cdot y_{\text{class}}|}{|\hat{y}_{\text{class}}| + |y_{\text{class}}|}$$

$$L_{\text{loc}} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_{\text{loc},i} - y_{\text{loc},i})^2}{n}}$$

$$L = L_{\text{class}} + 2 \cdot L_{\text{loc}}$$

The dice loss $L_{\text{class}}$ was applied to the first output layer corresponding to the spot probability. Compared with binary cross-entropy, the dice loss generally performs better in class imbalances [92]. The RMSE error $L_{\text{loc}}$, the square root of the MSE penalizing larger errors less severely, was applied to the spot coordinate returning layers. The final loss function, denoted by $L$, was obtained by combining the individual loss functions with a weight of 2 assigned to the RMSE term. This implies that the localization error was given greater importance in the overall loss function.

To measure the impact of these individual components in the network, ablation experiments were conducted, wherein each element was removed to assess its effect on model performance (Table 3.1). The most significant factor in improving model performance was the usage of dice loss. The emphasis on localization loss over detection loss also contributed to higher model performance. This may be due to the network's greater difficulty in regressing spot coordinates compared to detecting spot-containing grid-cells. Another significant factor was the inclusion of the squeeze block, which allows for finer adjustments on critical features by scaling convolutional feature maps [39]. Although other architectural blocks, such as inception [37] and residuals [8], have been used in similar computer vision tasks [93], they did not result in significant performance improvements. Maintaining a constant number of filters (64) across each layer of the network produced better results than the typical increase/decrease in filters for each downsampling/upsampling step, respectively, indicating that higher-level features found in deeper layers were less important for spot detection. This trade-off resulted in the selection of 64 filters for the network. In addition, the reduction of overfitting by spatial dropout layers [33] was reflected in decreased performance upon ablation. Finally, the bottom skip connection may allow for easier gradient flow, similar to residual blocks, if skipped layers do not contribute to overall accuracy.

**Table 3.1:** Ablation table on the smFISH and Receptor datasets showing F1 integral scores (mean $\pm$ standard deviation) and percentage decrease (%-dec) upon removal of singular features. Features were sorted in descending order of mean percentage decrease. The "Replacement" column describes how each feature was replaced.

| | | smFISH | | Receptor | | Mean |
|---|---|---|---|---|---|---|
| | Replacement | Score | %-dec | Score | %-dec | %-dec |
| Full network | | **0.905 $\pm$ 0.145** | | **0.683 $\pm$ 0.309** | | |
| Dice Loss | BCE | 0.806 $\pm$ 0.196 | 10.990 | 0.639 $\pm$ 0.347 | 6.460 | 8.730 |
| RMSE Weight | $1/10$ | 0.856 $\pm$ 0.129 | 5.450 | 0.675 $\pm$ 0.304 | 1.160 | 3.310 |
| Squeeze Block | Remove | 0.893 $\pm$ 0.140 | 1.300 | 0.656 $\pm$ 0.317 | 3.930 | 2.620 |
| Filters $2^n$ | 4 | 0.885 $\pm$ 0.145 | 2.200 | 0.673 $\pm$ 0.301 | 1.450 | 1.830 |
| Spatial Dropout | Dropout | 0.900 $\pm$ 0.143 | 0.610 | 0.666 $\pm$ 0.309 | 2.430 | 1.520 |
| RMSE Weight | $1/2$ | 0.890 $\pm$ 0.135 | 1.670 | 0.678 $\pm$ 0.307 | 0.700 | 1.190 |
| Filters $2^n$ | 5 | 0.890 $\pm$ 0.144 | 1.610 | 0.680 $\pm$ 0.309 | 0.450 | 1.030 |
| Bottom Skip Connection | Remove | 0.895 $\pm$ 0.170 | 1.120 | 0.680 $\pm$ 0.308 | 0.370 | 0.750 |
| Batch Size | 1 | 0.900 $\pm$ 0.148 | 0.570 | 0.677 $\pm$ 0.309 | 0.850 | 0.710 |
| Filter Arrangement | Symmetrical | 0.897 $\pm$ 0.141 | 0.850 | 0.679 $\pm$ 0.309 | 0.560 | 0.710 |
| Convolutional Blocks | Residual | 0.903 $\pm$ 0.147 | 0.210 | 0.675 $\pm$ 0.311 | 1.080 | 0.650 |
| Optimizer | ADAM | 0.894 $\pm$ 0.154 | 1.190 | 0.682 $\pm$ 0.312 | 0.060 | 0.630 |
| L2 Regularization | Remove | 0.896 $\pm$ 0.165 | 1.020 | 0.682 $\pm$ 0.310 | 0.060 | 0.540 |
| Conv2D Activation | ReLu | 0.897 $\pm$ 0.153 | 0.900 | 0.683 $\pm$ 0.312 | -0.010 | 0.450 |
| Convolutional Blocks | Inception | 0.898 $\pm$ 0.137 | 0.740 | 0.683 $\pm$ 0.309 | 0.000 | 0.370 |
| Filters $2^n$ | 7 | 0.903 $\pm$ 0.146 | 0.200 | 0.680 $\pm$ 0.312 | 0.400 | 0.300 |
| Batch Size | 4 | 0.902 $\pm$ 0.150 | 0.330 | 0.682 $\pm$ 0.309 | 0.090 | 0.210 |

Once trained, to compare different models or methods, a metric independent of the loss function is typically used. Here, since the scoring is performed on sets of xy coordinates which has not been done before, a new metric termed the "F1 integral score" was created. The F1 integral score takes into account both spot detection accuracy (i.e. how many spots were detected) as well as spot localization precision (i.e. how close to the ground truth true positive spots are). The score ranges between zero and one with one being perfect prediction and all spots assigned and zero being that no spots have been predicted within a 3px radius of the ground truth. The algorithmic overview of the F1 integral score can be found in Supplemental Algorithm 1, a more intuitive description is below:

- The Euclidean distance ($\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$) is calculated between all coordinate

pairs.

- At a given pixel-cutoff, a linear sum assignment is performed linking one and only one prediction to a ground truth based on the precomputed Euclidean distance.
- The resulting assignments are used to count true positives (tp), false positives (fp), and false negatives (fn).
- These values are used to calculate the F1 score:

$$\text{recall} = \text{tp}/(\text{tp} + \text{fn} + \epsilon)$$

$$\text{precision} = \text{tp}/(\text{tp} + \text{fp} + \epsilon)$$

$$\text{f1} = (2 * \text{precision} * \text{recall})/(\text{precision} + \text{recall} + \epsilon)$$

- This is repeated for a total of 50 cutoffs linearly distributed between 0 and 3 pixels.
- The area below this resulting curve (pixel cutoff vs. F1 score) is calculated using the trapezoidal rule of calculus. This value is normalized by the total area resulting in a score between 0 and 1.

To evaluate deepBlink's performance against previously published methods, a novel metric called the "F1 integral score" was developed that considers both spot detection and localization accuracy (see Methods and Figure 3.4). The F1 integral score ranges from zero to one, with perfect prediction being represented by a score of one and no predicted spot within a 3px radius of the ground truth spot positions represented by a score of zero. Furthermore, the classical F1 score with a 3px cutoff was used to evaluate detection efficiency together with the RMSE to evaluate localization precision. The F1 score measures the accuracy with which true positive spots are detected in a radius of 3px. The RMSE was calculated only for true positive spots and increases as localization precision decreases, with perfect localization being represented by an RMSE of zero.

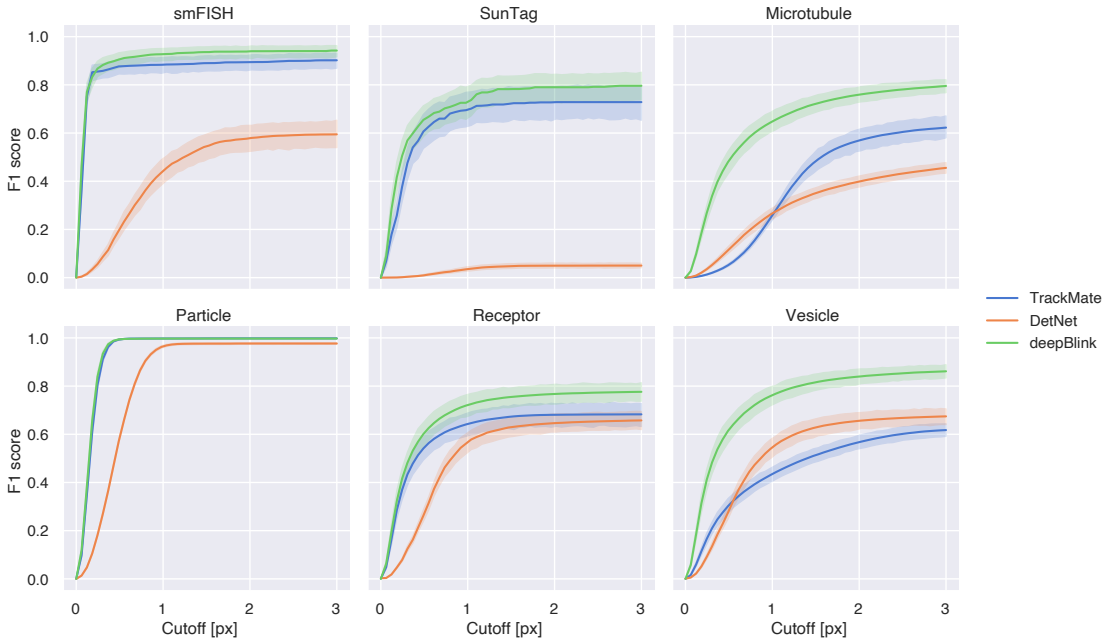



**Figure 3.4:** F1 score as a function of cutoffs on all datasets across and methods. Shaded areas correspond to the confidence interval of 95%.

To assess the performance of deepBlink on a range of spot sizes, shapes, densities, and SNRs, two microscopy datasets were employed that were manually labeled (smFISH [86] and SunTag live cell single particle tracking [87]) and four synthetic datasets (a custom Particle dataset generated using [85] and the Microtubule, Vesicle, and Receptor datasets from the ISBI particle tracking challenge [84]). Figure 3.5 presents sample images from each dataset. To minimize the labor-intensive nature of manual labeling, the number of labeled images required to achieve marginal improvements in performance when more images are added was assessed. As seen in Figure 3.6 A and B, results show that a training set size of about 100 images is sufficient for achieving acceptable prediction quality on datasets with narrow SNR distributions. However, datasets with broader SNR distributions require more labeled training examples to capture the greater variability, as observed in Figure 3.6 A and C.

The performance of deepBlink was compared against a popular classical method and a state-of-the-art deep learning-based method. The Laplacian of Gaussian (LoG) is used by TrackMate [70], which requires the setting of manual thresholds on an image-by-image basis for highly variable datasets. DetNet [83], a more recent deep learning-based method, has been shown to perform well on the synthetic ISBI datasets. Hyperparameters for both benchmarking methods (TrackMate and DetNet) were optimized on every dataset individually (see

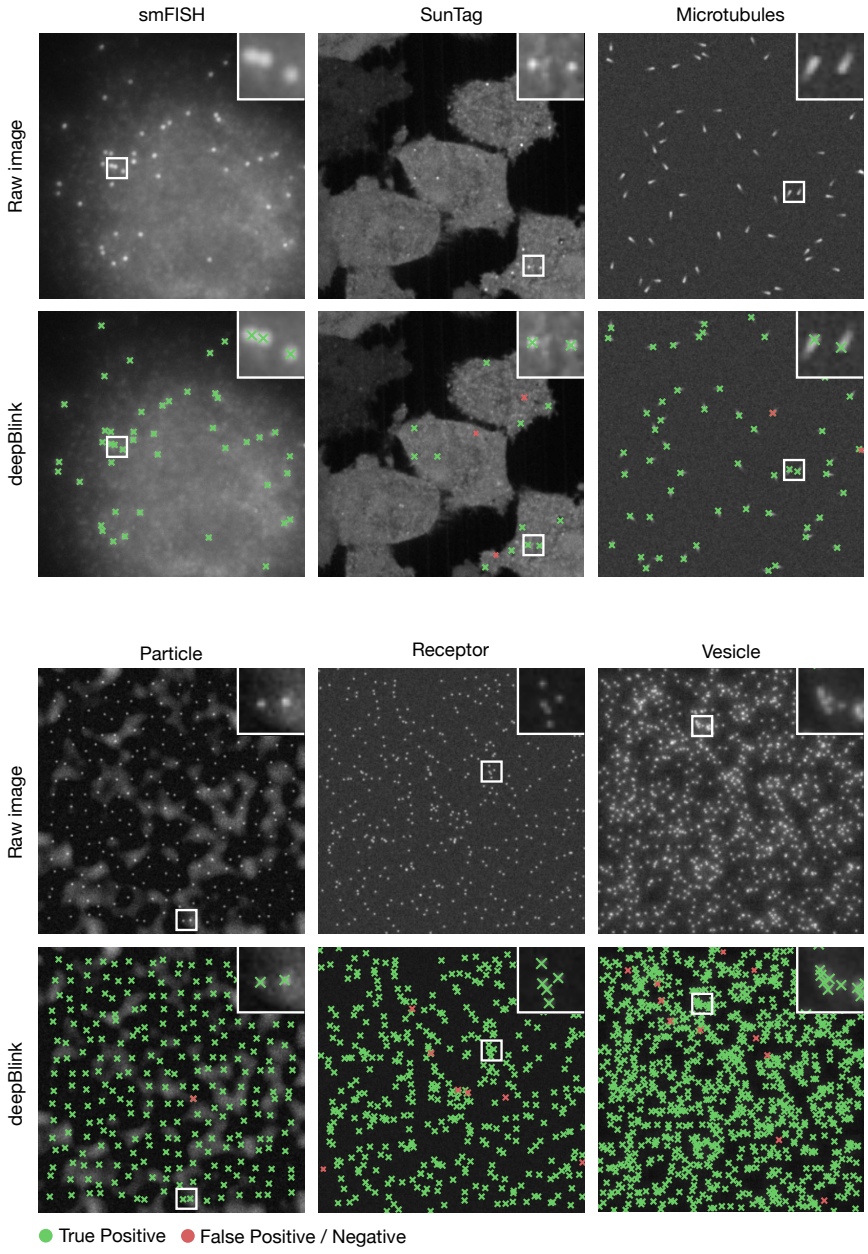



**Figure 3.5:** Representative images for all six datasets with their corresponding prediction.

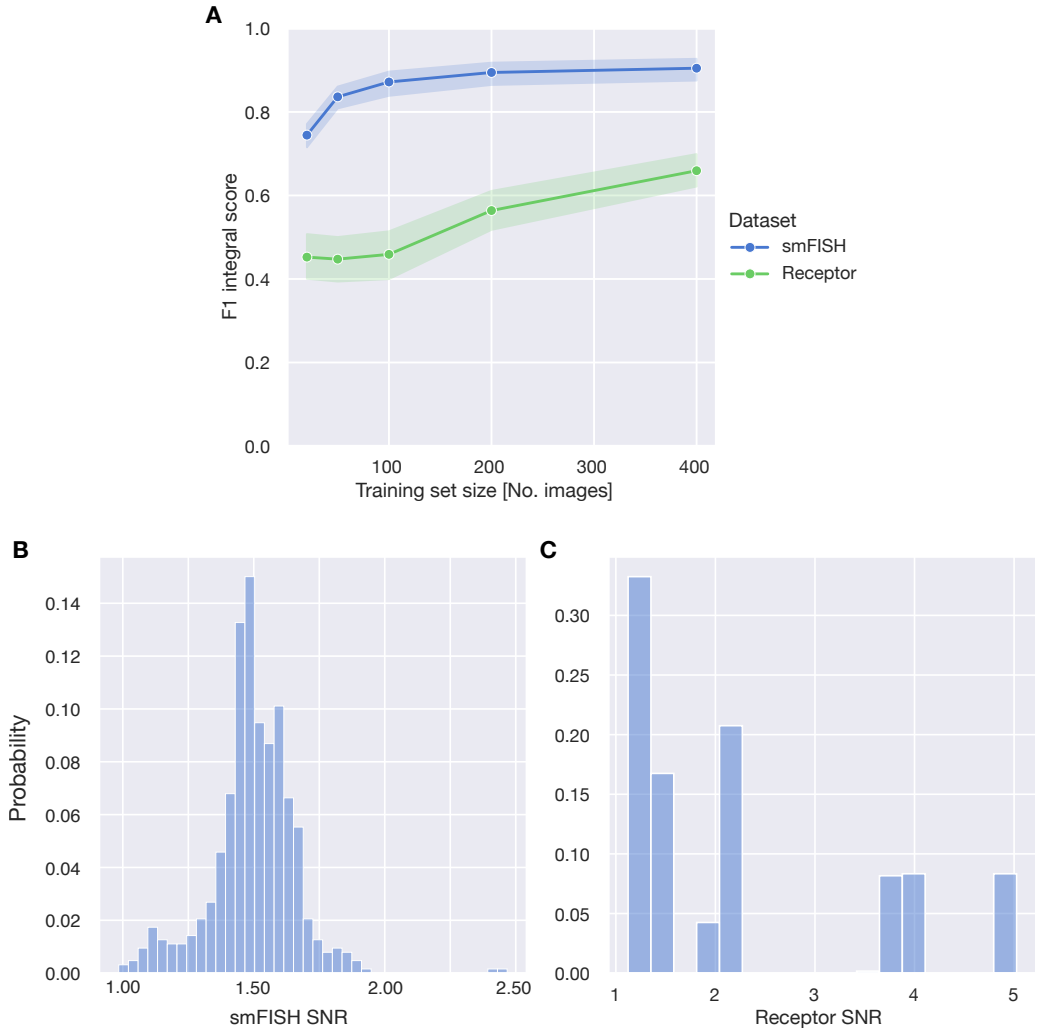



**Figure 3.6:** The impact of dataset size on the performance of the model. **A** displays the correlation between the size of the training set and the F1 integral score on the identical holdout test set, demonstrating this relationship for both the smFISH and Receptor datasets. **B** and **C** present the distribution of SNR throughout the entire dataset for the smFISH and Receptor datasets, respectively.

**Table 3.3:** F1 integral score (mean $\pm$ standard deviation) results for three methods across six datasets. Statistical significance was determined by a one-sided Wilcoxon signed-rank test (deepBlink greater than) with **: p < $1e^{-2}$, ****: p < $1e^{-4}$. Size of test set: 129 smFISH, 105 SunTag, 64 Particle, 240 Microtubule, 240 Receptor, and 240 Vesicle.

| | | TrackMate | | DetNet | | deepBlink |
|---|---|---|---|---|---|---|
| Real | smFISH | 0.865 $\pm$ 0.177 | **** | 0.442 $\pm$ 0.250 | **** | **0.905 $\pm$ 0.145** |
| | SunTag | 0.652 $\pm$ 0.328 | ** | 0.036 $\pm$ 0.040 | **** | **0.712 $\pm$ 0.279** |
| Synthetic | Particle | 0.941 $\pm$ 0.008 | **** | 0.828 $\pm$ 0.014 | **** | **0.944 $\pm$ 0.008** |
| | Microtubule | 0.380 $\pm$ 0.228 | **** | 0.298 $\pm$ 0.149 | **** | **0.637 $\pm$ 0.261** |
| | Receptor | 0.606 $\pm$ 0.344 | **** | 0.517 $\pm$ 0.258 | **** | **0.682 $\pm$ 0.310** |
| | Vesicle | 0.459 $\pm$ 0.198 | **** | 0.520 $\pm$ 0.245 | **** | **0.732 $\pm$ 0.259** |
| | Average | 0.651 $\pm$ 0.214 | | 0.440 $\pm$ 0.159 | | **0.769 $\pm$ 0.210** |

Methods) to improve the fairness of the comparison, while deepBlink parameters remained consistent.

All datasets were analyzed, and it was found that deepBlink significantly outperformed other methods on average (Table 3.3). It was observed that deepBlink demonstrated excellent performance on all datasets for both detection (average efficiency above 85%, Supplementary Table S2) and localization (average error below 0.5 pixels, Supplementary Table S3). The performance of TrackMate, which utilizes LoG filters for detecting and localizing spots, slightly exceeded deepBlink's performance in detection efficiency on the Particle dataset and localization precision on the Receptor dataset. However, deepBlink was found to outperform TrackMate on other particle-like datasets, namely the experimental smFISH and SunTag datasets, presumably because experimental microscopy images have higher image-to-image variability. In contrast, deepBlink outperformed DetNet, a more recent deep learning-based method, across all datasets for both detection and localization. These findings indicate that deepBlink is more accessible for a wide range of applications without the need for dataset-specific tuning. Additionally, fitting a Gaussian function on the original input data initialized by deepBlink's prediction output did not demonstrate any improvement in localization (Supplementary Table S3), further supporting deepBlink's localization precision.

In addition to localization and detection performance, deepBlink was also compared to TrackMate and DetNet in terms of execution time. The execution time of the methods was compared on a set of sample images, and it was found that deepBlink was slightly slower than TrackMate and DetNet (TrackMate: 30 ± 6 ms, DetNet: 253 ± 31 ms, deepBlink: 502 ± 27 ms). However, since TrackMate requires manual adjustment of parameters on an image-by-image basis, the gain in execution speed is offset by the time required for manual parameter

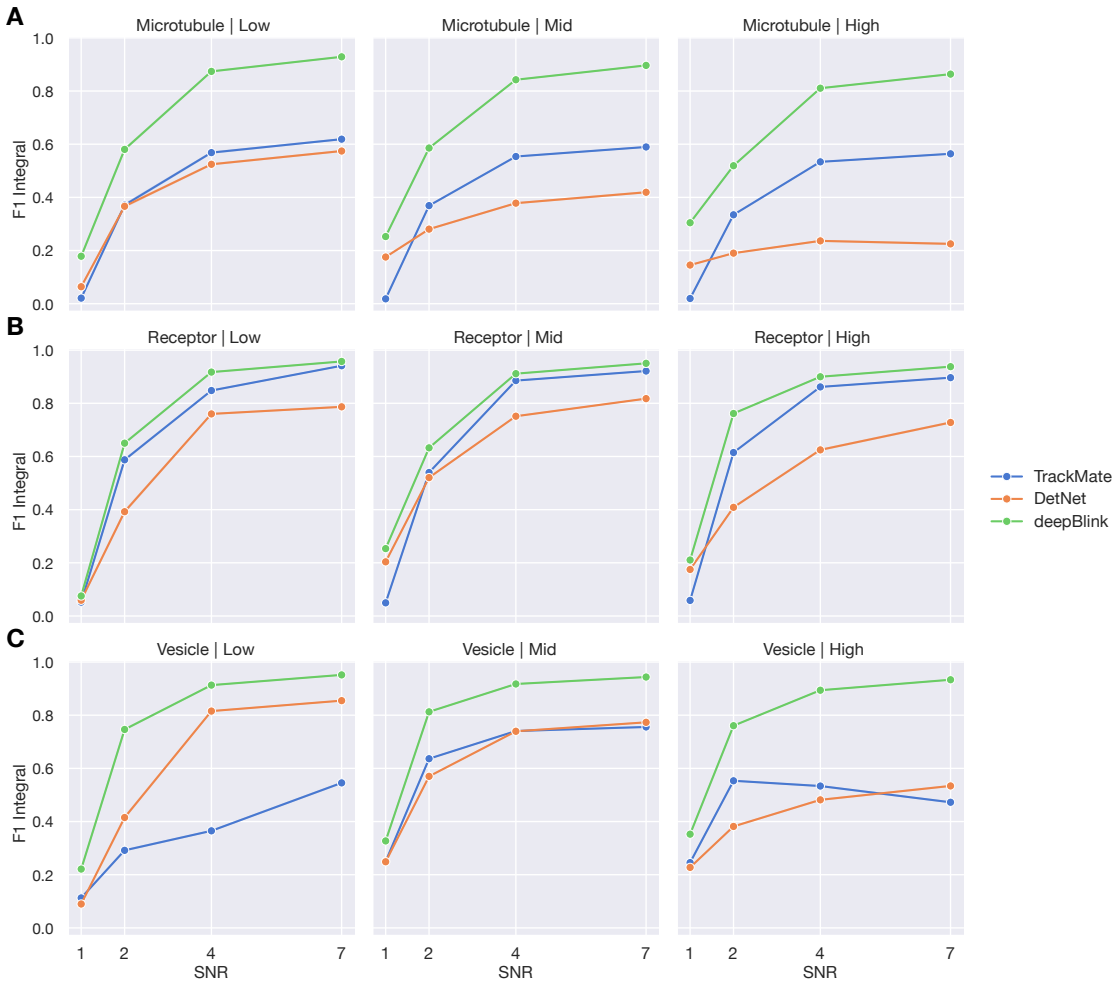



**Figure 3.7:** The F1 integral scores for all three methods were calculated at different spot densities and image SNRs for the **A** Microtubule, **B** Receptor, and **C** Vesicle datasets.

adjustment. On the other hand, DetNet is marginally faster than deepBlink, but its performance is markedly worse.

To investigate the effect of spot densities and SNRs on detection performance, the Microtubule, Receptor, and Vesicle datasets were divided into categories based on official categories from the ISBI particle tracking challenge (see Methods). It was observed that spot density and SNR generally do not affect the performance of deepBlink, except for very low SNRs, where no model is able to perform well (Supplementary Table S4, Supplementary Figure S1, and Figure 3.7).

In summary, deepBlink enables the automated detection and accurate localization of diffraction-

limited spots in a threshold-independent manner. The software comes with an easy-to-use CLI that allows both novice and expert users to efficiently analyze microscopy data. All pre-trained models can be downloaded free of charge, and the software has been benchmarked on six publicly available datasets containing synthetic images, including smFISH and live cell (SunTag) imaging data. These datasets feature a variety of spot sizes, shapes, densities, and SNRs. We demonstrated that deepBlink outperforms all benchmark methods in detection and localization across these datasets. This improved detection efficiency and localization precision can enhance current particle tracking analyses and make image processing more reliable and reproducibleSNRs.

## 3.5 Future developments in spot detection methods

Since deepBlink's publication in 2021, there have been a handful of other methods that perform similar functions but are substantially different to deepBlink:

- **N2V2**: N2V2 [75] provides an innovative new approach to denoise different types of images, compared to previous approaches [76], it does not require a set of high-resolution images for training making it more suitable for biomedical imaging. N2V2 can be combined with traditional, mathematical operator-based spot detection methods, however, subsequent settings of parameters are still prone to wrongful detections in areas of varying intensities as was the case in deepBlink's benchmarking with TrackMate [70].
- **Radial Symmetry-FISH (RS-FISH)**: RS-FISH [94] is an alternative Gaussian fitting method. While improving the detection efficiency and processing speed compared to other methods, there are two key differentiators. First, the continued reliance on thresholds makes runtime detection subjective and prone to errors and negates the time benefits achieved by a faster run-time. Secondly, their comparison with deepBlink did not involve any custom-trained models and excluded benchmarking on the only real data, which negated the key advantage of deepBlink.
- **DeepSpot**: DeepSpot [95] is another neural network-based architecture to enhance spots specifically which aims to perform spot enhancement enabling traditional spot detection downstream. In their benchmarking, deepBlink outperformed DeepSpot on real data.

The biggest usability downside of the current version of deepBlink is the reliance on a pre-trained model specific to the dataset at hand. While different microscope settings and cellular

conditions are generally not an issue, it still poses a reasonable up-front investment to set up a new analysis. There are two key approaches to address this shortcoming. First, a public database should be established for all users to be able to contribute labelled datasets and (if data privacy is an issue) the trained model. Secondly, unsupervised approaches may be explored to not specifically reduce data requirements in general but reduce time spent labelling by simply providing images without and images with spots. Although unsupervised object detection methods have shown great promise, they are still in the early stages of development and have not yet reached the level of performance achieved by supervised approaches. As research in this area continues, unsupervised methods will likely play an increasingly important role in various computer vision tasks enabling an unsupervised future for deepBlink.

## 3.6 Example use cases of an automated pipeline

With the advent of modern fluorescence microscopy, large amounts of data can be created with ease. To extract quantitative information, these images have to be analyzed objectively. In the past, manual analysis was often the only option due to the lack of sophisticated computational techniques and the complexity of the images themselves. When performed in a fully automated fashion, it enables researchers to analyze large quantities of data in a shorter amount of time and also eliminates the subjective bias that can be introduced by manual analysis.

In the upcoming chapter, the diverse range of applications for deepBlink will be explored in automated pipelines for various cellular microscopy datasets. All subsequent examples will be using the Python Application Programming Interface (API). However, deepBlink has also been made accessible via a straightforward CLI and a Konstanz Information Miner (KNIME) node (for prediction purposes only).

### 3.6.1 Single molecule FISH analysis in cells

A prototypical experiment requiring analysis of imaging data is smFISH. The principle of smFISH involves the use of short (typically 20-50nt long) oligonucleotide probes labeled with fluorophores that hybridize to a complementary target sequence of interest (see Appendix C). Each target Ribonucleic Acid (RNA) is visualized as a discrete fluorescent spot, and can thus provide information on RNA abundance and localization. By designing probes specific to different RNA species using commercial solutions or open-source alternatives such as eFISHent (see Appendix B), smFISH can be used to simultaneously visualize and quantify multiple RNA

molecules in a single sample. The technique is highly sensitive and allows for the detection of low-abundance transcripts, making it a valuable tool for studying gene expression and regulation at the single-cell level. Typical tasks in smFISH image analysis are cellular segmentation, and detection of RNA foci as well as potential marker segmentations when combined with Immunofluorescence (IF). An example analysis workflow relying on mathematical based thresholding (here Otsu [96]) for cellular and marker segmentation as well as a custom-trained deepBlink model for spot detection is shown in Figure 3.8. By lazily parallelizing this analysis, a typical experimental dataset consisting of 200+ images can be run in less than 10 minutes compared to multiple hours in a more manual approach.

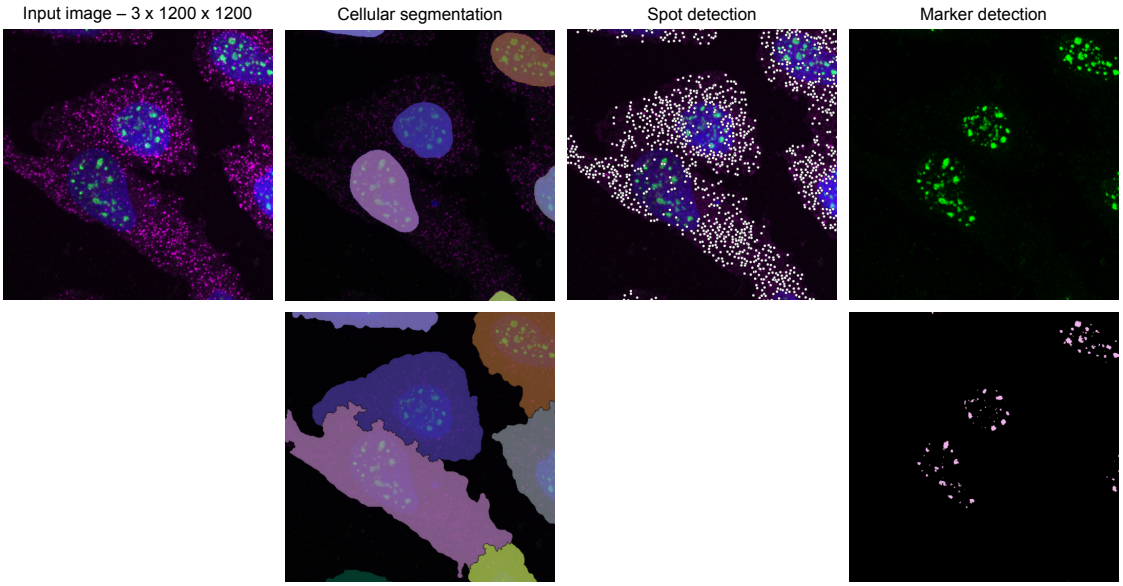



**Figure 3.8:** An example image of an extensible FISH-IF analysis workflow including both nuclear and cytoplasmic segmentation, spot detection, and segmentation of other channels.

### 3.6.2 Live cell SunTag analysis

Another common experiment using live cells is the analysis of stellar explosion SUperNova tag (SunTag) data. For SunTag, a repeated array expressing short GCN4 peptides is inserted into the mRNA of interest and - once translated - can be bound by co-expressed fluorescently labeled antibodies. Simultaneously, mRNAs are visualized in a different fluorescent color using the MS2-MCP system [97]. Once expressed, the SunTag system can be used to track the movement and localization of target proteins on a sub-cellular scale, providing insights into molecular dynamics and protein function (for more detail see Appendix A). This technique has been used to study a variety of cellular processes including translation [98] and mRNA trafficking [87]. For automated analysis of SunTag movies (Figure 3.9), only the cytoplasm

of cells is segmented using Cellpose [71]. Next, in both MS2 and GCN4 channels, spots are detected and tracked over time using two custom-trained deepBlink models. One key advantage of using deep learning-based models is that they do not detect any false positive spots in the high-intensity regions of the nucleus by selectively training on cytoplasmic spots only. Thereby, traditionally difficult-to-segment nuclei do not have to be excluded. Lastly, spot tracks are colocalized to provide insights into active translation sites.

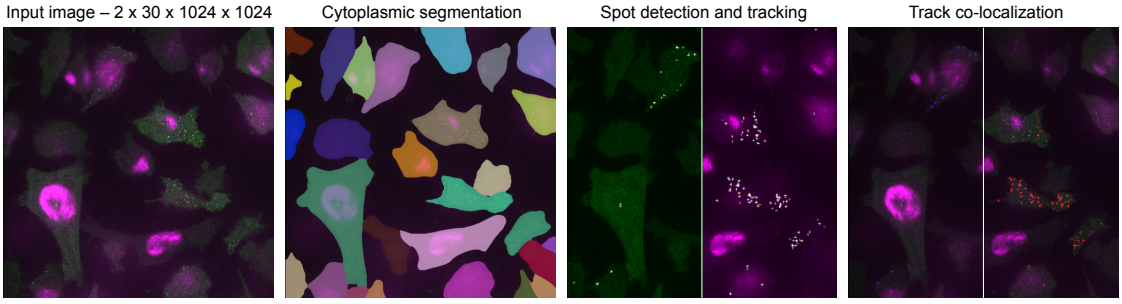



**Figure 3.9:** An example image illustrating the standard SunTag analysis workflow including cytoplasmic segmentation, spot detection, and tracking in both the MS2 and GCN4 channel, followed by track co-localization.

### 3.6.3 Single molecule FISH analysis in brain tissue

An expansion of traditional cellular smFISH involves the analysis of *Drosophila melanogaster* brains. Here, rather than a single (typically projected) 2D plane, a 3D image is required to capture the 3D organization of brains. Typical tasks involve the segmentation of very densely packed cells, spot detection of smFISH signal, and marker segmentation to more easily differentiate neurons. As shown in Figure 3.10, cellular segmentation is also done using Cellpose. Any non-deep learning-based tools are generally unable to differentiate closely packed cells from one another. deepBlink is employed for spot detection by iterating over all z-stacks followed by Non Maximum Supression (NMS). The basic idea behind NMS is to combine all detected spots in the z-stacks after examining if they are overlapping. The suppression of overlapping spots is done by keeping only the candidate with the highest intensity. A simple binary segmentation model can be trained to detect markers across all stacks, which can be further optimized by using pre-trained models on large image collections, thereby reducing the need for extensive training labels. Here, an Inception achitecture [37] pre-trained on ImageNet [10] was fine-tuned on manually labeled segmentation maps. By integrating these components, researchers can analyze a considerable number of fly brains each day making sample preparation and imaging the biggest bottleneck.

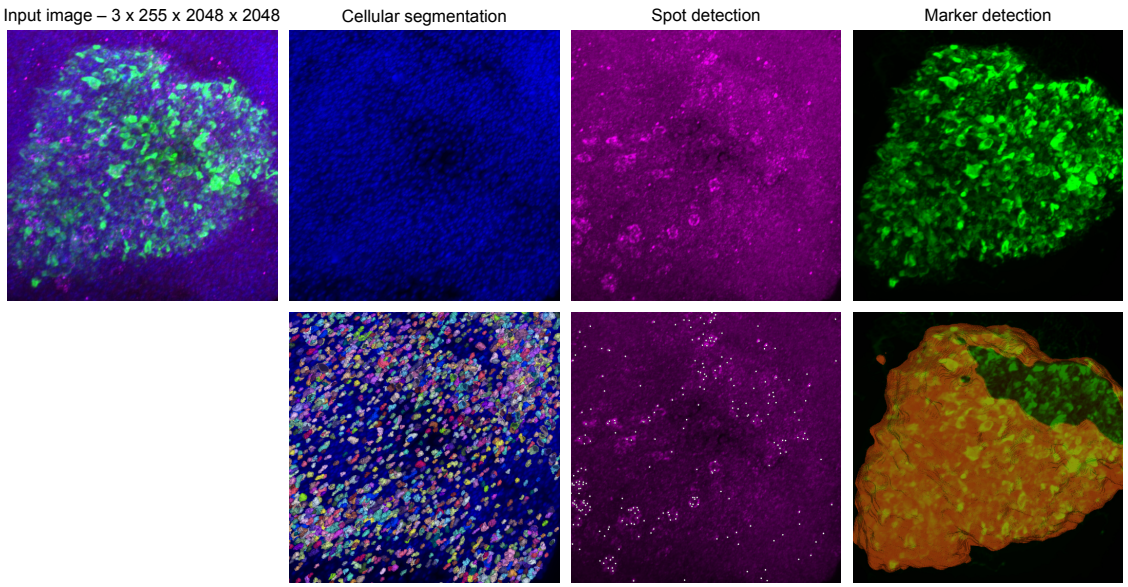



**Figure 3.10:** An overview of an exemplary workflow to analyze *Drosophila melanogaster* brains including cellular segmentation, spot detection, and marker detection.

Taken together, modern tools typically relying on deep learning are enabling researchers to analyze images more efficiently and in a less biased fashion. This has led to various breakthroughs, where researchers can now study large quantities of images with more confidence and accuracy [99, 100]. In addition, these tools have opened up new possibilities for identifying subtle patterns and relationships that were previously undetected, allowing for deeper insights into complex biological processes. As technology continues to advance and deep learning models become more sophisticated, the potential for image analysis in research will only continue to grow.

# 4

# Protein-protein interaction prediction

## 4.1 Protein structure prediction

The protein folding problem is one of the most challenging problems in computational biology involving the prediction of the 3D structure of a protein given only its amino acid sequence. The tertiary structure of a protein is essential for understanding its function and facilitates designing specific drugs. The challenge arises from the fact that the number of possible conformations a polypeptide can adopt is astronomically large, making an exhaustive search computationally infeasible. Additionally, the energy landscape of protein folding is complex, with multiple local minima, making it difficult to find the global energy minimum that represents the correct folded structure. The problem is further complicated by the fact that experimental techniques for determining protein structure, such as X-ray crystallography and cryogenic electron microscopy, are time-consuming, expensive, and not always feasible for all proteins. Therefore, developing accurate and efficient methods for predicting protein structure from sequence data is a major research goal in computational biology.

Natural Language Processing (NLP) refers to the field of computer science concerned with giving computers the ability to understand text and spoken words similar to the way human beings can. Since protein sequence can be interpreted in a similar way replacing words with amino acids, advancements in the field of NLP have opened up new possibilities for analyzing and predicting protein structure.

## 4.2 Natural language processing

NLP is a highly versatile field with numerous applications. Examples of these applications include spelling error identification, which can flag errors such as theses, and machine translation, which utilizes an encoder-decoder architecture to translate text from one language to another [101]. Here, the source language is typically encoded into a fixed-length vector

representation, similar to the one used in U-Net models, and then decoded to generate a translation in the target language. Advancements in NLP have led to significant improvements in speech-to-text translation, making it more accurate and reliable. As a result, individuals who may struggle with reading or have limited literacy skills can now benefit from modern technology [102].

NLP models can also be combined with computer vision approaches to solve complex problems such as visual question answering and image captioning. In visual question answering, an image and a natural language question are given to the model, which then provides an answer to the question [103]. In image captioning, an NLP model generates a textual description of an image. This technology has been particularly helpful in allowing visually impaired individuals to access news articles, social media posts, and other visual content that would have been difficult to understand without assistance, helping to bridge the gap in accessibility for these individuals [104].

### 4.2.1 Tokenization: from text to number

The first part in most NLP applications involves converting data into a numerical format for models to understand. This process called tokenization breaks down raw text into smaller meaningful units, called tokens. Tokens can be words, subwords, or characters, depending on the specific task and the approach taken. The tokenization process typically involves several steps. First, text is split into individual words or subwords, known as the tokenization unit. For example, the sentence "I love playing basketball" could be split into "I", "love", "playing", and "basketball" when the tokenization unit is set to words. In some cases, subword tokenization can be used to handle words not present in the vocabulary or to better capture the underlying meaning of the text. Here, words are split into smaller subunits based on their frequency of occurrence. For example, the word "basketball" might be split into "basket" and "ball", with each subword representing a separate token. After the tokenization unit is chosen, the next step is to clean and normalize the text. This involves removing punctuation, special characters, and other noise not relevant to the task at hand. For example, converting text to lowercase and replacing words with their canonical forms (e.g. "am" to "are"). Once the text has been cleaned and normalized, the final step is to map each token to a unique integer ID corresponding to its position in the vocabulary.

### 4.2.2 Embeddings: creating a vector representation

As part of the actual network, the embedding component of an NLP network refers to the process of representing words or phrases in a continuous, high-dimensional space. Compared with tokenization which divides text into individual units, embedding maps these tokens to high-dimensional vectors, enabling them to be processed by NLP algorithms. One of the most basic approaches to embedding is one-hot encoding. In this method, each word in the vocabulary is assigned a unique vector of all zeros except for a one in the position corresponding to the word's index in the vocabulary. One-hot encoding is simple and efficient but suffers from the curse of dimensionality since the resulting vectors are very high-dimensional and sparse. More sophisticated approaches to embedding involve learning the embeddings during training. Here, the embeddings are represented as vectors in a continuous space and are updated using backpropagation, so that similar words are mapped to similar regions in the embedding space. The resulting embeddings capture semantic relationships between words, such as synonyms and antonyms.

Some early models such as Word2Vec [48], which consists of a shallow, two-layer neural network use these embeddings to learn associations between words. This allowed for simple word substitutions and analyses of semantic and syntactic relationships, where for example, "King" - "Man" + "Woman" produces a vector representation close to "Queen".

### 4.2.3 Core NLP architectures

Recurrent Neural Networks (RNNs) are a class of neural networks designed for sequential data processing that have a feedback loop that enables information to persist over time. The core idea behind RNNs is to use the same set of weights contained in one "cell" across all time steps to compute the output of the network (see Figure 4.1). At each time step, the input is fed into the network, along with the previous hidden state $A$. The input can either be a new token $x_t$ or the previous output $y_{t-1}$ (e.g. in the case of text generation). The hidden state is a vector that represents the network's "memory" of previous inputs and is updated iteratively. Computation, therefore, takes into account information across past events. The most common type of an RNN cell is the Long-term short-term memory (LSTM) cell [105], which is designed to address the problem of vanishing gradients that occur in traditional RNNs. LSTMs use three gates termed input, forget, and output to control the flow of information through the cell, allowing it to selectively remember or forget information over time. CNNs have also been adapted for use in NLP tasks such as sentiment analysis [106], text classification [107], and others by treating text as a 1D input signal.

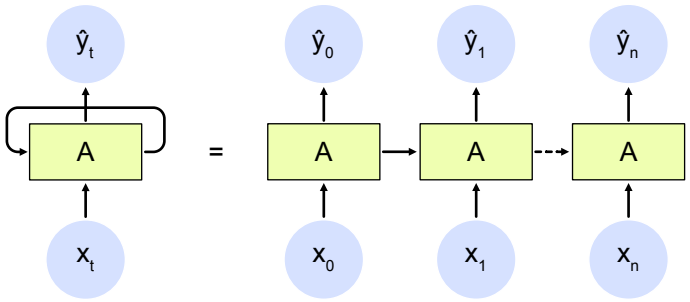



**Figure 4.1:** An overview of the functionality of a Recurrent Neural Network. Adapted from: colah.github.io and stanford.edu

While RNNs have been applied successfully in many areas, they have some inherent limitations such as slow computation due to repeated application of the same weights for each time step and the inability to capture longer-term dependencies and future events, resulting in the need to explore alternative options. As a solution, Transformer architectures have emerged as a newer and widely-adopted alternative. They use an attention mechanism introduced by Vaswani et al. in 2017 [108] to focus on the relevant parts of the input sequence during encoding and decoding. This enables them to capture long-range dependencies in text more efficiently than RNNs or LSTMs. Specifically, the Transformer architecture is based on the concept of self-attention. The self-attention layer takes a sequence of embeddings as input and calculates a set of attention weights for each input token based on its relationship with all the other tokens in the sequence. These attention weights are used to compute a weighted sum of the embeddings, which is passed through a feedforward neural network to produce the output for that token. The Transformer architecture also includes residual connections and layer normalization helping to alleviate the vanishing gradient problem.

### 4.2.4 Modern examples of NLP networks

Modern architectures primarily rely on transformers due to their advantages described above. Also, most architectures are improved using unsupervised pre-training strategies on a large corpus of text such as Wikipedia [109] or parts of the Internet [110]. The following are some of the most popular methods:

- Bidirectional Encoder Representations from Transformers (BERT) [103] uses bidirectional training to generate pre-trained language representations. Specifically, it employs masked language modeling and next-sentence prediction to generate contextualized word embeddings by unsupervised training on a large corpus of text.

- Embeddings from Language Models (ELMO) [111] is another transformer-based architecture that uses a deep bidirectional language model to generate contextual word embeddings. These embeddings capture both syntactic and semantic features of words. This is done using a task called "language modeling", where it tries to predict the next word in a sentence given the previous words.
- Generative Pre-trained Transformer (GPT) [112, 110] also tries to predict the next word in a sentence using autoregression. Compared with ELMO, GPT is uni-directional only allowing it to take into account the past tokens. Additionally, the changes in architecture makes GPT suitable for language generation as compared to tasks like text classification for BERT.

Other modern transformer-based architectures include XLNet [113], RoBERTa [114], and T5 [115]. These models build on the success of BERT and GPT and use various techniques such as masked language modeling and transfer learning to improve performance on a range of tasks. One of the most prominent recent examples is ChatGPT [15] which revolutionized the field of NLP by achieving state-of-the-art performance on a variety of tasks, including language modeling, question answering, and dialog generation.

> ChatGPT is a state-of-the-art language model that uses deep learning techniques to generate human-like responses to textual inputs. Trained on a massive amount of text data, ChatGPT is able to understand the context of a conversation and generate coherent and relevant responses. It works by using a transformer architecture, which allows it to handle long-range dependencies in text and learn relationships between words and phrases. ChatGPT can be fine-tuned for specific tasks, such as question-answering or text summarization, making it a versatile tool for natural language processing applications. Overall, ChatGPT is a powerful tool for generating human-like language and advancing the field of natural language processing.

This paragraph above was generated by ChatGPT after asking "Describe chatGPT in one paragraph" without any editing.

### 4.2.5 AlphaFold 2

AlphaFold 2 (AF2) is a transformer-based model predicting protein folding structures with high accuracy [116]. It was developed by a team of researchers at DeepMind in London,

and its performance has been described as a breakthrough in the field of computational biology [117]. The network comprises two main stages: the trunk, which processes the inputs using a novel block called Evoformer to produce an array that represents a processed Multiple Sequence Alignment (MSA) and an array that represents residue pairs; and the structure module, which introduces an explicit 3D structure for each residue of the protein.

Specifically, the input amino acid sequence is embedded as a MSA and pair representation before iteratively being passed through the Evoformer block. Here, the network uses new mechanisms to exchange information that enables direct reasoning about spatial and evolutionary relationships. The structure module then operates on the pair representation and the MSA representation from the trunk. This module is also run iteratively and outputs the predicted atom coordinates and per-residue confidence scores. Extensive parametrization is used to ensure that structures are physically feasible.

## 4.3 Predicting the effect of point mutations on bZIP domain-containing protein-interaction

A key question in human genetics and evolutionary biology is how mutations in different genes combine to alter phenotypes. Efforts to systematically map genetic interactions have mostly made use of gene deletions [118]. However, most genetic variation consists of point mutations of diverse and difficult-to-predict effects [119]. A comprehensive view of how point mutations in two genes combine to alter a molecular phenotype is therefore required.

Using a new sequencing-based protein interaction assay - Deep-sequencing-based Protein Fragment Complementation (deepPCA) - the effects of more than a million pairs of point mutations were measured [120]. deepPCA is a method that uses the principles of protein fragment complementation to detect Protein-Protein Interactions (PPIs) in living cells. In this approach, both interaction partners are each fused to a complementary fragment of a methotrexate-resistant Dihydrofolate (DHF) reductase reporter. When the two partners interact with each other, they bring the two fragments of the reporter protein in close proximity, leading to the reconstitution of a functional reporter protein. Since cells are grown in a selective medium containing methotrexate, only cells containing the reconstituted reporter protein are resistant to methotrexate and can grow. By paired-end deep sequencing, the changes in relative populational input and output frequencies can be quantified. The resulting PPI score is a normalized logarithmic Fold Change (logFC) score quantifying the interaction strength between both proteins.

Basic Leucine Zippers (bZIPs) domain-containing proteins (bZIP proteins) are a family of eukaryotic transcription factors that play a crucial role in regulating gene expression. They are comprised of two main parts: the leucine zipper and the DNA binding domain. The DNA binding domain is typically made from a high proportion of basic amino acids, such as arginine and lysine, which confer a positive charge to the protein and enable it to bind to DNA. The leucine zipper is a conserved stretch of amino acids rich in leucine forming a coiled-coil structure, which allows the protein to homo- or hetero-dimerize (reviewed in [121]). bZIP proteins are involved in a variety of cellular processes, including cell growth [122], differentiation [123], and stress response [124]. They regulate the expression of target genes by binding to specific DNA sequences in the promoter region and recruiting other proteins to form transcriptional complexes [125]. There are numerous bZIP proteins in mammals, including Maf, cAMP response element-binding protein (CREB), CCAAT-enhancer-binding proteins (C/EBP), and Activator protein 1 (AP-1). Each protein has a unique DNA-binding specificity and regulates a distinct set of target genes. For example, C/EBP proteins are involved in adipocyte differentiation [123], while AP-1 proteins are involved in the response to environmental stressors such as UV radiation and oxidative stress [124].

Due to their conserved nature and abundant interaction partners, bZIP proteins are a prime candidate for investigating PPI. Specifically, a library of 54 wild-type human bZIP proteins was screened in yeast against all point mutants in the leucine zipper domain (20 amino acids $\times$ 40 positions $\times$ 54 proteins $\approx$ 38'000 mutants). This resulted in 1.6 million data points (sequence pairs and their associated PPI score) usable for analysis.

### 4.3.1 Using AlphaFold 2 to predict the effect of point mutations

The simplest first approach to solving the problem of predicting PPI was to use AF2 to assist in predicting the structure of each point mutant and use a simple linear model to combine predicted free energy values. Specifically, all point mutants of the bZIP protein JunD were fed through the network and refined. As can be seen from Figure 4.2A, an overlay of all JunD mutants shows that the $\alpha$ carbons do not differ largely from each other.

Next, to predict interaction scores between two example proteins of JunD and Fos, the structures were predicted together using AlphaFold-Multimer [126]. After prediction, the unrelaxed (direct output from the AlphaFold prediction model) and relaxed (minimization of "unrelaxed" models with OpenMM+Amber forcefield, this is an optional step of the AlphaFold pipeline) structure models were scored using Rosetta's InterfaceAnalyzer [127]. This was done for a total of 2500 sequence pairs and plotted against the experimentally derived logFC

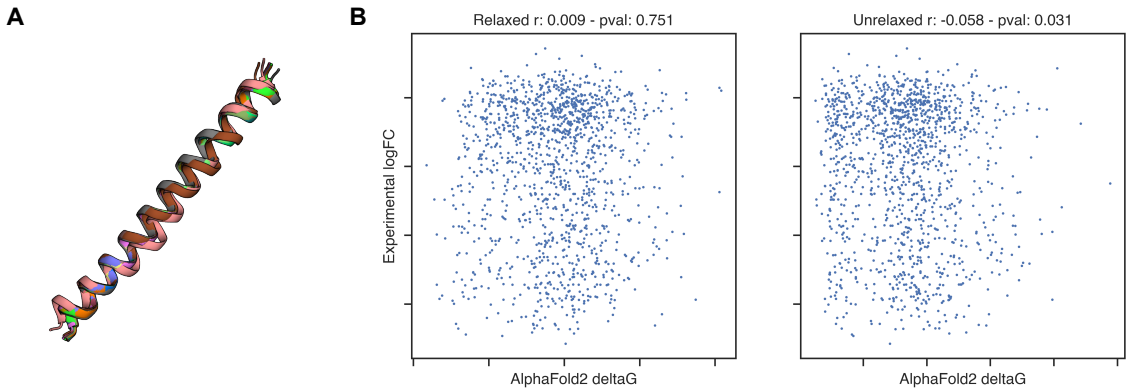



**Figure 4.2: A** An overlay of all point mutant predictions by AlphaFold 2 of protein JunD. **B** The correlation of delta G values extracted from the AlphaFold 2 prediction compared to the ground truth experimental data.

in Figure 4.2B. As can be seen from the figure, predicted delta G values do not correlate well with experimental scores confirming the initial expectation after the single mutant prediction. Recent publications have also shown that AF2 is unable to predict the effect of point mutations [128, 129].

### 4.3.2 Custom architecture search

To find a suitable model architecture, a variety of sweeps were formed which are subsequently described. The general architecture that was conceived is shown in Figure 4.3A. Two sequences with tokenized amino acid sequences are passed into a network, embedded to form a vector representation, and passed through a first set of layers with shared weights. Subsequently, these sequence representations are combined and passed through a second set of layers yielding a final output prediction score. The model was penalized with a MSE loss with L1 norm against the mean logFC across experimental replicates:

$$MSE_{L1} = \frac{1}{n}\sum(\frac{|y|}{\sum|y|} - \frac{|\hat{y}|}{\sum|\hat{y}|})^2 \tag{4.1}$$

All models were trained for 10 epochs with the Adam optimizer. Sweeps were performed screening for different learning rates, model width (number of filters/heads), and model depth (number of repetitive blocks). The score measured is the R2 score on the same validation dataset. The R2 score, also known as the coefficient of determination, is a measure frequently used to measure the goodness of fit of a regression model. It represents the proportion of the variance in the dependent variable (i.e., the predicted score) that is explained by the

independent variables (i.e., the ground truth). The R2 score takes values between 0 and 1, where 0 indicates that the model explains none of the variability in the dependent variable, and 1 indicates that the model explains all of the variability in the dependent variable. A higher R2 score indicates a better fit of the model to the data. The R2 score can be calculated as follows:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y})^2} \tag{4.2}$$

where $\bar{y}$ is the mean of the actual values. The numerator represents the sum of squared residuals (i.e., the difference between actual and predicted values), and the denominator represents the total sum of squares (i.e., the variance of the actual values). The mathematical upper R2 score bound is difficult to define globally but since the experiment was performed across multiple batches and replicates, sample means shared between batches can be scored against each other. Depending on a threshold set for input read count, the expected maximum R2 score lies around 0.85. Low read counts result in a higher variability and therefore make the measured logFC less reliable.

Different core layer architectures were tested ranging from traditional RNNs, over modern Transformers to CNNs. Unsurprisingly, as can be seen from Figure 4.3B, more modern Transformer and CNN architectures are significantly better than more dated RNN architectures. However, Transformers on average perform worse than simple CNNs. This suggests that the interaction between bZIP proteins can be efficiently captured simply by looking at local dependencies rather than including more long-range interactions. All subsequent sweeps were therefore performed using a standard CNN architecture.

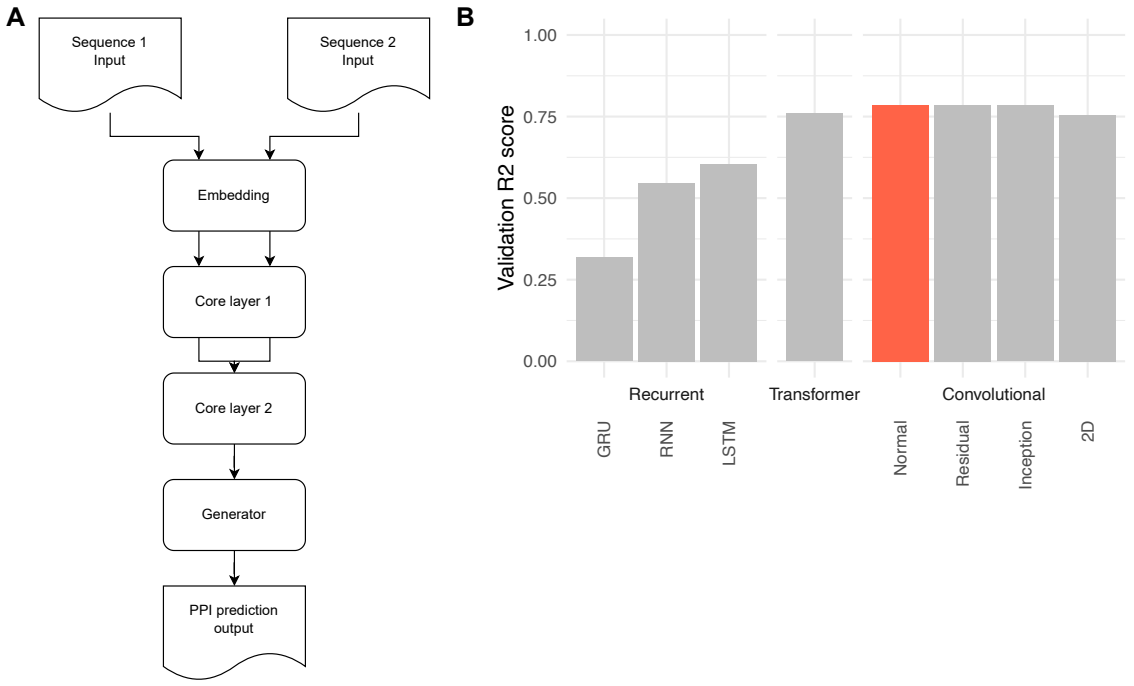



**Figure 4.3:** **A** The general architecture of the protein-protein interaction prediction model, takes two sequences as input and passes them through multiple layers leading to a single-valued output. **B** Results from an architectural sweep comparing Recurrent Neural Networks, Transformers and, convolutional approaches. The highest validation R2 score is plotted for each sweep. Higher values are better with the red column denoting the highest overall score.

Next, the type of embeddings were compared to one another. When running a sweep over four different methods:

- One-hot previously described in Chapter 4.2.2 and illustrated in Figure 4.4
- Positional which adds basic information on the current sequence position to learned token embeddings
- AminoAcid which works similarly to one-hot but instead of binary values uses a principle component analysis describing physicochemical properties of the current amino acid
- Transformer uses the original sin/cos positional and token embeddings described in Viswani et al. [108]. An example is shown in Figure 4.4

As is evident from Figure 4.4B, simpler methods perform better with one-hot taking a slight lead. Preliminary experiments using a Transformer architecture and embedding layer did not improve the score either.

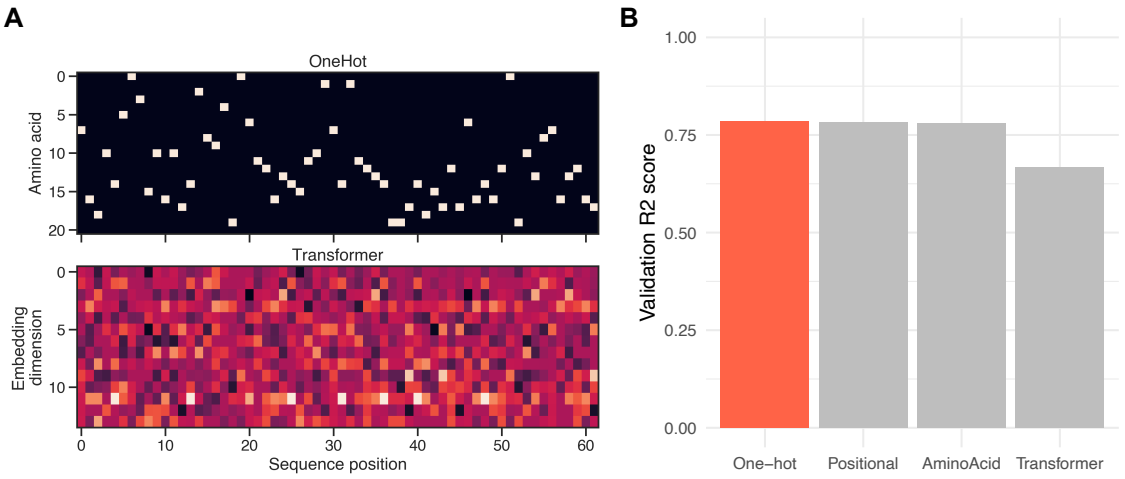



**Figure 4.4: A** The comparison of how one-hot encoding (top) and a learned embedding (bottom) return a sequence.**B** Results from an input encoding sweep comparing simpler one-hot and more complicated learned embeddings. The highest validation R2 score is plotted for each sweep. Higher values are better.

Since the interaction dataset is very detailed, it can be split in different ways (see Figure 4.5). To explore different splits and therefore different levels of information the model can see, five split methods were selected:

- Row which splits all data points randomly
- ID which keeps a mutation at a given position grouped. For example, all interaction data points collected for JunD with a Leucine mutation in position 24 will only appear in the train or validation set
- Mutation takes this a step further by keeping the same mutation and positional but joining them between mutant proteins. In the previous example, all Leucine mutations at position 24 will stay grouped.
- Position groups all mutations at a specific position and split them
- Protein groups the entire protein with all mutations and positions and splits them

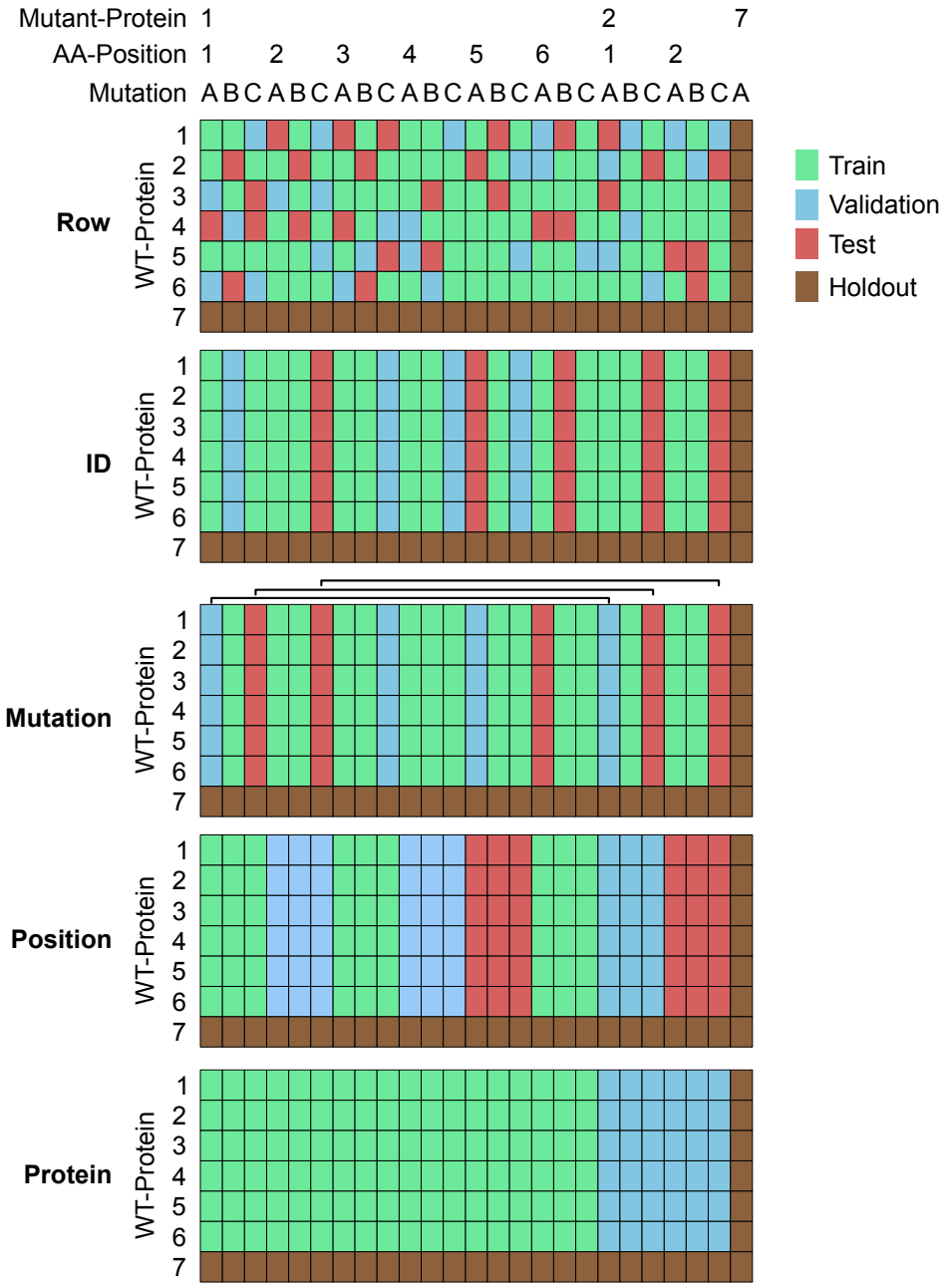



**Figure 4.5:** An overview of the different methods for splitting data ranging from more fine-grained splitting of all data points (Row) to the coarsest protein-level split (Protein).

These splits were performed with 5-fold cross-validation with averages of run-replicates being plotted. An additional factor to consider is the number of reads in the input sample. As is shown in Figure 4.6, the protein split routinely performs worst. However, all other methods of splitting achieve similar scores between k-folds. This suggests that the model can predict

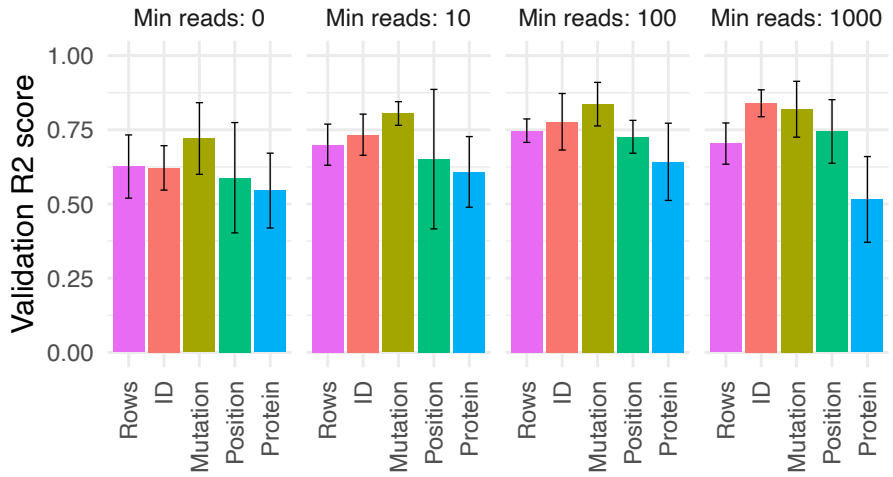



**Figure 4.6:** Results from a dataset split sweep comparing different strategies of partitioning data and filtering based on the total number of input reads across experimental replicates. The model architectures and hyperparameters were kept consistent. The average and standard deviation of validation R2 scores are plotted for a k=5 fold split. Higher values are better.

the effect of different positions and mutations within proteins but struggles to generalize to unseen protein sequences. Additionally, the increase in minimum read count improves validation performance. However, higher minimum input read counts discards parts of the data making generalizability even harder (as can be seen in the right-most protein split score).

Lastly, the best model's predictions were plotted against the experimental ground truth (Figure 4.7). The best model consisted of 140 convolutional kernels with a kernel size of 15, stacked three layers deep. The rather large kernel size coincides with two heptad repeats in the bZIP proteins meaning that longer-range dependencies are generally not required. To prevent overfitting, the model employed a 20% dropout rate and an L2 regularization with a weight of $5e^{-7}$. A reasonable performance in the test dataset suggests that the model can generalize well to unseen data and the model can capture underlying patterns and relationships between the input features and the output variable. This indicates that the model is likely to generalize well on new, unseen data. All plots show a deviation from a straight linear correlation around a model prediction of zero. Taking into account input read counts (see Figure 4.8), low mean read counts mainly accumulate in this area. This suggests that it is less of the model's fault for wrongful predictions but primarily for lower-quality experimental data.

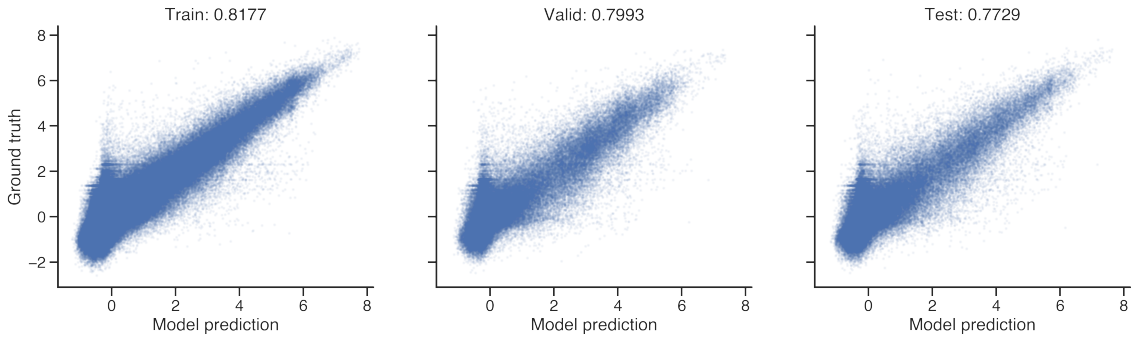



**Figure 4.7:** Correlation and R2 values showing the best model's accuracy on the train, validation, and test dataset splits using the "Mutation" method and a minimum input read count of zero.

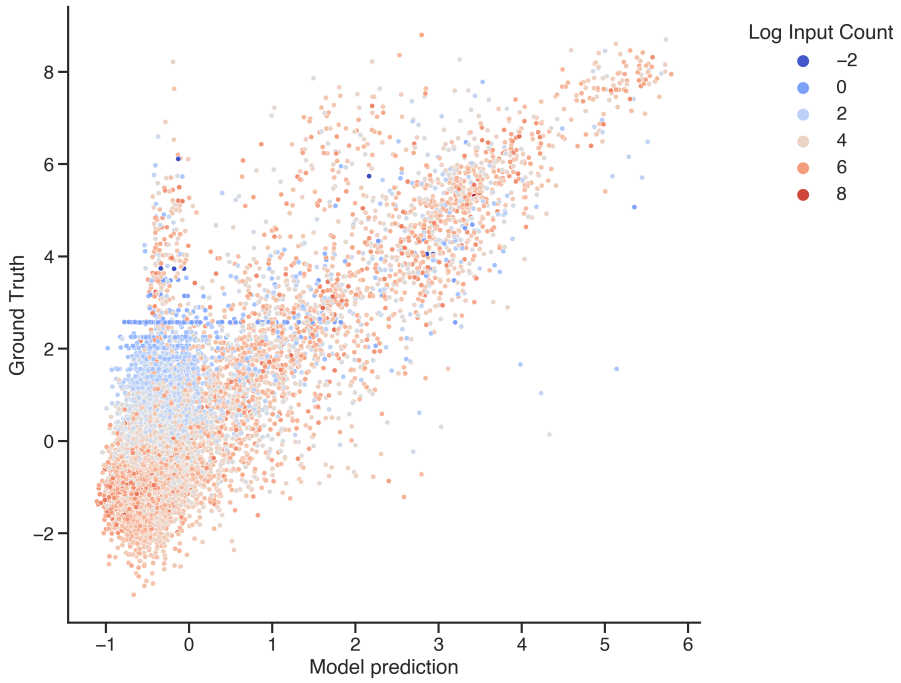



**Figure 4.8:** Correlation and R2 values showing the best model's accuracy on a random sample of 10'000 data points using the "Mutation" split method. The logarithm of average input reads is shown as hue with all zero inputs represented as -2.

Taken together, after conducting extensive experimentation and testing, a suitable CNN model architecture was identified that proved to be effective for predicting PPI. The model was designed to take two sequences as input and map them onto a predicted interaction score, based on their one-hot encodings. The best-performing model demonstrated excellent performance during training, however, it was found to struggle with generalizing to new proteins that it had not seen before, resulting in lower R2 scores during testing. This suggests that

the model was not able to effectively capture the complex patterns and relationships present in new protein sequences. To improve the model's generalization performance, further research and experimentation may be required. This could involve more extensive regularization or data augmentation techniques to prevent overfitting, or exploring different model architectures or hyperparameters to better capture the underlying patterns and relationships in the data. Additionally, it may be useful to collect more diverse training data taken from a larger array of proteins to improve the model's ability to generalize.

### 4.3.3 Model introspection

Despite having a model reasonably suited for making predictions, one challenge with DL models has always been to understand what they are learning [130]. To address this problem, a few approaches have been taken. First, weights associated with convolutional kernels were inspected before and after training. Figure 4.9A shows the change in weights of the first convolutional layer during the training process. For visualization purposes, a model with a shorter kernel size of 7 was selected (Validation R2: 0.741). As can be seen, most kernels have stayed relatively static but some (e.g. the fourth kernel in the first row) have had weight changes. Due to the one-hot encoding employed, a Softmax can be applied revealing which amino acids are weighted most heavily. The Softmax function converts these weights into a probability distribution over the different amino acids in the sequence. As shown in Figure 4.9B, this can be visualized as a sequence plot to illustrate the model's "interest" at different positions. Compared to computer vision problems, where kernel visualizations result in human-understandable image features, most of these plots are inherently difficult to comprehend. To further illustrate this problem, the left-most plot shows the kernel of a trained kernel (standard deviation above 0.1), the central plot shows an untrained kernel, and the right-most plot uses randomly generated probabilities.

Next, two models were trained only differing in their random seed for initialization. Both reached an almost identical R2 score of 0.783±0.001. The activation maps from the first convolutional layer were compared by feeding all wild type bZIP proteins through. As CNN kernels are positionally invariant, traditional correlation methods cannot be used for comparison. Instead, a Representational dissymilarity matrix (RDM) can be constructed to quantify the similarity between the feature maps generated. The RDM is constructed by computing the Euclidean distance between the activations. The resulting matrix provides a way to visualize the similarity relationships among the representations of each wild type bZIP protein. Figure 4.9C shows RDMs between untrained (weights before the first optimization step) and trained versions. As can be seen, there is more similarity in the same network as compared to

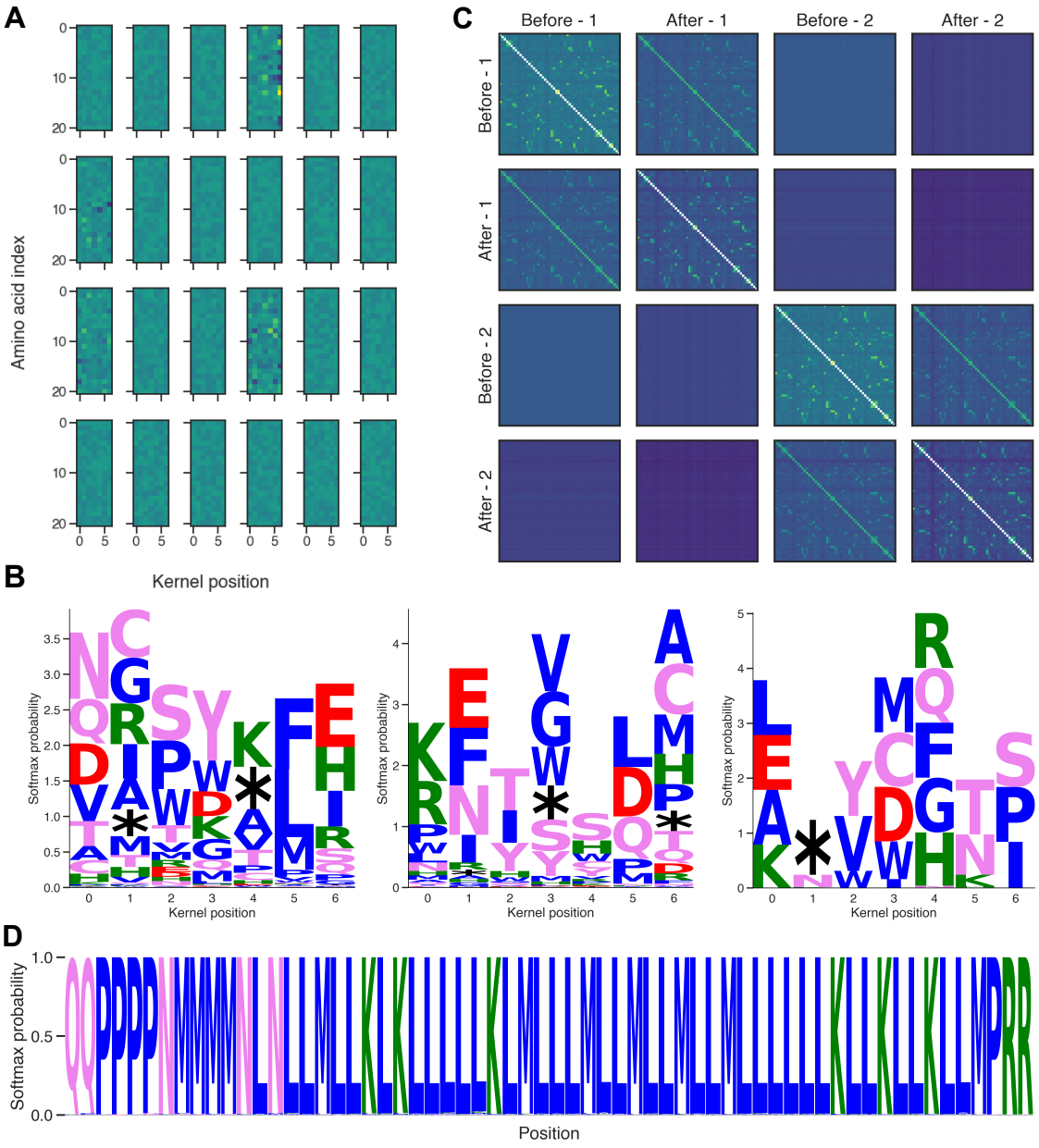



**Figure 4.9:** Model introspections performed on the first convolutional layer of a well-trained model: **A** Kernel weight changes (before and after training). **B** Softmax (100) probabilities for a trained, untrained, random kernel (from l.t.r.). **C** Representational dissimilarity matrix of the activation maps from two models trained the same way but with different weight initialization. Plotted are all wild type basic leucine zipper proteins. **D** Input optimization of the first layer.

across different networks. This suggests that different representations can be found separate from one another rather than all networks converging on a single "ground truth" source.

Lastly, the best artificial input sequence was predicted by minimizing the sum of squares of the activation map (for a single protein). Initially, a random input sequence was Softmax normalized to ensure only one amino acid was present at each position. The sequence was then optimized using gradient descent to find the sequence that was most activated by the given layer. Figure 4.9D shows that most positions in the sequence are occupied by leucine, which is expected as bZIP proteins primarily rely on leucines to form dimers through coiled-coil interactions. However, it is important to note that this particular protein would be unable to form a leucine zipper in real life. The lack of a suitable "protein grammar" makes it difficult to parameterize this input optimization and does not permit an easy, experimentally verifiable search.

Taken together, the methods described show that even though the network can predict PPI reasonably well, understanding how it learns remains difficult. One key challenge is finding features for humans to interpret while having a limited understanding of protein grammar. Nonetheless, the use of visualization techniques such as activation maps and RDMs provide insights into the network's ability to identify interaction partners. Additionally, the observation that different networks do not converge on similar representations suggests that there may be multiple solutions to the PPI prediction problem. Future work could explore ways to better understand the learned features of the network, for example through adversarial attacks [131], while also incorporating biological domain knowledge to guide the learning process.

# 5

# Summary and Outlook

Modern microscopy methods have significantly contributed to improving our understanding of biological systems by enabling the observation of structures and processes at increasingly higher resolutions. However, with the increase in resolution comes an increase in data, which poses a challenge for effective analysis and interpretation. To overcome this challenge, there is a need for better analysis methods. Traditional mathematical operators are often used for data analysis in microscopy studies, but have difficulties in enabling automated analysis and can introduce thresholding bias. Furthermore, these methods can be time-consuming and require significant expertise, which can limit their utility for large datasets. As a result, there has been a growing interest in the use of DL tools for more efficient and accurate analysis of microscopy data. Recently, several DL tools have been developed to enable automated single-molecule analyses in microscopy studies. Two such tools are Cellpose and deepBlink. Cellpose is a DL-based software that can segment cells with high accuracy. deepBlink, on the other hand, is a DL-based software that can detect spots without the need for thresholds. It has been shown to outperform traditional detection methods and can be used to analyze a wide range of biological processes involving spot detection or tracking in live-cell data. Overall, the use of DL tools in microscopy studies has the potential to revolutionize the way we analyze and interpret data. By enabling more efficient and accurate analysis, these tools can help researchers to gain new insights into biological systems and advance our understanding of complex biological processes.

Additionally, an investigation was carried out on the impact of point mutations on interactions between proteins belonging to the bZIP family. For the quantitative measurement of PPI, a sequencing-based protein assay (deepPCA) was employed, which generated over a million interaction scores of point mutations. An attempt was made to utilize AF2 for predicting the structure; however, it proved to be unsuccessful. Instead, a custom DL-based model was developed, which takes both amino acid sequences as input and returns a single interaction score. The selected architecture yielded high accuracy, although it encountered challenges while dealing with low-quality input data and did not exhibit good generalization ability to unseen proteins. Introspection of the results proved to be very difficult due to our lack of innate

understanding of protein grammar. These proof-of-concept experiments have demonstrated the inherent advantages and drawbacks of DL models. The future applications of such models could be facilitated for drug screenings where proteins could be designed to specifically target other proteins, once data has been generated for a larger array of protein families.

Deep learning has shown great promise in biomedical research, particularly in image analysis and PPI prediction as described above. However, the successful integration and application of deep learning in these domains are still subject to several key difficulties which are discussed below in more detail.

**Difficulty 1: Limited availability of high-quality data** One of the primary difficulties in biomedical research is obtaining large amounts of high-quality data for deep learning models. Privacy concerns, small sample sizes, and high variability in the data make it difficult to generate the amount of data necessary to train deep learning models. This issue can be addressed by improving unsupervised techniques that utilize vast amounts of data generated directly without the need for labels. Similarly, models can also be trained on a combination of real and synthetic data.

**Difficulty 2: Complex and heterogeneous data** Biomedical data, particularly in the realm of image analysis, can be incredibly complex and heterogeneous. Variations in imaging settings, contrast, and tissue characteristics can make it challenging to design deep learning models that can effectively handle this variability and generalize well to new samples. As such, there is a need for the development of more capable models that can better adapt to these complexities. One approach to addressing this challenge is to incorporate domain knowledge into model development which can provide a better understanding of the underlying biological processes, which can inform the design of more robust and effective models. Another strategy is to leverage transfer learning techniques, which involve using pre-trained models as a starting point for developing new models. Transfer learning allows for the transfer of knowledge and features learned from one dataset or task to another, which can greatly improve the performance of deep learning models.

**Difficulty 3: Interpretability and transparency** Deep learning models are often viewed as "black boxes" that are difficult to interpret or explain, making it challenging to understand why a certain prediction was made or to identify potential sources of bias in the model. This difficulty is particularly relevant in biomedical research, where the interpretability of the model's decision can be critical to understanding analyses or improving patient outcomes. One so-

lution to this difficulty is to develop models with more transparent architectures that enable the identification of the model's decision-making process. An example of such difficulties was described in the attempt to better understand the PPI model but ultimately lacked in understanding protein grammar.

**Difficulty 4: Integration with existing workflows and technologies** Deep learning models can be computationally intensive and require specialized hardware, which can be a challenge in biomedical research settings. For instance, segmenting a single *Drosophila melanogaster* brain using Cellpose [71] can require up to 10 GPU minutes, which translates to more than an hour on a CPU. To overcome this challenge, it is essential to develop lightweight models that can run on lower-end hardware or leverage cloud computing resources. Although integrating deep learning models into existing workflows and technologies can be difficult, developing open-source tools and frameworks that support deep learning integration can ease the process. Researchers may be accustomed to specific workflows, which can make adapting to new systems both a technological and pedagogical investment. Therefore, it is important to weigh the upfront investments required against the potential time savings in analyses and improved system reliability.

**Difficulty 5: Ethical considerations** Finally, the use of deep learning models in biomedical research raises important ethical considerations, such as privacy, fairness, and potential harm. Addressing these issues requires careful consideration and the establishment of appropriate safeguards when designing and applying deep learning models in this context. Ethical considerations are highly relevant when dealing with patient data, and models should be developed with a patient-centric approach that prioritizes ethical considerations.

In conclusion, deep learning has shown great promise in biomedical research, but several key difficulties must be addressed for its successful integration and application in tasks such as image analysis and PPI prediction. Addressing these difficulties requires interdisciplinary collaboration between computer scientists, biologists, and ethicists to develop solutions that advance biomedical research while addressing the challenges that come with it.

Deep learning has the potential to revolutionize biological research by providing powerful tools for analyzing and interpreting complex biological data. The ability of deep learning algorithms to learn complex patterns and relationships in data makes them well-suited for a wide range of applications in biology, including image analysis, genomics, drug discovery, and personalized medicine. One of the most promising applications of deep learning in biology is the analysis of medical images. Deep learning algorithms have been used to analyze images

from a variety of imaging modalities, including microscopy [71], magnetic resonance imaging [132], and computer tomography scans [133], with impressive results. These algorithms can detect patterns in the images that may not be visible to the human eye and can be used to diagnose diseases [134], monitor disease progression [135], and evaluate the efficacy of treatments [136].

Deep learning is also being applied to genomics research, where it is being used to analyze large-scale genomic data sets. These data sets include genomic sequences [17], gene expression data [137], and epigenomic data [138], among others. Deep learning algorithms can be used to analyze these data sets and identify patterns and relationships that may be relevant to disease diagnosis, drug discovery, and personalized medicine. In drug discovery, deep learning can be used to predict the properties of small molecules, including their binding affinities to proteins [139], their pharmacokinetic properties [140], and their toxicity profiles [141]. These predictions can help researchers identify promising drug candidates and optimize their properties for clinical use.

Overall, deep learning has the potential to transform the field of biological research by enabling the analysis of complex data sets and the discovery of new insights into biological systems. As the field of deep learning continues to advance, we will likely see even more exciting applications of these powerful algorithms in biology and medicine.

# Bibliography


[1] Waymo. https://waymo.com/.

[2] Siri. https://www.apple.com/siri/.

[3] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. Nature, 550(7676):354–359, October 2017. Number: 7676 Publisher: Nature Publishing Group.

[4] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 575(7782):350–354, November 2019. Number: 7782 Publisher: Nature Publishing Group.

[5] Piyabute Fuangkhon. An incremental learning preprocessor for feed-forward neural network. Artificial Intelligence Review, 41(2):183–210, February 2014.

[6] Simon Haykin, Simon S. Haykin, and SIMON AUTOR HAYKIN. Neural Networks: A Comprehensive Foundation. Prentice Hall, 1999. Google-Books-ID: bX4pAQAAMAAJ.

[7] William James. The principles of psychology. New York : Holt, 1890.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, Las Vegas, NV, USA, June 2016. IEEE.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg,

and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, January 2015. arXiv:1409.0575 [cs].

[11] 10 years later, deep learning 'revolution' rages on, say AI pioneers Hinton, LeCun and Li. https://venturebeat.com/ai/10-years-on-ai-pioneers-hinton-lecun-li-say-deep-learning-revolution-will-continue/, September 2022.

[12] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: an overview. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 8599–8603, Vancouver, BC, Canada, May 2013. IEEE.

[13] Afshan Latif, Aqsa Rasheed, Umer Sajid, Jameel Ahmed, Nouman Ali, Naeem Iqbal Ratyal, Bushra Zafar, Saadat Hanif Dar, Muhammad Sajid, and Tehmina Khalil. Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review. Mathematical Problems in Engineering, 2019:e9658350, August 2019. Publisher: Hindawi.

[14] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep Learning based Recommender System: A Survey and New Perspectives. ACM Computing Surveys, 52(1):1–38, January 2020. arXiv:1707.07435 [cs].

[15] Introducing ChatGPT. https://openai.com/blog/chatgpt.

[16] Livemint. ChatGPT's meteoric rise: 100 million users in two months. https://www.livemint.com/technology/tech-news/chatgpts-meteorical-rise-100-million-users-in-2-months-11677997670518.html, March 2023. Section: Technology.

[17] Xu Min, Ning Chen, Ting Chen, and Rui Jiang. DeepEnhancer: Predicting enhancers by convolutional neural networks. In 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 637–644, December 2016.

[18] Nathaniel D Heintzman and Bing Ren. Finding distal regulatory elements in the human genome. Current Opinion in Genetics & Development, 19(6):541–549, December 2009.

[19] Haoyang Zeng and David K. Gifford. Predicting the impact of non-coding variants on DNA methylation. Nucleic Acids Research, 45(11):e99, June 2017.

[20] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Deep Retinal Image Understanding. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, Lecture Notes in Computer Science, pages 140–148, Cham, 2016. Springer International Publishing.

[21] Morteza Amini, Mir Mohsen Pedram, AliReza Moradi, and Mahshad Ouchani. Diagnosis of Alzheimer's Disease Severity with fMRI Images Using Robust Multitask Feature Extraction Method and Convolutional Neural Network (CNN). Computational and Mathematical Methods in Medicine, 2021:5514839, April 2021.

[22] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks, October 2015. arXiv:1412.0767 [cs].

[23] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nature Neuroscience, 21(9):1281–1289, September 2018. Number: 9 Publisher: Nature Publishing Group.

[24] Tanmay Nath, Alexander Mathis, An Chi Chen, Amir Patel, Matthias Bethge, and Mackenzie Weygandt Mathis. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. Nature Protocols, 14(7):2152–2176, July 2019. Number: 7 Publisher: Nature Publishing Group.

[25] Louis K Scheffer, C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, Kazunori Shinomiya, Jeremy Maitlin-Shepard, Stuart Berg, Jody Clements, Philip M Hubbard, William T Katz, Lowell Umayam, Ting Zhao, David Ackerman, Tim Blakely, John Bogovic, Tom Dolafi, Dagmar Kainmueller, Takashi Kawase, Khaled A Khairy, Laramie Leavitt, Peter H Li, Larry Lindsey, Nicole Neubarth, Donald J Olbris, Hideo Otsuna, Eric T Trautman, Masayoshi Ito, Alexander S Bates, Jens Goldammer, Tanya Wolff, Robert Svirskas, Philipp Schlegel, Erika Neace, Christopher J Knecht, Chelsea X Alvarado, Dennis A Bailey, Samantha Ballinger, Jolanta A Borycz, Brandon S Canino, Natasha Cheatham, Michael Cook, Marisa Dreher, Octave Duclos, Bryon Eubanks, Kelli Fairbanks, Samantha Finley, Nora Forknall, Audrey Francis, Gary Patrick Hopkins, Emily M Joyce, SungJin Kim, Nicole A Kirk, Julie Kovalyak, Shirley A Lauchie, Alanna Lohff, Charli Maldonado, Emily A Manley, Sari McLin, Caroline Mooney, Miatta Ndama, Omotara Ogundeyi, Nneoma Okeoma, Christopher Ordish, Nicholas Padilla, Christopher M Patrick, Tyler Paterson, Elliott E Phillips, Emily M Phillips, Neha Rampally, Caitlin Ribeiro, Madelaine K Robertson, Jon Thomson Rymer, Sean M Ryan, Megan Sammons, Anne K Scott, Ashley L Scott, Aya Shinomiya, Claire Smith, Kelsey Smith, Natalie L Smith, Margaret A Sobeski, Alia Suleiman, Jackie Swift, Satoko Takemura, Iris Talebi, Dorota Tarnogorska, Emily Tenshaw, Temour Tokhi, John J Walsh, Tansy Yang, Jane Anne Horne, Feng Li, Ruchi Parekh, Patricia K Rivlin, Vivek Jayaraman, Marta Costa, Gregory SXE Jefferis, Kei Ito, Stephan Saalfeld, Reed George, Ian A Meinertzhagen, Gerald M Rubin, Harald F Hess, Viren Jain, and Stephen M Plaza. A con-

nectome and analysis of the adult Drosophila central brain. eLife, 9:e57443, September 2020. Publisher: eLife Sciences Publications, Ltd.

[26] Sandeep Ravindran. Five ways deep learning has transformed image analysis. Nature, 609(7928):864–866, September 2022. Bandiera_abtest: a Cg_type: Technology Feature Number: 7928 Publisher: Nature Publishing Group Subject_term: Microscopy, Imaging, Mathematics and computing, Computational biology and bioinformatics.

[27] Joshua I. Raji and Christopher J. Potter. The number of neurons in Drosophila and mosquito brains. PLOS ONE, 16(5):e0250381, May 2021. Publisher: Public Library of Science.

[28] Andong Wang, Qi Zhang, Yang Han, Sean Megason, Sahand Hormoz, Kishore R. Mosaliganti, Jacqueline C. K. Lam, and Victor O. K. Li. A novel deep learning-based 3D cell segmentation framework for future image-based disease detection. Scientific Reports, 12(1):342, January 2022. Number: 1 Publisher: Nature Publishing Group.

[29] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, January 2018. arXiv:1603.07285 [cs, stat].

[30] Kunihiko Fukushima. Cognitron: A self-organizing multilayered neural network. Biological Cybernetics, 20(3):121–136, September 1975.

[31] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4724–4733, Honolulu, HI, July 2017. IEEE.

[32] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, March 2015. arXiv:1502.03167 [cs].

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(56):1929–1958, 2014.

[34] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1):60, July 2019.

[35] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks, January 2018. arXiv:1608.06993 [cs].

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–9, June 2015. ISSN: 1063-6919.

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision, December 2015. arXiv:1512.00567 [cs] version: 3.

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. arXiv:2010.11929 [cs].

[39] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-Excitation Networks, May 2019. arXiv:1709.01507 [cs].

[40] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 249–256. JMLR Workshop and Conference Proceedings, March 2010. ISSN: 1938-7228.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, February 2015. arXiv:1502.01852 [cs].

[42] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, Proceedings of the 30th International Conference on Machine Learning, volume 28 of Proceedings of Machine Learning Research, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[43] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs].

[44] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions, April 2017. arXiv:1610.02357 [cs] version: 3.

[45] Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method, December 2012. arXiv:1212.5701 [cs].

[46] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12(61):2121–2159, 2011.

[47] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations, 2015.

[48] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, September 2020. arXiv:1905.11946 [cs, stat].

[49] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners are Image-Text Foundation Models, June 2022. arXiv:2205.01917 [cs].

[50] Ling Dai, Liang Wu, Huating Li, Chun Cai, Qiang Wu, Hongyu Kong, Ruhan Liu, Xiangning Wang, Xuhong Hou, Yuexing Liu, Xiaoxue Long, Yang Wen, Lina Lu, Yaxin Shen, Yan Chen, Dinggang Shen, Xiaokang Yang, Haidong Zou, Bin Sheng, and Weiping Jia. A deep learning system for detecting diabetic retinopathy across the disease spectrum. Nature Communications, 12(1):3242, May 2021. Number: 1 Publisher: Nature Publishing Group.

[51] Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, and Bo Xu. A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19). European Radiology, 31(8):6096–6104, August 2021.

[52] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, October 2014. arXiv:1311.2524 [cs].

[53] Ross Girshick. Fast R-CNN, September 2015. arXiv:1504.08083 [cs].

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, January 2016. arXiv:1506.01497 [cs].

[55] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 779–788, June 2016.

[56] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. volume 9905, pages 21–37. 2016. arXiv:1512.02325 [cs].

[57] Alireza Ghasemieh and Rasha Kashef. 3D object detection for autonomous driving: Methods, models, sensors, data, and challenges. Transportation Engineering, 8:100115, June 2022.

[58] Pawan Kumar Mishra and G. P. Saroha. A study on video surveillance system for object detection and tracking. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pages 221–226, March 2016.

[59] Hakan Karaoguz and Patric Jensfelt. Object Detection Approach for Robot Grasp Detection. In 2019 International Conference on Robotics and Automation (ICRA), pages 4953–4959, May 2019. ISSN: 2577-087X.

[60] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, International Conference on Medical Image Computing and Computer Assisted Intervention, pages 234–241, 2015.

[61] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, October 2017. ISSN: 2380-7504.

[62] Leon Amadeus Varga, Benjamin Kiefer, Martin Messmer, and Andreas Zell. SeaDronesSee: A Maritime Benchmark for Detecting Humans in Open Water. pages 2260–2270, 2022.

[63] Zhe Wang, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning Deep Representations for Scene Labeling with Semantic Context Guided Supervision, June 2017. arXiv:1706.02493 [cs].

[64] Xiang Wang, Zhiwu Qing, Ziyuan Huang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Changxin Gao, and Nong Sang. Proposal Relation Network for Temporal Action Detection, June 2021. arXiv:2106.11812 [cs] version: 1.

[65] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation, October 2022. arXiv:2204.12484 [cs] version: 3.

[66] Kok Hao Chen, Alistair N. Boettiger, Jeffrey R. Moffitt, Siyuan Wang, and Xiaowei Zhuang. RNA imaging. Spatially resolved, highly multiplexed RNA profiling in single cells. Science (New York, N.Y.), 348(6233):aaa6090, April 2015.

[67] Matsilele Mabaso, Daniel Withey, and Bhekisipho Twala. Spot Detection in Microscopy Images using Convolutional Neural Network with Sliding-Window Approach:. In Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies, pages 67–74, Funchal, Madeira, Portugal, 2018. SCITEPRESS - Science and Technology Publications.

[68] J. M. Halstead, J. H. Wilbertz, F. Wippich, T. Lionnet, A. Ephrussi, and J. A. Chao. TRICK: A Single-Molecule Method for Imaging the First Round of Translation in Living Cells and Animals. Methods in Enzymology, 572:123–157, 2016.

[69] Hui Kong, Hatice Cinar Akakin, and Sanjay E. Sarma. A Generalized Laplacian of Gaussian Filter for Blob Detection and Its Applications. IEEE Transactions on Cybernetics, 43(6):1719–1733, December 2013. Conference Name: IEEE Transactions on Cybernetics.

[70] Jean-Yves Tinevez, Nick Perry, Johannes Schindelin, Genevieve M. Hoopes, Gregory D. Reynolds, Emmanuel Laplantine, Sebastian Y. Bednarek, Spencer L. Shorte, and Kevin W. Eliceiri. TrackMate: An open and extensible platform for single-particle tracking. Methods, 115:80–90, February 2017.

[71] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. Nature Methods, 18(1):100–106, January 2021. Number: 1 Publisher: Nature Publishing Group.

[72] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell Detection with Star-convex Polygons. volume 11071, pages 265–273. 2018. arXiv:1806.03535 [cs].

[73] Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers. Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy. In 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 3655–3662, Snowmass Village, CO, USA, March 2020. IEEE.

[74] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void - Learning Denoising from Single Noisy Images, April 2019. arXiv:1811.10980 [cs].

[75] Eva Höck, Tim-Oliver Buchholz, Anselm Brachmann, Florian Jug, and Alexander Freytag. N2v2 – fixing noise2void checkerboard artifacts with modified sampling strategies and a tweaked network architecture, 2022.

[76] Martin Weigert, Uwe Schmidt, Tobias Boothe, Andreas Müller, Alexandr Dibrov, Akanksha Jain, Benjamin Wilhelm, Deborah Schmidt, Coleman Broaddus, Siân Culley, Mauricio Rocha-Martins, Fabián Segovia-Miranda, Caren Norden, Ricardo Henriques, Marino Zerial, Michele Solimena, Jochen Rink, Pavel Tomancak, Loic Royer, Florian Jug, and Eugene W. Myers. Content-aware image restoration: pushing the limits of fluorescence microscopy. Nature Methods, 15(12):1090–1097, December 2018. Number: 12 Publisher: Nature Publishing Group.

[77] David A. Van Valen, Takamasa Kudo, Keara M. Lane, Derek N. Macklin, Nicolas T. Quach, Mialy M. DeFelice, Inbal Maayan, Yu Tanouchi, Euan A. Ashley, and Markus W. Covert. Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments. PLOS Computational Biology, 12(11):e1005177, November 2016. Publisher: Public Library of Science.

[78] Erick Moen, Enrico Borba, Geneva Miller, Morgan Schwartz, Dylan Bannon, Nora Koe, Isabella Camplisson, Daniel Kyme, Cole Pavelchek, Tyler Price, Takamasa Kudo, Edward Pao, William Graf, and David Van Valen. Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning. preprint, Bioinformatics, October 2019.

[79] Erick Moen, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen. Deep learning for cellular image analysis. Nature Methods, 16(12):1233–1246, December 2019. Number: 12 Publisher: Nature Publishing Group.

[80] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. seqFISH Accurately Detects Transcripts in Single Cells and Reveals Robust Spatial Organization in the Hippocampus. Neuron, 94(4):752–758.e1, May 2017.

[81] Chenglong Xia, Jean Fan, George Emanuel, Junjie Hao, and Xiaowei Zhuang. Spatial transcriptome profiling by MERFISH reveals subcellular RNA compartmentalization and cell cycle-dependent gene expression. Proceedings of the National Academy of Sciences, 116(39):19490–19499, September 2019.

[82] Prabhakar R. Gudla, Koh Nakayama, Gianluca Pegoraro, and Tom Misteli. SpotLearn: Convolutional Neural Network for Detection of Fluorescence In Situ Hybridization (FISH) Signals in High-Throughput Imaging Approaches. Cold Spring Harbor Symposia on Quantitative Biology, 82:57–70, 2017.

[83] T. Wollmann, C. Ritter, J. N. Dohrke, J.-Y. Lee, R. Bartenschlager, and K. Rohr. Detnet: Deep Neural Network For Particle Detection In Fluorescence Microscopy Images. In International Symposium on Biomedical Imaging, pages 517–520, April 2019.

[84] Nicolas Chenouard, Ihor Smal, Fabrice de Chaumont, Martin Maška, Ivo F. Sbalzarini, Yuanhao Gong, Janick Cardinale, Craig Carthel, Stefano Coraluppi, Mark Winter, Andrew R. Cohen, William J. Godinez, Karl Rohr, Yannis Kalaidzidis, Liang Liang, James Duncan, Hongying Shen, Yingke Xu, Klas E. G. Magnusson, Joakim Jaldén, Helen M. Blau, Perrine Paul-Gilloteaux, Philippe Roudot, Charles Kervrann, François Waharte, Jean-Yves Tinevez, Spencer L. Shorte, Joost Willemse, Katherine Celler, Gilles P. van Wezel, Han-Wei Dan, Yuh-Show Tsai, Carlos Ortiz de Solórzano, Jean-Christophe Olivo-Marin, and Erik Meijering. Objective comparison of particle tracking methods. Nature Methods, 11(3):281–289, March 2014.

[85] Ihor Smal, Marco Loog, Wiro Niessen, and Erik Meijering. Quantitative comparison of spot detection methods in fluorescence microscopy. IEEE transactions on medical imaging, 29(2):282–301, February 2010.

[86] Ivana Horvathova, Franka Voigt, Anna V. Kotrys, Yinxiu Zhan, Caroline G. Artus-Revel, Jan Eglinger, Michael B. Stadler, Luca Giorgetti, and Jeffrey A. Chao. The Dynamics of mRNA Turnover Revealed by Single-Molecule Imaging in Single Cells. Molecular Cell, 68(3):615–625.e9, November 2017.

[87] Daniel Mateju, Bastian Eichenberger, Franka Voigt, Jan Eglinger, Gregory Roth, and Jeffrey A. Chao. Single-Molecule Imaging Reveals Translation of mRNAs Localized to Stress Granules. Cell, 183(7):1801–1812.e13, December 2020.

[88] H. W. Kuhn. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2(1-2):83–97, 1955.

[89] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep Neural Networks for Object Detection. In Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc., 2013.

[90] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the Convergence of Adam and Beyond, April 2019. arXiv:1904.09237 [cs, math, stat] version: 1.

[91] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, February 2017. arXiv:1609.04836 [cs, math].

[92] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. volume 10553, pages 240–248. 2017. arXiv:1707.03237 [cs].

[93] N. Aloysius and M. Geetha. A review on deep convolutional neural networks. In International Conference on Communication and Signal Processing, pages 0588–0592, April 2017.

[94] Ella Bahry, Laura Breimann, Marwan Zouinkhi, Leo Epstein, Klim Kolyvanov, Nicholas Mamrak, Benjamin King, Xi Long, Kyle I. S. Harrington, Timothée Lionnet, and Stephan Preibisch. RS-FISH: precise, interactive, fast, and scalable FISH spot detection. Nature Methods, 19(12):1563–1567, December 2022. Number: 12 Publisher: Nature Publishing Group.

[95] Emmanuel Bouilhol, Anca F. Savulescu, Edgar Lefevre, Benjamin Dartigues, Robyn Brackin, and Macha Nikolski. DeepSpot: A deep neural network for RNA spot enhancement in single-molecule fluorescence in-situ hybridization microscopy images. Biological Imaging, 2:e4, 2022. Publisher: Cambridge University Press.

[96] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1):62–66, January 1979. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics.

[97] Edouard Bertrand, Pascal Chartrand, Matthias Schaefer, Shailesh M. Shenoy, Robert H. Singer, and Roy M. Long. Localization of ASH1 mRNA Particles in Living Yeast. Molecular Cell, 2(4):437–445, October 1998. Publisher: Elsevier.

[98] Xiaowei Yan, Tim A. Hoek, Ronald D. Vale, and Marvin E. Tanenbaum. Dynamics of Translation of Single mRNA Molecules In Vivo. Cell, 165(4):976–989, May 2016.

[99] Noah F. Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J. McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, Cole Pavelchek, Sunny Cui, Isabella Camplisson, Omer Bar-Tal, Jaiveer Singh, Mara Fong, Gautam Chaudhry, Zion Abraham, Jackson Moseley, Shiri Warshawsky, Erin Soon, Shirley Greenbaum, Tyler Risom, Travis Hollmann, Sean C. Bendall, Leeat Keren, William Graf, Michael Angelo, and David Van Valen. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. Nature Biotechnology, 40(4):555–565, April 2022. Number: 4 Publisher: Nature Publishing Group.

[100] Lars E. Borm, Alejandro Mossi Albiach, Camiel C. A. Mannens, Jokubas Janusauskas, Ceren Özgün, David Fernández-García, Rebecca Hodge, Francisca Castillo, Charlotte R. H. Hedin, Eduardo J. Villablanca, Per Uhlén, Ed S. Lein, Simone Codeluppi, and Sten Linnarsson. Scalable in situ single-cell profiling by electrophoretic capture of mRNA using EEL FISH. Nature Biotechnology, 41(2):222–231, February 2023. Number: 2 Publisher: Nature Publishing Group.

[101] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc., 2014.

[102] Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. Fairseq S2T: Fast Speech-to-Text Modeling with Fairseq. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations, pages 33–39, Suzhou, China, December 2020. Association for Computational Linguistics.

[103] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In 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), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[104] Burak Makav and Volkan Kılıç. A New Image Captioning Approach for Visually Impaired People. In 2019 11th International Conference on Electrical and Electronics Engineering (ELECO), pages 945–949, November 2019.

[105] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735–1780, November 1997.

[106] Shiyang Liao, Junbo Wang, Ruiyun Yu, Koichi Sato, and Zixue Cheng. CNN for situations understanding based on sentiment analysis of twitter data. Procedia Computer Science, 111:376–381, January 2017.

[107] Muhammad Zain Amin and Noman Nadeem. Convolutional Neural Network: Text Classification Model for Open Domain Question Answering System, October 2019. arXiv:1809.02479 [cs].

[108] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[109] Wikimedia Foundation. Wikimedia Downloads. https://dumps.wikimedia.org.

[110] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[111] Vidur Joshi, Matthew Peters, and Mark Hopkins. Extending a Parser to Distant Domains Using a Few Dozen Partially Annotated Examples, May 2018. arXiv:1805.06556 [cs].

[112] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[113] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding, January 2020. arXiv:1906.08237 [cs].

[114] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019. arXiv:1907.11692 [cs].

[115] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, July 2020. arXiv:1910.10683 [cs, stat].

[116] 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 Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873):583–589, August 2021. Number: 7873 Publisher: Nature Publishing Group.

[117] Zeeya Merali. AlphaFold developers win US$3-million Breakthrough Prize. Nature, 609(7929):889–889, September 2022. Bandiera_abtest: a Cg_type: News Number: 7929 Publisher: Nature Publishing Group Subject_term: Quantum information, Molecular biology, Mathematics and computing, Machine learning.

[118] Anastasia Baryshnikova, Michael Costanzo, Chad L. Myers, Brenda Andrews, and Charles Boone. Genetic Interaction Networks: Toward an Understanding of Heritability. Annual Review of Genomics and Human Genetics, 14(1):111–133, August 2013.

[119] Jay Shendure and Joshua M. Akey. The origins, determinants, and consequences of human mutations. Science, 349(6255):1478–1483, September 2015.

[120] Guillaume Diss and Ben Lehner. The genetic landscape of a physical interaction. eLife, 7:e32472, April 2018.

[121] Andreas G. Bader and Peter K. Vogt. Leucine Zipper Transcription Factors: bZIP Proteins. In Encyclopedic Reference of Genomics and Proteomics in Molecular Medicine, pages 964–967. Springer, Berlin, Heidelberg, 2006.

[122] Jorge Blanco, Jacob C Cooper, and Nicholas E Baker. Roles of C/EBP class bZip proteins in the growth and cell competition of Rp ('Minute') mutants in Drosophila. eLife, 9:e50535, January 2020. Publisher: eLife Sciences Publications, Ltd.

[123] Liang Guo, Xi Li, and Qi-Qun Tang. Transcriptional Regulation of Adipocyte Differentiation: A Central Role for CCAAT/Enhancer-binding Protein (C/EBP) β*. Journal of Biological Chemistry, 290(2):755–761, January 2015.

[124] Jochen Hess, Peter Angel, and Marina Schorpp-Kistner. AP-1 subunits: quarrel and harmony among siblings. Journal of Cell Science, 117(25):5965–5973, December 2004.

[125] Benoit Miotto and Kevin Struhl. Differential gene regulation by selective association of transcriptional coactivators and bZIP DNA-binding domains. Molecular and Cellular Biology, 26(16):5969–5982, August 2006.

[126] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with AlphaFold-Multimer. preprint, Bioinformatics, October 2021.

[127] Steven M. Lewis and Brian A. Kuhlman. Anchored design of protein-protein interfaces. PloS One, 6(6):e20872, 2011.

[128] Gwen R. Buel and Kylie J. Walters. Can AlphaFold2 predict the impact of missense mutations on structure? Nature Structural & Molecular Biology, 29(1):1–2, January 2022.

[129] Marina A. Pak, Karina A. Markhieva, Mariia S. Novikova, Dmitry S. Petrov, Ilya S. Vorobyev, Ekaterina S. Maksimova, Fyodor A. Kondrashov, and Dmitry N. Ivankov. Using AlphaFold to predict the impact of single mutations on protein stability and function. preprint, Bioinformatics, September 2021.

[130] Neil Savage. Breaking into the black box of artificial intelligence. Nature, March 2022. Bandiera_abtest: a Cg_type: Outlook Publisher: Nature Publishing Group Subject_term: Machine learning, Computer science, Information technology.

[131] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding Adversarial Attacks on Deep Learning Based Medical Image Analysis Systems. Pattern Recognition, 110:107332, February 2021. arXiv:1907.10456 [cs, eess].

[132] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on MRI. Zeitschrift für Medizinische Physik, 29(2):102–127, May 2019.

[133] Nathalie Lassau, Samy Ammari, Emilie Chouzenoux, Hugo Gortais, Paul Herent, Matthieu Devilder, Samer Soliman, Olivier Meyrignac, Marie-Pauline Talabard, Jean-Philippe Lamarque, Remy Dubois, Nicolas Loiseau, Paul Trichelair, Etienne Bendjebbar, Gabriel Garcia, Corinne Balleyguier, Mansouria Merad, Annabelle Stoclin, Simon Jegou, Franck Griscelli, Nicolas Tetelboum, Yingping Li, Sagar Verma, Matthieu Terris, Tasnim Dardouri, Kavya Gupta, Ana Neacsu, Frank Chemouni, Meriem Sefta, Paul Jehanno, Imad Bousaid, Yannick Boursin, Emmanuel Planchet, Mikael Azoulay, Jocelyn Dachary, Fabien Brulport, Adrian Gonzalez, Olivier Dehaene, Jean-Baptiste Schiratti, Kathryn Schutte,

Jean-Christophe Pesquet, Hugues Talbot, Elodie Pronier, Gilles Wainrib, Thomas Clozel, Fabrice Barlesi, Marie-France Bellin, and Michael G. B. Blum. Integrating deep learning CT-scan model, biological and clinical variables to predict severity of COVID-19 patients. Nature Communications, 12(1):634, January 2021. Number: 1 Publisher: Nature Publishing Group.

[134] Dong Jin Park, Min Woo Park, Homin Lee, Young-Jin Kim, Yeongsic Kim, and Young Hoon Park. Development of machine learning model for diagnostic disease prediction based on laboratory tests. Scientific Reports, 11(1):7567, April 2021. Number: 1 Publisher: Nature Publishing Group.

[135] Loredana Storelli, Matteo Azzimonti, Mor Gueye, Carmen Vizzino, Paolo Preziosa, Gioachino Tedeschi, Nicola De Stefano, Patrizia Pantano, Massimo Filippi, and Maria A. Rocca. A Deep Learning Approach to Predicting Disease Progression in Multiple Sclerosis Using Magnetic Resonance Imaging. Investigative Radiology, 57(7):423–432, July 2022.

[136] Khoa A. Tran, Olga Kondrashova, Andrew Bradley, Elizabeth D. Williams, John V. Pearson, and Nicola Waddell. Deep learning in cancer diagnosis, prognosis and treatment selection. Genome Medicine, 13(1):152, September 2021.

[137] Jan Zrimec, Christoph S. Börlin, Filip Buric, Azam Sheikh Muhammad, Rhongzen Chen, Verena Siewers, Vilhelm Verendel, Jens Nielsen, Mats Töpel, and Aleksej Zelezniak. Deep learning suggests that gene expression is encoded in all parts of a co-evolving interacting gene regulatory structure. Nature Communications, 11(1):6141, December 2020. Number: 1 Publisher: Nature Publishing Group.

[138] Shushan Toneyan, Ziqi Tang, and Peter K. Koo. Evaluating deep learning for predicting epigenomic profiles. Nature Machine Intelligence, 4(12):1088–1100, December 2022. Number: 12 Publisher: Nature Publishing Group.

[139] Derek Jones, Hyojin Kim, Xiaohua Zhang, Adam Zemla, Garrett Stevenson, W. F. Drew Bennett, Daniel Kirshner, Sergio E. Wong, Felice C. Lightstone, and Jonathan E. Allen. Improved Protein–Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. Journal of Chemical Information and Modeling, 61(4):1583–1592, April 2021. Publisher: American Chemical Society.

[140] Xiting Wang, Meng Liu, Lan Zhang, Yun Wang, Yu Li, and Tao Lu. Optimizing Pharmacokinetic Property Prediction Based on Integrated Datasets and a Deep Learning Approach. Journal of Chemical Information and Modeling, 60(10):4603–4613, October 2020. Publisher: American Chemical Society.

[141] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. DeepTox: Toxicity Prediction using Deep Learning. Frontiers in Environmental Science, 3, 2016.

# 6

# Supplementary Figures

**Table S1:** Description of network blocks in tabular format. Parameters that were not specifically listed were set to the default values of TensorFlow version 2.2.

**(a)** List of parameters in every "Conv2D" layer.

| Parameter | Value |
|---|---|
| Kernel | 3x3 |
| Kernel initializer | He et al. |
| Padding | Same |
| Bias regularizer | L2 $10^{-6}$ |
| Kernel regularizer | L2 $10^{-6}$ |

**(b)** Layers in the "Convolutional block".

| Layer name | Operator | Input |
|---|---|---|
| conv2d_1 | Conv2D | input |
| activation_1 | LReLU | conv2d_1 |
| conv2d_2 | Conv2D | activation_1 |
| activation_2 | LReLU | conv2d_2 |
| conv2d_3 | Conv2D | activation_2 |
| activation_3 | LReLU | conv2d_3 |

**(c)** Layers in the "Squeeze and excitation block".

| Layer name | Operator | Input |
|---|---|---|
| gap2d | GlobalAveragePooling2D | input |
| dense_1 | Dense(units = max(n//8, 1)) | gap2d |
| activation_1 | LReLU | dense_1 |
| dense_2 | Dense(units = n) | activation_1 |
| activation_2 | Sigmoid | dense_2 |
| multiply | Multiply | activation_2, input |

**Algorithm 1:** Algorithmic description of the F1 integral score

**Data:** Two lists of coordinates $\hat{y}$ (prediction), $y$ (ground truth)

**Result:** F1 integral score

**Function** `hungarian_assignment(V, W, cost)`**:**

$l_w$ = `len(`*W*`)`

$l_v$ = `len(`*V*`)`

$L$ = `min(`$l_w$, $l_v$`)`

$P_w$ = `permutations(`*W*`)`

$P_v$ = `permutations(`*V*`)`

**return** $(w, v)$ **for** $w \in P_w$, $v \in P_v$ such that $\sum_{i=0}^{L-1}$`cost(`$w_i, v_i$`)` is minimized

**Function** `assignment(`$V, W$`, cost, cutoff)`**:**

$(v, w)$ = `hungarian_assignment(`$V, W$`, cost)`

**return** [($v_i$, $w_i$) **for** $v_i$, $w_i$ in $(v, w)$ **if** cost($v_i, w_i$) $\leqslant$ cutoff]

**Function** `calculate_f1_at_cutoff(`$\hat{y}, y$`, cost, cutoff)`**:**

pred_true = `assignment(`$\hat{y}, y$`, cost, cutoff)` `/* assignment of` $\hat{y}$ `to` $y$ `*/`

true_pred = `assignment(`$y, \hat{y}$`, cost, cutoff)` `/* assignment of` $y$ `to` $\hat{y}$ `*/`

$t_p$ = `len(`*true_pred*`)`

$f_n$ = `len(`$\hat{y}$`)` - `len(`*true_pred*`)`

$f_p$ = `len(`$y$`)` - `len(`*pred_true*`)`

F1 = $2 * (P * R)/(P + R)$, where $\begin{cases} P = t_p/(t_p + f_p) \\ R = t_p/(t_p + f_n) \end{cases}$

**return** F1

min_cutoff = 0

max_cutoff = 3

num_cutoff = 50

cutoffs = `linspace(`*min=min_cutoff, max=max_cutoff, num=num_cutoff*`)`

scores = [`calculate_f1_at_cutoff(`$\hat{y}, y$, euclidean_dist, c`)` **for** c in cutoffs]

F1$_{integral}$ = `trapezoidal_rule(`*scores*`)`

F1$_{integral}$ score /= max_cutoff `/* normalize between 0 and 1 */`

**Table S2:** F1 scores at 3px cutoff of deepBlink and benchmarking methods (mean ± standard deviation). Shows values for three methods across six datasets. Higher value is better.

| | | TrackMate | DetNet | deepBlink |
|---|---|---|---|---|
| Real | smFISH | 0.902 ± 0.183 | 0.595 ± 0.319 | **0.942 ± 0.141** |
| | SunTag | 0.728 ± 0.362 | 0.050 ± 0.053 | **0.796 ± 0.297** |
| Synthetic | Particle | **0.998 ± 0.002** | 0.977 ± 0.014 | 0.997 ± 0.002 |
| | Microtubule | 0.623 ± 0.366 | 0.456 ± 0.169 | **0.795 ± 0.214** |
| | Receptor | 0.683 ± 0.367 | 0.658 ± 0.288 | **0.776 ± 0.300** |
| | Vesicle | 0.617 ± 0.216 | 0.674 ± 0.258 | **0.861 ± 0.213** |
| | Average | 0.759 ± 0.250 | 0.568 ± 0.183 | **0.861 ± 0.195** |

**Table S3:** RMSE values of deepBlink and benchmarking methods (mean ± standard deviation). RMSE was only calculated on true positive spots. Shows values for three methods across four synthetic datasets, where the exact spot position is known. Lower value is better.

| | TrackMate | DetNet | deepBlink + Gaussian | deepBlink |
|---|---|---|---|---|
| Particle | 0.167 ± 0.020 | 0.438 ± 0.016 | 0.207 ± 0.021 | **0.157 ± 0.020** |
| Microtubule | 1.017 ± 0.221 | 0.887 ± 0.228 | 0.850 ± 0.223 | **0.588 ± 0.339** |
| Receptor | **0.428 ± 0.255** | 0.661 ± 0.203 | 0.517 ± 0.302 | 0.435 ± 0.294 |
| Vesicle | 0.658 ± 0.324 | 0.684 ± 0.223 | 0.644 ± 0.312 | **0.459 ± 0.311** |
| Average | 0.568 ± 0.205 | 0.667 ± 0.167 | 0.554 ± 0.215 | **0.410 ± 0.241** |

**Table S4:** F1 integral score (mean $\pm$ standard deviation) for images of SNR equal to one. Shows values for three methods across three datasets.

| Dataset | Density | TrackMate | DetNet | deepBlink |
|---|---|---|---|---|
| Microtubule | Low | $0.021 \pm 0.017$ | $0.064 \pm 0.014$ | $\mathbf{0.179 \pm 0.022}$ |
| | Mid | $0.018 \pm 0.008$ | $0.176 \pm 0.015$ | $\mathbf{0.253 \pm 0.012}$ |
| | High | $0.020 \pm 0.006$ | $0.145 \pm 0.012$ | $\mathbf{0.305 \pm 0.007}$ |
| Receptor | Low | $0.052 \pm 0.029$ | $0.059 \pm 0.013$ | $\mathbf{0.075 \pm 0.013}$ |
| | Mid | $0.050 \pm 0.010$ | $0.204 \pm 0.013$ | $\mathbf{0.253 \pm 0.013}$ |
| | High | $0.059 \pm 0.010$ | $0.175 \pm 0.009$ | $\mathbf{0.211 \pm 0.010}$ |
| Vesicle | Low | $0.113 \pm 0.016$ | $0.090 \pm 0.011$ | $\mathbf{0.221 \pm 0.031}$ |
| | Mid | $0.250 \pm 0.016$ | $0.249 \pm 0.010$ | $\mathbf{0.327 \pm 0.013}$ |
| | High | $0.246 \pm 0.008$ | $0.228 \pm 0.008$ | $\mathbf{0.353 \pm 0.008}$ |

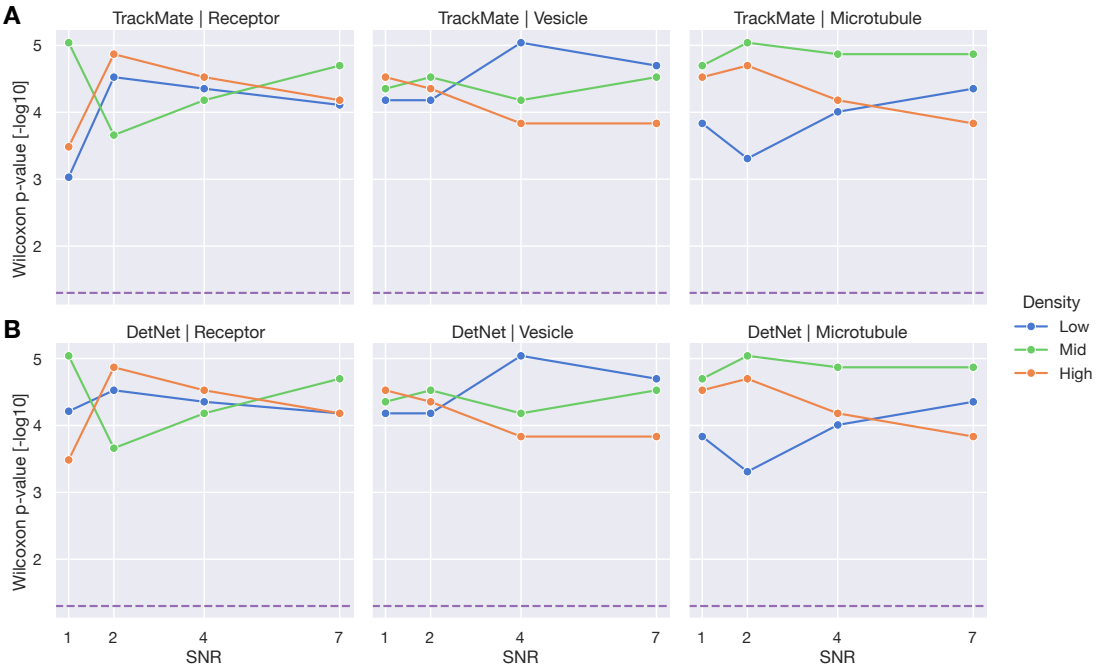
**Table S5:** The set of optimized parameters used for each benchmarking method and for each dataset.

**(a)** TrackMate

| Dataset | Diameter | Quantile |
|---|---|---|
| smFISH | 4.0 | 0.871 |
| SunTag | 1.0 | 0.170 |
| Particle | 4.0 | 0.001 |
| Microtubule | 4.0 | 1.091 |
| Receptor | 3.0 | 0.795 |
| Vesicle | 3.0 | 1.342 |

**(b)** DetNet

| Dataset | Alpha |
|---|---|
| smFISH | 0.4737 |
| SunTag | 0.6842 |
| Particle | 0.5263 |
| Microtubule | 0.2105 |
| Receptor | 0.8946 |
| Vesicle | 0.5789 |




**Supplementary Figure S1:** P-Values from one-sided Wilcoxon signed-rank tests measuring how significantly deepBlink performs better than **A** TrackMate and **B** DetNet at different spot densities and image SNRs. The purple dotted line indicates a significant p-value of 0.05.

# A

# A glimpse into the birth, life and death of single mRNAs in cells

Annual Review of Cell and Developmental Biology. (Accepted) 2023 January 15th; Vol. 39


- **Bastian Th. Eichenberger**[*]
- Esther Griesbach[*]
- Jessica Mitchell[*]
- Jeffrey A. Chao[#]

[*] These authors have contributed equally to this work

[#] Corresponding authors


I wrote the sections on "Recent developments for imaging single mRNAs in fixed and live cells" and "Imaging RNA molecules in live cells" and designed Figure 1.

*Appendix A. A glimpse into the birth, life and death of single mRNAs in cells*

**Abstract**

Recent advances in single-molecule imaging of mRNAs in fixed and living cells have enabled the lives of mRNAs to be studied with unprecedented spatial and temporal detail. These approaches have moved beyond simply being able to observe specific events and have begun to allow understanding how regulation is coupled between steps in the mRNA lifecycle. Additionally, these methodologies are now being applied in multicellular systems and animals to provide more nuanced insights into the physiological regulation of RNA metabolism.

**Keywords**

fluorescence microscopy, transcription, splicing, RNA localization, translation, mRNA decay

*Appendix A. A glimpse into the birth, life and death of single mRNAs in cells*

**1. Introduction**

When investigating a biological question, as with any other discipline that requires analyzing information and problem solving, it is often prudent to consider how various experiments answer the fundamental questions: Who, What, When, Where, How and Why? For instance, while genetic screens are able to identify the proteins (*who*) involved in a process, they usually provide limited detail into if these proteins assemble into a complex or if they function directly. Similarly, biochemical approaches can resolve differences between distinct molecular mechanisms (*how*), but almost always require the molecules involved to be already known. While often overlooked, understanding *when* and *where* in a cell molecules interact can help decipher the complexity that exists in biology. The recent developments in single-molecule fluorescence microscopy methodologies for imaging individual mRNA transcripts, however, have enabled researchers to move beyond obtaining only temporal and spatial information and have begun to allow biochemical measurements to be made inside cells. This unique ability to directly interrogate cellular biochemistry has already led to a number of unanticipated findings and will continue to make significant contributions as these methodologies mature. In this review, we have organized the sections into a summary of recent technological developments in RNA imaging, the application of these techniques for investigating RNA biology with a particular focus on new insights into how steps in the mRNA life cycle are coupled and the challenges of applying these methodologies in multicellular systems and animals.

**2. Recent developments for imaging single mRNAs in fixed and live cells**

**2.1. Single-molecule fluorescence *in situ* hybridization**

While there have been tremendous advances since the initial development of single-molecule fluorescence in situ hybridization (smFISH) by Singer and colleagues over 25 years ago, the fundamental concept has remained unchanged since its introduction (Femino et al., 1998). In this method, fluorescently-labeled antisense oligonucleotide probes are hybridized to a transcript of interest under stringent conditions and non-specifically bound probes are removed during washes resulting in the accumulation of sufficient fluorescent signal to enable detection of individual RNAs within single cells **(Figure 1a)**. Continued improvements in the cost and ease of implementation, amplification of the fluorescent signal and scale of the number of transcripts

that can be detected within each cell have resulted, in part, in the recent revolution in spatial transcriptomics.

In order for smFISH to become more widely adopted by the community, easier access to fluorescently labeled probes was a crucial first step. An important advance was the development of ~20nt probes that were singly fluorescently-labeled at the 5′-end and could be purchased commercially (Raj et al., 2006). While ~48 probes are necessary to achieve single-molecule resolution, they facilitated experiments to be performed in a variety of experimental systems (Raj et al., 2008). These probes, however, were still not affordable for many labs and, consequently, more cost-effective approaches were developed. One approach was to take advantage of the terminal deoxynucleotidyl transferase enzyme and fluorescently-labeled dideoxy UTP to specifically add a fluorophore to the 3′-end of an oligonucleotide (Gaspar et al., 2017). While this results in an untemplated uridine nucleotide at the 3′-end, this addition can be taken into account during probe design and a single mismatched base pair in this position has not been reported to affect hybridization specificity. An alternative approach (single-molecule inexpensive FISH, smiFISH) was to design unlabelled primary bipartite oligonucleotides that contained both a sequence for gene-specific hybridization and an additional sequence complementary to a secondary oligonucleotide that was fluorescently labeled (Tsanov et al., 2016). Recently, this strategy was further modified to use RNA primary probes and the smFISH protocol was adapted to 96-well plates to enable higher throughput screens with single-molecule resolution (Safieddine et al., 2023).

The use of bipartite primary oligonucleotides has also led to the development of signal amplification strategies that enable smFISH experiments to be performed on a wider variety of microscopy modalities and sample specimens. One of the first approaches was based on harnessing the ability of *in situ* hybridization chain reactions (HCR) to amplify the signal by the self-assembly of two fluorophore-labeled hairpin oligonucleotides with overhanging sequences that initiate elongating chain reactions (Choi et al., 2010; Dirks & Pierce, 2004) **(Figure 1a)**. In the latest version, the protocol has been adapted to reduce background noise levels using split-initiator probes that enable measurements in thicker tissue slices previously limited by autofluorescence (Choi et al., 2018). Similarly, signal amplification by exchange reaction (SABER) uses *in vitro* primer exchange reactions to synthesize long stretches of concatemers that can be used with traditional fluorescently labeled probes (Kishi et al., 2019). Signal amplification can also be obtained using branched DNA oligonucleotides that can form

elaborate tree-like structures using four oligonucleotides (sequence specific probe, pre-amplifier, amplifier and fluorophore-labeled) and sequential hybridizations (Battich et al., 2013) **(Figure 1a)**. Another approach called ClampFISH uses amplification cycles relying on click chemistry reactions with padlock probes to provide increased binding sites for readout probes (Dardani et al., 2022). Compared with the previous generation of ClampFISH (Rouhanifard et al., 2018), it no longer requires gene region-specific oligonucleotides to be modified in the padlock thus reducing cost and protocol duration. An important consideration, however, for any amplification approach is the increased potential for false-negatives as any hybridization or polymerization step may not be completely efficient resulting in loss of accurate transcript numbers.

In order to detect multiple RNA species within a single cell, both sequential and multiplexed smFISH approaches have been developed **(Figure 1a)**. Initial experiments used only a few spectrally distinct fluorescent dyes or pseudo-colors to detect ~10 distinct mRNAs (Levsky et al., 2002; Lubeck et al., 2014). A major breakthrough that utilized an error-correcting barcode and combinatorial labeling scheme (multiplexed error robust FISH, MERFISH) enabled the identification of up to 140 mRNAs in individual cells (K. H. Chen et al., 2015). There have since been improvements in the encoding scheme of MERFISH allowing transcriptome wide analyses (1000s of mRNAs) with high detection efficiency and low misidentification rate (Xia et al., 2019). These techniques can be used in combination with expansion microscopy to counteract physical constraints imposed by diffraction limitations (F. Chen et al., 2015). Recent MERFISH-based approaches have employed branched DNA probes to improve spot brightness (Xia et al., 2019). Another approach (seqFISH) uses sequential labeling and a different encoding scheme to measure transcriptome-level profiling of both nascent and mature mRNAs (Eng et al., 2019; Shah et al., 2018). A multiplexed approach (split-FISH) has also been devised that uses two primary sequence-specific probes that form the binding site for a bridging probe that can then be detected with a fluorescent oligonucleotide in order to reduce background and false-positives (Goh et al., 2020).

Thicker tissues have posed challenges due to longer reaction kinetics and increased autofluorescence. Enhanced ELectric Fluorescence *in situ* Hybridization (EEL FISH) provides an innovative approach to transfer RNA from tissue sections onto an indium tin oxide-coated coverslip using electrophoresis (Borm et al., 2022). The tissue can then be digested allowing for increased imaging acquisition while preserving spatial organization. Expansion-Assisted Iterative Fluorescence *in situ* Hybridization (EASI-FISH) combines previously described

methods including expansion microscopy and HCR with an automated data processing pipeline to create a 3D characterization of different cell types in thick brain slices (Y. Wang et al., 2021). Similarly, advances in image analysis allowed smiFISH to be used to investigate whole embryos at single-molecule resolution (Calvo et al., 2021).

**2.2 *In situ* sequencing technologies**

A limitation of *in situ* hybridization techniques is that there is an inherent bias that only transcripts that have been specifically selected prior to the experiment can be detected. DNA sequencing, however, is a powerful discovery tool and a variety of approaches have enabled its use directly on RNAs within cells. One of the first methods for *in situ* sequencing used padlock probes, rolling circle amplification and sequencing by ligation chemistry (Ke et al., 2013) **(Figure 1b)**. In this approach, transcripts were reverse-transcribed to cDNA using random decamer primers to which either gap-filled or barcoded padlock probes were hybridized. The DNA circle is subsequently amplified and bases are called through sequential rounds of sequencing by ligation. This approach has been fully automated by employing a microfluidics platform thus reducing protocol time and improving throughput (Maïno et al., 2019). Hybridization based *in situ* sequencing using barcoded padlock probes (hybISS) is a similar padlock probe-based sequencing approach using unique ID sequences on each padlock that can then be detected by hybridization (Gyllborg et al., 2020). To improve the readout of cellular barcodes, BaristaSeq coupled padlock probes with Illumina sequencing chemistry that resulted in a five-fold increase in amplification efficiency (X. Chen et al., 2018). Spatially-resolved transcript amplicon readout mapping (STARmap) utilized hydrogel tissue clearing, dual padlock probes to increase specificity and *in situ* sequencing to detect between 160-1020 mRNA species into $mm^3$ volumes (X. Wang et al., 2018).

To fully leverage RNA sequencing to identify the position of all cellular transcripts, an alternative approach without was developed called Fluorescent *In Situ* SEQuencing (FISSEQ) (J. H. Lee et al., 2015). Here, RNA sequences are reverse-transcribed with the addition of amine-modifications to crosslink cDNA fragments to the cellular protein matrix and adapter sequence-tagged random hexamers. cDNA is amplified with CircLigases using rolling circle amplification analogous to padlock-based approaches but also crosslinked to create a mesh of amplicons. Imaging is performed by hybridization of a sequencing primer followed by the ligation of a fluorescent octamer to investigate the adjacent dinucleotide pair. Three nucleotides are

cleaved after imaging allowing ligation cycles to be repeated providing information for every fifth base position. This process is then repeated with four other sequencing primers with a single base offset filling in the previously hidden dinucleotide pairs. With seven cycles of ligation-cleavage, 32 nucleotides can be interrogated. FISSEQ has also been coupled with expansion microscopy to enable higher spatial resolution sequencing of RNAs in crowded environments (Alon et al., 2021).

**3. Imaging RNA molecules in live cells**

**3.1 Molecular Beacons**

While conceptually similar to smFISH, several challenges needed to be overcome to promote specific hybridization of a molecular beacon to its target mRNA in living cells **(Figure 1c)**. In order to prevent sequestration of molecular beacons in the nucleus, they have been combined to either tRNA or siRNA-like scaffolds (A. K. Chen et al., 2010; Mhlanga et al., 2005). Another approach is to use molecular beacons linked with particles such as quantum dots or NeutrAvidin that are physically unable to enter the nuclear pore channels due to their size (Alami et al., 2014; A. K. Chen et al., 2007). Recently, molecular beacons have enabled tracking of single *Camk2a, β-actin* and *Psd95* mRNAs in dendrites of primary rat hippocampal neurons suggesting that this approach may become more routinely applied (Donlin-Asp et al., 2021; Klimek et al., 2021).

**3.2 Cas9 and Cas13-based RNA imaging**

The discovery of Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) and the CRISPR-associated protein 9 (Cas9) has led to a revolution in genome-engineering due to its ability to program the Cas9 endonuclease to site-specifically cleave a DNA target (Ran et al., 2013). This system was engineered to allow specific recognition of single-stranded RNA, by directing Cas9 to an mRNA targeted by a guide RNA by providing a protospacer adjacent motif (PAM) sequence as a separate oligonucleotide (O'Connell et al., 2014). By using a nuclease-dead mutant of Cas9 (dCas9) fused to a fluorescent protein in combination with a synthetic guide RNA and PAM sequence complementary to an mRNA of interest, the movement of mRNAs can be monitored in live cells (Nelles et al., 2016) **(Figure 1c)**. An alternative approach to detect RNAs in live cells was developed using a different catalytically dead Cas

protein (dCas13a) (Abudayyeh et al., 2017). The two approaches (dCas9 and dCas13a) were combined to allow the imaging of two distinct RNA species in living cells and additional Cas proteins have been identified for multi-color RNA imaging (Yang et al., 2019, 2022). While these approaches hold tremendous promise for imaging endogenous mRNAs, their application for single-molecule imaging requires further development.

**3.3. Engineered RNA-protein complexes**

The highly specific interaction between the MS2 bacteriophage coat protein and its cognate RNA stem-loop was first developed to image the localization of *ASH1* mRNA in yeast (Bertrand et al., 1998). In this system, a reporter mRNA containing the *ASH1* 3′-UTR was modified to also contain six MS2 stem-loops that could be bound by an MS2 coat protein (MCP), which was engineered to prevent capsid formation, fused to GFP. Since interaction between MCP and its stem-loops is orthogonal to any host RNA-protein complex, the system has been widely adopted for imaging RNAs in a variety of experimental systems. This approach was further refined to allow for single-molecule imaging of mRNAs (Fusco et al., 2003), sequence-optimized to reduce recombination of the repetitive stem-loop and coat proteins sequences (Wu et al., 2015), engineered to reduce accumulation of degradation intermediates in yeast (Tutucci et al., 2018), limit background fluorescence using split fluorescent proteins (S. Y. Park et al., 2020; Wu et al., 2014) and destabilization of reporter transcripts (W. Li et al., 2022) **(Figure 1d)**. Additionally, other RNA-protein complexes based on the PP7 bacteriophage coat protein (Hocine et al., 2013; Larson et al., 2011), lambda N peptide (Daigle & Ellenberg, 2007), human U1A (Brodsky & Silver, 2002) and bacterial BglG (J. Chen et al., 2009) have also been engineered for single-molecule RNA imaging that has allowed multicolor experiments to be performed in living cells.

**3.4 Fluorogenic RNA Aptamers**

The introduction of both a fluorescent RNA-binding protein and a modified reporter RNA can be challenging in some experimental systems as both components must be expressed at appropriate levels to allow for imaging of single mRNAs. RNAs, however, can be evolved through multiple rounds of selection and amplification for specific properties including emission of fluorescence upon binding a ligand (Tuerk & Gold, 1990). One of the first evolved RNAs (aptamers) for live-cell imaging was Spinach, an aptamer that showed selective fluorescence

upon binding to 3,5-difluoro-4-hydroxybenzylidene imidazolinone (DFHBI), which is a small molecule that mimics the GFP fluorophore (Paige et al., 2011) **(Figure 1d)**. Similar approaches have been used to construct a palette of aptamers (Broccoli, Mango, Peach) with different fluorescent properties (Dolgosheina et al., 2014; Filonov et al., 2014; Kong et al., 2021). A modified Mango2 array enabled live-cell imaging of single RNAs with similar sensitivity to the MS2-based system (Cawte et al., 2020). Recently, the RhoBAST aptamer was described that binds to a fluorogenic rhodamine dye with rapid association and dissociation kinetics that is suitable for super-resolution studies with 10-nm localization precision (Sunbul et al., 2021).

**3.5 Imaging translation of single mRNAs**

In order to distinguish an mRNA that has never been translated from one that has undergone at least one round of translation, Translating RNA Imaging by Coat protein Knock-off (TRICK) was developed (Halstead et al., 2015) **(Figure 1e)**. In this approach, a transcript is fluorescently labeled using both the PP7 (coding sequence) and MS2 (3′-UTR) systems with spectrally distinct fluorescent proteins. In this experiment, the timing and location of the first round of translation is directly observed by the displacement of the PP7 signal from the coding sequence of the transcript by the translating ribosome resulting in singly MS2 labeled transcripts. Since the PP7 coat protein (PCP) is fused to a nuclear localization sequence as well as a fluorescent protein, the displaced PCP fusion protein does not accumulate in the cytoplasm and thereby prevents rebinding after the first round of translation.

While TRICK is a sensitive measurement of the first round of translation, it is unable to monitor additional rounds of translation. Two methodologies that rely on amplification of the fluorescent signal of a single protein have enabled translation to be further characterized by monitoring the signal arising from nascent polypeptides in polysomes (Pichon et al., 2016; C. Wang et al., 2016; Wu et al., 2016; Yan et al., 2016). In the first approach, the SUperNova tag (SunTag) cassette (Tanenbaum et al., 2014), which is a repetitive sequence consisting of 24 GCN4 epitopes, is placed at the N-terminus of a protein of interest **(Figure 1e)**. The SunTag cassette is then recognized as it emerges from the ribosome during translation and is bound by its cognate single-chain antibody fused to GFP. An orthogonal system to SunTag based on a gp41 peptide (MoonTag) and corresponding antibody has since been developed and has allowed the heterogeneity of translation to be further characterized (Boersma et al., 2019). The second approach utilized either 10 FLAG or HA epitopes that were grafted into a non-fluorescent GFP

beta-barrel (Viswanathan et al., 2015) in order to image translation site dynamics (Morisaki et al., 2016). The main difference between this approach and the one based on SunTag is that recombinant anti-FLAG or anti-HA antibody fragments had to be bead-loaded into cells. An HA-binding single-chain variable fragment that can be genetically encoded, however, was developed which has expanded the possibilities of this approach (Zhao et al., 2019).

**3.6 Single-molecule imaging of mRNA decay**

Using fluorescence microscopy to image mRNA degradation is inherently challenging since the process results in the loss of the molecule being studied. One approach called 3′-RNA end accumulation during turnover (TREAT) uses a flaviviral RNA pseudoknot that blocks Xrn1 (Chapman et al., 2014), the cytoplasmic 5′-3′ exonuclease, to specifically stabilize a degradation intermediate (Horvathova et al., 2017) **(Figure 1f)**. Placement of these tandem pseudoknots between PP7 and MS2 stem-loops in the 3′-UTR of a reporter mRNA enables both intact transcripts (PP7 and MS2) and stabilized degradation fragments (MS2-only) to be visualized in living cells. It should be noted that while MCP and PCP bound to their respective stem-loops were found to inhibit Xrn1-mediated degradation in some contexts in yeast (J. F. Garcia & Parker, 2015), this effect was not observed in either human (Horvathova et al., 2017) or mouse cell lines (Kim et al., 2019).

**4. Imaging coupling of steps in gene expression**

**4.1 Transcription factor-binding and transcription**

Understanding the kinetics of transcription is crucial to dissect the molecular mechanisms that underlie gene activation. Single-molecule imaging approaches have revealed that some genes are constitutively transcribed whereas others are expressed in bursts of transcriptional activity (Bahar Halpern et al., 2015; Chubb et al., 2006; Darzacq et al., 2007; Larson et al., 2011; Zenklusen et al., 2008). To study how transcriptional activation is triggered, single-molecule imaging technologies have been developed to measure the dynamics of transcription factor (TF) binding to promoters (C.-Y. Lee & Myong, 2021; Liu & Tjian, 2018). Live imaging of transcription sites (TS) is accompanied by technical challenges as it is usually performed in a single focal plane and TS intensity vary due to both transcription and cell movement. Donovan et al. have overcome these limitations by making use of an orbital tracking microscope to simultaneously

track TSs and TFs in three dimensions (3D). To visualize transcription, the authors incorporated PP7 stem-loops into the 5′-UTR of either *GAL3* or *GAL10* and for simultaneous visualization of TF binding, they tagged Gal4 with a HaloTag (**Figure 2a**). By tracking TSs and TFs in 3D over time, the authors were able to correlate TF binding with transcription activation at a single locus (Donovan et al., 2019). Higher affinity TF binding sites resulted in longer TF dwell times at the promoter and increased transcription (Donovan et al., 2019).

RNAPII occupancy and transcription activation at a specific locus has also been measured in mammalian cells (Cho et al., 2016; J. Li et al., 2019). In a recent study using Target-Locking 3D stimulated emission depletion (STED) nanoscopy (J. Li et al., 2019), a mini-gene system (Tet-Off) was established to simultaneously image RNAPII (SNAP-Rbp1), nascent RNA (PP7) and the genomic location (TetR-RFP) in human U2OS cells. This allowed a temporal and spatial link between RNAPII accumulation and transcription activity and enabled visualization of the different steps of the transcription cycle (initiation, elongation, termination). Using inhibitors to block transcription initiation or promoter-proximal pause release into active elongation, it was shown that the RNAPII signal follows kinetics similar to elongation and RNA release. This indicated that RNAPII molecules accumulate at the mini-gene during elongation or termination, instead of accumulating before initiation, as observed for the endogenous mouse $\beta$-actin gene (Cho et al., 2016). These results may reflect gene-specific variability or differences between artificial and endogenous promoters. Li et al. extended their analysis to mouse embryonic stem cells, where nascent RNA (MS2) and RNAPII or different regulatory factors (RF) (SNAP) were imaged, which allowed correlations between focal RF accumulation at enhancer regions, RNAPII occupancy at the promoter and transcriptional activity to be measured (J. Li et al., 2019). While these methods pave the way to dissect the complexity of gene expression regulation in living cells, future developments will increase resolution even further to visualize real-time promoter-enhancer interactions or follow single RNAPII molecules during a complete transcription cycle.

It is still unclear how intranuclear localization of gene loci affects gene expression, but several studies suggest that association to nuclear speckles (NS) enhances gene expression (Bhat et al., 2023; Y. Chen et al., 2018; Y. Chen & Belmont, 2019; Faber et al., 2022; Guo et al., 2019; Takei et al., 2021; L. Zhang et al., 2020). NSs are dynamic membraneless subcompartments in the nucleus enriched for RNA processing factors and RNA (Fei et al., 2017; Hall et al., 2006; Misteli et al., 1997). The main two components of nuclear speckles, and essential for their

architecture, are the spliceosome-associated proteins SRRM2 and SON (Dopie et al., 2020; Ilik et al., 2020). A transgene for the heat shock protein 70 (Hsp70) gene family associated with NSs through a sequence element in its promoter that was independent of genomic location (Hu et al., 2010). In a later study, the HSP70 locus (EGFP-LacI, strong signal), nascent HSPA1A mRNA (MS2) and nuclear speckles (EGFP-SON, weak signal) were imaged simultaneously in the same cells to show that NS-association is followed by increased mRNA expression (Khanna et al., 2014) **(Figure 2b)**. Interestingly, HSP70 locus association to NSs was dependent on nuclear actin polymerisation, although the molecular mechanism remains unknown. The authors also made the important observation that expression of MS2-coat protein (MCP) alters the localization of transcripts containing MS2 stem-loops with respect to NSs. When MCP was expressed, transcripts localized at the NSs periphery rather than inside NSs, as shown for cells devoid of MCP (Khanna et al., 2014). This highlights how mRNA tagging can alter mRNA localization and the importance for appropriate controls.

In order to provide a broader view of how genomic location impacts transcription, MERFISH was used to image hundreds of endogenous RNAs, their genomic loci and NSs in the same cells. This showed that proximity to NSs is associated with increased transcriptional bursting (Su et al., 2020). Additionally, smFISH was used to show that p53, through direct DNA binding, mediates localization of a subset of its target genes to NSs to boost their transcription by increasing burst size (Alexander et al., 2021). Interestingly, this association to nuclear speckles is facilitated by transcription-independent mechanisms as it was mediated through the DNA binding domain of p53 rather than its transactivation domain. Future studies are required to show which nuclear speckle-associated factors are responsible for this effect and it also remains to be seen if locus localization to NSs is a general regulatory mechanism employed by TFs to fine tune gene expression or if it is a function limited to specific TFs.

**4.2. Transcription and splicing**

Single-molecule imaging has allowed visualization of splicing in fixed and live cells by labeling exons and introns with different fluorophores **(Figure 2c)**. Vargas et al. found that splicing mostly occurs co-transcriptionally but can be uncoupled when the intronic polypyrimidine tract (Py) that precedes the 3′ splice site is embedded in a strong secondary structure (Vargas et al., 2011). Substituting a strong for a weak Py also resulted in a slower splicing rate (Martin et al., 2013). These experiments show that splice site accessibility and strength influence splicing

efficiency, possibly reflecting changes in spliceosome recruitment dynamics. Increasing the number of introns in a reporter mRNA led to increased splicing factor recruitment but delayed splicing until after transcription termination but still in close proximity to the TS (Brody et al., 2011). Recent work from the same group to study splicing factor dynamics in the nucleus using live imaging, showed that increasing the availability of active splicing factors in the nucleoplasm allows for more efficient splicing and faster transcript release from the TS (Hochberg-Laufer et al., 2019). Coulon et al. found that transcript release from the TS is stochastic and the time required to excise the last intron can occur either before or after the RNA leaves the TS, as a result of kinetic competition between 3′-end processing and splicing. Interestingly, post-release splicing was found to be faster than splicing at the TS. Kinetic competition was abolished upon splicing factor (U2AF1) mutation, which resulted in strictly post-transcriptional splicing (Coulon et al., 2014). In order to expand these studies to multiple genes, introns of endogenous human genes were tagged (MS2/PP7) using a GeneTrap strategy (Wan et al., 2021). All 442 analyzed genes showed transcriptional bursting but each burst was small, only producing a few transcripts. Intron removal kinetics were highly variable, consistent with stochastic splice site selection, and often occurred progressively by recursive splicing. By labeling a long intron of the RAB7A gene with two different colors, it was shown that the upstream part of the intron (MS2) was removed first and the downstream part (PP7) in a consecutive splicing reaction, in a co-transcriptional manner (Wan et al., 2021). Consequently, while annotated splice sites define the intron boundaries they are not the only splice sites that are used and thus reflect only a fraction of splicing activity in the cell.

Given the diffraction limit for optical microscopy, it remains a debate whether RNAs localized close to the TS are actually still bound to transcribing RNAPII or if they remain close to the gene body even though transcription is concluded. A recent study to address this issue, using smFISH combined with expansion microscopy, showed that the proportion of post-transcriptionally spliced introns varies from intron to intron within a single gene and between cells. Interestingly, all four tested human genes showed some degree of post-transcriptional splicing (Coté et al., 2021). For long introns, 5′ and 3′ splice sites can be separated by thousands of nucleotides, however, it remains unclear how these two ends are brought into close proximity and how the transcription machinery facilitates this process. Interactions between transcribing polymerase and components of the U1 snRNP have been suggested to position the 5′ splice site close to the RNA exit site of RNAPII and thus close to the newly transcribed 3′ splice site (Harlen et al., 2016; Nojima et al., 2018; S. Zhang et al., 2021). A

recent study using smFISH combined with structured illumination microscopy (SIM) shows that as pre-mRNAs emerge from transcribing RNAPII, introns are packaged into compact particles with the 5′ and 3′ ends in close proximity **(Figure 2d)**. By visualizing partially transcribed introns using probe sets for different parts of a long intron, the authors show that the intron has a looped conformation that is altered during transcription, reflected by a changing separation between distal regions of the intron with the length to which the intron has been transcribed (Adivarahan et al., 2021). This finding suggests that the intron 5′ end is indeed tethered to transcribing RNAPII, consistent with the U1 snRNP-RNAPII tethering model. They further show that this conformational organization is dependent on correct spliceosome assembly, as splicing inhibition with Pladienolide B, leads to separation of intron 5′ and 3′ ends, consistent with a less compact conformation of the intron.

Splicing factors are enriched in NSs, which led to the hypothesis that splicing occurs in these nuclear compartments (Dias et al., 2010; Lamond & Spector, 2003; Spector & Lamond, 2011). However, recent studies show that NSs can act as storage bodies that supply splicing factors to sites of active transcription (Faber et al., 2022; Hasenson & Shav-Tal, 2020; Lewis & Tollervey, 2000). Interestingly, transcriptionally active gene loci have been shown to localize close to NSs thereby enabling more efficient splicing of their nascent pre-mRNAs (Bhat et al., 2023; Takei et al., 2021). Using a combination of crosslinking methods (RNA-DNA SPRITE and psoralen-mediated crosslinking) (Calvet & Pederson, 1981; Quinodoz et al., 2022) and seqFISH+ (Takei et al., 2021), Bhat et al. showed that spliceosome U snRNPs (U1, U2, U4, U6) are more enriched on nascent pre-mRNAs expressed from genes that are close to NSs (Bhat et al., 2023). To further validate their results, the authors developed a reporter-based assay to test if splicing is enhanced through localization to NSs. The reporter pre-mRNA contained MS2 stem-loops in the intron and was designed to express GFP only when spliced. To bring this construct into close proximity to NSs, it was integrated into cells that expressed MCP fused to NS markers or non-speckle associated proteins as negative controls. Only when the pre-mRNA was associated with NSs, but not when localized to the nuclear lamina or the nucleoplasm, its splicing efficiency was enhanced (Bhat et al., 2023).

**4.3 Splicing and RNA export**

Using single-molecule imaging in live cells, it has been shown that mRNPs move from the TS to the NPC via regular or corralled diffusion through interchromatin channels, whereas

translocation through the pore is facilitated and rapid (Biswas et al., 2021; Grünwald & Singer, 2010; Mor et al., 2010; Politz et al., 1999; Shav-Tal et al., 2004). The latter occurs with the 5′-end entering the NPC first, as confirmed by smFISH with probes labeled with different colors for the mRNA 3′-end, the middle of the transcript and the 5′-end (Ashkenazy-Titelman et al., 2022; Daneholt, 2001) **(Figure 2e)**. Mor et al. used single-molecule imaging to study how splicing affects mRNA export. They compared reporter mRNAs of different lengths, with or without introns, tagged with MS2 stem-loops. Intron-containing reporter mRNAs were spliced co-transcriptionally and more efficiently exported compared to their intron-lacking counterparts, however, splicing had no effect on nucleoplasmic diffusion rates of these mRNAs nor their transit through the pore (Mor et al., 2010). The authors conclude that the transcripts without introns are retained in the nucleus for longer due to the lack of suitable export signals.

**4.4. RNA localization and translation**

Localizing specific mRNA transcripts within relevant subcellular compartments is an important mechanism by which gene expression is spatially regulated. This is perhaps most spectacularly demonstrated in the developing *Drosophila* oocyte and embryo in which specific mRNAs are spatially localized to define developmentally-relevant patterns of gene expression. Live imaging of mRNA has been instrumental in uncovering the mechanistic basis of transcript localization and a number of mechanisms have emerged including passive diffusion and localized entrapment (Forrest & Gavis, 2003) or stabilization (Bashirullah et al., 2001), while in other cases, transcripts are actively transported directly along the microtubules of the cytoskeleton (Bullock & Ish-Horowicz, 2001; Weil et al., 2006; Zimyanin et al., 2008) **(Figure 2f)**. mRNAs can also be trafficked through indirect interactions with the cytoskeleton by hitchhiking on actively transported lysosomes via ANXA11, which acts as a molecular tether (Liao et al., 2019). These diverse localization mechanisms are orchestrated by *cis*-acting localization elements, which are recognised by mRNA binding proteins that mediate interactions with the transport machinery as well as regulatory proteins. Localization sequences are commonly found within the 3′-UTR and alternative polyadenylation site usage can be used to generate 3′-UTR isoforms that specify different localization patterns (Middleton et al., 2019).

Live translation site imaging has begun to reveal unexpected insights into the coupling of mRNA localization and protein synthesis. For instance, the localization of mRNAs to SGs was previously thought to universally repress translation, however, SunTag imaging revealed that SG

localization did non inhibit translation and that elongation rates were similar for mRNAs in SGs and in the cytosol of stressed as well as for mRNAs in unstressed cells (Mateju et al., 2020) **(Figure 2g)**. Similarly, mRNAs were thought to be transported in translationally-repressed states, but live imaging experiments have revealed that mRNAs can engage in active transport while being translated (C. Wang et al., 2016; Wu et al., 2016)**.** Additionally, mRNAs were found to localize with late endosomes in axonal projections and were actively translated on these vesicles (Cioni et al., 2019). These experiments highlight the strength of live imaging approaches in understanding translation regulation within cells.

**4.5. Translation and RNA decay**

Recent advances in single-molecule imaging methods have enabled the effect of translation on mRNA decay to be measured with single-molecule resolution (Cialek et al., 2022; Dave et al., 2023; Kobayashi & Singer, 2022). Dave et al. engineered a TREAT reporter mRNA construct **(Figure 2h)** containing an iron-response element (IRE) (Hentze et al., 1987) in the 5′-UTR to specifically activate translation of this construct by adding iron to the culture media (Dave et al., 2023). Active translation of the reporter mRNA resulted in faster mRNA decay and by combining smFISH, SunTag imaging and mathematical modeling, the study found that translation-dependent decay is mediated by ribosome flux indicating that translation promotes a basal level of mRNA decay. When this approach was applied to a reporter transcript that contained a premature stop codon, it was found that mRNA degradation induced by the nonsense mediated decay surveillance pathway was much faster than destabilization caused by ribosome flux. Interestingly, the cellular abundance of these PTC-containing mRNAs could be best modeled by both an NMD and a ribosome flux component indicating that both pathways were contributing to the destabilization of these transcripts (Hoek et al., 2019). Interestingly, when three let7 or miR21 miRNA binding sites were incorporated into the 3′-UTR, it was found that these miRNAs could destabilize the reporter transcript without prior inhibition of translation. Two other recent studies using single-molecule imaging, however, found that translation inhibition occurs before mRNA decay suggesting that the identity of the specific miRNA, its abundance or the target site within the mRNA could affect the mechanism of miRNA-mediated repression (Cialek et al., 2022; Kobayashi & Singer, 2022).

**5. Challenges in live imaging of RNAs in multicellular systems**

The first efforts to image mRNAs live within a multicellular system were in *Drosophila* oocytes (Forrest & Gavis, 2003; Jaramillo et al., 2008). In the 20 years since live mRNA imaging was first established in multicellular systems, only a few papers from a small number of labs have successfully followed suit. In short, this is because establishing the tools for live imaging requires the optimization of several variables. Moreover, the imaging can be technically challenging and there is no standardized software for analysis so individual labs must establish their own image analysis pipelines. In this section, we discuss the major challenges of setting up live mRNA and translation site imaging in multicellular systems and how to address them.

When designing a live imaging experiment, the first decision is which mRNA to visualize. To date, candidates have been selected based on known localization patterns (*β-actin*) (Lionnet et al., 2011; H. Y. Park et al., 2014), developmentally-induced (*twist*, *eve*) (Bothma et al., 2014; Dufourt et al., 2021), or neuronal activity-induced (*Arc*) (Das et al., 2018; B. H. Lee et al., 2022) expression patterns. Once a candidate transcript is selected, it needs to be engineered to contain MS2 or PP7 stem-loops for mRNA imaging or SunTag epitopes for translation site imaging. There are essentially two approaches: introducing reporter sequences into an endogenous genomic locus or creating reporter transgenes. In the *Drosophila* embryo, many developmentally-regulated genes are expressed in precisely-defined spatial and temporal patterns to establish the blueprint of the adult fly (Gregor et al., 2014) and many of the promoter and enhancer regions driving these expression patterns have been well-characterized. A popular strategy for live mRNA imaging, therefore, has been to clone minimal promoter fragments upstream of a MS2 reporter transcript (Bothma et al., 2014; Dufourt et al., 2021; H. G. Garcia et al., 2013; Lucas et al., 2013; Pimmett et al., 2021). This requires that the promoter has been determined and even then, reporter transgenes lack many endogenous regulatory elements and can fail to fully recapitulate endogenous expression patterns (Bothma et al., 2014; Dufourt et al., 2021; Lucas et al., 2013). Nonetheless, live transcription site imaging of reporter transgenes in the early *Drosophila* embryo has been used for studying transcription dynamics (Bothma et al., 2014; H. G. Garcia et al., 2013) and enhancer-promoter interactions (Lim & Levine, 2021). Dual imaging of reporter genes, one containing an MS2 array and the other containing a PP7 array, has revealed that single enhancers can co-activate the transcription of linked genes separated by relatively large genomic distances (Fukaya et al., 2016; Levo et al., 2022) or even when located on different homologous chromosomes (Heist et al., 2019; Lim et al., 2018).

The alternative to reporter transgenes is to integrate the MS2/PP7 and SunTag arrays into endogenous loci. The integration of MS2 sequences into the endogenous 3′-UTR of the mouse *β-actin* gene led to the first mouse model for live mRNA imaging (Lionnet et al., 2011). Initial imaging experiments were limited to fibroblast cells and hippocampal neurons cultured from the *β-actin*-MS2 mouse, but this was later expanded to brain slices imaged ex-vivo (H. Y. Park et al., 2014). More recently, a second mouse model was generated integrating PP7 stem-loops into the 3′-UTR of the activity regulated gene *Arc* (Das et al., 2018). *Arc-PP7* transcription sites were then imaged within the brain of head-fixed anesthetized mice following contextual fear conditioning representing a landmark achievement for the field (B. H. Lee et al., 2022). The introduction of exogenous sequences into a gene risks altering some aspect of its proper expression, therefore, integration sites must be selected with care to avoid disrupting functionally important sequences. One strategy for minimizing deleterious effects is to target evolutionarily non-conserved regions of the gene in the 3′-UTR, though our current understanding of *cis*-acting elements is limited. Consequently, there will always remain an element of unpredictability that the insertion will alter proper expression and it is imperative to compare endogenous spatial and temporal expression patterns with that of an modified allele.

Visualizing MS2 or PP7 labeled mRNA and SunTag nascent polypeptides requires the binding of fluorescent coat proteins and single chain antibodies, respectively. The fluorescently-labeled coat proteins that bind MS2/PP7 stem-loops contain a nuclear localization signal, which reduces cytoplasmic background fluorescence and enables the labeling of nascent RNAs as soon as they are transcribed. In contrast, the fluorescent single chain antibody fragments that bind epitopes in nascent polypeptides are diffuse throughout the cytoplasm. Expression levels must be controlled so that there are sufficient fluorescent proteins to saturate binding sites on mRNA transcripts or nascent polypeptides, but not so many that background fluorescence precludes the imaging of single molecules. It is trivial to sort cultured cells based on expression levels, but in living animals this is more difficult. The optimization of fluorescent protein expression levels in multicellular systems is achieved via the choice of promoter the transgene is expressed from and may be different for cell types within an animal or tissue

The type of fluorophore attached to the coat protein or single chain antibody is also important. To date, all live mRNA and translation site imaging studies in multicellular systems have employed genetically-encoded fluorescent protein fusions (e.g. GFP, YFP, TagRFPT, mCherry

etc.) (H. Chen et al., 2018; Dufourt et al., 2021; Fukaya et al., 2016; Heist et al., 2019; Hoppe et al., 2020; B. H. Lee et al., 2022; Levo et al., 2022; Lim et al., 2018; H. Y. Park et al., 2014; Pimmett et al., 2021; Trcek et al., 2020). Fluorescent proteins can be prone to aggregation and they photobleach rapidly, which precludes the tracking of single mRNAs or nascent polypeptides over extended timeframes (Dufourt et al., 2021). In contrast, the quantum yield and photostability of organic fluorophores are far superior to that of fluorescent proteins. For this reason, coat protein fusions with the HaloTag, which bind cell-permeable synthetic ligands labeled with organic fluorophores, are a much better option for high-resolution imaging (Grimm et al., 2015). Since the fluorescent ligand must be provided, this approach may not be suitable for all experimental systems.

Imaging single-molecules within a tissue requires an objective with a long working distance and high numerical aperture, but there is a tradeoff between these two parameters. Therefore, in order to achieve high-resolution single-molecule imaging within a complex multicellular system it may be necessary to increase the fluorescence of each single transcript. This can be achieved by increasing the number of stem-loops and hence fluorescent molecules recruited to a single transcript. A 128xMS2 array has been created (Tantale et al., 2016) and used to visualize single transcripts in *Drosophila* embryos (Dufourt et al., 2021). Increasing the size of the stem-loop insert, however, may increase the risk of introducing deleterious effects on some aspect of the mRNAs lifecycle. Indeed, the 128xMS2 array caused the unanticipated persistence of transcription foci through mitosis (Dufourt et al., 2021).

**6. Microscopy developments for imaging RNAs in multicellular systems**

Biological tissues strongly scatter light due to refractive index variations throughout a tissue, which makes capturing the small amount of fluorescence emitted from mRNAs and nascent polypeptides difficult when imaging deep into a tissue. In addition, biological tissues have high levels of autofluorescence, which produces non-specific background signals. In fixed samples, smFISH can be imaged by several imaging modalities. Widefield microscopy is often adequate for thinner tissue samples or sections. Whereas for thicker samples, the optical sectioning capability of confocal microscopy is preferable since fluorescence from beyond the imaging plane is blocked by the pinhole, thereby reducing background fluorescence. Point scanning confocal microscopy, however, can be slow when imaging sizable volumes with multiple fluorescent channels. Multi-point scanning systems, such as the spinning disk, reduce image

acquisition time considerably and may therefore be preferable. smFISH labeled mRNA has also been imaged with structured illumination microscopy (SIM), a super-resolution microscopy technique that uses periodic illumination patterns to achieve high spatial resolution (Heintzmann & Huser, 2017). In the ribonucleoprotein granules of the *Drosophila* germplasm, SIM was used to resolve the spatial distribution and organization of various densely-packed mRNA species (Trcek et al., 2015).

Irrespective of the imaging setup, detecting the fluorescence emitted from single-molecules located deep within tissues remains challenging. Currently the limit for imaging smFISH signals is ~50μm. Imaging beyond this limit has been achieved only via the amplification of the fluorescence signal associated with transcripts coupled with tissue clearing (Shah et al., 2016). Clearing refers to a variety of techniques that aim to alter the optical properties of a tissue in order to improve light propagation by minimizing inherent refractive index mismatches (Richardson et al., 2021). Most clearing protocols aim to remove lipids and water from a sample, replacing them with a solution that approximates the refractive index of the remaining tissue components (Weiss et al., 2021). In order to reduce autofluorescence, tissues can be photobleached prior to hybridization by exposing them to a broadband light source for 24h (Duong & Han, 2013). In mammalian tissues, autofluorescence can also be reduced to some extent by eliminating dietary sources of chlorophyll-containing substrates (H. Y. Park et al., 2014). Alternatively, autofluorescence can simply be avoided by selecting dyes with emission spectra outside of the major wavelengths of tissue autofluorescence.

Live mRNA imaging within multicellular systems is even more challenging than in fixed tissues. To date, live mRNA imaging in multicellular systems has been achieved using either confocal (H. Chen et al., 2018; Dufourt et al., 2021; Fukaya et al., 2016; Heist et al., 2019; Hoppe et al., 2020; Levo et al., 2022; Lim et al., 2018; Pimmett et al., 2021; Trcek et al., 2020) or 2-photon microscopy (B. H. Lee et al., 2022; H. Y. Park et al., 2014). While translation sites have been imaged with light sheet microscopy (Dufourt et al., 2021). 2-photon microscopy can achieve superior imaging depths and is widely used for imaging *in vivo* calcium responses deep within living brains. 2-photon setups have even been miniaturized such that freely moving animals can wear head-mounted scopes and calcium transients can be recorded while the animal performs various behavioral tasks (Zong et al., 2022). Impressively, *Arc-PP7* transcription sites in the brain of head-fixed mice were imaged at a depth of 400μm using a 2-photon setup (B. H. Lee et al., 2022).

**7. Outlook**

While single-molecule methodologies for imaging RNAs in fixed and living cells have made significant progress in recent years, there are still several challenges for the field. Understanding the molecular mechanism used by *trans*-acting factors to regulate gene expression at a single-molecule level is at the forefront of current experiments and the combinatorial action of multiple proteins acting either independently or in complexes is largely unknown. Additionally, experiments in animals are only beginning to shed light on how RNA metabolism is regulated within tissues and how this activity is coordinated between different cell types within an organ or between organs will be an important goal.

**Acknowledgements**

This work was supported by the Novartis Research Foundation (J.A.C), the SNF-NCCR RNA & Disease Network #51NF40-182880 (J.A.C.) and an EMBO fellowship ALTF 793-2021 (E.G.). We apologize to our colleagues whose work we were not able to discuss due to space limitations.

**Figure legends**

**Figure 1.** An overview of methodologies used in single-molecule imaging in fixed cells using A) single-molecule in situ hybridization or B) in-situ sequencing. Approaches to image C) endogenous or D) exogenous reporter RNAs in live cells. Methodologies for imaging E) translation or F) decay of single mRNAs.

**Figure 2.** Single-molecule imaging sheds light on the coupling of the different steps in mRNA metabolism. a) Increased dwell time of the transcription factor (TF) at the gene promoter correlates with increased transcriptional activity. b) Increased proximity of gene loci to nuclear speckles (NSs) leads to increased transcription activity, faster recruitment of splicing factors and more efficient splicing. c) Splicing is mostly co-transcriptional but some introns can be spliced post-transcriptionally. Splicing enhances export and export of some mRNAs can be timed through regulated intron retention. d) The intron 5' end is tethered to transcribing RNAPII and introns are compact particles with 5' and 3' ends in close proximity. e) mRNA export through the

nuclear pore complex is directional, 5'-end first. f) mRNAs can actively be transported to specific locations in the cytoplasm for localized translation. g) Translation can occur in stress granules (SGs). h) mRNA translation promotes decay.

**References**

Abudayyeh, O. O., Gootenberg, J. S., Essletzbichler, P., Han, S., Joung, J., Belanto, J. J., Verdine, V., Cox, D. B. T., Kellner, M. J., Regev, A., Lander, E. S., Voytas, D. F., Ting, A. Y., & Zhang, F. (2017). RNA targeting with CRISPR-Cas13. *Nature*, *550*(7675), 280–284.

Adivarahan, S., Kalhara Abeykoon, A. M., & Zenklusen, D. (2021). Single-molecule imaging suggests compact and spliceosome dependent organization of long introns. In *bioRxiv* (p. 2021.10.27.466141). https://doi.org/10.1101/2021.10.27.466141

Alami, N. H., Smith, R. B., Carrasco, M. A., Williams, L. A., Winborn, C. S., Han, S. S. W., Kiskinis, E., Winborn, B., Freibaum, B. D., Kanagaraj, A., Clare, A. J., Badders, N. M., Bilican, B., Chaum, E., Chandran, S., Shaw, C. E., Eggan, K. C., Maniatis, T., & Taylor, J. P. (2014). Axonal transport of TDP-43 mRNA granules is impaired by ALS-causing mutations. *Neuron*, *81*(3), 536–543.

Alexander, K. A., Coté, A., Nguyen, S. C., Zhang, L., Gholamalamdari, O., Agudelo-Garcia, P., Lin-Shiao, E., Tanim, K. M. A., Lim, J., Biddle, N., Dunagin, M. C., Good, C. R., Mendoza, M. R., Little, S. C., Belmont, A., Joyce, E. F., Raj, A., & Berger, S. L. (2021). p53 mediates target gene association with nuclear speckles for amplified RNA expression. *Molecular Cell*. https://doi.org/10.1016/j.molcel.2021.03.006

Alon, S., Goodwin, D. R., Sinha, A., Wassie, A. T., Chen, F., Daugharthy, E. R., Bando, Y., Kajita, A., Xue, A. G., Marrett, K., Prior, R., Cui, Y., Payne, A. C., Yao, C.-C., Suk, H.-J., Wang, R., Yu, C.-C. J., Tillberg, P., Reginato, P., … Boyden, E. S. (2021). Expansion sequencing: Spatially precise in situ transcriptomics in intact biological systems. *Science*, *371*(6528). https://doi.org/10.1126/science.aax2656

Ashkenazy-Titelman, A., Atrash, M. K., Boocholez, A., Kinor, N., & Shav-Tal, Y. (2022). RNA export through the nuclear pore complex is directional. *Nature Communications*, *13*(1), 5881.

Bahar Halpern, K., Tanami, S., Landen, S., Chapal, M., Szlak, L., Hutzler, A., Nizhberg, A., & Itzkovitz, S. (2015). Bursty gene expression in the intact mammalian liver. *Molecular Cell*, *58*(1), 147–156.

Bashirullah, A., Cooperstock, R. L., & Lipshitz, H. D. (2001). Spatial and temporal control of RNA stability. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(13), 7025–7028.

Battich, N., Stoeger, T., & Pelkmans, L. (2013). Image-based transcriptomics in thousands of single human cells at single-molecule resolution. *Nature Methods*, *10*(11), 1127–1133.

Bertrand, E., Chartrand, P., Schaefer, M., Shenoy, S. M., Singer, R. H., & Long, R. M. (1998). Localization of ASH1 mRNA particles in living yeast. *Molecular Cell*, *2*(4), 437–445.

Bhat, P., Chow, A., Emert, B., Ettlin, O., Quinodoz, S. A., Takei, Y., Huang, W., Blanco, M. R., & Guttman, M. (2023). 3D genome organization around nuclear speckles drives mRNA splicing efficiency. In *bioRxiv* (p. 2023.01.04.522632). https://doi.org/10.1101/2023.01.04.522632

Biswas, J., Li, W., Singer, R. H., & Coleman, R. A. (2021). Imaging Organization of RNA Processing within the Nucleus. *Cold Spring Harbor Perspectives in Biology*, *13*(12). https://doi.org/10.1101/cshperspect.a039453

Boersma, S., Khuperkar, D., Verhagen, B. M. P., Sonneveld, S., Grimm, J. B., Lavis, L. D., & Tanenbaum, M. E. (2019). Multi-Color Single-Molecule Imaging Uncovers Extensive Heterogeneity in mRNA Decoding. *Cell*, *178*(2), 458–472.e19.

Borm, L. E., Mossi Albiach, A., Mannens, C. C. A., Janusauskas, J., Özgün, C., Fernández-García, D., Hodge, R., Castillo, F., Hedin, C. R. H., Villablanca, E. J., Uhlén, P., Lein, E. S., Codeluppi, S., & Linnarsson, S. (2022). Scalable in situ single-cell profiling by

electrophoretic capture of mRNA using EEL FISH. *Nature Biotechnology*. https://doi.org/10.1038/s41587-022-01455-3

Bothma, J. P., Garcia, H. G., Esposito, E., Schlissel, G., Gregor, T., & Levine, M. (2014). Dynamic regulation of eve stripe 2 expression reveals transcriptional bursts in living Drosophila embryos. *Proceedings of the National Academy of Sciences of the United States of America*, *111*(29), 10598–10603.

Brodsky, A. S., & Silver, P. A. (2002). Identifying proteins that affect mRNA localization in living cells. *Methods* , *26*(2), 151–155.

Brody, Y., Neufeld, N., Bieberstein, N., Causse, S. Z., Böhnlein, E.-M., Neugebauer, K. M., Darzacq, X., & Shav-Tal, Y. (2011). The in vivo kinetics of RNA polymerase II elongation during co-transcriptional splicing. *PLoS Biology*, *9*(1), e1000573.

Bullock, S. L., & Ish-Horowicz, D. (2001). Conserved signals and machinery for RNA transport in Drosophila oogenesis and embryogenesis. *Nature*, *414*(6864), 611–616.

Calvet, J. P., & Pederson, T. (1981). Base-pairing interactions between small nuclear RNAs and nuclear RNA precursors as revealed by psoralen cross-linking in vivo. *Cell*, *26*(3 Pt 1), 363–370.

Calvo, L., Ronshaugen, M., & Pettini, T. (2021). smiFISH and embryo segmentation for single-cell multi-gene RNA quantification in arthropods. *Communications Biology*, *4*(1), 352.

Cawte, A. D., Unrau, P. J., & Rueda, D. S. (2020). Live cell imaging of single RNA molecules with fluorogenic Mango II arrays. *Nature Communications*, *11*(1), 1283.

Chapman, E. G., Costantino, D. A., Rabe, J. L., Moon, S. L., Wilusz, J., Nix, J. C., & Kieft, J. S. (2014). The structural basis of pathogenic subgenomic flavivirus RNA (sfRNA) production. *Science*, *344*(6181), 307–310.

Chen, A. K., Behlke, M. A., & Tsourkas, A. (2007). Avoiding false-positive signals with nuclease-vulnerable molecular beacons in single living cells. *Nucleic Acids Research*, *35*(16), e105.

Chen, A. K., Davydenko, O., Behlke, M. A., & Tsourkas, A. (2010). Ratiometric bimolecular beacons for the sensitive detection of RNA in single living cells. *Nucleic Acids Research*, *38*(14), e148.

Chen, F., Tillberg, P. W., & Boyden, E. S. (2015). Optical imaging. Expansion microscopy. *Science*, *347*(6221), 543–548.

Chen, H., Levo, M., Barinov, L., Fujioka, M., Jaynes, J. B., & Gregor, T. (2018). Dynamic interplay between enhancer-promoter topology and gene activity. *Nature Genetics*, *50*(9), 1296–1303.

Chen, J., Nikolaitchik, O., Singh, J., Wright, A., Bencsics, C. E., Coffin, J. M., Ni, N., Lockett, S., Pathak, V. K., & Hu, W.-S. (2009). High efficiency of HIV-1 genomic RNA packaging and heterozygote formation revealed by single virion analysis. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(32), 13535–13540.

Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S., & Zhuang, X. (2015). RNA imaging. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science*, *348*(6233), aaa6090.

Chen, X., Sun, Y.-C., Church, G. M., Lee, J. H., & Zador, A. M. (2018). Efficient in situ barcode sequencing using padlock probe-based BaristaSeq. *Nucleic Acids Research*, *46*(4), e22.

Chen, Y., & Belmont, A. S. (2019). Genome organization around nuclear speckles. *Current Opinion in Genetics & Development*, *55*, 91–99.

Chen, Y., Zhang, Y., Wang, Y., Zhang, L., Brinkman, E. K., Adam, S. A., Goldman, R., van Steensel, B., Ma, J., & Belmont, A. S. (2018). Mapping 3D genome organization relative to nuclear compartments using TSA-Seq as a cytological ruler. *The Journal of Cell Biology*, *217*(11), 4025–4048.

Choi, H. M. T., Chang, J. Y., Trinh, L. A., Padilla, J. E., Fraser, S. E., & Pierce, N. A. (2010). Programmable in situ amplification for multiplexed imaging of mRNA expression. *Nature Biotechnology*, *28*(11), 1208–1212.

Choi, H. M. T., Schwarzkopf, M., Fornace, M. E., Acharya, A., Artavanis, G., Stegmaier, J., Cunha, A., & Pierce, N. A. (2018). Third-generation in situ hybridization chain reaction: multiplexed, quantitative, sensitive, versatile, robust. *Development* , *145*(12). https://doi.org/10.1242/dev.165753

Cho, W.-K., Jayanth, N., English, B. P., Inoue, T., Andrews, J. O., Conway, W., Grimm, J. B., Spille, J.-H., Lavis, L. D., Lionnet, T., & Cisse, I. I. (2016). RNA Polymerase II cluster dynamics predict mRNA output in living cells. *eLife*, *5*. https://doi.org/10.7554/eLife.13617

Chubb, J. R., Trcek, T., Shenoy, S. M., & Singer, R. H. (2006). Transcriptional pulsing of a developmental gene. *Current Biology: CB*, *16*(10), 1018–1025.

Cialek, C. A., Galindo, G., Morisaki, T., Zhao, N., Montgomery, T. A., & Stasevich, T. J. (2022). Imaging translational control by Argonaute with single-molecule resolution in live cells. *Nature Communications*, *13*(1), 3345.

Cioni, J.-M., Lin, J. Q., Holtermann, A. V., Koppers, M., Jakobs, M. A. H., Azizi, A., Turner-Bridger, B., Shigeoka, T., Franze, K., Harris, W. A., & Holt, C. E. (2019). Late Endosomes Act as mRNA Translation Platforms and Sustain Mitochondria in Axons. *Cell*, *176*(1), 56–72.e15.

Coté, A., Coté, C., Bayatpour, S., Drexler, H. L., Alexander, K. A., Chen, F., Wassie, A. T., Boyden, E. S., Berger, S., Stirling Churchman, L., & Raj, A. (2021). pre-mRNA spatial distributions suggest that splicing can occur post-transcriptionally. In *bioRxiv* (p. 2020.04.06.028092). https://doi.org/10.1101/2020.04.06.028092

Coulon, A., Ferguson, M. L., de Turris, V., Palangat, M., Chow, C. C., & Larson, D. R. (2014). Kinetic competition during the transcription cycle results in stochastic RNA processing. *eLife*, *3*. https://doi.org/10.7554/eLife.03939

Daigle, N., & Ellenberg, J. (2007). LambdaN-GFP: an RNA reporter system for live-cell imaging. *Nature Methods*, *4*(8), 633–636.

Daneholt, B. (2001). Assembly and transport of a premessenger RNP particle. *Proceedings of*

*the National Academy of Sciences of the United States of America*, *98*(13), 7012–7017.

Dardani, I., Emert, B. L., Goyal, Y., Jiang, C. L., Kaur, A., Lee, J., Rouhanifard, S. H., Alicea, G. M., Fane, M. E., Xiao, M., Herlyn, M., Weeraratna, A. T., & Raj, A. (2022). ClampFISH 2.0 enables rapid, scalable amplified RNA detection in situ. *Nature Methods*, *19*(11), 1403–1410.

Darzacq, X., Shav-Tal, Y., de Turris, V., Brody, Y., Shenoy, S. M., Phair, R. D., & Singer, R. H. (2007). In vivo dynamics of RNA polymerase II transcription. *Nature Structural & Molecular Biology*, *14*(9), 796–806.

Das, S., Moon, H. C., Singer, R. H., & Park, H. Y. (2018). A transgenic mouse for imaging activity-dependent dynamics of endogenous Arc mRNA in live neurons. *Science Advances*, *4*(6), eaar3448.

Dave, P., Roth, G., Griesbach, E., Mateju, D., Hochstoeger, T., & Chao, J. A. (2023). Single-molecule imaging reveals translation-dependent destabilization of mRNAs. *Molecular Cell*. https://doi.org/10.1016/j.molcel.2023.01.013

Dias, A. P., Dufu, K., Lei, H., & Reed, R. (2010). A role for TREX components in the release of spliced mRNA from nuclear speckle domains. *Nature Communications*, *1*, 97.

Dirks, R. M., & Pierce, N. A. (2004). Triggered amplification by hybridization chain reaction. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(43), 15275–15278.

Dolgosheina, E. V., Jeng, S. C. Y., Panchapakesan, S. S. S., Cojocaru, R., Chen, P. S. K., Wilson, P. D., Hawkins, N., Wiggins, P. A., & Unrau, P. J. (2014). RNA mango aptamer-fluorophore: a bright, high-affinity complex for RNA labeling and tracking. *ACS Chemical Biology*, *9*(10), 2412–2420.

Donlin-Asp, P. G., Polisseni, C., Klimek, R., Heckel, A., & Schuman, E. M. (2021). Differential regulation of local mRNA dynamics and translation following long-term potentiation and depression. *Proceedings of the National Academy of Sciences of the United States of*

*America*, *118*(13). https://doi.org/10.1073/pnas.2017578118

Donovan, B. T., Huynh, A., Ball, D. A., Patel, H. P., Poirier, M. G., Larson, D. R., Ferguson, M. L., & Lenstra, T. L. (2019). Live-cell imaging reveals the interplay between transcription factors, nucleosomes, and bursting. *The EMBO Journal*, *38*(12). https://doi.org/10.15252/embj.2018100809

Dopie, J., Sweredoski, M. J., Moradian, A., & Belmont, A. S. (2020). Tyramide signal amplification mass spectrometry (TSA-MS) ratio identifies nuclear speckle proteins. *The Journal of Cell Biology*, *219*(9). https://doi.org/10.1083/jcb.201910207

Dufourt, J., Bellec, M., Trullo, A., Dejean, M., De Rossi, S., Favard, C., & Lagha, M. (2021). Imaging translation dynamics in live embryos reveals spatial heterogeneities. *Science*, *372*(6544), 840–844.

Duong, H., & Han, M. (2013). A multispectral LED array for the reduction of background autofluorescence in brain tissue. *Journal of Neuroscience Methods*, *220*(1), 46–54.

Eng, C.-H. L., Lawson, M., Zhu, Q., Dries, R., Koulena, N., Takei, Y., Yun, J., Cronin, C., Karp, C., Yuan, G.-C., & Cai, L. (2019). Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH. *Nature*, *568*(7751), 235–239.

Faber, G. P., Nadav-Eliyahu, S., & Shav-Tal, Y. (2022). Nuclear speckles - a driving force in gene expression. *Journal of Cell Science*, *135*(13). https://doi.org/10.1242/jcs.259594

Fei, J., Jadaliha, M., Harmon, T. S., Li, I. T. S., Hua, B., Hao, Q., Holehouse, A. S., Reyer, M., Sun, Q., Freier, S. M., Pappu, R. V., Prasanth, K. V., & Ha, T. (2017). Quantitative analysis of multilayer organization of proteins and RNA in nuclear speckles at super resolution. *Journal of Cell Science*, *130*(24), 4180–4192.

Femino, A. M., Fay, F. S., Fogarty, K., & Singer, R. H. (1998). Visualization of Single RNA Transcripts in Situ. In *Science* (Vol. 280, Issue 5363, pp. 585–590). https://doi.org/10.1126/science.280.5363.585

Filonov, G. S., Moon, J. D., Svensen, N., & Jaffrey, S. R. (2014). Broccoli: rapid selection of an

RNA mimic of green fluorescent protein by fluorescence-based selection and directed evolution. *Journal of the American Chemical Society*, *136*(46), 16299–16308.

Forrest, K. M., & Gavis, E. R. (2003). Live imaging of endogenous RNA reveals a diffusion and entrapment mechanism for nanos mRNA localization in Drosophila. *Current Biology: CB*, *13*(14), 1159–1168.

Fukaya, T., Lim, B., & Levine, M. (2016). Enhancer Control of Transcriptional Bursting. *Cell*, *166*(2), 358–368.

Fusco, D., Accornero, N., Lavoie, B., Shenoy, S. M., Blanchard, J.-M., Singer, R. H., & Bertrand, E. (2003). Single mRNA Molecules Demonstrate Probabilistic Movement in Living Mammalian Cells. In *Current Biology* (Vol. 13, Issue 2, pp. 161–167). https://doi.org/10.1016/s0960-9822(02)01436-7

Garcia, H. G., Tikhonov, M., Lin, A., & Gregor, T. (2013). Quantitative imaging of transcription in living Drosophila embryos links polymerase activity to patterning. *Current Biology: CB*, *23*(21), 2140–2145.

Garcia, J. F., & Parker, R. (2015). MS2 coat proteins bound to yeast mRNAs block 5’ to 3' degradation and trap mRNA decay products: implications for the localization of mRNAs by MS2-MCP system. *RNA* , *21*(8), 1393–1395.

Gaspar, I., Wippich, F., & Ephrussi, A. (2017). Enzymatic production of single-molecule FISH and RNA capture probes. *RNA* , *23*(10), 1582–1591.

Goh, J. J. L., Chou, N., Seow, W. Y., Ha, N., Cheng, C. P. P., Chang, Y.-C., Zhao, Z. W., & Chen, K. H. (2020). Highly specific multiplexed RNA imaging in tissues with split-FISH. *Nature Methods*, *17*(7), 689–693.

Gregor, T., Garcia, H. G., & Little, S. C. (2014). The embryo as a laboratory: quantifying transcription in Drosophila. *Trends in Genetics: TIG*, *30*(8), 364–375.

Grimm, J. B., English, B. P., Chen, J., Slaughter, J. P., Zhang, Z., Revyakin, A., Patel, R., Macklin, J. J., Normanno, D., Singer, R. H., Lionnet, T., & Lavis, L. D. (2015). A general

method to improve fluorophores for live-cell and single-molecule microscopy. *Nature Methods*, *12*(3), 244–250, 3 p following 250.

Grünwald, D., & Singer, R. H. (2010). In vivo imaging of labelled endogenous β-actin mRNA during nucleocytoplasmic transport. *Nature*, *467*(7315), 604–607.

Guo, Y. E., Manteiga, J. C., Henninger, J. E., Sabari, B. R., Dall'Agnese, A., Hannett, N. M., Spille, J.-H., Afeyan, L. K., Zamudio, A. V., Shrinivas, K., Abraham, B. J., Boija, A., Decker, T.-M., Rimel, J. K., Fant, C. B., Lee, T. I., Cisse, I. I., Sharp, P. A., Taatjes, D. J., & Young, R. A. (2019). Pol II phosphorylation regulates a switch between transcriptional and splicing condensates. *Nature*, *572*(7770), 543–548.

Gyllborg, D., Langseth, C. M., Qian, X., Choi, E., Salas, S. M., Hilscher, M. M., Lein, E. S., & Nilsson, M. (2020). Hybridization-based in situ sequencing (HybISS) for spatially resolved transcriptomics in human and mouse brain tissue. *Nucleic Acids Research*, *48*(19), e112.

Hall, L. L., Smith, K. P., Byron, M., & Lawrence, J. B. (2006). Molecular anatomy of a speckle. *The Anatomical Record. Part A, Discoveries in Molecular, Cellular, and Evolutionary Biology*, *288*(7), 664–675.

Halstead, J. M., Lionnet, T., Wilbertz, J. H., Wippich, F., Ephrussi, A., Singer, R. H., & Chao, J. A. (2015). Translation. An RNA biosensor for imaging the first round of translation from single cells to living animals. *Science*, *347*(6228), 1367–1671.

Harlen, K. M., Trotta, K. L., Smith, E. E., Mosaheb, M. M., Fuchs, S. M., & Churchman, L. S. (2016). Comprehensive RNA Polymerase II Interactomes Reveal Distinct and Varied Roles for Each Phospho-CTD Residue. *Cell Reports*, *15*(10), 2147–2158.

Hasenson, S. E., & Shav-Tal, Y. (2020). Speculating on the Roles of Nuclear Speckles: How RNA-Protein Nuclear Assemblies Affect Gene Expression. *BioEssays: News and Reviews in Molecular, Cellular and Developmental Biology*, *42*(10), e2000104.

Heintzmann, R., & Huser, T. (2017). Super-Resolution Structured Illumination Microscopy. *Chemical Reviews*, *117*(23), 13890–13908.

Heist, T., Fukaya, T., & Levine, M. (2019). Large distances separate coregulated genes in living *Drosophila* embryos. *Proceedings of the National Academy of Sciences of the United States of America*, *116*(30), 15062–15067.

Hentze, M. W., Caughman, S. W., Rouault, T. A., Barriocanal, J. G., Dancis, A., Harford, J. B., & Klausner, R. D. (1987). Identification of the iron-responsive element for the translational regulation of human ferritin mRNA. *Science*, *238*(4833), 1570–1573.

Hochberg-Laufer, H., Neufeld, N., Brody, Y., Nadav-Eliyahu, S., Ben-Yishay, R., & Shav-Tal, Y. (2019). Availability of splicing factors in the nucleoplasm can regulate the release of mRNA from the gene after transcription. *PLoS Genetics*, *15*(11), e1008459.

Hocine, S., Raymond, P., Zenklusen, D., Chao, J. A., & Singer, R. H. (2013). Single-molecule analysis of gene expression using two-color RNA labeling in live yeast. *Nature Methods*, *10*(2), 119–121.

Hoek, T. A., Khuperkar, D., Lindeboom, R. G. H., Sonneveld, S., Verhagen, B. M. P., Boersma, S., Vermeulen, M., & Tanenbaum, M. E. (2019). Single-Molecule Imaging Uncovers Rules Governing Nonsense-Mediated mRNA Decay. *Molecular Cell*, *75*(2), 324–339.e11.

Hoppe, C., Bowles, J. R., Minchington, T. G., Sutcliffe, C., Upadhyai, P., Rattray, M., & Ashe, H. L. (2020). Modulation of the Promoter Activation Rate Dictates the Transcriptional Response to Graded BMP Signaling Levels in the Drosophila Embryo. *Developmental Cell*, *54*(6), 727–741.e7.

Horvathova, I., Voigt, F., Kotrys, A. V., Zhan, Y., Artus-Revel, C. G., Eglinger, J., Stadler, M. B., Giorgetti, L., & Chao, J. A. (2017). The Dynamics of mRNA Turnover Revealed by Single-Molecule Imaging in Single Cells. *Molecular Cell*, *68*(3), 615–625.e9.

Hu, Y., Plutz, M., & Belmont, A. S. (2010). Hsp70 gene association with nuclear speckles is Hsp70 promoter specific. *The Journal of Cell Biology*, *191*(4), 711–719.

Ilik, İ. A., Malszycki, M., Lübke, A. K., Schade, C., Meierhofer, D., & Aktaş, T. (2020). SON and SRRM2 are essential for nuclear speckle formation. *eLife*, *9*.

https://doi.org/10.7554/eLife.60579

Jaramillo, A. M., Weil, T. T., Goodhouse, J., Gavis, E. R., & Schupbach, T. (2008). The dynamics of fluorescently labeled endogenous gurken mRNA in Drosophila. *Journal of Cell Science*, *121*(Pt 6), 887–894.

Ke, R., Mignardi, M., Pacureanu, A., Svedlund, J., Botling, J., Wählby, C., & Nilsson, M. (2013). In situ sequencing for RNA analysis in preserved tissue and cells. *Nature Methods*, *10*(9), 857–860.

Khanna, N., Hu, Y., & Belmont, A. S. (2014). HSP70 transgene directed motion to nuclear speckles facilitates heat shock activation. *Current Biology: CB*, *24*(10), 1138–1144.

Kim, S. H., Vieira, M., Kim, H.-J., Kesawat, M. S., & Park, H. Y. (2019). MS2 Labeling of Endogenous Beta-Actin mRNA Does Not Result in Stabilization of Degradation Intermediates. *Molecules and Cells*, *42*(4), 356–362.

Kishi, J. Y., Lapan, S. W., Beliveau, B. J., West, E. R., Zhu, A., Sasaki, H. M., Saka, S. K., Wang, Y., Cepko, C. L., & Yin, P. (2019). SABER amplifies FISH: enhanced multiplexed imaging of RNA and DNA in cells and tissues. *Nature Methods*, *16*(6), 533–544.

Klimek, R., Donlin-Asp, P. G., Polisseni, C., Hanff, V., Schuman, E. M., & Heckel, A. (2021). Visible light-activatable Q-dye molecular beacons for long-term mRNA monitoring in neurons. *Chemical Communications* , *57*(94), 12683–12686.

Kobayashi, H., & Singer, R. H. (2022). Single-molecule imaging of microRNA-mediated gene silencing in cells. *Nature Communications*, *13*(1), 1435.

Kong, K. Y. S., Jeng, S. C. Y., Rayyan, B., & Unrau, P. J. (2021). RNA Peach and Mango: Orthogonal two-color fluorogenic aptamers distinguish nearly identical ligands. *RNA* , *27*(5), 604–615.

Lamond, A. I., & Spector, D. L. (2003). Nuclear speckles: a model for nuclear organelles. *Nature Reviews. Molecular Cell Biology*, *4*(8), 605–612.

Larson, D. R., Zenklusen, D., Wu, B., Chao, J. A., & Singer, R. H. (2011). Real-time observation

of transcription initiation and elongation on an endogenous yeast gene. *Science*, *332*(6028), 475–478.

Lee, B. H., Shim, J. Y., Moon, H. C., Kim, D. W., Kim, J., Yook, J. S., Kim, J., & Park, H. Y. (2022). Real-time visualization of mRNA synthesis during memory formation in live mice. *Proceedings of the National Academy of Sciences of the United States of America*, *119*(27), e2117076119.

Lee, C.-Y., & Myong, S. (2021). Probing steps in DNA transcription using single-molecule methods. *The Journal of Biological Chemistry*, *297*(3), 101086.

Lee, J. H., Daugharthy, E. R., Scheiman, J., Kalhor, R., Ferrante, T. C., Terry, R., Turczyk, B. M., Yang, J. L., Lee, H. S., Aach, J., Zhang, K., & Church, G. M. (2015). Fluorescent in situ sequencing (FISSEQ) of RNA for gene expression profiling in intact cells and tissues. *Nature Protocols*, *10*(3), 442–458.

Levo, M., Raimundo, J., Bing, X. Y., Sisco, Z., Batut, P. J., Ryabichko, S., Gregor, T., & Levine, M. S. (2022). Transcriptional coupling of distant regulatory genes in living embryos. *Nature*, *605*(7911), 754–760.

Levsky, J. M., Shenoy, S. M., Pezo, R. C., & Singer, R. H. (2002). Single-cell gene expression profiling. *Science*, *297*(5582), 836–840.

Lewis, J. D., & Tollervey, D. (2000). Like attracts like: getting RNA processing together in the nucleus. *Science*, *288*(5470), 1385–1389.

Liao, Y.-C., Fernandopulle, M. S., Wang, G., Choi, H., Hao, L., Drerup, C. M., Patel, R., Qamar, S., Nixon-Abell, J., Shen, Y., Meadows, W., Vendruscolo, M., Knowles, T. P. J., Nelson, M., Czekalska, M. A., Musteikyte, G., Gachechiladze, M. A., Stephens, C. A., Pasolli, H. A., … Ward, M. E. (2019). RNA Granules Hitchhike on Lysosomes for Long-Distance Transport, Using Annexin A11 as a Molecular Tether. *Cell*, *179*(1), 147–164.e20.

Li, J., Dong, A., Saydaminova, K., Chang, H., Wang, G., Ochiai, H., Yamamoto, T., & Pertsinidis, A. (2019). Single-Molecule Nanoscopy Elucidates RNA Polymerase II Transcription at

Single Genes in Live Cells. *Cell*, *178*(2), 491–506.e28.

Lim, B., Heist, T., Levine, M., & Fukaya, T. (2018). Visualization of Transvection in Living Drosophila Embryos. *Molecular Cell*, *70*(2), 287–296.e6.

Lim, B., & Levine, M. S. (2021). Enhancer-promoter communication: hubs or loops? *Current Opinion in Genetics & Development*, *67*, 5–9.

Lionnet, T., Czaplinski, K., Darzacq, X., Shav-Tal, Y., Wells, A. L., Chao, J. A., Park, H. Y., de Turris, V., Lopez-Jones, M., & Singer, R. H. (2011). A transgenic mouse for in vivo detection of endogenous labeled mRNA. *Nature Methods*, *8*(2), 165–170.

Liu, Z., & Tjian, R. (2018). Visualizing transcription factor dynamics in living cells. *The Journal of Cell Biology*, *217*(4), 1181–1191.

Li, W., Maekiniemi, A., Sato, H., Osman, C., & Singer, R. H. (2022). An improved imaging system that corrects MS2-induced RNA destabilization. *Nature Methods*, *19*(12), 1558–1562.

Lubeck, E., Coskun, A. F., Zhiyentayev, T., Ahmad, M., & Cai, L. (2014). Single-cell in situ RNA profiling by sequential hybridization [Review of *Single-cell in situ RNA profiling by sequential hybridization*]. *Nature Methods*, *11*(4), 360–361.

Lucas, T., Ferraro, T., Roelens, B., De Las Heras Chanes, J., Walczak, A. M., Coppey, M., & Dostatni, N. (2013). Live imaging of bicoid-dependent transcription in Drosophila embryos. *Current Biology: CB*, *23*(21), 2135–2139.

Maïno, N., Hauling, T., Cappi, G., Madaboosi, N., Dupouy, D. G., & Nilsson, M. (2019). A microfluidic platform towards automated multiplexed in situ sequencing. *Scientific Reports*, *9*(1), 3542.

Martin, R. M., Rino, J., Carvalho, C., Kirchhausen, T., & Carmo-Fonseca, M. (2013). Live-cell visualization of pre-mRNA splicing with single-molecule sensitivity. *Cell Reports*, *4*(6), 1144–1155.

Mateju, D., Eichenberger, B., Voigt, F., Eglinger, J., Roth, G., & Chao, J. A. (2020).

Single-Molecule Imaging Reveals Translation of mRNAs Localized to Stress Granules. *Cell*, *183*(7), 1801–1812.e13.

Mhlanga, M. M., Vargas, D. Y., Fung, C. W., Kramer, F. R., & Tyagi, S. (2005). tRNA-linked molecular beacons for imaging mRNAs in the cytoplasm of living cells. *Nucleic Acids Research*, *33*(6), 1902–1912.

Middleton, S. A., Eberwine, J., & Kim, J. (2019). Comprehensive catalog of dendritically localized mRNA isoforms from sub-cellular sequencing of single mouse neurons. *BMC Biology*, *17*(1), 5.

Misteli, T., Cáceres, J. F., & Spector, D. L. (1997). The dynamics of a pre-mRNA splicing factor in living cells. *Nature*, *387*(6632), 523–527.

Mor, A., Suliman, S., Ben-Yishay, R., Yunger, S., Brody, Y., & Shav-Tal, Y. (2010). Dynamics of single mRNP nucleocytoplasmic transport and export through the nuclear pore in living cells. *Nature Cell Biology*, *12*(6), 543–552.

Morisaki, T., Lyon, K., DeLuca, K. F., DeLuca, J. G., English, B. P., Zhang, Z., Lavis, L. D., Grimm, J. B., Viswanathan, S., Looger, L. L., Lionnet, T., & Stasevich, T. J. (2016). Real-time quantification of single RNA translation dynamics in living cells. *Science*, *352*(6292), 1425–1429.

Nelles, D. A., Fang, M. Y., O'Connell, M. R., Xu, J. L., Markmiller, S. J., Doudna, J. A., & Yeo, G. W. (2016). Programmable RNA Tracking in Live Cells with CRISPR/Cas9. *Cell*, *165*(2), 488–496.

Nojima, T., Rebelo, K., Gomes, T., Grosso, A. R., Proudfoot, N. J., & Carmo-Fonseca, M. (2018). RNA Polymerase II Phosphorylated on CTD Serine 5 Interacts with the Spliceosome during Co-transcriptional Splicing. *Molecular Cell*, *72*(2), 369–379.e4.

O'Connell, M. R., Oakes, B. L., Sternberg, S. H., East-Seletsky, A., Kaplan, M., & Doudna, J. A. (2014). Programmable RNA recognition and cleavage by CRISPR/Cas9. *Nature*, *516*(7530), 263–266.

Paige, J. S., Wu, K. Y., & Jaffrey, S. R. (2011). RNA mimics of green fluorescent protein. *Science*, *333*(6042), 642–646.

Park, H. Y., Lim, H., Yoon, Y. J., Follenzi, A., Nwokafor, C., Lopez-Jones, M., Meng, X., & Singer, R. H. (2014). Visualization of dynamics of single endogenous mRNA labeled in live mouse. *Science*, *343*(6169), 422–424.

Park, S. Y., Moon, H. C., & Park, H. Y. (2020). Live-cell imaging of single mRNA dynamics using split superfolder green fluorescent proteins with minimal background. *RNA* , *26*(1), 101–109.

Pichon, X., Bastide, A., Safieddine, A., Chouaib, R., Samacoits, A., Basyuk, E., Peter, M., Mueller, F., & Bertrand, E. (2016). Visualization of single endogenous polysomes reveals the dynamics of translation in live human cells. *The Journal of Cell Biology*, *214*(6), 769–781.

Pimmett, V. L., Dejean, M., Fernandez, C., Trullo, A., Bertrand, E., Radulescu, O., & Lagha, M. (2021). Quantitative imaging of transcription in living Drosophila embryos reveals the impact of core promoter motifs on promoter state dynamics. *Nature Communications*, *12*(1), 4504.

Politz, J. C., Tuft, R. A., Pederson, T., & Singer, R. H. (1999). Movement of nuclear poly(A) RNA throughout the interchromatin space in living cells. *Current Biology: CB*, *9*(6), 285–291.

Quinodoz, S. A., Bhat, P., Chovanec, P., Jachowicz, J. W., Ollikainen, N., Detmar, E., Soehalim, E., & Guttman, M. (2022). SPRITE: a genome-wide method for mapping higher-order 3D interactions in the nucleus using combinatorial split-and-pool barcoding. *Nature Protocols*, *17*(1), 36–75.

Raj, A., Peskin, C. S., Tranchina, D., Vargas, D. Y., & Tyagi, S. (2006). Stochastic mRNA synthesis in mammalian cells. *PLoS Biology*, *4*(10), e309.

Raj, A., van den Bogaard, P., Rifkin, S. A., van Oudenaarden, A., & Tyagi, S. (2008). Imaging individual mRNA molecules using multiple singly labeled probes. *Nature Methods*, *5*(10), 877–879.

Ran, F. A., Hsu, P. D., Wright, J., Agarwala, V., Scott, D. A., & Zhang, F. (2013). Genome engineering using the CRISPR-Cas9 system. *Nature Protocols*, *8*(11), 2281–2308.

Richardson, D. S., Guan, W., Matsumoto, K., Pan, C., Chung, K., Ertürk, A., Ueda, H. R., & Lichtman, J. W. (2021). TISSUE CLEARING. *Nature Reviews. Methods Primers*, *1*(1). https://doi.org/10.1038/s43586-021-00080-9

Rouhanifard, S. H., Mellis, I. A., Dunagin, M., Bayatpour, S., Jiang, C. L., Dardani, I., Symmons, O., Emert, B., Torre, E., Cote, A., Sullivan, A., Stamatoyannopoulos, J. A., & Raj, A. (2018). ClampFISH detects individual nucleic acid molecules using click chemistry-based amplification. *Nature Biotechnology*. https://doi.org/10.1038/nbt.4286

Safieddine, A., Coleno, E., Lionneton, F., Traboulsi, A.-M., Salloum, S., Lecellier, C.-H., Gostan, T., Georget, V., Hassen-Khodja, C., Imbert, A., Mueller, F., Walter, T., Peter, M., & Bertrand, E. (2023). HT-smFISH: a cost-effective and flexible workflow for high-throughput single-molecule RNA imaging. *Nature Protocols*, *18*(1), 157–187.

Shah, S., Lubeck, E., Schwarzkopf, M., He, T.-F., Greenbaum, A., Sohn, C. H., Lignell, A., Choi, H. M. T., Gradinaru, V., Pierce, N. A., & Cai, L. (2016). Single-molecule RNA detection at depth by hybridization chain reaction and tissue hydrogel embedding and clearing. *Development* , *143*(15), 2862–2867.

Shah, S., Takei, Y., Zhou, W., Lubeck, E., Yun, J., Eng, C.-H. L., Koulena, N., Cronin, C., Karp, C., Liaw, E. J., Amin, M., & Cai, L. (2018). Dynamics and Spatial Genomics of the Nascent Transcriptome by Intron seqFISH. *Cell*, *174*(2), 363–376.e16.

Shav-Tal, Y., Darzacq, X., Shenoy, S. M., Fusco, D., Janicki, S. M., Spector, D. L., & Singer, R. H. (2004). Dynamics of single mRNPs in nuclei of living cells. *Science*, *304*(5678), 1797–1800.

Spector, D. L., & Lamond, A. I. (2011). Nuclear speckles. *Cold Spring Harbor Perspectives in Biology*, *3*(2). https://doi.org/10.1101/cshperspect.a000646

Su, J.-H., Zheng, P., Kinrot, S. S., Bintu, B., & Zhuang, X. (2020). Genome-Scale Imaging of the

3D Organization and Transcriptional Activity of Chromatin. *Cell*, *182*(6), 1641–1659.e26.

Sunbul, M., Lackner, J., Martin, A., Englert, D., Hacene, B., Grün, F., Nienhaus, K., Nienhaus, G. U., & Jäschke, A. (2021). Super-resolution RNA imaging using a rhodamine-binding aptamer with fast exchange kinetics. *Nature Biotechnology*, *39*(6), 686–690.

Takei, Y., Yun, J., Zheng, S., Ollikainen, N., Pierson, N., White, J., Shah, S., Thomassie, J., Suo, S., Eng, C.-H. L., Guttman, M., Yuan, G.-C., & Cai, L. (2021). Integrated spatial genomics reveals global architecture of single nuclei. *Nature*, *590*(7845), 344–350.

Tanenbaum, M. E., Gilbert, L. A., Qi, L. S., Weissman, J. S., & Vale, R. D. (2014). A protein-tagging system for signal amplification in gene expression and fluorescence imaging. *Cell*, *159*(3), 635–646.

Tantale, K., Mueller, F., Kozulic-Pirher, A., Lesne, A., Victor, J.-M., Robert, M.-C., Capozi, S., Chouaib, R., Bäcker, V., Mateos-Langerak, J., Darzacq, X., Zimmer, C., Basyuk, E., & Bertrand, E. (2016). A single-molecule view of transcription reveals convoys of RNA polymerases and multi-scale bursting. *Nature Communications*, *7*, 12248.

Trcek, T., Douglas, T. E., Grosch, M., Yin, Y., Eagle, W. V. I., Gavis, E. R., Shroff, H., Rothenberg, E., & Lehmann, R. (2020). Sequence-Independent Self-Assembly of Germ Granule mRNAs into Homotypic Clusters. *Molecular Cell*, *78*(5), 941–950.e12.

Trcek, T., Grosch, M., York, A., Shroff, H., Lionnet, T., & Lehmann, R. (2015). Drosophila germ granules are structured and contain homotypic mRNA clusters. *Nature Communications*, *6*, 7962.

Tsanov, N., Samacoits, A., Chouaib, R., Traboulsi, A.-M., Gostan, T., Weber, C., Zimmer, C., Zibara, K., Walter, T., Peter, M., Bertrand, E., & Mueller, F. (2016). smiFISH and FISH-quant - a flexible single RNA detection approach with super-resolution capability. *Nucleic Acids Research*, *44*(22), e165.

Tuerk, C., & Gold, L. (1990). Systematic Evolution of Ligands by Exponential Enrichment: RNA Ligands to Bacteriophage T4 DNA Polymerase. In *Science* (Vol. 249, Issue 4968, pp.

505–510). https://doi.org/10.1126/science.2200121

Tutucci, E., Vera, M., Biswas, J., Garcia, J., Parker, R., & Singer, R. H. (2018). An improved MS2 system for accurate reporting of the mRNA life cycle. *Nature Methods*, *15*(1), 81–89.

Vargas, D. Y., Shah, K., Batish, M., Levandoski, M., Sinha, S., Marras, S. A. E., Schedl, P., & Tyagi, S. (2011). Single-molecule imaging of transcriptionally coupled and uncoupled splicing. *Cell*, *147*(5), 1054–1065.

Viswanathan, S., Williams, M. E., Bloss, E. B., Stasevich, T. J., Speer, C. M., Nern, A., Pfeiffer, B. D., Hooks, B. M., Li, W.-P., English, B. P., Tian, T., Henry, G. L., Macklin, J. J., Patel, R., Gerfen, C. R., Zhuang, X., Wang, Y., Rubin, G. M., & Looger, L. L. (2015). High-performance probes for light and electron microscopy. *Nature Methods*, *12*(6), 568–576.

Wang, C., Han, B., Zhou, R., & Zhuang, X. (2016). Real-Time Imaging of Translation on Single mRNA Transcripts in Live Cells. *Cell*, *165*(4), 990–1001.

Wang, X., Allen, W. E., Wright, M. A., Sylwestrak, E. L., Samusik, N., Vesuna, S., Evans, K., Liu, C., Ramakrishnan, C., Liu, J., Nolan, G. P., Bava, F.-A., & Deisseroth, K. (2018). Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science*, *361*(6400). https://doi.org/10.1126/science.aat5691

Wang, Y., Eddison, M., Fleishman, G., Weigert, M., Xu, S., Wang, T., Rokicki, K., Goina, C., Henry, F. E., Lemire, A. L., Schmidt, U., Yang, H., Svoboda, K., Myers, E. W., Saalfeld, S., Korff, W., Sternson, S. M., & Tillberg, P. W. (2021). EASI-FISH for thick tissue defines lateral hypothalamus spatio-molecular organization. *Cell*, *184*(26), 6361–6377.e24.

Wan, Y., Anastasakis, D. G., Rodriguez, J., Palangat, M., Gudla, P., Zaki, G., Tandon, M., Pegoraro, G., Chow, C. C., Hafner, M., & Larson, D. R. (2021). Dynamic imaging of nascent RNA reveals general principles of transcription dynamics and stochastic splice site selection. *Cell*, *184*(11), 2878–2895.e20.

Weil, T. T., Forrest, K. M., & Gavis, E. R. (2006). Localization of bicoid mRNA in late oocytes is

maintained by continual active transport. *Developmental Cell*, *11*(2), 251–262.

Weiss, K. R., Voigt, F. F., Shepherd, D. P., & Huisken, J. (2021). Tutorial: practical considerations for tissue clearing and imaging. *Nature Protocols*, *16*(6), 2732–2748.

Wu, B., Chen, J., & Singer, R. H. (2014). Background free imaging of single mRNAs in live cells using split fluorescent proteins. *Scientific Reports*, *4*, 3615.

Wu, B., Eliscovich, C., Yoon, Y. J., & Singer, R. H. (2016). Translation dynamics of single mRNAs in live cells and neurons. *Science*, *352*(6292), 1430–1435.

Wu, B., Miskolci, V., Sato, H., Tutucci, E., Kenworthy, C. A., Donnelly, S. K., Yoon, Y. J., Cox, D., Singer, R. H., & Hodgson, L. (2015). Synonymous modification results in high-fidelity gene expression of repetitive protein and nucleotide sequences. *Genes & Development*, *29*(8), 876–886.

Xia, C., Babcock, H. P., Moffitt, J. R., & Zhuang, X. (2019). Multiplexed detection of RNA using MERFISH and branched DNA amplification. *Scientific Reports*, *9*(1), 7721.

Yang, L.-Z., Gao, B.-Q., Huang, Y., Wang, Y., Yang, L., & Chen, L.-L. (2022). Multi-color RNA imaging with CRISPR-Cas13b systems in living cells. *Cell Insight*, *1*(4), 100044.

Yang, L.-Z., Wang, Y., Li, S.-Q., Yao, R.-W., Luan, P.-F., Wu, H., Carmichael, G. G., & Chen, L.-L. (2019). Dynamic Imaging of RNA in Living Cells by CRISPR-Cas13 Systems. *Molecular Cell*, *76*(6), 981–997.e7.

Yan, X., Hoek, T. A., Vale, R. D., & Tanenbaum, M. E. (2016). Dynamics of Translation of Single mRNA Molecules In Vivo. *Cell*, *165*(4), 976–989.

Zenklusen, D., Larson, D. R., & Singer, R. H. (2008). Single-RNA counting reveals alternative modes of gene expression in yeast. *Nature Structural & Molecular Biology*, *15*(12), 1263–1271.

Zhang, L., Zhang, Y., Chen, Y., Gholamalamdari, O., Wang, Y., Ma, J., & Belmont, A. S. (2020). TSA-seq reveals a largely conserved genome organization relative to nuclear speckles with small position changes tightly correlated with gene expression changes. *Genome*

*Research*, *31*(2), 251–264.

Zhang, S., Aibara, S., Vos, S. M., Agafonov, D. E., Lührmann, R., & Cramer, P. (2021). Structure of a transcribing RNA polymerase II-U1 snRNP complex. *Science*, *371*(6526), 305–309.

Zhao, N., Kamijo, K., Fox, P. D., Oda, H., Morisaki, T., Sato, Y., Kimura, H., & Stasevich, T. J. (2019). A genetically encoded probe for imaging nascent and mature HA-tagged proteins in vivo. *Nature Communications*, *10*(1), 2947.

Zimyanin, V. L., Belaya, K., Pecreaux, J., Gilchrist, M. J., Clark, A., Davis, I., & St Johnston, D. (2008). In vivo imaging of oskar mRNA transport reveals the mechanism of posterior localization. *Cell*, *134*(5), 843–853.

Zong, W., Obenhaus, H. A., Skytøen, E. R., Eneqvist, H., de Jong, N. L., Vale, R., Jorge, M. R., Moser, M.-B., & Moser, E. I. (2022). Large-scale two-photon calcium imaging in freely moving mice. *Cell*, *185*(7), 1240–1256.e30.

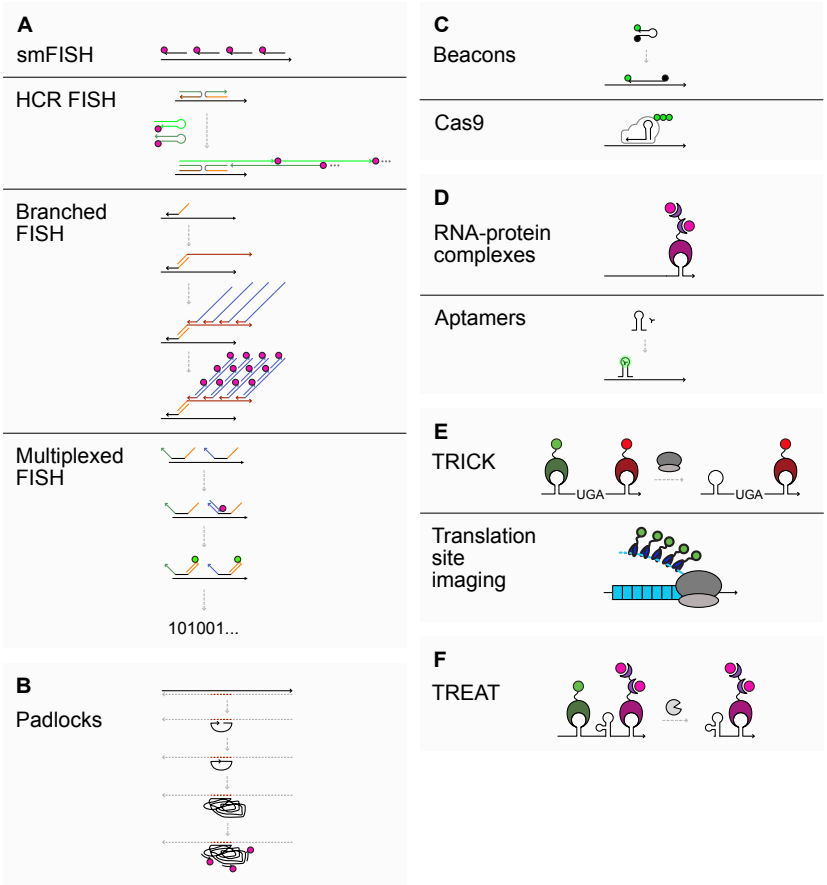



**Figure 1.** An overview of methodologies used in single-molecule imaging in fixed cells using **A)** single-molecule in situ hybridization or **B)** in-situ sequencing. Approaches to image **C)** endogenous or **D)** exogenous reporter RNAs in live cells. Methodologies for imaging **E)** translation or **F)** decay of single mRNAs.

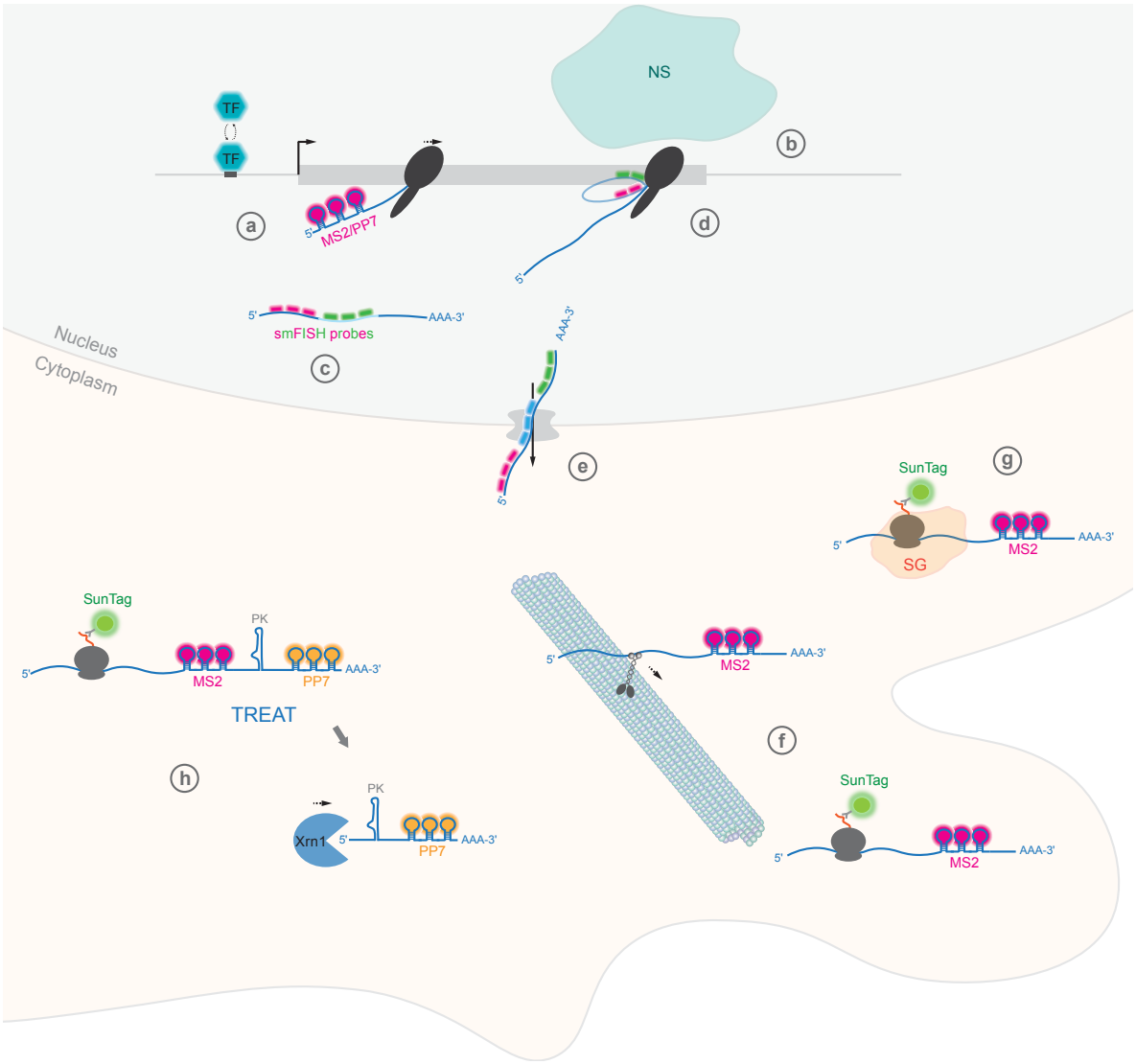



**Figure 2. Single-molecule imaging sheds light on the coupling of the different steps in mRNA metabolism. a)** Increased dwell time of the transcription factor (TF) at the gene promoter correlates with increased transcriptional activity. **b)** Increased proximity of gene loci to nuclear speckles (NSs) leads to increased transcription activity, faster recruitment of splicing factors and more efficient splicing. **c)** Splicing is mostly co-transcriptional but some introns can be spliced post-transcriptionally. Splicing enhances export and export of some mRNAs can be timed through regulated intron retention. **d)** The intron 5' end is tethered to transcribing RNAPII and introns are compact particles with 5' and 3' ends in close proximity. **e)** mRNA export through the nuclear pore complex is directional, 5'-end first. **f)** mRNAs can actively be transported to specific locations in the cytoplasm for localized translation. **g)** Translation can occur in stress granules (SGs). **h)** mRNA translation promotes decay.

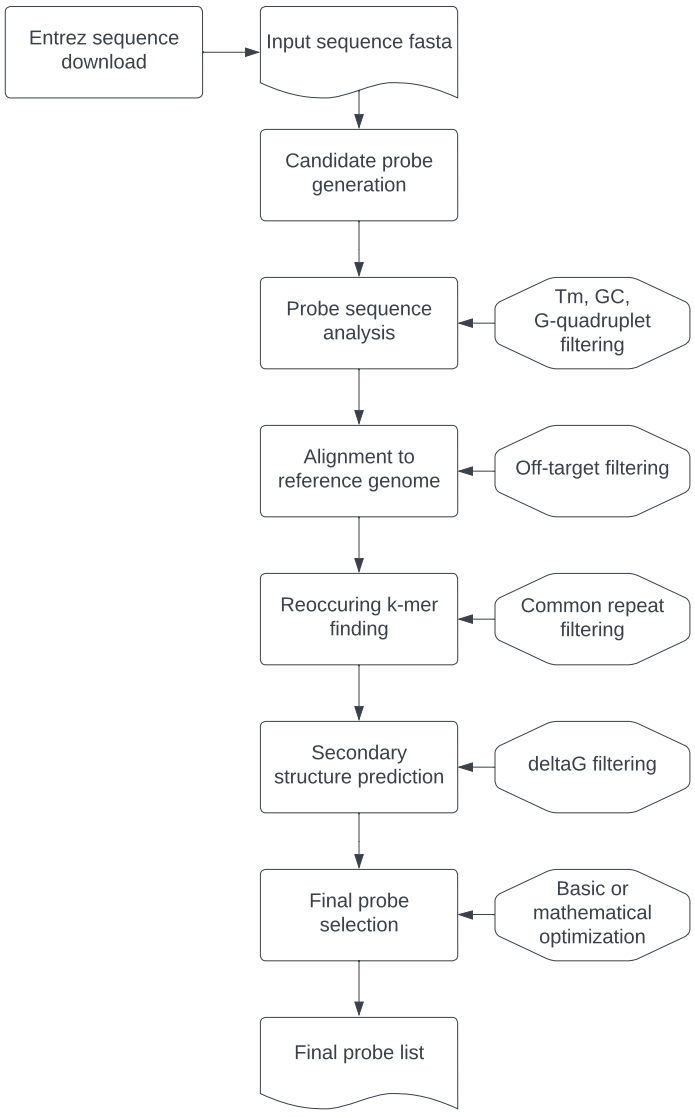
# B

# eFISHent: an efficient RNA smFISH probe design tool

**RNA fluorescence in-situ hybridization (RNA FISH) is a widely used technique enabling the visualization and localization of RNA in fixed cells or tissues. RNA FISH uses a set of fluorescently labeled oligonucleotide probes that specifically bind to the gene of interest. Here we present eFISHent, a CLI-based tool to allow non-experts to design specific probes with an unprecedented level of customizability.**

The development of single-molecule fluorescence in situ hybridization (smFISH) has revolutionized the study of gene expression at the cellular level. This technique enables the detection and quantification of individual RNA molecules in situ, allowing for the examination of RNA expression patterns and dynamics with unprecedented resolution. However, designing and optimizing the RNA probes required for smFISH can be a time-consuming and challenging process. Here, we introduce eFISHent, an efficient RNA smFISH probe design tool that makes generating optimized probe sets for any RNA target of interest simpler than ever before. eFISHent has been made available as a simple CLI-based tool and is accessible as a one-line installation using conda (available through bioconda).

There are two main established probe designers: Stellaris [C1] and OligoMiner [C2]. While Stellaris offers a straightforward web interface for probe design, it lacks significantly in terms of customizability regarding probe lengths, custom genomes, endogenous/exogenous sequences, and other probe parameters. Additionally, the closed-source nature of Stellaris makes understanding why certain probes have been chosen more difficult. OligoMiner [C2] and its online interface OligoMinderApp [C6] are open-sourced but are based on out-of-date Python and third-party package versions. Furthermore, some key features such as off-target expression weighting and multithreading are lacking. eFISHent's modern workflow and easy interface process contribute to making eFISHent a valuable addition to the toolkit of researchers interested in studying gene expression at the single-molecule level.

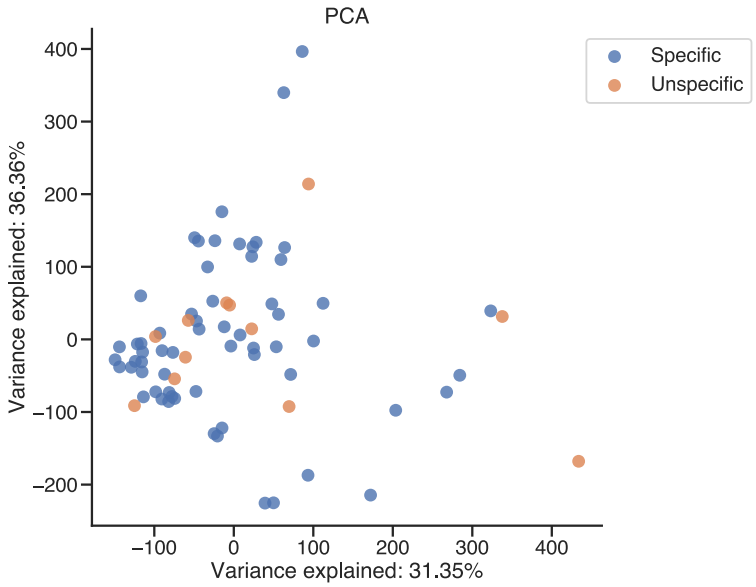



**Appendix Figure B1:** Diagram outlining the basic steps in the eFISHent probe generation workflow

eFISHent works by iteratively selecting probes passing various filtering steps as outlined in Figure B1. Specifically, a list of all candidate probes is generated from an input FASTA file containing the gene sequence. This sequence file can be passed manually or downloaded automatically from NCBI when providing a gene and species name. Subsequently, the first round of filtering removes any probes not passing basic sequence-specific criteria including melting temperature as given formamide and salt concentrations, GC content, and G-quadruplets. After this initial filtering, probes are aligned to the reference genome using bowtie [C3] and candidates with off-targets are removed. In case of shorter genes or if off-targets are unavoidable, off-targets can be weighted using a encode count table to remove highly expressed genes. Next, the targets are divided into short k-mers and discarded if they appear above a determined threshold in the reference genome using Jellyfish [C4]. Lastly, the secondary structure of each candidate is predicted using a nearest neighbor thermodynamic model [C5] and filtered if the free energy is too high which could result in motifs hindering hybridization. This gives the set of all viable candidates which are still overlapping. The final step is to use mathematical or greedy optimization to maximize probe non-overlapping coverage across the gene sequence. This workflow is implemented using Python and is wrapped in a simple CLI. It uses multithreading wherever possible improving runtime with little additional overhead, making it well-suited for high-throughput design. An example command used to run eFISHent is shown below:

```
eFISHent\
        --reference-genome ./hg-38.fa\
        --gene-name "norad"\
        --organism-name "homo sapiens"\
        --is-plus-strand True\
        --min-length 45\
        --max-length 50\
        --formamide-concentration 45\
        --threads 8
```

Here, probes targeting the human gene "norad" of length 45-50 nucleotides are designed at a 45% formamide hybridization buffer concentration. Since no gene sequence is provided directly, the sequence is downloaded from Entrez. While eFISHent allows for extensive customization of parameters, one open question regards the optimal set of values for a given gene of interest. To address this problem, a database of more than 70 probe sets containing 2500 probes has been created. Each probe set was scored from 0 to 6 based on specificity and sensitivity in fixed-cell experiments. Next, all filtering parameters in the pipeline were




**Appendix Figure B2:** Principle component analysis of the specificity of 74 smFISH probe sets using as parameters filtering options used in eFISHent.

calculated for each oligonucleotide in the probe sets. These include secondary structure, nucleotide sequence information, as well as k-mers and general off-targets. As can be seen from Figure B2, there is no clear separation of specific and unspecific probes. This suggests that traditional probe parameters do not have any major impacts on resultant specificity. Similarly, individual probe parameters are not significantly correlating with specificity (GC-content yielding the highest Pearson's correlation coefficient of 0.18). Taken together, the development of a probe set database is a step forward in addressing the issue of finding the optimal set of values for a given gene of interest using eFISHent. However, the lack of clear separation between specific and unspecific probes suggests that additional research is necessary to fully understand the factors that influence specificity.

[C1] Stellaris.

[C2] Brian J. Beliveau, Jocelyn Y. Kishi, Guy Nir, Hiroshi M. Sasaki, Sinem K. Saka, Son C. Nguyen, Chao-ting Wu, and Peng Yin. OligoMiner provides a rapid, flexible environment for the design of genome-scale oligonucleotide in situ hybridization probes. Proceedings of the National Academy of Sciences, 115(10):E2183–E2192, March 2018. Publisher: Proceedings of the National Academy of Sciences.

[C3] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biology, 10(3):R25, March 2009.

[C4] Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics, 27(6):764–770, March 2011.

[C5] David H. Mathews, Walter N. Moss, and Douglas H. Turner. Folding and Finding RNA Secondary Structure. Cold Spring Harbor Perspectives in Biology, 2(12):a003665, December 2010.

[C6] Marco Passaro, Martina Martinovic, Valeria Bevilacqua, Elliot A Hershberg, Grazisa Rossetti, Brian J Beliveau, Raoul J P Bonnal, and Massimiliano Pagani. OligoMinerApp: a web-server application for the design of genome-scale oligonucleotide in situ hybridization probes through the flexible OligoMiner environment. Nucleic Acids Research, 48(W1):W332–W339, July 2020.

# C

# Design, Labeling, and Application of Probes for RNA smFISH

Methods in Molecular Biology. 2022 July 22nd; Vol. 2537; Chapter 10 Nucleic Acids Research. 2021 July 1st; Vol. 49, No. 13; 7292 - 7297


- Ewa Piskaldo
- **Bastian Th. Eichenberger**
- Luca Giorgetti
- Jeffrey A. Chao


I wrote the protocol for RNA smFISH and imaging and made the nextflow pipeline more easily installable through extensive online documentation and installation workflows.

# Chapter 10

## Design, Labeling, and Application of Probes for RNA smFISH

**Ewa Piskadlo, Bastian Th. Eichenberger, Luca Giorgetti, and Jeffrey A. Chao**

### Abstract

Visualization of single mRNA molecules in fixed cells can be achieved using single molecule fluorescent in situ hybridization (smFISH). This approach enables accurate quantification of mRNA numbers and localization at a single-cell level. To ensure reliable results using smFISH, it is critical to use fluorescent probes that are highly specific to their RNA target. To facilitate probe design, we have created anglerFISH, a user-friendly command-line based pipeline. In this chapter, we present how to perform a smFISH experiment using user-designed and labeled probes.

**Key words** Single-molecule fluorescence in situ hybridization (smFISH), Fluorescence labeling of oligonucleotides, FISH probe design, Command line interface (CLI), Nextflow workflow

### 1 Introduction

Single-molecule RNA fluorescence in situ hybridization (smFISH) enables the quantification of the number and location of individual transcripts within their native cellular environment. This has enabled the entire life cycle of mRNAs from transcription to degradation to be visualized in single cells [1]. Additionally, when imaged in combination with fluorescent antibodies, smFISH can also be used to characterize RNA-protein interactions [2].

smFISH can be performed on a variety of specimens ranging from bacteria to the entire Drosophila brain [3–5]. All of these specimens are fixed and subsequently permeabilized. This allows for RNA transcripts to become accessible to fluorescently labeled probes that hybridize to the target thus revealing its position. Probes are single-stranded DNA oligonucleotides that are typically between 20 and 50 nucleotides in length and are complementary to the target RNA [6, 7]. Alternative approaches using branched DNA oligonucleotide technology have also been used for smFISH [8]. Irrespective of the exact design of the probes, the hybridization

Peter Scheiffele and Oriane Mauger (eds.), *Alternative Splicing: Methods and Protocols*,
Methods in Molecular Biology, vol. 2537, https://doi.org/10.1007/978-1-0716-2521-7_10,

of multiple fluorescently labeled oligonucleotides to a single RNA amplifies the signal and enables distinguishing it from background signals.

While smFISH provides information that is not readily obtainable from ensemble or even single-cell RNA sequencing experiments, it is currently limited in the number of genes that can be visualized at one time. This limitation stems from the challenges in the specific detection of spectrally distinct fluorophores. Recent methods, however, employing barcoding techniques and sequential rounds of hybridization and bleaching have been able to visualize up to ~10,000 unique RNA transcripts within a single cell [9].

Sequences for smFISH probes have to be selected carefully taking into account binding specificity, probe length, spacing, and melting temperatures. Additionally, designing probes can be challenging as multiple bioinformatics tools are required. This chapter introduces anglerFISH, a workflow-based, command-line tool that allows nonexpert users to design smFISH probes for RNA targets of interest. Designing custom probes reduces experimental costs and allows for more flexibility in fluorescent dyes and hybridization conditions. In addition, this chapter explains how to fluorescently label these probes in the lab and contains a protocol for performing RNA smFISH.

## 2 Materials

### 2.1 Designing Custom Probes

1. A POSIX compatible operating system (Linux or macOS). Windows users are advised to install the free Linux operating system Ubuntu via a virtual machine software, such as Oracle VM VirtualBox.
2. Bash (version $\geq$3.2).
3. Java (version $\geq$8, 1.8).
4. Miniconda 3 or Anaconda 3.
5. Git.
6. GNU Make.

### 2.2 Probe Labeling

1. Silica gel in a container with a lid.
2. Anhydrous DMSO.
3. Nuclease-free water.
4. 100% ethanol.
5. Nanodrop.
6. Centrifuge and nuclease-free tubes (1.5 mL and PCR).
7. Thermocycler.

8. ddUTP-dye conjugation mix: 10 µL of 20 mM amino-11-ddUTP (*see* **Note 1**), 2 µL of 1 M $NaHCO_3$ (pH 8.4), 10 µL of 40 mM amine-dye (in DMSO).
9. ddUTP-dye-oligonucleotides labeling reaction per one labeling reaction: 10 µL of 100 µM oligonucleotide mix, 1 µL of 5 mM ddUTP–dye conjugation mix (keep in darkness), 1 µL of 20 U/µL of terminal deoxynucleotidyl transferase (TdT), 3 µL of 5× TdT reaction buffer.
10. Amber tubes.
11. 1 M sodium acetate, pH 5.5.
12. 5 mg/mL linear acrylamide.

***2.3 RNA FISH Fixation and Hybridization (for 4 Coverslips)***

1. Phosphate-buffered saline (PBS).
2. Incubator with shaker set to 37 °C.
3. Glass coverslips #1 thickness 18 mm diameter.
4. Prolong Gold Antifade Reagent with DAPI.
5. Transparent nail polish.
6. Fixation solution: 10 mL of 20% paraformaldehyde, 5 mL of 10× PBS, 35 mL of $H_2O$.
7. Hybridization buffer: Gently thaw probes on ice and prepare mixture on ice as well. Mix: 100 µL of 20% dextran sulfate (*see* **Note 2**), 20 µL of 20× SSC, 20 µL of formamide brought to room temperature (RT) before opening to reduce oxidation, 200 nM of FISH probes (approximately 1–2 µL). Fill $H_2O$ up to 200 µL total volume. Gently pipette up and down to mix.
8. Wash buffer: 5 mL of 20× SSC, 5 mL of formamide brought to RT before opening, 40 mL of $H_2O$. Store in the dark by wrapping with aluminum.
9. Hybridization chamber: place a sheet of Parafilm on the bottom of a petri dish. Tape an inverted cap of a 50 mL conical tube to the edge of the chamber. Fill the inverted cap with PBS to create a humid environment. Prepare both a sheet of Parafilm to seal and aluminum foil to cover the chamber.
10. 70% ethanol.

## 3 Methods (*See* Fig. 1)

***3.1 Designing Custom Probes***

We have created anglerFISH [10], a custom Nextflow [11] workflow, based primarily on OligoMiner software [12], as well as several other common tools to design RNA smFISH probes [13–15]. This generates highly specific antisense probes against either endogenous or exogenous RNA targets in a reproducible and automated manner with a single command. Due to utilizing

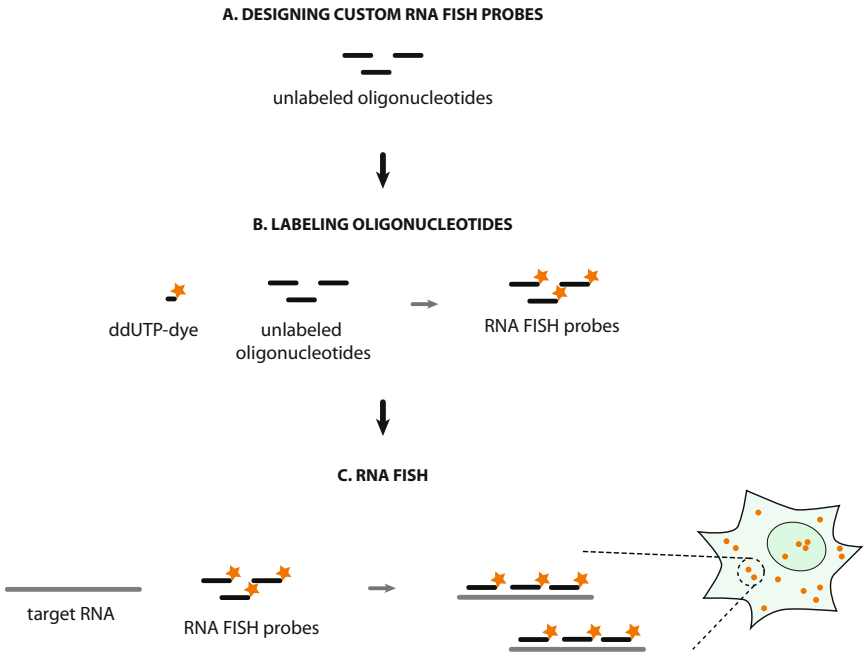



**Fig. 1** RNA smFISH with custom-designed and labeled probes. (**a**) Design sets of custom probes (ssDNA oligonucleotides) against an RNA of interest. (**b**) The probes with sequences determined in the previous step are labeled with ddUTP-dyes to create fluorescently labeled RNA FISH probes. (**c**) Cells are fixed with paraformaldehyde and permeabilized to enable FISH probes to hybridize to their RNA targets, allowing visualization of single molecules of RNA

OligoMiner software, the workflow provides the researcher with control over probe parameters including probe length range, probe spacing and overlap, hybridizing temperature, buffer composition and GC content. Refer to the README.md file in the GitHub repository https://github.com/episkadlo/anglerFISH for more detailed instructions and examples, or consult the workflow's help message. Tips for probe design parameters can be found in **Note 3**.

*3.1.1 Download and Configure the Workflow*

1. Download the repository containing the workflow from GitHub: go to https://github.com/episkadlo/anglerFISH, click on the green "Code" button, and select the "Download ZIP" option, then unpack the file in the location of your choice. Rename the folder to "anglerFISH."
2. In the terminal, go inside the directory you just unpacked (anglerFISH), by typing the following path.

```
cd Path/To/Your/anglerFISH/
```

When inside the directory, install and configure all the necessary software and packages by typing the following command.

```
make
```

3. To test if the pipeline is ready to be used and to see the help message with instructions of use, run the following.

```
./nextflow main.nf --help
```

### 3.1.2 Prepare Genome Indexes for Specificity Filtering

1. Download a fasta (.fa) file for the genomic sequence of the organism you want to design probes against. Genomic sequences can be found at the UCSC Genome Browser website (https://hgdownload.soe.ucsc.edu/downloads.html). If the fasta is compressed (fa.gz), unpack it to a .fa manually (Linux: in the command line with gunzip <file>.fa.gz, Mac: unpacks by default). Place the unpacked genome fasta file in the directory "anglerFISH/UPLOAD_HERE/genome_raw/".
2. From the anglerFISH/ directory, build the genome indexes for HISAT2 and Jellyfish programs. *See* **Note 4** for more details on generating indexes and parameters.

```
./nextflow main.nf --genome_index <genome name> --createIn-
dexes true --l <min length> --L <max length>
```

The indexes will be automatically placed in a new directory "anglerFISH/UPLOAD_HERE/genome_index/<genome_index>". The indexes are kept in this directory and can be used repeatedly to design sets of probes. There is no need to repeat this step until new target organism must be used.

### 3.1.3 Run the Workflow to Generate RNA FISH Probes

1. Copy the target mRNA sequence in fasta format to the directory "anglerFISH/UPLOAD_HERE/fasta".
2. Once the indexes have been created (Subheading 3.1.2), run the pipeline to design probes, specifying the parameters of the desired probes.

```
./nextflow main.nf --genome_index <genome> --input_file
<RNA_sequence_base> <optional parameters>
```

Carefully read the help message of the workflow (./nextflow main.nf --help) to learn about all the possible options, modes, and default parameters of the probes.

After a successful run of the pipeline (*see* **Note 5**), a zip file with all the output files produced by the workflow can be found in the directory "anglerFISH/UPLOAD_HERE/results". Unpack the zip file, which should contain the following.

(a) Fasta and tab files with designed probes.

(b) Fasta file with reverse complementary sequences of the probes.

(c) Alignment file(s) of the final probes against the genome.

(d) File with a table of melting temperatures (Tm) of the probes.

(e) Log File with basic information about the run.

3. Inspect the number of probes the workflow returned. For reliable detection of single RNAs, we recommend using at least 40 probes made of 18–23 nucleotides or at least 25 probes made of 45–50 nucleotides. If your run returned fewer probes, try rerunning the workflow with broader parameters.

4. (Optional) Visualization of probes against endogenous RNAs.

   Go to a web application version of the Integrative Genomic Viewer at https://igv.org/app/ and click on the *Genome > selector* and select the same genome version that was used to filter and map your probes (*see* **Note 6**). Click on *Tracks > Local File* to load the probe alignment files created by the pipeline. Select both the .bam and .bam.bai file at the same time. Search for the name of your gene in the search box to zoom in to the gene and view the aligned probes.

### 3.2 Probe Labeling

A detailed description of the probe labeling can be found in the original publication describing TdT probe labeling [16].

#### 3.2.1 Conjugating ddUTP and Dye

1. Assemble a moisture-free environment by enclosing silica beads in a box with a tightly closed lid for around an hour.
2. In the moisture-free environment, resuspend the amine-dye in anhydrous DMSO to a final concentration of 40 mM.
3. Assemble ddUTP-dye conjugation mix as described in Subheading 2 and incubate for 2 h at room temperature (RT) in the dark.
4. Add 18 μL of nuclease-free water to obtain 40 μL of 5 mM stock solution of ddUTP-dye mix. Aliquot to amber tubes and store at −20 °C.

*3.2.2 Conjugating Oligonucleotides and ddUTP-Dye*

1. Order the oligonucleotide ssDNAs with sequences designed by the workflow at a concentration of 100 µM and desalted.
2. In a PCR tube, assemble the ddUTP-dye-oligos labeling reaction described in Subheading 2. Vortex and briefly spin down the mixture in a minicentrifuge. Incubate the mixture for 16–24 h in a thermocycler set to 37 °C.
3. Add 60 µL 1 M sodium acetate (pH 5.5), 125 µL nuclease-free water, and 1.5 µL linear acrylamide to the PCR tube. Mix well and transfer to a 1.5 mL tube. Add 800 µL of 100% ethanol (−20 °C). Mix the contents of the tube by inverting several times. Transfer to −80 °C for 30–60 min.
4. Centrifuge for 30 min (16,000 × *g* at 4 °C) after which a colored pellet should be visible at the bottom of the tube. Keep the tubes on ice at all times.
5. Remove the supernatant, add 1 mL of 80% ethanol (4 °C) to the tube. Gently pipette the liquid up and down several times to detach the pellet from the bottom of the tube, trying to avoid fragmenting the pellet into smaller pieces. Centrifuge for 5 min (16,000 × *g* at 4 °C). Repeat this ethanol washing step three times in total, after the first wash transfer the entire ethanol with the floating pellet to a clean 1.5 mL tube.
6. Remove as much of the supernatant as possible and let the pellet dry for 5–10 min or until all of the liquid evaporated. Resuspend the pellet with 30 µL of nuclease-free water by pipetting up and down until completely dissolved (*see* **Note 7**).

*3.2.3 Assessing the Labeling Efficiency*

1. Measure the absorbance (optical density, OD) at 260 nm ($OD_{260}$) and at the maximum absorption of the dye ($OD_{dye}$) of the probe solution with Nanodrop. To calculate the degree of labeling, enter the values in the *DegreeOfLabeling.xlsx* spreadsheet attached in the anglerFISH directory or the anglerFISH GitHub repository. Alternatively, follow the step-by-step instructions below.
2. Calculate the mean molar extinction coefficient of the oligonucleotides mix ($\varepsilon_{oligos}$) by taking the mean of the molar extinction coefficient of each oligonucleotide. Add 9000 $mol^{-1}$ $cm^{-1}$ to correct for the extra UTP added to labeled oligonucleotide ($\varepsilon_{corr}$).
3. Calculate the concentration of the oligonucleotides.

$$c_{oligos} = \frac{OD_{260} - cf_{260} \times OD_{dye}}{\varepsilon_{corr}\left(mol^{-1}cm^{-1}\right) \times \text{light path (cm)}}$$

where $cf_{260}$ is the contribution of the dye to OD at 260 nm (consult dye manual), $\varepsilon_{dye}$ is the dye's extinction coefficient

(consult dye manual). The standard light path length of the Nanodrop is 0.1 cm.

4. Calculate the concentration of the dye:

$$c_{\text{dye}} = \frac{\text{OD}_{\text{dye}}}{\varepsilon_{\text{dye}}\left(\text{mol}^{-1}\text{cm}^{-1}\right) \times \text{light path (cm)}}$$

5. Calculate the degree of labeling.

$$\text{degree of labeling} = \frac{c_{\text{dye}}}{c_{\text{oligo}}}$$

We consider a probe set to be sufficiently well-labeled for smFISH experiments when the degree of labeling is above 0.9 (*see* **Note 8**).

### 3.3 RNA smFISH

#### 3.3.1 Fixation of Samples

1. Seed between 10,000 and 40,000 cells per well onto coverslips placed in a 12-well plate and return to the incubator to allow cells to attach and become well spread (~1–2 days). Avoid allowing cells to grow to confluence (*see* **Note 9**).
2. Aspirate the culture medium and wash twice with 1 mL of PBS.
3. Add 1 mL per well of fixation solution and incubate for 10 min at RT while gently rocking (*see* **Note 10**).
4. Wash thrice with PBS.
5. To permeabilize the cells, add 1 mL of fresh 70% w/v ethanol and incubate at 4 °C overnight.

#### 3.3.2 Probe Hybridization

1. Aspirate ethanol and wash the wells once with PBS.
2. Add 1 mL per well of the wash buffer and incubate for 5 min at RT while gently rocking. Repeat the wash once more.
3. Pipette 45 µL of hybridization solution into a drop on a sheet of Parafilm in the hybridization chamber (*see* Subheading 2).
4. Remove the coverslip with tweezers from the wash buffer and gently dry it by tapping the side on absorbent tissue. Invert the coverslip (cell side down) onto the drop of hybridization solution.
5. Seal the chamber with Parafilm and cover with aluminum foil before hybridizing at 37 °C for 4 h up to overnight.
6. In a new 12-well plate, add 1 mL per well of prewarmed (37 °C) wash buffer.
7. Transfer the coverslips back to the 12-well plate using tweezers, placing them cell side up.
8. Incubate in the dark at 37 °C for 30 min, while gently rocking to wash out any unhybridized probes.
9. Repeat wash once more.

10. Wash twice with PBS.
11. Prepare the glass slides by centrally placing a drop of Prolong Gold Antifade Reagent with DAPI.
12. Use tweezers to remove coverslips from PBS and slightly dry by tapping the side on absorbent tissue.
13. Place the coverslip cell side down onto the drop.
14. Let the mounting medium solidify by storing the samples in a dark place at RT overnight.
15. Seal coverslips with nail polish around the edges and proceed to image or store for up to a week at 4 °C.

### 3.4 Image Acquisition

1. smFISH images can be collected on a wide-field epifluorescence microscope equipped with a metal-halide lamp, a high numerical aperture oil objective, and a CCD camera.
2. The DAPI channel should be used to identify regions for imaging where cells are well dispersed and to avoid any bias in data collection.
3. Acquire optical sections spanning 5–6 μm using a 100× objective with a z-step sampling close to Nyquist rate, which is typically 200–300 nm depending on objective's numerical aperture. Set exposure times resulting in a satisfactory signal histogram.

## 4 Notes

1. We have successfully generated working probes with ATTO 565 and ATTO 633 dyes.
2. Prepare the 20% dextran sulfate solution prior to use as it takes several hours to dissolve.
3. As a general starting point for probe design, we suggest designing 18–23 nucleotides long probes, with melting temperatures between 47–60 °C, in 10% formamide. We found those conditions to provide a broad selection of probes. In case the number of probes designed by the workflow is very high and only a subset of probes needs to be selected for synthesis, we suggest either narrowing the desired melting temperatures of probes closer to 57 °C and selecting probes with similar melting temperatures or alternatively, picking probes based on the sequence alignment. We suggest selecting probes targeting the 5′ end of the target RNA, or probes that target the suitable isoforms or other important features.
4. Genome indexes, built from reference genome files, allow mapping and processing sequences more efficiently. In the case of anglerfish, the hisat2 index is used for aligning probes

to the genome and the jellyfish index is used for finding and filtering out common k-mer sequences. The jellyfish index must be built for each k-mer length separately. In order to define k-mers (probe lengths) to build jellyfish indexes, select the range of probe lengths that you want to design and indicate the range by specifying a minimal (--l) and maximal (--L) length of the probes. For example, if you plan to design probes with lengths ranging from 18–23 nucleotides, specify --l 18 --L 23.

In case you have already built the indexes in the past and want to expand the range of k-mer jellyfish indexes (probe lengths), you can use the --jf-only flag to only make additional jellyfish indexes, without making the hisat2 indexes again. For example, if you decide later to also design probes between 24 and 25 nucleotides, to add jellyfish indexes for these k-mers run the following.

```
./nextflow main.nf --genome_index <genome name> --createIn-
dexes true --l 24 --L 25 --jf_only
```

5. To save disk space, regularly remove the content of the "anglerFISH/work/" directory, which contains intermediate files used in the process of designing probes and are obsolete after the workflow finishes designing a probe set. Alternatively, you can also run "make clean."
6. You can also upload any genome of interest.
7. If the pellet is difficult to resuspend at first, allow it to soak the water for a few minutes, then try to pipet again.
8. Occasionally, the labeling efficiency can be low (<90%). Usually, simply repeating the labeling of the unlabeled oligonucleotides solves the issue. The second major factor that we found affecting probe labeling is the batch of the TdT enzyme, which should be replaced if the degree of labeling is consistently low. If the probe labeling efficiency is still suboptimal, refer to the original labeling publication [16] for more details on troubleshooting this issue.
9. The number of cells and the time will depend on the experimental cell line.
10. Perform this step under the hood to prevent inhaling toxic paraformaldehyde fumes.

## Acknowledgments

The authors would like to thank all the members of the Chao lab and the Giorgetti lab for their support in establishing the probe

designing and experimental pipeline. We thank Daniel Mateju for critically reading the manuscript. In addition, we thank the Facility for Advanced Imaging and Microscopy at FMI for help in image acquisition. Research in the Chao lab is funded by the Novartis Research Foundation, the Swiss National Science Foundation (310031A_182314) and the SNF NCCR RNA & Disease network. Research in the Giorgetti lab is funded by the Novartis Research Foundation and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation (grant agreement no. 759366, 'BioMeTre').

**References**

1. Chao JA, Lionnet T (2018) Imaging the life and death of mRNAs in single cells. Cold Spring Harb Perspect Biol 10:a032086
2. Eliscovich C, Shenoy SM, Singer RH (2017) Imaging mRNA and protein interactions within neurons. Proc Natl Acad Sci 114
3. Skinner SO, Sepúlveda LA, Xu H et al (2013) Measuring mRNA copy number in individual Escherichia coli cells using single-molecule fluorescent in situ hybridization. Nat Protoc 8:1100–1113
4. Yang L, Titlow J, Ennis D et al (2017) Single molecule fluorescence in situ hybridisation for quantitating post-transcriptional regulation in Drosophila brains. Methods 126:166–176
5. Long X, Colonell J, Wong AM et al (2017) Quantitative mRNA imaging throughout the entire Drosophila brain. Nat Methods 14: 703–706
6. Femino AM et al (1998) Visualization of single RNA transcripts in situ. Science 280:585–590
7. Raj A, van den Bogaard P, Rifkin SA et al (2008) Imaging individual mRNA molecules using multiple singly labeled probes. Nat Methods 5:877–879
8. Battich N, Stoeger T, Pelkmans L (2013) Image-based transcriptomics in thousands of single human cells at single-molecule resolution. Nat Methods 10:1127–1133
9. Xia C, Fan J, Emanuel G et al (2019) Spatial transcriptome profiling by MERFISH reveals subcellular RNA compartmentalization and cell cycle-dependent gene expression. Proc Natl Acad Sci 116:19490–19499
10. Piskadlo E, Eichenberger B (2021) anglerFISH: initial anglerFISH release. Zenodo 4427720
11. Tommaso PD, Chatzou M, Floden EW et al (2017) Nextflow enables reproducible computational workflows. Nat Biotechnol 35: 316–319
12. Beliveau BJ, Kishi JY, Nir G et al (2018) OligoMiner provides a rapid, flexible environment for the design of genome-scale oligonucleotide in situ hybridization probes. Proc Natl Acad Sci 115
13. Marçais G, Kingsford C (2011) A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics 27: 764–770
14. Kim D, Paggi JM, Park C et al (2019) Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat Biotechnol 37:907–915
15. Zadeh JN, Steenberg CD, Bois JS et al (2010) NUPACK: analysis and design of nucleic acid systems. J Comput Chem 32:170–173
16. Gaspar I, Wippich F, Ephrussi A (2017) Enzymatic production of single-molecule FISH and RNA capture probes. RNA 23:1582–1591

# Colophon

This document was created using LaTeX and BibTeX typesetting originally developed by Leslie Lamport, based on TeX created by Donald Knuth.
Theme adapted from a design by Tony Beltramelli.
Edited in Visual Studio Code.
The text is set in Helvetica.