# Enabling high-throughput image analysis with deep learning-based tools

Dissertation zur Erlangung des Doktorgrades an der
Fakultät für Biologie der LMU München

**David Bunk**

23. Januar 2023

Erstgutachter: Prof. Dr. Heinrich Leonhardt
Zweitgutachter: Prof. Dr. Dirk Metzler
Datum der Abgabe: 23.01.2023
Datum der mündlichen Prüfung: 27.06.2023

*Truth is sought for its own sake.*
*And those who are engaged upon the quest*
*for anything for its own sake*
*are not interested in other things.*
*Finding the truth is difficult,*
*and the road to it is rough.*

IBN AL-HAYTHAM

## Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass die vorgelegte Dissertation von mir selbständig und ohne unerlaubte Hilfe angefertigt ist.

München, den ...28.06.2023... ...David Bunk...

(Unterschrift)

## Erklärung

Hiermit erkläre ich, *

☒ dass die Dissertation nicht ganz oder in wesentlichen Teilen einer anderen Prüfungskommission vorgelegt worden ist.

☒ dass ich mich anderweitig einer Doktorprüfung ohne Erfolg **nicht** unterzogen habe.

☐ dass ich mich mit Erfolg der Doktorprüfung im Hauptfach ..................................

und in den Nebenfächern ..................................................................................

bei der Fakultät für .................................... der ...........................................................

(Hochschule/Universität)

unterzogen habe.

☐ dass ich ohne Erfolg versucht habe, eine Dissertation einzureichen oder mich der Doktorprüfung zu unterziehen.

München, den...28.06.2023... ...David Bunk...

(Unterschrift)

*) Nichtzutreffendes streichen

# Abstract

Microscopes are a valuable tool in biological research, facilitating information gathering with different magnification scales, samples and markers in single-cell and whole-population studies. However, image acquisition and analysis are very time-consuming, so efficient solutions are needed for the required speed-up to allow high-throughput microscopy.

Throughout the work presented in this thesis, I developed new computational methods and software packages to facilitate high-throughput microscopy. My work comprised not only the development of these methods themselves but also their integration into the workflow of the lab, starting from automating the microscopy acquisition to deploying scalable analysis services and providing user-friendly local user interfaces.

The main focus of my thesis was YeastMate, a tool for automatic detection and segmentation of yeast cells and sub-type classification of their life-cycle transitions. Development of YeastMate was mainly driven by research on quality control mechanisms of the mitochondrial genome in *S. cerevisiae*, where yeast cells are imaged during their sexual and asexual reproduction life-cycle stages. YeastMate can automatically detect both single cells and life-cycle transitions, perform segmentation and enable pedigree analysis by determining origin and offspring cells. I developed a novel adaptation of the Mask R-CNN object detection model to integrate the classification of inter-cell connections into the usual detection and segmentation analysis pipelines.

Another part of my work focused on the automation of microscopes themselves using deep learning models to detect wings of *D. melanogaster*. A microscope was programmed to acquire large overview images and then to acquire detailed images at higher magnification on the detected coordinates of each wing. The implementation of this workflow replaced the process of manually imaging slides, usually taking hours to do so, with a fully automated, end-to-end solution.

# Contents

**Eidesstattliche Erklärung** iii

**Abstract** v

**Contents** vii

**1 Introduction** 1

- 1.1 Machine Learning 6
  - 1.1.1 Origins of Machine Learning 6
  - 1.1.2 Data Preparation 7
  - 1.1.3 Model Training 9
- 1.2 Computer Vision 14
  - 1.2.1 Segmentation 14
  - 1.2.2 Image to image translation 17
  - 1.2.3 Classification 20
  - 1.2.4 Detection 20
- 1.3 Artificial Neural Networks 22
  - 1.3.1 Fully connected networks 23
  - 1.3.2 Deep Learning 24
  - 1.3.3 Faster R-CNN 28
- 1.4 Machine Learning in Biology 31
  - 1.4.1 Adapting models to biological problems 31
  - 1.4.2 Custom Faster R-CNN heads 33

1.4.3 Creating tools for the daily lab routine . . . . . . . . . . . . 36

**2 Original Papers 37**

2.1 Encoding of variation in spatial enhancer activity . . . . . . . . . . 37

2.1.1 Main Paper . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

2.1.2 Supplementary information . . . . . . . . . . . . . . . . . . . 50

2.2 Quality control of the mitochondrial genome . . . . . . . . . . . . . 65

2.2.1 Main Paper . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

2.2.2 Supporting Information . . . . . . . . . . . . . . . . . . . . . 79

2.3 YeastMate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114

2.3.1 Main Paper . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114

2.3.2 Supplementary information . . . . . . . . . . . . . . . . . . . 118

**3 Discussion 145**

3.1 Encoding of variation in spatial enhancer activity . . . . . . . . . . 146

3.2 Quality control of the mitochondrial genome . . . . . . . . . . . . . 149

3.2.1 Cristae morphology influences mtDNA quality control . . . 149

3.2.2 YeastMate . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150

Detection model . . . . . . . . . . . . . . . . . . . . . . . . . . 151

Software architecture . . . . . . . . . . . . . . . . . . . . . . . 154

**A Bibliography 157**

**B Abbreviations 171**

**C List of Publications 173**

**D Curriculum Vitae 175**

# Chapter 1

# Introduction

The pursuit of knowledge has been a fundamental aspect of human civilisation since the beginning of time. The human mind seeks to understand the world around it and builds a mental model of its perceived reality (Anderson, 2009). Science, sometimes defined as knowledge derived from the facts of experience (Chalmers, 2013), is a framework to create objective models of our reality based on verifiable evidence obtained through observation and experimentation. This approach contrasts with other, very human, bases for creating models such as faith or intuition. Most definitions of the scientific method within experimental frameworks include steps about creating a hypothesis about reality, either from prior knowledge or a new observation, creating and conducting a fitting experiment, and then analysing the results to accept or reject the hypothesis (Gauch Jr & Gauch, 2003). Here, the theoretical part of science creates new models from our past observations. In contrast, the experimental part puts them to the test, opening the way for new observations and thus closing the circle. This interplay creates an iterative process through which knowledge is refined step by step (Godfrey-Smith, 2009).

However, gathering observations to further this cycle can be a complicated process. As our knowledge about the world grows, so do our models. Having a complete understanding of every part of a scientific model has become almost impossible for a researcher (Ioannidis et al., 2015), increasing pressure to specialise (Casadevall & Fang, 2014). Biological systems are exceedingly complex; an organism can be viewed through many different lenses, for example, the structure and function of an organism's genome or, with epigenomics, the mechanisms that control gene expression. However, not just the research object can be complex, but the techniques used to study them can be equally so. Looking at the field of proteomics, for example, many different techniques can be used to study the structure and function of proteins. These techniques include x-ray crystallography, microscopy, flow cytometry, and mass spectrometry (Breker & Schuldiner, 2014; Yee et al., 2005). Each combination between analysis technique and research topic slice might give a subset, or slice, of facts, which might even clash with facts observed in a different slice. Because of this, each slice of a biological system studied with a particular technique is just one piece of a much larger puzzle. Exploring all layers of biological phenomena with the right combination of approach and technique is essential.

Microscopy, the technique highlighted in this work, can hold a unique, emotional appeal to researchers. Humans are visual animals (Kaas, 2013), and microscopy allows us to engage with the objects of their study directly (Hacking, 1985). The ability to see and interpret the results of experiments with our own eyes can provide a sense of understanding that is impossible with other forms of scientific inquiry. With microscopy, we can zoom on the core building blocks of biological systems, visualising single cells or even single proteins (Longchamp et al., 2017). Furthermore, the use of microscopy images allows researchers to showcase the complexity of biological systems to the general public in an understandable and more relatable way, which might provide a more intuitive understanding than numerical results.

While microscopy is a powerful tool for studying biological systems, it has limitations. One of the critical limitations of microscopy is the so-called Abbe limit, which states that the resolution of a microscope is limited by the wavelength of light used to illuminate the sample (Stelzer, 2002). This means that a traditional microscope cannot resolve features smaller than roughly half of the wavelength of light used to illuminate the sample. While the Abbe limit imposes a fundamental limit on the resolution of traditional microscopy, it is possible to overcome this limit using super-resolution microscopy techniques. There are several different types of super-resolution microscopy, including techniques such as stimulated emission depletion (STED) microscopy (Hell & Wichmann, 1994) and structured illumination microscopy (SIM) (Gustafsson, 2000). These techniques push physical boundaries to overcome the Abbe limit and obtain images with higher resolution than possible with a traditional microscope. Super-resolution microscopy techniques often do not immediately generate interpretable images and often rely on complex computational algorithms to either process or visualise the data. For example, SIM microscopy uses structured illumination patterns to excite the sample and then uses computational methods to combine multiple images with rotated and shifted illumination patterns into a super-resolution, interpretable image. A similar principle is used in localization microscopy, like photoactivated localization microscopy (PALM) (Betzig et al., 2006) or stochastic optical reconstruction microscopy (STORM) (Rust et al., 2006), where many images per view are acquired, with each excitating and capturing a subset of all molecules, and are then combined into a single super-resolution image. This also comes with the drawback of not being able to look through the microscopes with your own eyes. In this case, as in many others, we must rely on new computational tools to observe new facts.

The Big Data revolution (Kitchin, 2014) and the advent of deep learning (DL) (Ching et al., 2018) a decade ago had a major impact on many fields and did not spare the biological sciences. The availability of large amounts of data and the ability to process them has opened up new research opportunities, but it has also presented new challenges (Leonelli, 2014). In the case of microscopy, acquisition, storage and analysis of more and bigger image acquisitions became possible, and with this, the field of high-throughput microscopy (HTM) (Pepperkok & Ellenberg, 2006). With HTM, image acquisition became both faster and easier, enabling methods like high content screening (HCS), where the effects of many different compounds on a biological sample can be captured within a single experiment run (Pegoraro & Misteli, 2017). While HCS aims to analyse many variations in an experiment quickly, the inverse is also possible. When dealing with rare biological events, manually finding enough to reach a statistically significant sample size can be difficult; these can be more readily found if a broad set of images is acquired. An example of this is found in my work on *S. cerevisiae*, where sexual mating events have to be located (Bunk et al., 2022; Jakubke et al., 2021).

Independent from how it was acquired, an extensive dataset usually precludes its analysis by the human eye, removing one of the hallmarks of microscopy. While a single microscopy image can still be seen and analysed manually, its context and meaning may only arise in union with every piece of the dataset. Thus, new tools are required to handle such datasets and extract observable facts for the researcher. Whether such tools would pre-sort, classify or even completely analyse our images, pipelines to reduce the dimensionality of data have to be created.

The ways to make machines help us extract information from images can be put under the umbrella term of Computer Vision (CV). Its most known definition is:

> The goal of Computer Vision is to make useful decisions about real physical objects and scenes based on sensed images.
> (Shapiro, Stockman et al., 2001)

This includes all necessary steps from beginning to end, from transforming raw microscope output to images to pre-processing these and extracting information out of images for these useful decisions. While in the end, the researcher will have to make the decisions, everything up to that could be automated. The applications of CV to biological research also were the central part of my work, with a big focus on providing tools for HTM using DL.

With the following sections, I hope to introduce the realms of CV, machine and DL and its applications in the biological field, including my own tools.

## 1.1 Machine Learning

### 1.1.1 Origins of Machine Learning

The term machine learning (ML) encompasses a wide array of mathematical techniques and is not synonymous with the nowadays ubiquitous terms of DL and artificial intelligence (AI). In essence, ML describes that a model or program is created to perform a task when given corresponding data without relying on specific, predetermined rules; one could also describe this as the model *learning* from previously seen data. While some ML methods like the *Least Squares* algorithm (Stigler, 1981) or *Linear Regression* (Stanton, 2001) can already be traced back to the $18^{\text{th}}$ or $19^{\text{th}}$ century respectively, the field of ML originated from the 1950s when Alan Turing proposed a theoretical model of learning in 1950 (Turing, 2009) and from the works of A.L. Samuel who developed the first self-learning program for computers and coined the term *machine learning* (ML) (Samuel, 1959). While the 1950s and 1960s produced necessary groundwork such as the Perceptron (Rosenblatt, 1958) and Backpropagation (Linnainmaa, 1970), as well as new models like the Nearest Neighbor algorithm (Cover & Hart, 1967), the 1970s became known as the *AI winter*. Following the book *Perceptrons* (Minsky & Papert, 1969), which criticised Rosenblatt's Perceptron and claimed its inherent limitations, coupled with other AI projects ending up underwhelming, interest in AI died down and AI research barely progressed. Interest in ML increased again in the 1990s, with many new publications, including support vector machines (SVMs) (Cortes & Vapnik, 1995) and Random Forests (Ho, 1995), which are still commonly used almost twenty years later. With the increase in both data availability and amount, and computational resources, interest steadily increased in the 2000s. When DL was proven feasible for image classification tasks in 2012 with GPUs providing the necessary compute power for it (Krizhevsky et al., 2012), progress continued every year, with considerable improvements in existing areas and new use cases for AI.

ML has been used in biology for a long time. In 1943, McCulloch and Pitts already proposed a method to study the human brain by building a simplified model of neuronal structures. Other examples from the 20th include the analysis of nerve cell recordings (Linsker, 1988), identification of genes by analysing DNA sequences (Baldi & Brunak, 2001) and prediction of the secondary structure of proteins (Salamov & Solovyev, 1995). What sets biology apart from every other field impacted by ML (Xu et al., 2021) is the close relation between ML and learning mechanisms in neurobiological systems (Hasson et al., 2020). While early iterations of artificial neural networks and learning mechanisms were inspired by the brain (Macpherson et al., 2021), technologies like DL now allow more rapid progress in some areas of neurobiology (Richards et al., 2019). This circle is closed again by new insights on brain structures and learning, which might influence the development of new artificial models again (Hassabis et al., 2017).

This work focuses on the applications of DL, a subset of ML, on images. Analysing images is challenging, especially in biology, where images can get so big that specific software is needed even to visualise them (Hörl et al., 2019; Pietzsch et al., 2015). Thus, DL, as the state-of-the-art approach for many problems, remained my method of choice for image analysis. While DL and each specialised model come with their own set of requirements and implementations, many of the basic principles of general ML described below still apply to DL and create the basis of how learning in computational systems is achieved.

### 1.1.2 Data Preparation

While model architectures are plentiful and diverse, there are some basic paradigms about data preparation that apply to most model architectures. ML models require data to learn the (usually multi-dimensional) distribution of data points to capture the underlying structure and to allow the model to generalize to new, unseen data. There are *unsupervised* models, which are provided with input data without corresponding labels indicating the correct output for each input. Instead, the model uses various approaches to find the underlying structure and patterns in the data (Zhou, 2021). Some models may achieve this without prior knowledge or assumptions. At the same time, other methods like k-means clustering allow the user to include basic assumptions, such as the number of different classes in the dataset (MacQueen, 1967). On the other side stand *supervised* models, which are ML models trained on labelled data. These models are fed with input data and corresponding labels indicating each input's correct output. A ML algorithm sets the parameters of its model using this labelled data to capture the relationship between the input and the output and can then be used to make predictions on new, unknown data.

One of the critical disadvantages of supervised ML is that it requires a large amount of labelled data to train the algorithm adequately. Acquiring new ground truth can be difficult and time-consuming, particularly in biomedical research, where the data is often complex and difficult to label. While standard datasets for DL-based, non-biological image analysis contain millions of images (Deng et al., 2009), a dataset containing hundreds of annotated biological images is seen as a large dataset, as seen with the big Data Science Bowl 2018 competition, which dataset contained 841 images (Caicedo et al., 2019). Additionally, the quality of the labelled data can significantly impact the algorithm's accuracy. As biomedical images are often difficult to assess for human experts, the dataset will almost always include incorrectly annotated ground-truth (Le et al., 2022).

Splitting the data into separate training, validation, and test sets adds to this data scarcity. The training set is used to train the ML algorithm and typically consists of most of the data. The validation set is used to fine-tune the model and to ensure it is not overfitting to the training data. Finally, the test set is used to evaluate the performance of the trained model on unseen data and to estimate its generalisation performance. Data splitting is a common practice in ML (Raykar & Saha, 2015) to ensure that the algorithm is trained on a representative sample of the data and is tested on unseen data to provide an unbiased evaluation of its performance. The necessity of an unbiased evaluation to have valid scientific insights also mandates a big enough test set size.

### 1.1.3 Model Training

When first initialised, a model will produce, depending on implementation, either random or insufficiently fitted output, as it still needs to tune its response to any input. Two steps are integral for model training; defining a loss function and propagating this loss back to the model's weights. A loss function usually is the distance between generated and desired output (Goodfellow et al., 2016). In supervised learning, this will be a metric of difference between generated output and the labelled ground-truth point of the input; in unsupervised learning, the loss might either be defined by a direct relation of output to input, as with, e.g. autoencoders or by analysing the structure of outputs, like in some clustering models. Two simple loss functions summing up the absolute difference between predicted and real data points are the mean absolute error (MAE) and the mean squared error (MSE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \qquad MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where $N$ is the total number of examples in the dataset, $y_i$ is the true label of the $i^{th}$ example, and $\hat{y}_i$ is the predicted label of the $i^{th}$ example.

While the MAE measures the absolute difference, it fails to punish large differences between data points. The similar MSE, which squares the difference, remedies this problem by focusing more on those data points. Just as the MAE could be adapted for a different use case, there are various losses for different models and applications. While the MAE and the MSE are predominantly used for regression, other tasks like classification call for other losses like the binary cross-entropy (BCE) loss. Some models are built on specific losses, like SVM, which uses the hinge loss:

$$hinge = \frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - y_i \cdot \hat{y}_i)$$

where $N$ is the total number of examples in the dataset, $y_i$ is the true label of the $i^{th}$ example, and $\hat{y}_i$ is the predicted output of the model for the $i^{th}$ example with $-1$ for the negative and $+1$ for the positive class. This classification loss punishes wrong classifications more the further they are away from the threshold line between a binary classification. It also slightly punishes correct classifications close to the threshold and thus closer to being incorrect.

The calculated loss must be applied to the model in a second step to improve it. The process of applying error metrics varies greatly between model architectures. While training Decision Trees as part of a Random Forest, the error metric, in this case often the Gini impurity, is applied in every step of iteratively creating the different layers of the model, with the model itself undergoing only one iteration. Starting at the top at the root node, the algorithm considers all possible data splits according to their features. The quality of each split is then evaluated by using the error metric. The split that results in the most significant improvement in this metric is chosen, and the data is split accordingly. This process is repeated for each of the resulting child nodes until the decision tree is fully grown (Rokach & Maimon, 2005).

This process is more difficult for artificial neural networks (ANNs) , which will take the biggest part of this work. ANNs are fully and randomly initialised in the first step and then iteratively trained. As an ANN contains many weights to tweak every iteration, determining which parts of the model to tweak to decrease loss requires methods called gradient descent and backpropagation. Gradient descent is an optimisation algorithm to find the minimum of a differentiable loss function; it iteratively moves a parameter in the direction opposite to the gradient of the objective function at the current point to move closer to the minimum. A single update for a model parameter can be calculated with:

$$\theta_k \leftarrow \theta_k - \alpha \cdot \nabla_{\theta_k} L$$

where $\theta_k$ is a parameter of the model, $L$ is the loss function, $\nabla_{\theta_k} L$ is the gradient of the loss function corresponding to the parameter $\theta_k$, and $\alpha$ is the learning rate. The learning rate is one of the most essential hyperparameters in model training and is a scalar multiplier to the step in each iteration; it defines how much the model's weights can be changed in each iteration and strongly influences training stability and speed. The gradient is a vector that indicates the direction of the steepest ascent of the loss function. Thus, moving in the opposite direction of the gradient will decrease the value of the loss function.

The backpropagation algorithm is specific to ANNs and is a way to propagate the gradient from the end of the network back to its beginning. It uses the chain rule of calculus to calculate the gradient of the loss function through the network's weights. Starting with the output layer and going backwards through the hidden layers of the network, it uses the weights of the connections between the neurons to propagate the error backwards and calculate the gradient for each weight in the network. The gradient can then be used to update the weights in the direction that reduces the error.

The use of backpropagation in model training constitutes a big discrepancy between ANNs and the way biological brains are thought to work. There is little evidence that the mechanisms modifying synaptic connections between neurons work in the same way (Crick, 1989; Grossberg, 1987). While approximations of backpropagation have been suggested to exist within the brain (Lillicrap et al., 2020), other learning processes like predictive coding (Bastos et al., 2012), feedback alignment (Lillicrap et al., 2016) or equilibrium propagation (Scellier & Bengio, 2017) have been suggested. While optimiser algorithms for ANNs currently use backpropagation and gradient descent to train a network, the increasing understanding of how learning functions in biological systems might help build better ML models.

Common optimiser algorithms using backpropagation and gradient descent are stochastic gradient descent (SGD) (Robbins & Monro, 1951), Adam (Kingma & Ba, 2015), and Adagrad (Duchi et al., 2011). While these pursue the same goal, their implementations differ and have unique advantages and disadvantages. SGD for example, while similar to standard gradient descent, uses a small batch of training examples to estimate the gradient instead of the entire training dataset, reducing the required memory footprint of the model. This allows SGD to compute the gradient more efficiently and make more frequent updates to the network parameters. This brings the advantages of being computationally efficient and easy parallelisation, makes it well-suited for training on large datasets, and can help reduce the training time of the network.

However, SGD also has its disadvantages. Because it uses a small batch of training examples to estimate the gradient, the gradient estimate can be noisy and less accurate than the gradient computed using the entire training dataset. This can lead to more unstable and noisy updates, making converging to the optimal solution slower and less stable. Additionally, SGD can be more difficult to tune than other optimisation algorithms and may require careful selection of the learning rate and other hyperparameters for good performance.

Other optimisers like Adam or Adagrad offer techniques like adaptive learning rates to improve the stability and convergence of the optimisation process. This means that the learning rate is automatically adjusted based on the characteristics of the data and the gradient, which can help to avoid the challenges of choosing the learning rate manually. Other techniques, such as momentum in the case of Adam and weight decay in the case of Adagrad can improve the stability and convergence of the optimisation process. These techniques can help to smooth the optimisation process and to avoid local minima and other challenges that can arise during training. These methods make the training process much more robust; a model trained with an Adam optimiser will usually reach convergence even with default settings (Bock & Weiß, 2019). For this reason, an SGD optimiser is usually not used as a starting point. However, if adequately fine-tuned, it might still be the best choice because of its better performance in generalisation than other optimisers (Keskar & Socher, 2018).

## 1.2 Computer Vision

Computer vision is a research field that focuses on developing methods to facilitate computers to process visual data from the world around them and infer quantitative information about the contents of a presented image. Computer vision has a long and rich history, dating back to the early days of computing when researchers first began to explore the potential of using computers to process and analyse visual information. The origins of computer vision can be traced back to the 1960s when researchers began to think about teaching computers to analyse visual data (Papert, 1966). Understanding the visual world around them was seen as an essential step in creating intelligent robots, so computer vision held a vital position within AI research (Szeliski, 2022). Early efforts in the field focused on developing simple algorithms for recognising basic shapes and patterns in images and applying these algorithms to practical tasks such as character recognition. This would allow the extraction of numerical information about an image that can be used in quantitative research. Nowadays, computer vision tasks are predominantly solved with AI tools. The field of computer vision has undergone significant evolution and growth over the past years, driven by advances in computer hardware and software and the increasing availability of large datasets and powerful ML algorithms (Voulodimos et al., 2018).

The central CV tasks arising from biomedical image analysis are segmentation, image-to-image transformation, classification and detection. These tasks are deeply intertwined and often need to be solved in tandem to reach desirable results.

### 1.2.1 Segmentation

Segmentation of images can also be described as classifying each pixel in it. In its simplest use-case, the binary segmentation, each pixel can be differentiated into either a foreground (usually 1) or a background (usually 0) pixel. Visualising

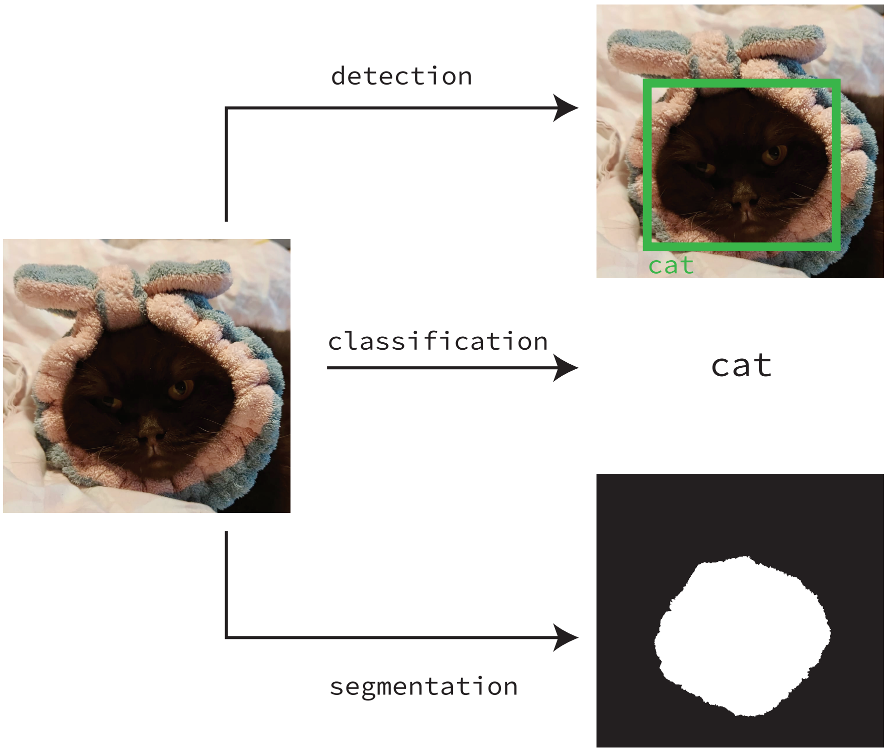


Figure 1.1: An image of a cat can either be *classified* as containing a cat; the exact location of the cat including the classification that it is a cat can be *detected*; or an exact *segmentation* of each pixel corresponding to the cat can be created.

this result as a binary image shows a so-called *semantic segmentation* mask (Fig. 1.1). This mask can then be used to extract information from the pixels of all foreground objects. As it is only a binary mask with no information about the image potentially containing multiple objects, it is not directly possible to do any single-cell analysis in this case.

A widespread, simple segmentation algorithm is Otsu-Thresholding (Otsu, 1979). Here, a threshold value, meaning that pixels above it will be categorised as foreground, and pixels below as background, is iteratively determined. The algorithm below is an optimisation method called the Otsu algorithm that seeks to find the threshold value that maximises the separation between the two classes of pixels in the image.

$$\arg\min_{t} \sigma^2(t) = \omega_{bg}(t)\sigma^2_{bg}(t) + \omega_{fg}(t)\sigma^2_{fg}(t)$$

where $\sigma^2(t)$ is the weighted sum of variances for the background and foreground classes at a given threshold value $t$. The terms $\omega_{bg}(t)$ and $\omega_{fg}(t)$ represent the proportions of pixels in the background and foreground classes at this threshold value, while $\sigma^2_{bg}(t)$ and $\sigma^2_{fg}(t)$ represent the variances of the pixel intensities within the background and foreground classes, respectively. By finding the threshold value that minimizes this weighted sum of variances, the algorithm can automatically determine an appropriate threshold value for performing image segmentation.

When thresholding, the classification of a pixel does not directly impact the classification of the next pixel. An exceptionally bright single pixel in the background might be erroneously classified as a foreground object by thresholding, even though there are no adjacent positive pixels; the opposite might also happen. Most of the time, this behaviour is not intended, so additional post-processing of the segmentation mask is necessary. Common operations include filling small holes that are entirely enclosed by foreground pixels, removing tiny objects, or dilating or eroding objects if their borders are not classified correctly. Most of these operations can be summarised as binary morphological operations (Haralick et al., 1987).

Because most binary classification methods can be adapted for multi-class classification, it is possible to define different classes of objects in an image and classify the pixels accordingly. As this increases the complexity and difficulty of the task, it is common to have pixels of different classes within each object, making correct post-processing even more critical. As binary morphological operations cannot directly be applied to multi-class masks, these masks need to either be split class-wise into binary images again or, if the use-case allows it, multi-colour implementations could be used (Comer & Delp III, 1999).

In many cases, being able to extract the pixel values for each cell separately instead of analysing the whole semantic mask can be advantageous to extract more detailed information. Additionally, object segmentations can be more easily improved if the segmentation of each object can be inspected and modified separately instead of on a per-image basis. The process of generating individual masks for each object is called *instance segmentation* and can be achieved through two opposed approaches. One approach is to generate a semantic segmentation mask as described above and then split this mask into different object masks. This task becomes trivial if objects are non-touching and well-segmented, as each can be defined as a cluster of fully adjoined pixels. Multiple touching objects would count as a single one, necessitating steps to separate object boundaries. A standard solution is one of the many implementations of the watershed transform (Roerdink & Meijster, 2000). Another method, usually used with ML or DL models, is to let the model predict additional semi-structured output, for example a vector pointing towards the object boundary for each pixel like in the StarDist architecture (Schmidt et al., 2018) or topological maps, similar to the watershed transform, as used in the CellPose tool (Stringer et al., 2021). The second approach, described below, is an inverse of the workflow; objects can also be detected separately in the first step and only then are the segmentation masks generated on an image crop for each object.

### 1.2.2 Image to image translation

Segmentation can also be viewed as a translation of an image into a mask image; in contrast, images can also undergo image-to-image translation from a source image to an arbitrary target image of the same dimensions (Liu et al., 2017). Instead of treating this as a classification problem with categorical output values, the distance between the expected and retrieved image can usually be measured with regression metrics as the output values are continuous (Pang et al., 2021). While image translation can be seen as a regression task when applying ML algorithms, the simplest form of it is applying an image filter. Blurring and smoothing image filters such as the gaussian filter can be used remove noise from an image (Fan et al., 2019), while filters like the sobel (Kanopoulos et al., 1988) or hessian filter (Frangi et al., 1998) filter can be used to highlight specific features, in this case, edges or local extrema, in an image (Fig. 1.2). This can be enough to detect and process some biological structures; e.g. curvilinear structures such as blood vessels or bronchial trees can be extracted with the application of the sato filter, which highlights ridges in an image (Sato et al., 1998).

Denoising and image restoration are common preprocessing tasks for biological microscopy images, as a good signal-to-noise ratio (SNR) can be difficult to achieve in many applications; meanwhile, it is paramount to improve the SNR to extract valid data points and/or maintain a robust analysis pipeline. This also gets exacerbated by the frequent trade-off in microscopy between SNR and sample health (Scherf & Huisken, 2015). A good SNR can be achieved by imaging with high light intensity for a longer time; this might decrease sample health by, for example, bleaching fluorophores in fluorescent microscopy (Lichtman & Conchello, 2005) or by damaging or influencing the assayed organism (Magidson & Khodjakov, 2013).

Content-aware image restoration, utilising a DL model, tackles this issue by acquiring a set of images containing pairs of high-intensity and corresponding low-intensity, low-toxicity images (Weigert et al., 2018). This dataset can be used to capture the connection between the two sub-sets, allowing the model to transform unseen low-light images into high-quality images, thus increasing the SNR without sacrificing sample health. Other approaches include Noise2Void (Krull et al., 2019), which aims to achieve similar restoration results on image sets where no high-quality

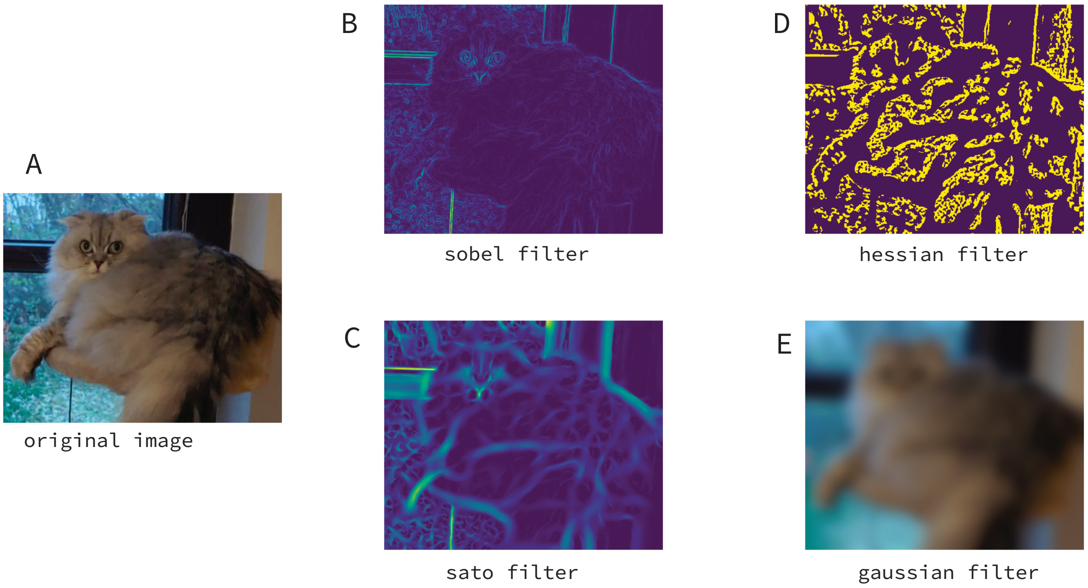


Figure 1.2: Different filters applied on a picture of a cat (A). B: Sobel filter highlighting edges. C: Sato filter highlighting ridges. D: Hessian filter detecting local extrema. E: Gaussian filter blurring the image.

equivalent of an image can be acquired, or PhaseStain (Rivenson et al., 2019), which can translate images of biological structures into images containing the stained equivalent of these.

While image translation with DL methods can be helpful in many use cases, the possible insertion of instabilities and perturbations into the reconstructed image has to be taken into account (Antun et al., 2020). New patterns might be found that are just artefacts of the translation process, while other expected details might be lost during it. Thus, the viability of these techniques must be carefully considered for each project.

### 1.2.3 Classification

Image classification is a fundamental task in computer vision and refers to the process of automatically assigning a label to an image based on its content. There are many different approaches to image classification, each with its strengths and limitations. Traditional approaches often rely on hand-crafted features extracted from the image using domain-specific knowledge. These features are then used as input to a classifier, which makes a prediction based on the learned characteristics of each class. However, these approaches can be limited by the quality and expressiveness of the features and specific domain knowledge required to design them. Additionally, a statistical approach can analyse all pixel values in the input image and use this information to assign a class label to the image, including techniques such as histogram or texture analysis. While whole-image classification is very hard for classical approaches, it is, in comparison with other tasks like segmentation or detection, one of the easier tasks for modern approaches like DL; with the model architectures used in classification often re-used as a base for other tasks. This can be seen with the classic VGG-16 architecture (Simonyan & Zisserman, 2015) being extended for segmentation in the U-Net model (Ronneberger et al., 2015), or the ResNet architecture (He et al., 2016) being commonly used as the backbone for detection architectures like Faster R-CNN (Ren et al., 2015).

### 1.2.4 Detection

In microscopy, we often deal with relatively simple and homogeneous images. Especially in fluorescence microscopy, where bright signals against a dark background result in high-contrast images, cells or signals can often be detected consistently if they are not touching. Object detection becomes a much more challenging task in other modalities where objects are not clearly separated from the background. A good example is tissue imaging, where the definition of background depends on what has to be extracted, and the objects are part of the structure that builds the background. Of course, segmentation on these images might get more complicated, making a pipeline of detection and segmentation appealing.

Object detection can also be described as the classification of arbitrary sub-regions of the image into either containing or not containing an object. Without DL, it is a difficult task with a limited amount of possible approaches. Either specific features can be engineered that can be used to extract objects, e.g. above segmentation masks, or specific colour and shape constraints that stay roughly equal between objects (a typical toy example is the detection of coke cans, which are specifically rectangle and red). Alternatively, if these features are not manually specified, an example object can be given and compared to the image in a process called Template Matching (Brunelli, 2009). Here, a smaller template image of the sought object is slid across the image while the correlation between the template and the corresponding image area is calculated. A local maximum in correlation can then be treated as a matched object if the correlation surpasses a set threshold. Template matching can be helpful if the objects are very similar or even identical to each other, and has been used in finding and classifying viruses in electron microscopy (Sintorn et al., 2004).

## 1.3 Artificial Neural Networks

Even though the public usually refers to artificial neural networks when talking about AI, ANNs are just a subset of AI and ML. They mimic the structure and function of the brain, being composed of many interconnected processing units called *neurons* (Krenker et al., 2011). The strength, also called *weights* of these connections, are the variables of the model, which change during model training, similar to how connections between real neurons can grow and strengthen. Of course, this concept of a neuron is an oversimplification as computational neurons cannot build new connections and lack the many understood, and also not yet understood properties of real biological neurons (Davidson & Furber, 2021). Nevertheless, this architecture makes the model capable of learning from data and making predictions on unseen samples.

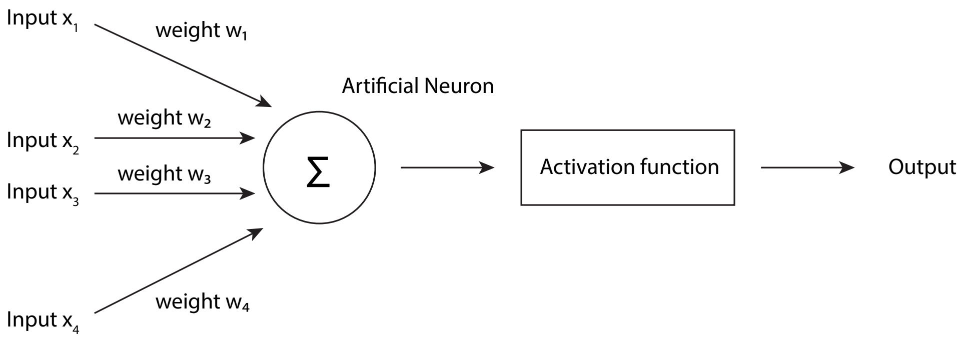


Figure 1.3: Structure of a single artificial neuron. Inputs are multiplied separately by their corresponding weights and then summed up together. A non-linear activation function is then applied to its output.

### 1.3.1 Fully connected networks

Artificial neurons are the key building block of artificial neural networks, which are mathematical models inspired by how biological central nervous systems are structured. They are the basic computation units in a neural network responsible for processing and transmitting information. An artificial neuron receives input from other neurons or external sources, such as the input features of a dataset. The input is multiplied by a set of weights, which determines the importance of each input. The weighted input is then summed and passed through a non-linear activation function, which produces the neuron's output (Fig. 1.3):

$$f(x) = \max(0, x)$$
$$y = f(\sum_{i=1}^{n} w_i x_i + b)$$

where $y$ is the output of the neural network, $x_i$ are the inputs to the neural network, $w_i$ are the weights applied to the inputs, $b$ is the bias term and $f(x)$ is the activation function, in this example the commonly used rectified linear unit (ReLU) function (Agarap, 2018).

After the activation function, an artificial neuron's output is transferred to other neurons in the network or is used as the network's final output. In this way, neurons can be connected by building an artificial neural network (Fig. 1.4), forming a complex network of computational units that can learn and model complex data distributions (Goodfellow et al., 2016).

The activation function of an artificial neuron is a fundamental design choice that determines the properties of the neuron and the behaviour of the network. Common examples of activation functions include the ReLU, the sigmoid function and the hyperbolic tangent (tanh) (Fig. 1.5). Each function has different properties, such

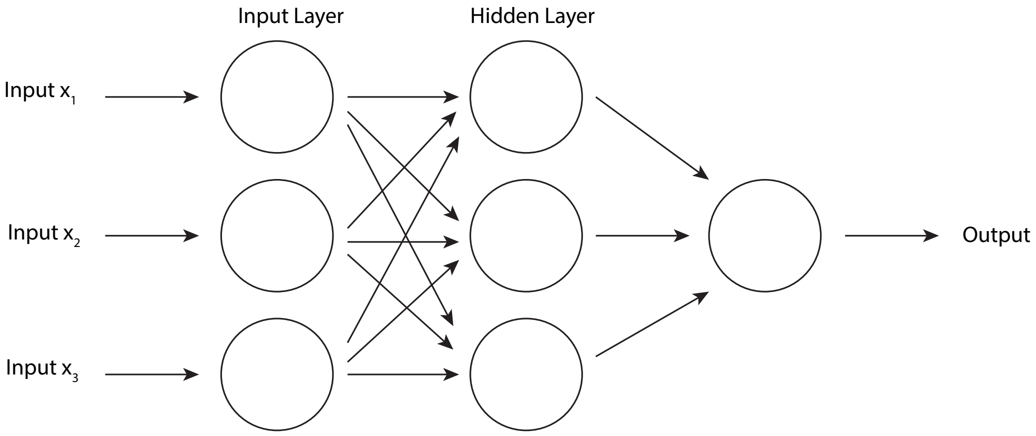


Figure 1.4: Diagram of a fully connected artificial neural network with 3 input neurons, 3 hidden neurons, and 1 output neuron. Arrows represent the weighted connections between neurons.

as being bounded or unbounded and smooth or non-smooth. The shape of the activation functions determines its use cases; the sigmoid function, for example, is often used on the last layer of output neurons as it is bound between 0 and 1, thus able to model probabilities (Narayan, 1997). Without non-linear activation functions, the output of a neural network would be a linear function of the inputs. Thus a network consisting of multiple layers of ANNs could be reduced to a single layer again, negating the advantages of a deeper network (Huang & Babri, 1998).

### 1.3.2 Deep Learning

While artificial neural networks consisting of neuronal units described above can handle some tasks and modalities well enough, they are unsuitable for image analysis. There are 262,144 pixels in an average-sized 512x512 image; having a neuron for each input pixel and the same amount in the first hidden layer would

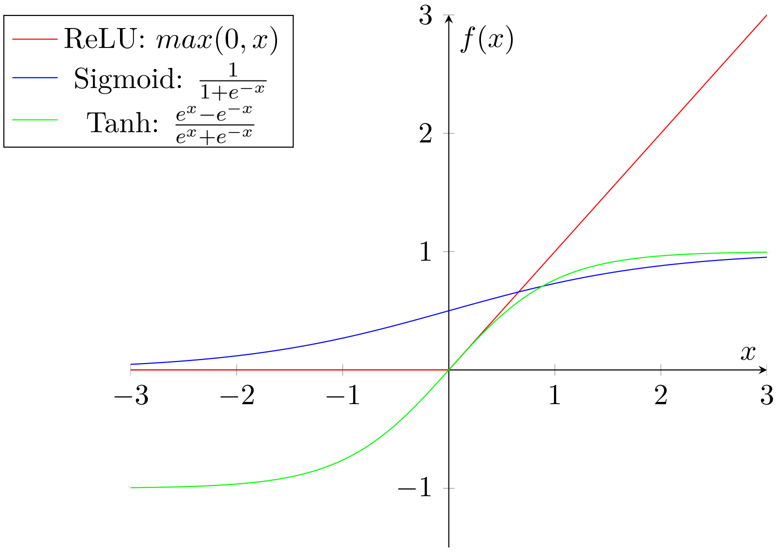


Figure 1.5: Different activation functions for an ANN.

result in almost 69 billion parameters in the first layer. As of 2022, the biggest model in use, GPT-3 for language modelling (Brown et al., 2020), has comparable 175 billion parameters with approximate compute costs of 5 million dollars per training run; this is a rare exception and training such massive networks is not feasible for general use. Thus, a method that does not encode every pixel separately is required. The use of convolutional layers to encode and extract information in spatial modalities like images was already suggested in the 1980s by David Marr (1982) and again by Yann LeCun (1995). A convolutional layer applies a convolution operation to the input data to extract local features and patterns; the discrete convolutional operation can be expressed as:

$$(f * g)[n] = \sum_{k=-\infty}^{\infty} f[k] \cdot g[n-k]$$

where $f$ and $g$ are discrete signals (arrays of numbers), with $f$ as the input or intermediary image and $g$ as the kernel, which is used as a weight in the network. The convolution operation takes these two signals and produces a third signal that represents the overlap between the two signals as one of them is shifted over the other. The resulting signal is indexed by $n$. As the non-zero part of the kernel is limited, the sum is not bounded from $-\infty$ to $\infty$ in practice.

This can also be described simplified as the kernel being moved as a small window across the image to detect and highlight local patterns and thus transforming the image into a filtered version. Different kernels can be used to recognise specific features, such as edges, ridges, or textures; thus different filters can highlight different features of an image (similar to the filters in Fig. 1.2). In convolutional neural networks, the numerical contents of the kernels represent the weights of the network and are thus not manually selected, but automatically fit during model training. As each kernel is used across the whole image, the same set of weights is used at each position in the input data, dramatically decreasing the number of network parameters. This feature falls under the term *parameter sharing*, and also improves the *translation invariance* of convolutional neural networks (O'Shea & Nash, 2015). As the same kernels are applied to e.g. the top and bottom half of the image, image features should be recognised independent of their global location, with some exception to boundary effects at the borders of an image (Kayhan & Gemert, 2020).

This transformed representation of the original image is called the feature map, which encodes the local patterns and features detected by the filters. By applying multiple convolutional layers to the input data, a DL model can learn to extract and combine increasingly complex and abstract features. Convolutional layers are usually alternated with pooling layers, which compress the spatiality of an image (Nagi et al., 2011). This enlarges the receptive field of the convolutional layer, meaning that the convolutional kernel now receives information on a larger part of the image (Araujo et al., 2019). Through this architecture, the shallow layers will have a small receptive field and encode for simpler features like edges (Fig. 1.2). In comparison, the deeper layers will have larger receptive fields and encode for more global features and bigger shapes, e.g. directly highlighting eyes or faces in an image.

While a new architecture for general tasks and image analysis called Transformer emerged recently (Carion et al., 2020), the majority of DL image analysis is still done with  convolutional nteural networks (CNNs). CNNs for image analysis had their breakthrough a decade ago when Krizhevsky et al. presented AlexNet (2012) and Simonyan et al. presented VGG16 (2015), which both are feasible model architectures for image classification with DL. Ronneberger et al. presented the U-Net architecture for image segmentation (2015), which instantly became a staple in the biomedical imaging community. Its architecture is still commonly used by itself, with modifications, or as a part of more extensive networks as in the Mask R-CNN described below (He et al., 2017).

The U-Net model is a type of encoder-decoder architecture. It has two main parts: an encoder that extracts features from the input image and a decoder that uses the extracted features to generate the segmentation map. The encoder consists of convolutional and pooling layers, which downsample the input image and extract hierarchical features at different scales. The decoder uses these features to upsample the image and generate the segmentation map. The U-Net model also uses skip connections, which connect the encoder and decoder at different levels, to combine features at different scales and provide a more detailed and accurate segmentation.

### 1.3.3 Faster R-CNN

New architectures for object detection quickly emerged as well, with the two most common ones being YOLO (Redmon et al., 2016) and Faster R-CNN (Ren et al., 2015). While the former focuses mainly on real-time detection, sacrificing accuracy for speed, the latter does the opposite. It lends itself to tasks in the biomedical domain, where almost instantaneous outputs are usually unnecessary. Faster R-CNN went through fast iterations with it, and its predecessors R-CNN (Girshick et al., 2014), and Fast R-CNN (Girshick, 2015) being developed in the timespan between 2014 and 2015. It consists of four main components: a feature extractor, a region proposal network (RPN), and lastly a box head consisting of a bounding box regressor and a classifier (Fig. 1.6, with an additional mask head).

The feature extractor is a typical CNN, similar to VGG16, extracting features from the image. This model *backbone* can be freely changed between smaller and bigger architectures, opening the possibility for a similar speed/accuracy tradeoff as YOLO. The RPN is a fully convolutional network that runs in parallel with the CNN and uses its feature maps to generate proposals. These proposals represent the possibility that an object is present at a location, regarding all objects belonging to the same foreground class. The fact that the number of objects in an image is unknown to the model is not easily reconciled with the fact that it needs a consistent output shape. This differs from classification or segmentation tasks, as the model will always output a predefined number of classifications or an output segmentation mask of a defined shape. This necessitates using a predefined number of anchors, which are boxes of different sizes and aspect ratios placed at regular intervals across the input image. At each anchor and box, a prediction can be made whether it contains a foreground object, thus generating the proposals.

Once the proposals have been generated, the classifier uses the features from the CNN and the proposals from the RPN to classify each region as belonging to a

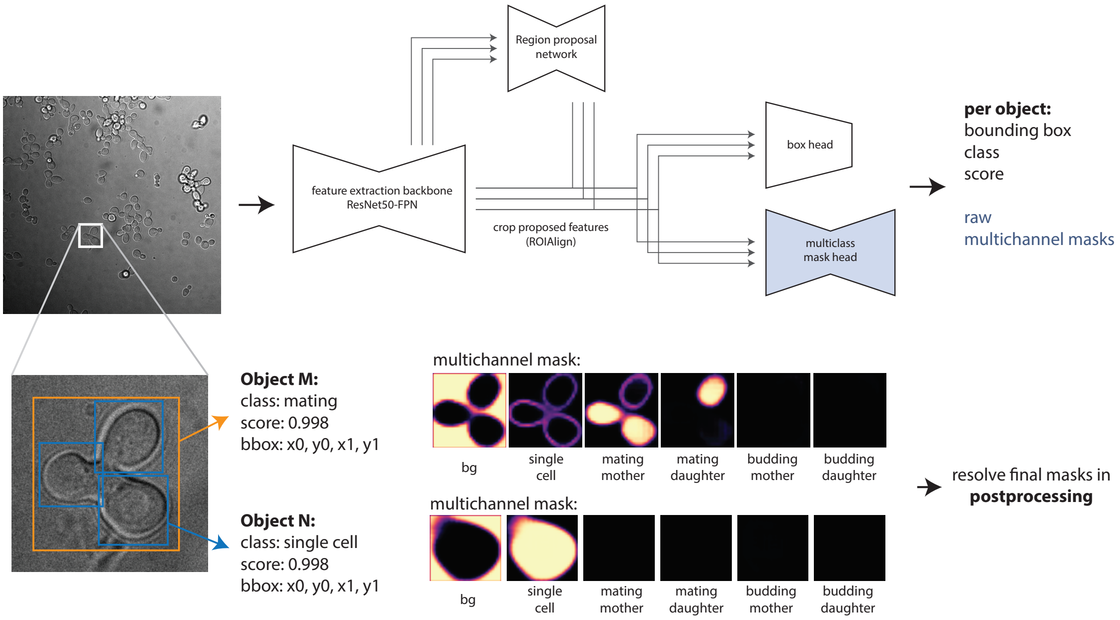


Figure 1.6: Adapted Mask R-CNN architecture containg an additional multi-mask header (Bunk et al., 2022). The box and mask headers are stacked on top of the model backbone and RPN.

specific object class or as background. To already discard unlikely matches and save computational resources, only the top-k proposals are forwarded from the RPN. The classifier and regressor are called model heads and are stacked on top of the backbone and the RPN. They are modular and can be easily switched and modified independently from the rest of the model. The classifier is trained to assign class labels to the proposals and uses the features from the CNN to make these predictions. The bounding box regressor uses the features and proposals to refine the locations and sizes of the bounding boxes around the objects. This can improve the accuracy of the bounding boxes and make the detection results more precise.

While it also introduces much complexity to the model, the modularity of the Faster R-CNN architecture also means that it can be easily extended beyond the originally proposed object detection task. While switching the backbone of the model can greatly impact its performance, it mainly arises from the use of task-specific network headers on top of the backbone and RPN. The most prominent example is Mask R-CNN (He et al., 2017). Here, the original object detection header is complemented by an additional semantic segmentation header, which is similar, but smaller than the U-Net architecture (Ronneberger et al., 2015). A binary mask is generated for each box, which can then be stitched into a whole instance segmentation image with the information from the detection header.

The topic of instance/semantic segmentation and object detection are closely related, so one might not be surprised to see this use-case implemented. However, the same paper also describes another custom header for keypoint detection (Keypoint R-CNN). Here specific keypoints are detected in an image; this is mainly used to find keypoints for human pose estimation, e.g. joints or landmarks of the face like the nose or edges of the mouth. Custom headers are also used for even more objectives, like improving Few-Shot training (Qiao et al., 2021).

## 1.4 Machine Learning in Biology

While constant progress is happening in DL and CV (Sapoval et al., 2022), it is not easily transferred into the biological domain. Biology asks unique questions, often requiring unique tools to answer them. On the one hand, computer vision is heavily focused on specific, outlined problems and benchmarks that do not translate well to biological problems. On the other hand, creating tools from well-performing models that can easily be integrated into the daily routine of the lab poses its own set of challenges.

### 1.4.1 Adapting models to biological problems

In the past ten years since the viable training of an image classification CNN by (Krizhevsky et al., 2012), AI models have experienced a renaissance with massive interest from both the commercial and the scientific side. A new research area has surfaced, and biological or medical applications are just a small subset. Many problems with applying general AI research to biological problems arise from the stark difference in input data that goes into the models. While microscopy images can be broadly seen as pictures, as the human eye can view them just like a picture taken from a phone, they differ significantly in contents and shape.

Image analysis networks are usually built to accept images in a three-dimensional shape of $[channels, height, width]$ with $channels = 3$, one each for each colour of the common RGB-format (Gulli & Pal, 2017). While RGB images are used in some biological areas, for example, when imaging histopathological slices, they are instead often recorded as grayscale images with one single channel. In the case of fluorescence microscopy, recordings of different fluorescent channels are often stacked in an image, resulting in sometimes even more than three channels. This will require either modification to the input layers of a CNN to process the images or changing the input images themselves to conform to the model structure.

Images of even higher dimensionality, such as three-dimensional images or focus series require even more pre-processing or even dedicated model architectures. Additionally, the content, and thus the embeddings of a picture or a microscopy image, differ significantly. In microscopy, objects are kept in focus as much as possible, while it is customary in pictures to have out-of-focus objects and different perspectives. Inherent statistics of an image dataset, like pixel value distribution and contrast, greatly impact how a model performs. A common difference is the lower SNR in microscopy images compared to well-lit pictures. Thus, more tweaks on a model may be required to compensate for changes in these statistics.

These discrepancies make transfer learning, where models are pre-trained on a larger, general dataset and then fine-tuned on new data, difficult. Transfer learning is a big part of current state-of-the-art models. It helps when the amount of training data in a new dataset is limited, as it allows the model to leverage the knowledge of a previous domain containing many training examples to improve its performance on the new task (Zhuang et al., 2020). While transfer learning from non-biological domains to a biological task can be done to success (Maqsood et al., 2019), its performance will still be a bit degraded compared to in-domain transfer learning. An ideal setup, for example, in fluorescence microscopy, would be to have a large, general dataset containing various fluorescent cells, which can then be transferred to tasks examining specific structures. A good example of in-domain transfer learning is the transfer of knowledge gained on a dataset of the well-understood model organism *Caenorhabditis elegans* to other organisms to predict mRNA splice sites (Schweikert et al., 2008).

Limited ground truth and the difficulty of acquiring new annotations to create these kinds of big datasets is a recurring problem in biomedical AI research. Especially when compared to big, non-biological datasets like COCO (Lin et al., 2014) with 328,000 pictures or ImageNet (Deng et al., 2009) with over 14 million pictures, the effort to build such a basis seems insurmountable. The Broad Bioimage Benchmark Collection (Ljosa et al., 2012) is a big driver in this effort, especially with their dataset from the Kaggle DataScienceBowl 2018, which provided 841 images of different nuclei acquisitions (Caicedo et al., 2019). The lack of common datasets also precludes the existence of benchmarks on those, making reproducibility in biological AI studies even more of a problem than it already is (Wagner et al., 2022). Experimental setups vary wildly, so the performance of AI models might do the same, even if the general task and organism are the same.

The lack of computational resources can also inhibit the use of AI in biology. Training a bigger AI model can be computationally expensive and usually requires GPU capacities to train them within an acceptable time-frame (Wang et al., 2019). As the current trend with transformer networks reaches unseen model sizes (Floridi & Chiriatti, 2020), it is unlikely for this to change in the near future. While research groups from bioinformatical areas are not strangers to this issue, as high-performance computing was already part of their research (Hey & Trefethen, 2003), the average biological researcher might not have these computational capacities. A good approach is to move model training into the cloud, and while this might still be prohibitively expensive, efforts have been made to offer a free alternative within Google's Colab platform to biological researchers (von Chamier et al., 2021). Another good approach is to take bigger models such as DETR, a detection model based on the transformer architecture (Carion et al., 2020), and reduce the model's size while preserving its performance for biomedical tasks as done with GasHis-Transformer (Chen et al., 2022).

### 1.4.2 Custom Faster R-CNN heads

In my work YeastMate (Bunk et al., 2022), I adapted the common Mask R-CNN model architecture to a specific problem in the biological domain. *S. cerevisiae* cells can reproduce both asexually and sexually; in both, a new daughter cell will emerge from either the mother cell or the two fused mother cells. To analyse these reproduction events, it is important not only to find them but also to be able to differentiate between mother and daughter cells. This might be a trivial task if the daughter cell is in its earliest stage and much smaller than the mother cells. However, to correctly classify them in their later stages, the conformation of the whole reproduction event must be considered.

We solved this problem with an implementation reminiscent of the mask header from Mask R-CNN. Instead of just generating a binary mask and assigning it a class value transmitted by the detection header, the segmentation head is trained to produce a multi-categorical mask containing information on the class value of the included objects (Fig. 1.6). Here, every mask layer is returned instead of reducing the mask header output to the predicted mask of the class specified by the detection head. Equally, the model is trained with multi-dimensional masks containing the sub-classifications of the reproduction event. To facilitate this change in training objective, the usual BCE loss for binary image segmentation is replaced by a categorical cross-entropy (CE) loss:

$$\mathrm{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right)$$

$$\mathrm{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j} \log \hat{y}_{i,j}$$

where $N$ is the number of samples, $y_i$ is the true label for sample $i$, $\hat{y}_i$ encodes the probabilities of the predicted labels for sample $i$, and $C$ is the number of classes.

The multiclass segmentation mask allows for postprocessing the acquired detections into their respective sub-classes (Fig. 1.7). Instances of segmented single-cell objects can be compared with the scores of the multiclass segmentation at the same pixel locations. The instance of the single cell can then be matched to the sub-class using linear sum assignment. Additional constraints can be defined, such as the minimum and the maximum number of mother or daughter cells. When setting the acceptable amount range of daughter cells to 0-1, a second daughter cell will be rejected, even with a high confidence score. In the inverse, a daughter cell might be rejected if its confidence score is too low, as having no daughter cell lies within our constraints. This helps avoid false positives, as even if an unrelated cell sits at the location a daughter cell would be expected, the multiclass segmentation will try to reject it based on texture and edges.

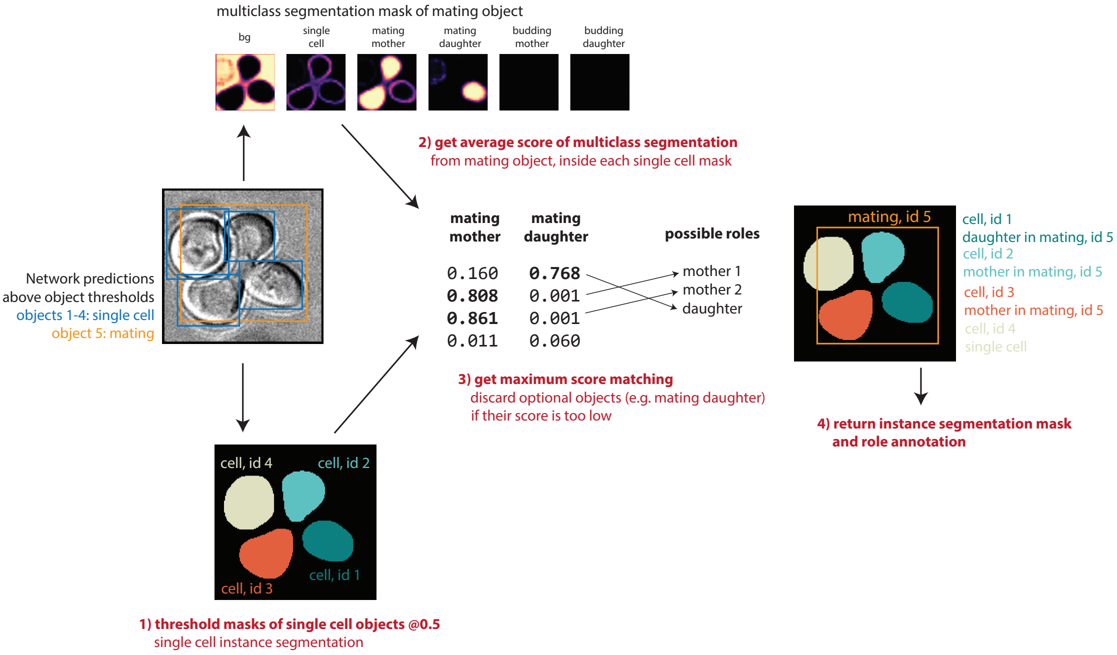


Figure 1.7: Post-processing steps of the generated yeast mating multi-masks (Bunk et al., 2022).

The problem of identifying and tracking cell divisions and reproduction events is a unique biological problem. Especially when the imaged cells are in a colony or very close together, the resulting image will not have an equivalent in everyday pictures like in ImageNet (Deng et al., 2009). This example shows that while domain-specific adaptions to DL models are sometimes required, they can usually still be solved with minimal changes and interference so as not to diverge too far away from the constant progress in DL.

### 1.4.3 Creating tools for the daily lab routine

DL itself in its current form is not even a decade old at this point, and the knowledge of how to efficiently deploy trained models in production, called MLOps, even more so (Alla & Adari, 2021). That might translate to having dedicated servers that can be easily used for routine high-throughput applications or the knowledge of how to package models and provide them to the user without any further requirements. However, in the field of biomedical research, there is often a lack of expertise and knowledge in this area, which can make it challenging to deploy DL models reliably and effectively. Another challenge is the high barrier to entry for users with limited programming experience. Many researchers in the field of biomedical research may have little or no programming experience and may not be familiar with the tools and technologies used in DL. This can make it difficult for them to use DL models and tools and can limit their ability to apply these methods to their research.

In order to overcome these challenges, it is essential for researchers and tool developers to focus on making DL tools and models more accessible and user-friendly. This can include providing easy installation, low system requirements, wide availability, and good documentation. By lowering the entrance barrier and making DL tools more widely available, researchers can expand the use of these methods in biomedical research. The past years have seen some very useful tools that keep the user experience in mind, like DeepLabCut (Mathis et al., 2018), ilastik (Berg et al., 2019) and CellPose (Stringer et al., 2021). Still, the onus is on us as the model developers and tool providers to open up our tools to an audience as broad as possible.

# Chapter 2

# Original Papers

## 2.1 Regulatory encoding of quantitative variation in spatial activity of a *Drosophila* enhancer

### 2.1.1 Main Paper

SCIENCE ADVANCES | RESEARCH ARTICLE

DEVELOPMENTAL BIOLOGY

# Regulatory encoding of quantitative variation in spatial activity of a *Drosophila* enhancer

**Yann Le Poul[1]*, Yaqun Xin[1]*, Liucong Ling[1], Bettina Mühling[1], Rita Jaenichen[1], David Hörl[2], David Bunk[2], Hartmann Harz[2], Heinrich Leonhardt[2], Yingfei Wang[3], Elena Osipova[1], Mariam Museridze[1], Deepak Dharmadhikari[1], Eamonn Murphy[1], Remo Rohs[3], Stephan Preibisch[4,5], Benjamin Prud'homme[6†], Nicolas Gompel[1†]**

Copyright © 2020 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution NonCommercial License 4.0 (CC BY-NC).

**Developmental enhancers control the expression of genes prefiguring morphological patterns. The activity of an enhancer varies among cells of a tissue, but collectively, expression levels in individual cells constitute a spatial pattern of gene expression. How the spatial and quantitative regulatory information is encoded in an enhancer sequence is elusive. To link spatial pattern and activity levels of an enhancer, we used systematic mutations of the *yellow spot* enhancer, active in developing *Drosophila* wings, and tested their effect in a reporter assay. Moreover, we developed an analytic framework based on the comprehensive quantification of spatial reporter activity. We show that the quantitative enhancer activity results from densely packed regulatory information along the sequence, and that a complex interplay between activators and multiple tiers of repressors carves the spatial pattern. Our results shed light on how an enhancer reads and integrates trans-regulatory landscape information to encode a spatial quantitative pattern.**

## INTRODUCTION

Enhancers constitute a particular class of cis-regulatory elements that control in which cells a gene is transcribed, when, and at which rate (*1*, *2*). Notably, enhancers play a central role during development in plants and animals (*3*), generating patterns of gene expression that delineate embryonic territories and prefigure future forms (*4*). How the information determining these patterns is encoded in a developmental enhancer has therefore been at the center of attention for several decades. Enhancers integrate spatial information from transcription factors (TFs) bound to them, and the number, affinity, and arrangement of TF binding sites (TFBSs) in the enhancer sequence are relevant to the enhancer spatial activity [reviewed in (*5*)]. However, the logic of TFBS organization that determines a spatial pattern is not sufficiently understood to reliably design a functional synthetic enhancer driving correct expression levels (*6*, *7*).

The study of developmental enhancers has been polarized by two conceptions of gene expression patterns. Until recently, most studies have referred to enhancer activities in qualitative terms exclusively, where the notion of spatial pattern evokes discrete and relatively homogeneous domains of gene expression (*8*). With the rise of genomics from the early 2000s, it has become possible to precisely measure gene expression and, by extension, enhancer activity. However, whether it is measured in a given tissue or in single cells, this quantification of gene expression is done at the expense of losing spatial information [e.g., (*9*–*11*)], with few exceptions [e.g., (*12*, *13*)]. It is nevertheless critical to appreciate that the overall levels and the spatial pattern of activity in a given tissue are intrinsically linked. Therefore, to understand how a spatial pattern of gene expression is encoded in the sequence of an enhancer, it is necessary to measure quantitative variation of gene expression in space in the tissue where the enhancer is active. Leading this endeavor, recent studies have quantified spatial enhancer activity but without considering the pattern itself as a quantitative object (*13*–*18*).

To pursue this effort of measuring quantitative variation in spatial gene expression, we have analyzed the structure and the functional logic of a compact *Drosophila* enhancer sequence with quantitative measurements of its spatial activity in fly wings. The so-called *spot*$^{196}$ enhancer, from the *yellow* gene of the fruit fly *Drosophila biarmipes*, drives a patterned gene expression in pupal wings with heterogeneous expression levels among cells (*19*–*21*). The *spot*$^{196}$ enhancer sequence contains at least four TFBSs for the activator Distal-less (Dll) and at least one TFBS for the repressor Engrailed (En) (Fig. 1A) (*19*, *20*). Together, these inputs were considered to be sufficient to explain the spatial activity of *spot*$^{196}$ in the wing, with activation in the distal region and repression in the posterior wing compartment (*19*, *20*). Grafting TFBSs for these factors on a naïve sequence in their native configuration, however, proved insufficient to produce regulatory activity in wings. This prompted us to dissect the *spot*$^{196}$ element further to identify what determines its regulatory activity, considering simultaneously spatial pattern and activity levels.

We first introduced systematic small-scale mutations along the 196 base pairs (bp) of the enhancer sequence to test the necessity of the mutated positions; we then randomized large blocks of the enhancer sequence to test the sufficiency of the remaining intact sequence to drive activity. To assess the activity of each mutant enhancer, we devised a pipeline that uses comprehensive descriptors to quantify variations in reporter activity levels across the wing of *Drosophila melanogaster* transgenic lines. Our quantitative analysis revealed a

---

[1]Evolutionary Ecology, Ludwig-Maximilians Universität München, Fakultät für Biologie, Biozentrum, Grosshaderner Strasse 2, 82152 Planegg-Martinsried, Germany. [2]Human Biology and Bioimaging, Ludwig-Maximilians Universität München, Fakultät für Biologie, Biozentrum, Grosshaderner Strasse 2, 82152 Planegg-Martinsried, Germany. [3]Quantitative and Computational Biology, Departments of Biological Sciences, Chemistry, Physics and Astronomy, and Computer Science, University of Southern California, Los Angeles, CA 90089, USA. [4]Berlin Institute for Medical Systems Biology, Max Delbrück Center for Molecular Medicine, Robert-Rössle-Str. 10, 13092 Berlin, Germany. [5]Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, VA 20147, USA. [6]Aix-Marseille Université, CNRS, IBDM, Institut de Biologie du Développement de Marseille, Campus de Luminy Case 907, 13288 Marseille Cedex 9, France.
*These authors contributed equally to this work.
†Corresponding author. Email: benjamin.prudhomme@univ-amu.fr (B.P.); gompel@bio.lmu.de (N.G.)

Downloaded from https://www.science.org on January 26, 2022

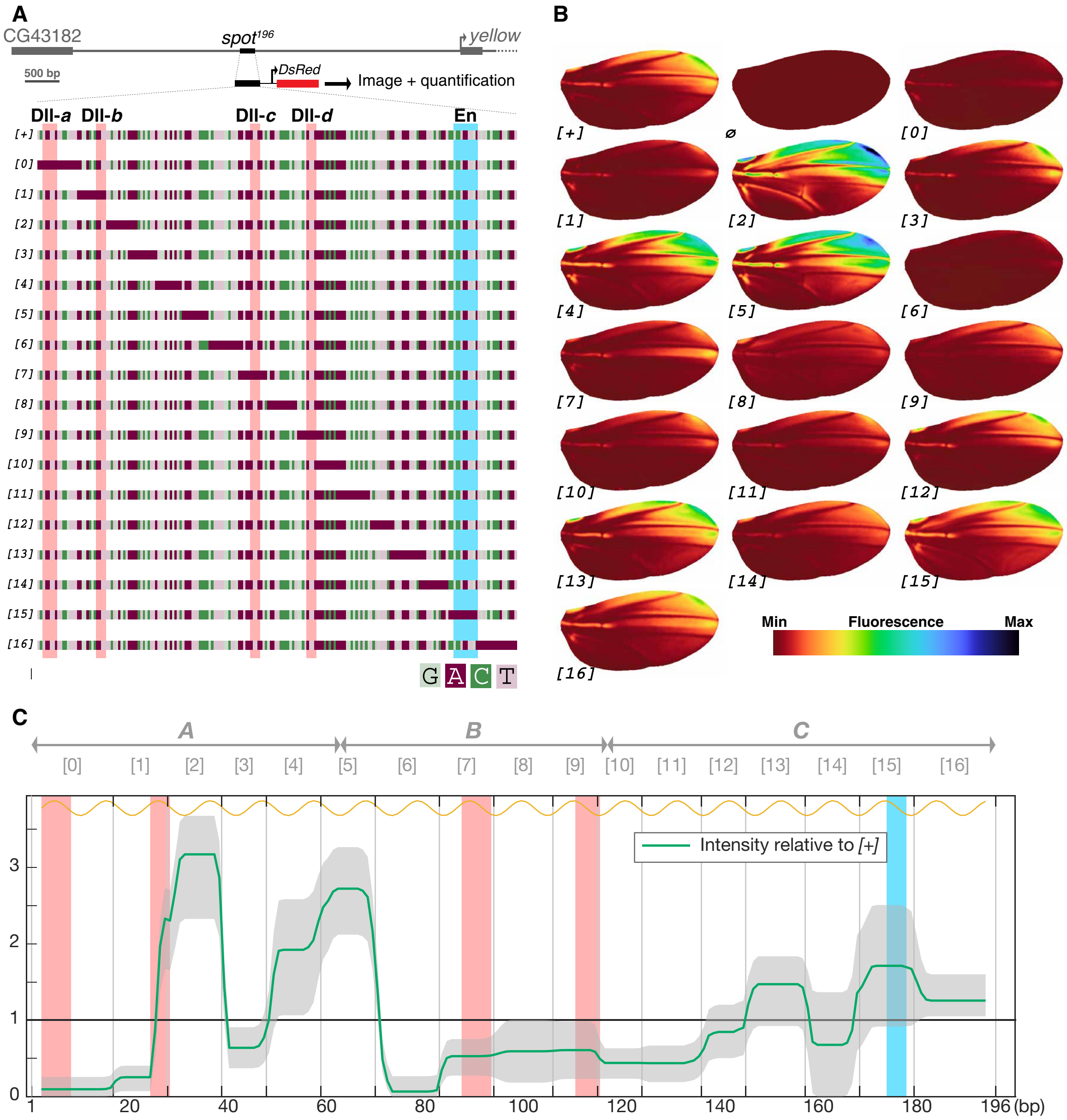


**Fig. 1. A mutational scan of the *D. biarmipes spot*$^{196}$ enhancer with a quantitative reporter assay.** (**A**) Wild-type (*[+]*) and mutant (*[0]* to *[16]*) versions of the *spot*$^{196}$ enhancer from the *D. biarmipes yellow* locus (depicted at the top) were cloned upstream of a DsRed reporter to assay their respective activities in transgenic *D. melanogaster*. Each mutant targets a position of the enhancer, where the native sequence was replaced by an A-tract (color code: light green, guanine; purple, adenine; dark green, cytosine; pink, thymine). Four characterized binding sites for the TF Distal-less (Dll-a, Dll-b, Dll-c, and Dll-d) (*19*) are highlighted in red, and a single binding site for the TF Engrailed (*20*) is highlighted in blue across all constructs. (**B**) Average wing reporter expression for each construct depicted in (A) and an empty reporter vector (ø). Each wing image is produced from 11 to 77 individual wing images (38 on average; data file S2), aligned onto a unique wing model. The average image is smoothened, and intensity levels are indicated by a colormap. (**C**) Mutational effect on intensity of activity along the *spot*$^{196}$ sequence. The phenotypic effect of each mutation described in (A) along the *spot*$^{196}$ sequence (*x* axis) is plotted as the average level of expression across the wing relative to the wild-type average levels. Shaded gray areas around the curve represent the 95% confidence interval of the average levels per position. "1" on the *y* axis represents the mean wild-type intensity of reporter expression. The graph shows how each construct departs from the wild-type activity (see Materials and Methods). Mutation positions in constructs *[0]* to *[16]* are indicated above the graph. The locations of blocks *A*, *B*, and *C*, analyzed in Fig. 3, are also indicated above the graph. The yellow curve above the graph indicates the helical phasing.

high density of regulatory information, with all mutated positions along the *spot*$^{196}$ enhancer sequence contributing significantly to the activity levels. It also outlined an unanticipated regulatory logic for this enhancer, where the spatial pattern in the wing results from a complex interplay between activators and multiple tiers of repressors carving a spatial pattern.

## RESULTS

### Regulatory information distributed along the entire *spot*$^{196}$ enhancer contributes to its quantitative spatial activity in the wing

We first systematically evaluated the potential role of all positions along the *spot*$^{196}$ enhancer sequence to produce an activity pattern

Downloaded from https://www.science.org on January 26, 2022

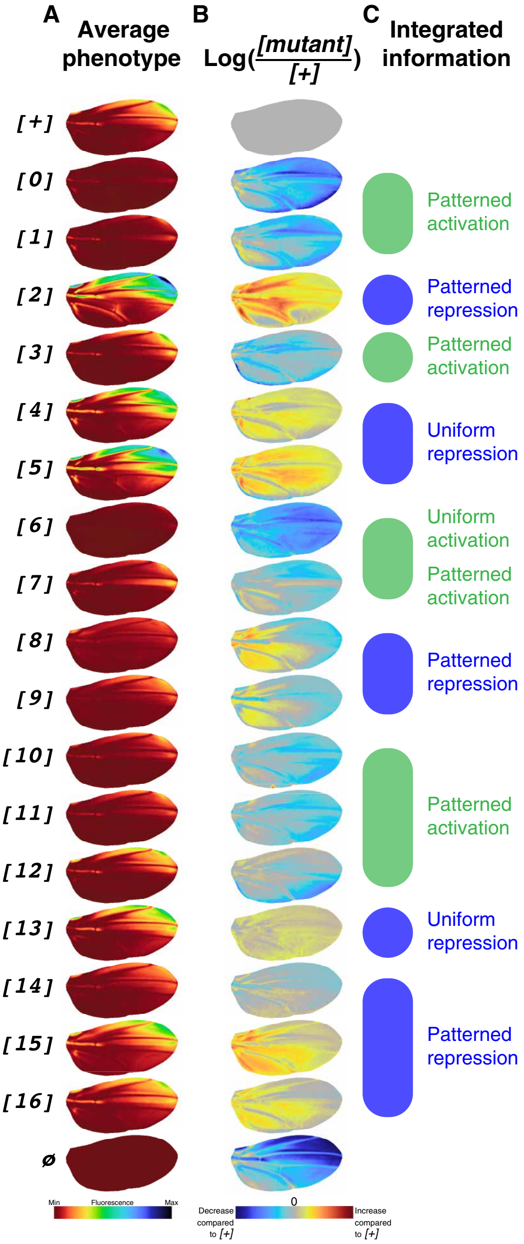


**Fig. 2. Trans-regulatory integration along the *spot*$^{196}$ sequence.** (**A**) Average phenotypes reproduced from Fig. 1B. (**B**) *logRatio* images [log(*[mutant]*/*[+]*) for intensity values of each pixel of registered wing images] reveal what spatial information is integrated by each position along the enhancer sequence. For instance, a blue region on an image indicates that the enhancer position contains information for activation in this region. When mutated, this enhancer position results in lower activity than *[+]* in this region of the wing. Note that *logRatio* illustrates local changes between *[+]* and mutants far better than image differences (fig. S3) in regions of relatively low activity. (**C**) Summary of spatial information integrated along the enhancer sequence.

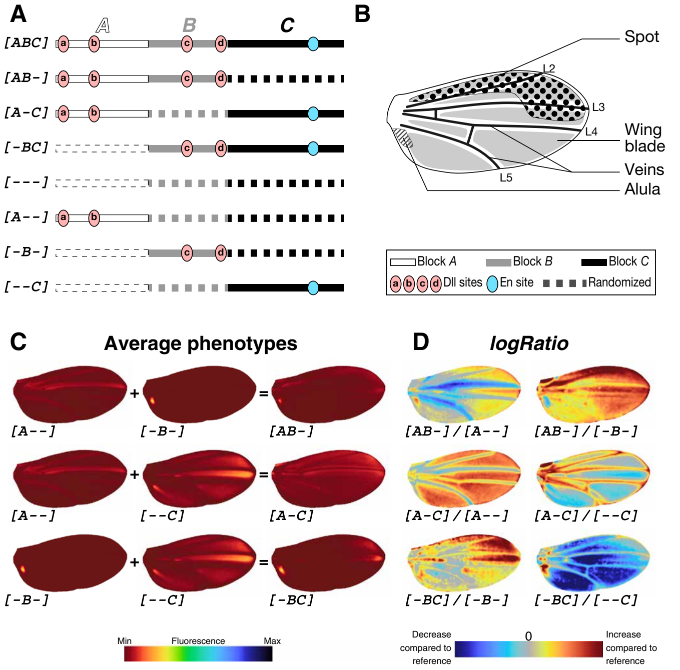


**Fig. 3. Regulatory interactions in the *spot*$^{196}$ sequence.** (**A**) Schematics of constructs with block randomizations. The *spot*$^{196}$ sequence was arbitrarily divided into three blocks (A, 63 bp; B, 54 bp; C, 79 bp). In each construct, the sequence of one, two, or all three blocks was randomized. (**B**) Terminology for parts of the wing where constructs from (A) drive reporter expression. (**C**) Average phenotypes resulting from constructs in (A). Constructs where single blocks remain indicate the sufficiency of these blocks to promote wing activity: *A* in the veins, *B* in the alula, and *C* at high levels across the wing blade. Constructs with two nonrandomized blocks show the effect of one block on the other. For instance, *B* is sufficient to suppress the wing blade activation promoted by *C*, as seen by comparing *[-B-]*, *[--C]*, and *[-BC]*. Colormap of average phenotypes normalized for all constructs of the block series, including block permutations of Fig. 4B. (**D**) Block interactions are best visualized with *logRatio* images of construct phenotypes shown in (C). For each *logRatio*, the denominator is the reference construct, and the image shows on a logarithmic scale how much the construct in the numerator changes compared to this reference. For instance, log(*[-BC]*/*[--C]*) shows the effect of *B* on *C*, a global repression, except in the spot region. Colormap indicates an increase or a decrease of activity compared to the reference (denominator). For an overview of all comparisons, particularly the relative contribution of each block to the entire enhancer activity, see fig. S4 (C to F).

and wild-type levels of gene expression. We generated a series of mutants scanning the element and thereby testing the necessity of short adjacent segments to the enhancer function. Notably, we made no prior assumption (e.g., predicted TFBSs) on the function of the mutated nucleotides. We maximized the disruption of sequence information by introducing stretches of 10 to 18 bp (11.5 bp on average) of poly(dA:dT), also known as A-tracts (*22*), at adjacent positions along the sequence (Fig. 1A). Thus, the sequence of each of the 17 constructs (*spot*$^{196\ [0]}$ to *spot*$^{196\ [16]}$, or *[0]* to *[16]* in short; Fig. 1A) is identical to the wild-type *spot*$^{196}$ (*[+]* in short), except for one segment where the sequence was replaced by the corresponding number of adenines. These mutations affect the local sequence composition, without changing distances or helical phasing in the rest of the enhancer. We measured activities of each mutant enhancer in the wing of the corresponding reporter construct line of *D. melanogaster*, here used as an experimental recipient for site-specific integration. In brief, for each reporter construct line, we imaged individually around 30 male wings (1 wing per fly) under bright-field and fluorescent light. We detected the venation on the bright-field images of all wings and used it to compare reporter activity across wings. For this, we applied a deformable model to

Downloaded from https://www.science.org on January 26, 2022

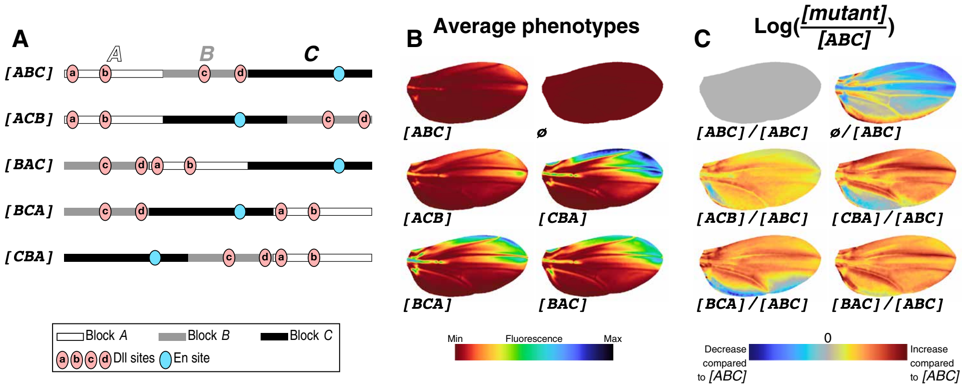


**Fig. 4. Block permutations scale the activity of the *spot*$^{196}$ enhancer.** (**A**) Schematics of constructs with block permutations. In this series, the same blocks of sequences as in Fig. 3A were permutated. (**B**) Average phenotypes resulting from constructs in (A). Colormap of average phenotypes normalized for all constructs of the block series, including block randomizations of Fig. 3C and fig. S4B. (**C**) Average phenotypes in (B) compared to the average phenotype of the wild-type *[ABC]* (*logRatio*). Note that, in contrast to constructs with randomized blocks (Fig. 3), constructs with block permutations result in near-uniform changes of activity across the wing. Colormap indicates an increase or a decrease of activity compared to the wild-type enhancer *[ABC]*.

warp the fluorescent image of each wing, using landmarks placed along the veins of the corresponding bright-field image and aligning them to a reference venation (see Materials and Methods for details). The resulting dataset is a collection of fluorescence images for which the venation of all specimens is perfectly aligned. These images, represented as the list of fluorescence intensity of all pixels, constitute the basis of all our quantitative dissection. To assess whether the activity driven by a given enhancer sequence significantly differs from any other, wild type or mutant, we used the scores produced by principal components analysis (PCA) that comprehensively summarizes the variation of the pixel intensities across wings. To visualize the reporter activity per line, we used images representing the average activity per pixel (hereafter average phenotype).

The activity of each mutant (Fig. 1B) differs significantly from that of *[+]*, as measured in the PCA space (fig. S1 and data file S1). This means that the activity of each mutant had some features, more or less pronounced, that significantly differentiate its activity from *[+]*, revealing the high density of regulatory information distributed along the sequence of *spot*$^{196}$. The magnitude and direction of the effects, however, vary widely among mutants, ranging from activity levels well above those of *[+]* to a near-complete loss of activity.

The average activity levels of each mutant construct in the wing relative to the average activity levels of *[+]* show how effect directions and intensities are distributed along the enhancer sequence (Fig. 1C). This distribution of regulatory information and the magnitude and direction of the effects, including several successions of overexpressing and underexpressing mutants, suggest a more complex enhancer structure than previously thought (*20*). The density of regulatory information is also reminiscent of what has been found for other enhancers (*13*, *23*, *24*).

In principle, the localized mutations we introduced can affect the *spot*$^{196}$ enhancer function through nonexclusive molecular mechanisms. Mutations may affect TF-DNA interactions by disrupting TFBS cores or by influencing TF binding at neighboring TFBSs [for instance, by altering DNA shape properties (*25*, *26*)]. A-tract mutations may also influence nucleosome positioning and thereby the binding of TFs at adjacent sites (*27*). Not exclusively, because of stacking interactions between adjacent As and Ts, they increase local DNA rigidity (*22*, *28*, *29*) and may thereby hinder or modulate TF interactions. These changes in rigidity, which we have evaluated for our mutant series (fig. S2A), may affect TF-TF interactions (fig. S2B). Regardless of the precise molecular mechanisms underlying the mutations we introduced in the *spot*$^{196}$ sequence, we wanted to assess how they affect the integration of spatial information along the enhancer sequence.

## An enhancer's view on the wing trans-regulatory landscape revealed by *logRatio* images

We have introduced a spatial visualization of the intensity of effect of a mutation on the enhancer activity. We computed the pixel-wise log of the ratio between two average phenotypes (single mutants over *[+]*) at every pixel (*30*), hereafter noted *logRatio*. The advantages of using *logRatio* are detailed in the Supplementary Materials and briefly summarized here. *logRatio* images show visually how much a mutant affects the enhancer activity across the wing proportionally to the local activity level. By contrast, the absolute difference in expression is generally locally linked to the level of expression. Therefore, effects in areas of high activity tend to be much more visible than those in areas of low activity (compare Fig. 2 and fig. S3). *logRatio* images instead represent the local proportional effects and are therefore suitable to reveal the variety of spatial effects of mutations, irrespective of the expression pattern itself.

Depending on how TF integration is modified by a mutant, *logRatio* images can also reflect the distribution of the individual spatial inputs received and integrated along the *spot*$^{196}$ sequence. They can be particularly informative when both a TFBS and the spatial distribution of the cognate TF are known, as they shed light on how directly the TF information is integrated. This is the case for En and Dll, for which TFBSs have been previously characterized in *spot*$^{196}$ (*19*, *20*). The disruption of an En binding site (Fig. 1, A and B, construct *[15]*) resulted in a proportional increase of activity in the posterior wing compartment (75%, $F_{1,124} = 77.8$, $P = 8.8818 \times 10^{-15}$). The log(*[15]*/*[+]*) image (Fig. 2) shows that mutant *[15]* proportionally affects the activity mostly in the posterior wing. The effect

Downloaded from https://www.science.org on January 26, 2022

correlates with En distribution (*20*) and is consistent with the repressive effect of its TF. Contrary to what the average phenotypes suggested (Fig. 1C), mutant *[16]* shows a very similar *logRatio* to that of *[15]*, albeit with only 25% increase in activity. The effect of mutant *[16]* was barely discernible when considering the variation in the overall fluorescence signal (Fig. 1C), illustrating the power of the *logRatio* analysis to detect local effects in areas of low activity. Mutations that disrupted characterized Dll binding sites (Fig. 1, A and B, constructs *[0]*, *[1]*, *[7]*, and *[9]*) resulted in strong reduction in reporter expression (90%, $F_{1,74} = 143.3$, $P = 0$; 75%, $F_{1,78} = 109.3$, $P = 2.2204 \times 10^{-16}$; 47%, $F_{1,107} = 75.4$, $P = 4.8073 \times 10^{-14}$; and 39%, $F_{1,74} = 23.2$, $P = 7.6363 \times 10^{-6}$, respectively; data file S1). The *logRatio* images for mutants *[0]*, *[1]*, and, to a lesser extent, *[7]* show a patterned decrease of activity in line with Dll distribution in the wing (Fig. 2) (*19*), with a proportionally stronger loss of activity toward the distal wing margin. This corroborates previous evidence that Dll binds to these sites. The respective *logRatio* images for segments *[0]* and *[1]* correlate with levels of Dll across the wing. This suggests that these sites individually integrate mostly Dll information and do so in a near-linear fashion. Site *[9]*, which produces a relatively different picture with areas showing overexpression, is discussed below. Mutations of Dll sites, however, have nonadditive effects, as mutants *[0]*, *[1]*, *[7]*, and *[9]* result in a decrease of activity levels by 90, 75, 47, and 39% compared to *[+]*, respectively. This nonadditivity could be explained by a strong cooperative binding of Dll at these sites or, alternatively, by considering that these Dll TFBS are interacting with other sites in the sequence.

In addition, we noted that, despite mutating a Dll TFBS, mutant *[9]* showed a substantially different *logRatio* than *[0]* and *[1]* but similar to *[8]*, with a repressing activity in the posterior wing compartment, proximally, and a distal activation (Fig. 2B). This dual effect could be explained by the disruption of the Dll site along with a distinct TFBS for a posterior repressor. Alternatively, a single TFBS could be used by different TFs with opposite activities. In this regard, we note that the homeodomains of Dll and En have similar binding motifs (*31*) and could both bind the Dll TFBS disrupted by *[9]* (and possibly *[8]*). The posterior repression of En and the distal activation of Dll seem compatible with this hypothesis.

### Unraveling trans-regulatory integration along the *spot*$^{196}$ sequence

Following the same approach, we next analyzed the information integrated in other segments. Apart from the known Dll and En TFBSs, the enhancer scan in Fig. 1C identified several segments with strong quantitative effects on the regulatory activity. Between the two pairs of Dll TFBSs, we found an alternation of activating sites [*[3]* and *[6]*, reducing overall levels by 36% ($F_{1,69} = 17.6$, $P = 7.8336 \times 10^{-5}$) and 93% ($F_{1,98} = 284.9$, $P = 0$) compared to *[+]*, respectively] and strong repressing sites [*[2]*, *[4]*, and *[5]*, with an overall level increase of 3.2-fold ($F_{1,72} = 511.5$, $P = 0$), 1.9-fold ($F_{1,85} = 103.2$, $P = 2.2204 \times 10^{-16}$), and 2.7-fold ($F_{1,82} = 426.5$, $P = 0$) compared to *[+]*, respectively]. Construct *[3]* proportionally decreases the expression mostly around the wing veins (Fig. 2B), suggesting that this segment integrates information from an activator of the vein regions. We had found a similar activity for this region of *yellow* from another species, *Drosophila pseudoobscura*, where no other wing blade activity concealed it (*20*). The *logRatio* of mutant *[6]*, with a stronger, more uniform effect than for the other mutants that repress the activity, suggests a different trans-regulatory integration than Dll sites. We have recently shown that this site regulates the chromatin state of the enhancer (*21*). Regarding segments with a repressive effect, mutants *[4]* and *[5]* result in a fairly uniform relative increase in expression, different from the activity of *[2]*, indicating that the information integrated by these two regions (*[2]* versus *[4]* and *[5]*) likely involves different TFs. Three segments, *[6]*, *[0]*, and *[1]* (the last two containing previously known Dll binding sites), each decrease the activity levels by 75% or more. Finding additional strong repressive sites (*[2]*, *[4]*, and *[5]*) with a global effect on the enhancer activity across the wing is also unexpected.

The analysis revealed another activating stretch of the sequence, between 116 and 137 bp, as mutated segments *[10]* and *[11]* decreased activity by 56% relative to *[+]* and showed very similar *logRatios*. Mutant *[12]* showed a mixed effect, with practically, in absolute terms, no effect in the anterior distal wing quadrant. Last, segments *[13]*, *[14]*, and *[15]* showed a succession of repressing and activating sites, as we have seen for segments *[2]* to *[6]*, although with a lower amplitude. Mutant *[13]* caused an overall increase in activity (1.4-fold relative to *[+]*) with, proportionally, a uniform effect across the wing (*logRatio*). By contrast, mutant *[14]* decreased the overall activity by 36%, with a *logRatio* indicating an activating effect in the spot region and a repressive effect in the proximal part of the posterior wing compartment, similarly to mutants *[8]* and *[9]* but with lesser effects.

Together, this first dissection, focusing on the necessity of segments for the enhancer activity at the scale of a TFBS, which is typically 10 bp long (*32*), suggested a much higher density of regulatory information in the *spot*$^{196}$ enhancer than previously described (*19*, *20*). The nonadditivity of effects at Dll binding sites, three repressing and four activating and previously unidentified segments distributed in alternation along the enhancer, and the variety of their effects pointed to a complex regulatory logic, involving more (possibly six to eight) factors than just Dll and En. We resorted to a different approach to further probe the regulatory logic of *spot*$^{196}$.

### An interplay of activating and repressing inputs produces a spatial pattern of enhancer activity

The first series of mutations informed us on the contribution of the different elementary components of the *spot*$^{196}$ enhancer sequence to its regulatory activity. However, it failed to explain how these components integrated by each segment interact to produce the enhancer activity. To unravel the regulatory logic of this enhancer, it is required to understand not only which segments are sufficient to drive expression but also how elementary components underlying the regulatory logic influence each other. To evaluate the sufficiency of, and interactions between, different segments, we would require to test all possible combinations of mutated segments, namely, a combinatorial dissection. Doing this at the same segment resolution as above is unrealistic, because the number of constructs grows with each permutation. Instead, we used three sequence blocks of comparable sizes in the *spot*$^{196}$ enhancer—*A*, *B*, and *C*, defined arbitrarily (Fig. 3A)—and produced constructs where selected blocks were replaced by a randomized sequence (noted "-"). This second series, therefore, consists of eight constructs, including all combinations of one, two, or three randomized blocks, a wild-type *[ABC]* (which has strictly the same sequence as *[+]* from the first series), and a fully randomized sequence, *[---]*.

Downloaded from https://www.science.org on January 26, 2022

With these constructs, we can track which segments, identified in the first series as necessary for activation in the context of the whole *spot*$^{196}$, are also sufficient to drive activity (table S3; see Fig. 1C for the correspondence between the two series of mutations). Of the three blocks (constructs *[A--]*, *[-B-]*, and *[--C]*), only block *C* is sufficient to produce activity levels comparable to those of the wild-type *spot*$^{196}$ in the wing blade, although with a different pattern from *[ABC]* (fig. S4, A to C). Reciprocally, randomizing block *C* (construct *[AB-]*) results in a uniform collapse of the activity (fig. S4, A to C). We concluded that the sequence of block *C* contains information necessary and sufficient to drive high levels of activity in the wing in the context of our experiment. This is particularly interesting because *C* does not contain previously identified Dll TFBSs or strong activating segments. By contrast, blocks *A* and *B*, although they each contain two Dll sites, do not drive wing blade expression. The activating segments in block *C* revealed in the first dissection, particularly segments *[10]* and *[11]*, are therefore candidates to drive the main activity of *spot*$^{196}$ in the context of these reporter constructs.

Block *A* alone (*[A--]*) produces high levels of expression in the veins (fig. S4, A to C). Combined with block *C* (construct *[A-C]*), it also increases the vein expression compared to *C* alone. We concluded that *A* is sufficient to drive expression in the veins. Segment *[3]*, which proportionally decreased the activity mostly in the veins, could therefore be the necessary counterpart for this activation.

Block *B* alone drives expression only near the wing hinge, in a region called the alula (*[-B-]*; Fig. 3, B to D). The first dissection series, however, did not identify a mutated segment within block *B* that affected specifically the alula.

The necessity of Dll binding sites (in segments *[0]*, *[1]*, *[7]*, and *[9]*) and of segment *[6]*, and their insufficiency to drive activity in the wing blade in the context of block *A* alone, block *B* alone, or blocks *A* and *B* combined, suggest that these sites with a strong activation effect function as permissive sites. We next focused on understanding the interplay between repressing and activating sites to shed light on how the *spot*$^{196}$ patterning information is built. In the first series of constructs, we identified several strong repressing segments in block *A* (*[2]* and *[4]*) and block *B* (*[5]*). Using sufficiency reasoning with the second series of constructs, we further investigated how these inputs interacted with other parts of the enhancer (Fig. 3). These interactions are best visualized with *logRatios*, comparing this time double-block constructs to single-block constructs used as references (Fig. 3D and fig. S4, D to F). Block *B* has a strong repressive effect on block *C* throughout the wing, except at the anterior distal tip, where *C* activity is nearly unchanged [log(*[-BC]*/*[--C]*); Fig. 3D]. Likewise, log(*[AB-]*/*[A--]*) shows that *B* also represses the vein expression driven by *A*. Similarly, block *A* represses the *C* activity across the wing blade, except in the spot region log(*[A-C]*/*[--C]*). We have seen above that blocks *A* and *B* both contain not only strong repressing segments but also known Dll TFBSs. Because both *A* and *B* show a repressive effect on block *C*, except in the spot region, we submit that the apparent patterned activation by Dll may result from its repressive effect on direct repressors of activity, mostly at the wing tip. This indirect activation model would explain the nonadditivity of the individual Dll binding sites observed in the first construct series and why grafting Dll TFBSs on a naïve DNA sequence is not sufficient to create a wing spot pattern. Together, these results outline an unexpectedly complex regulatory logic that contrasts with the simple model we had initially proposed (*19*, *20*) and involves multiple activators and several tiers of repressors.

### Sequence reorganization affects activity levels of the *spot*$^{196}$ enhancer, not its spatial output

In a final series of experiments, we wondered whether the complex regulatory architecture uncovered by the first two mutant series was sensitive to the organization of the inputs. To test the effect of changes in the organization of enhancer logical elements, we introduced new constructs with permutations of blocks *A*, *B*, and *C* (Fig. 4A). These permutations preserve the entire regulatory content of the enhancer, except at the junction of adjacent blocks where regulatory information may be lost or created. All permutations that we have tested (four of five possible permutations) drive significantly higher levels of expression than the wild-type *[ABC]* [*[ACB]*: 2.9-fold ($F_{1,98} = 191.8$, $P = 0$); *[BAC]*: 6-fold ($F_{1,93} = 589.1$, $P = 0$); *[BCA]*: 5.8-fold ($F_{1,93} = 589.1$, $P = 0$); *[CBA]*: 8.4-fold ($F_{1,93} = 1664.2$, $P = 0$); Fig. 4B] yet with minor effects on the activity distribution proportionally to the wild type (Fig. 4C). We concluded from these experiments that, in terms of pattern, the regulatory output is generally resilient to large-scale rearrangements. As long as all inputs are present in the sequence, the spatial activity is deployed in a similar pattern, yet its quantitative activity is strongly modulated. Because they have little influence on the activity pattern, the rearrangements may not change the nature of the interactions within the enhancer or with the core promoter. Although we would need to challenge this conclusion with additional constructs and blocks with different breakpoints, we speculate that, molecularly, the block randomization perturbates the action of some of the uniformly repressing elements. It highlights the robustness of the enhancer logic to produce a given patterned activity.

## DISCUSSION

With this work, we have set to decipher the regulatory logic of an enhancer, *spot*$^{196}$. The viewpoint presented here is the information that the enhancer integrates along its sequence. Combined with the quantitative measurement of enhancer activity in a tissue, the wing, this information reveals the enhancer regulatory logic and how it reads the wing trans-regulatory environment to encode a spatial pattern. The strength of our arguments stems from the introduction of two complementary aspects of the method (discussed in the following sections): one to combine the assessment of necessity and sufficiency of regulatory information in our analysis and another to compare the spatial activity of enhancer variants (*logRatio*).

### Regulatory necessity and regulatory sufficiency

When dissecting a regulatory element, it is straightforward to assess the necessity of a TFBS or any stretch of the sequence to the activity, by introducing mutations. It is generally more difficult to assess whether the same sequence is sufficient to promote regulatory activity at all, and most enhancer dissections are focusing on necessity analysis [see, for instance, (*12*, *17*, *19*, *20*, *23*, *33*–*37*)]. However, our study shows that, to decipher regulatory logic and eventually design synthetic enhancers, understanding which regulatory components are sufficient to build an enhancer activity is key.

Downloaded from https://www.science.org on January 26, 2022

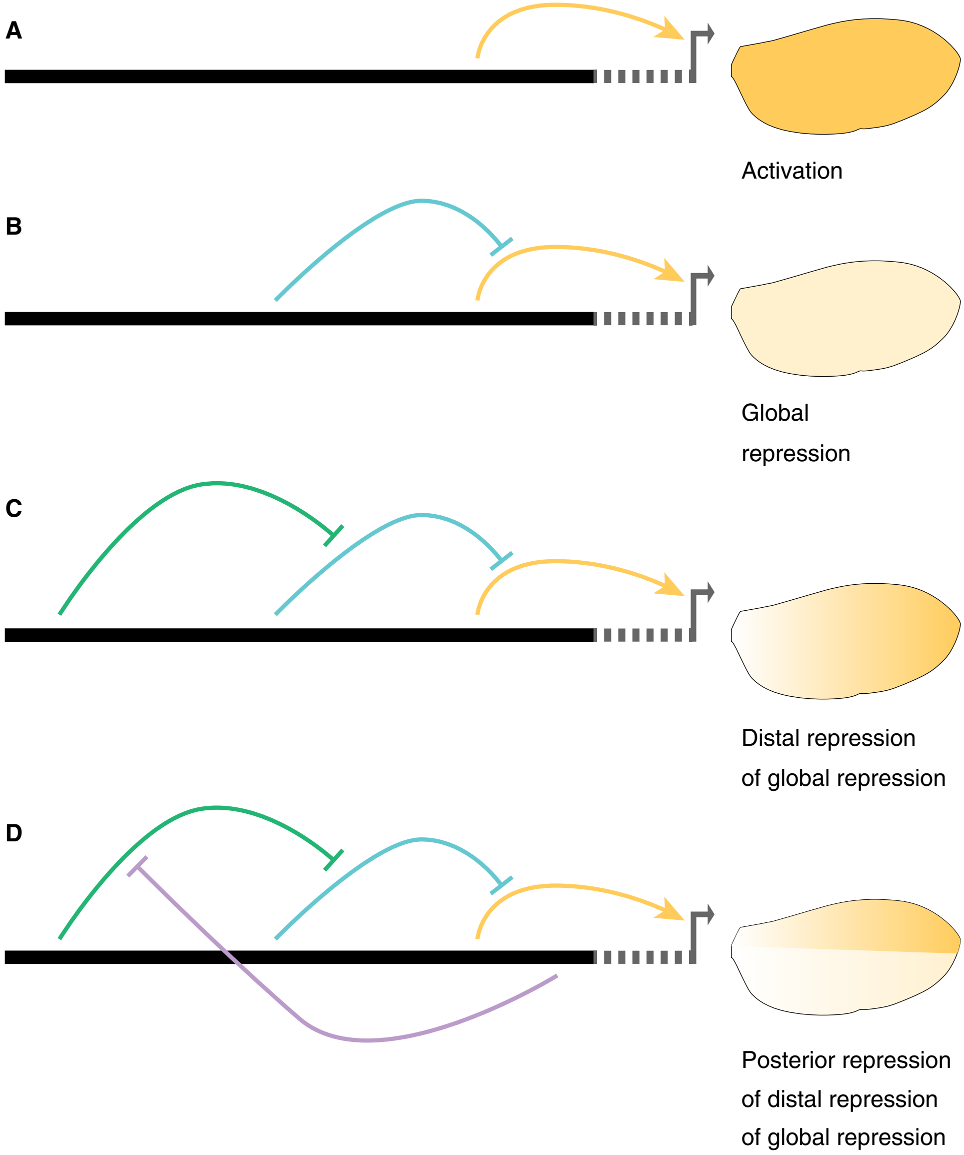


**Fig. 5. A model of the regulatory logic governing the *spot*$^{196}$ enhancer.** (**A** to **D**) The schematics show step by step how regulatory information and interactions integrated along the enhancer sequence produce a spatial pattern of activity. (A) Three independent inputs, respectively, in blocks *A*, *B*, and *C* promote activity (arrows) in the wing veins, the alula, and the wing blade, as illustrated with average phenotypes of constructs *[A--]*, *[-B-]*, and *[--C]*, respectively. Note that activity levels in the wing blade, stemming from block *C*, match the final levels of the *spot*$^{196}$ enhancer activity in the spot region. (B) A first set of repressive inputs suppresses activity in the wing blade (stemming from blocks *A* and *B*) and the veins (stemming from block *B*). The overall combined output of the initial activation and the global repressive inputs is a near-complete loss of activity, except in the alula. (C) A second set of repressive inputs, whose action is localized in the distal wing region, counters the global repression, thereby carving a pattern of distal activity promoted by block *C*. (D) The distal activity is repressed in the posterior wing compartment, likely through the repressive action of Engrailed, resulting in a final pattern of activity in the spot region.

Downloaded from https://www.science.org on January 26, 2022

### A visual tool to compare spatial activities driven by enhancer variants

We introduced a new representation to compare activities between enhancer variants, typically a wild type and a mutant. Proportional effects, or local fold changes, as revealed by *logRatio* produce representations that are independent from the distribution of the reference activity. They also better reflect the distribution of factors in trans and their variations as seen by the enhancer (here, across the wing) than differential comparisons (compare Fig. 2 and fig. S3). Differential comparisons are dominated by regions of high activities and thereby focusing our attention to the regions of high variation of activity. By contrast, *logRatios* reveal strong effects in regions of low activity that would hardly be visible using differential comparisons, highlighting some cryptic components of the regulatory logic. When additional knowledge about TFBSs and TF distribution will become available, they will also inform us on the contribution of the TF in the regulatory logic. In this respect, the introduction of *logRatios* in our analysis has proven useful and could be adapted to any system where image alignment is possible, such as *Drosophila* blastoderm embryos (*38*) or developing mouse limbs (*39*).

### A-tracts did not disrupt the major effect of TF-TF interactions

A-tracts are known to change local conformational properties of DNA. Hence, our A-tract mutations could influence the regulatory logic not only by directly disrupting the information contained in the sequence they replaced but also, indirectly, by introducing more changes than wanted. As an alternative, sequence randomization, however, is more likely to create spurious TFBSs, which is difficult to control for, especially if all the determinants of the enhancer activity are not known. The possible occurrence of undesired and undetected TFBSs would have biased our interpretation of the effect of individual segments and, consequently, of the regulatory logic of the enhancer. The chance that A-tracts introduce new TFBSs in the enhancer sequence is quite low compared to sequence randomization, which is why we favored this mutational approach for the analysis of short, individual segments. However, A-tracts can modify various physical properties of the DNA molecule and, in turn, influence interactions between TFs binding the enhancer. The disruption of a TF-TF interaction due to the introduction of an A-tract between two TFBSs (fig. S2B) would be revealed if mutating a particular segment would have an effect similar to the effect of mutating immediately adjacent flanking segments. We note, however, that we do not have such situation in our dataset. This suggests that the A-tracts we introduced, if anything, only mildly altered TF-TF interactions through changes in the physical properties of *spot*$^{196}$. Instead, we think that the effects of A-tract mutations are mostly due to disrupted TFBSs along the enhancer sequence.

### The regulatory logic underlying *spot*$^{196}$ enhancer activity

The main finding of our study is that the *spot*$^{196}$ enhancer likely integrates six to eight distinct regulatory inputs, with multiple layers of cross-interactions (Fig. 5). We had previously proposed that the spot pattern resulted from the integration of only two spatial regulators: the activator Dll and the repressor En (*19*, *20*). The regulatory density that we reveal here (Figs. 1C and 2) is reminiscent of what has been found for other enhancers (*13*, *23*, *24*). A logical analysis of systematic mutations along the enhancer gives a different status to the factors controlling *spot*$^{196}$. The main levels of *spot*$^{196}$ activity across the wing blade seem to result mostly from two unknown activators: one promoting a relatively uniform expression in the wing blade, and another along the veins (Fig. 5A). This activation is, in turn, globally repressed throughout the wing by an unknown repressor whose action masks that of the global activator (Fig. 5B). Upon these first two regulatory layers, the actual spot pattern of activity is carved by two local repressions. A distal repression counteracts the effect of the global repressor in the distal region of the wing (Fig. 5C), but the spatial range of this repression is limited to the anterior wing compartment by another repressor acting across the posterior wing compartment (Fig. 5D). The former local repression could be mediated by Dll itself, a hypothesis compatible with the nonadditive effects of Dll TFBS mutations, whereas the latter is almost certainly due to En. Thus, the pattern of activity results not so much from local activation but from multiple tiers of repressors.

One would expect this complex set of interactions between TFs that bind along the enhancer sequence to be vulnerable to sequence reorganization. We unexpectedly find that shuffling blocks of the sequence resulted in marked changes in activity levels with little effect on the activity pattern. Similarly, many of the mutations still produced a pattern of activity quite similar to the one of *[+]*. This suggests that the exact organization of the different inputs and the absence of some of these inputs do not affect the TF-enhancer and TF-TF interactions required for a patterned activity, which here translates mainly to the role of Dll in repressing global repressors and the repressing role of En. The frequency of these interactions, or the interactions with the core promoter, may, however, change significantly upon sequence modifications, affecting transcription rate. In other words, the regulatory logic described above is robust to changes for the production of a spatial pattern but less so for the tuning of enhancer activity levels.

The regulatory logic of this enhancer perhaps reflects the evolutionary steps of the emergence of *spot*$^{196}$. The *spot*$^{196}$ element evolved from the co-option of a preexisting *wing blade* enhancer (*20*). The sequences of this ancestral *wing blade* enhancer and the evolutionary-derived *spot*$^{196}$ overlap and share at least one common input (*21*). This perspective is consistent with the idea that a novel pattern emerged by the progressive evolution of multiple tiers of repression carving a spot pattern from a uniform regulatory activity in the wing blade. To further deconstruct the regulatory logic governing the *spot*$^{196}$ enhancer and its evolution, one first task will be to investigate how some of the mutations we introduced affect the activity of a broader fragment containing the entire *spot* activity (and the *wing blade* enhancer), closer to the native context of this enhancer. Another challenging step will be to identify the direct inputs integrated along its sequence. It will also be necessary to characterize their biochemical interactions with DNA and with one another. Ultimately, to fully grasp the enhancer logic will mean to be able to recreate these interactions in a functional synthetic regulatory element.

## MATERIALS AND METHODS

### Fly husbandry

Our *D. melanogaster* stocks were maintained on standard cornmeal medium at 25°C with a 12:12 day-night light cycle.

### Transgenesis

All reporter constructs were injected as in (*19*). We used ϕC31-mediated transgenesis (*40*) and integrated all constructs at the genomic *attP* site *VK00016* (*41*) on chromosome 2. All transgenic lines were genotyped to ascertain that the enhancer sequence was correct.

### Molecular biology

All 196-bp constructs derived from the *D. biarmipes spot*$^{196}$ sequence were synthesized in vitro by a biotech company (Integrated DNA Technologies, Coralville, USA; catalog no. 121416). Table S1 provides a list of all constructs and their sequences. Each construct was cloned by In-Fusion (Takara, Mountain View, USA) in our pRedSA vector [a custom version of the transformation vector pRed H-Stinger (*42*) with a 284-bp *attB* site for ϕC31-mediated transgenesis (*40*) cloned at the Avr II site of pRed H-Stinger]. All constructs in Fig. 1 were cloned by cutting pRedSA with Kpn I and Nhe I and using the following homology arms for In-Fusion cloning: 5′-GAGCCCGGGCGAATT-3′ and 5′-GATCCCTCGAGGAGC-3′. Likewise, constructs in Fig. 3 were cloned by cutting pRedSA with Bam HI and Eco RI and using the following homology arms for In-Fusion cloning: 5′-GAGCCCGGGCGAATT-3′ and 5′-GATCCCTCGAGGAGC-3′.

Downloaded from https://www.science.org on January 26, 2022

### Wing preparation and imaging

All transgenic wings imaged in this study were homozygous for the reporter construct. Males were selected at emergence from pupa, a stage that we call "post-emergence," when their wings are unfolded but still slightly curled. When flies were massively emerging from an amplified stock, we collected every 10 min and froze staged flies at −20°C until we had reached a sufficient number of flies. In any case, staged flies were processed after a maximum of 48 hours at −20°C. We dissected a single wing per male. Upon dissection, wings were immediately mounted onto a microscope slide coated with transparent glue (see below) and fixed for 1 hour at room temperature in 4% paraformaldehyde diluted in phosphate-buffered saline–1% Triton X-100 (PBST). Slides with mounted wings were then rinsed in PBST and kept in a PBST bath at 4°C until the next day. Slides were then removed from PBST, and the wings were covered with Vectashield (Vector Laboratories, Burlingame, USA). The samples were then covered with a coverslip. Preparations were stored for a maximum of 48 hours at 4°C until image acquisition.

The glue-coated slides were prepared immediately before wing mounting by dissolving adhesive tape (Tesa brand, tesafilm, ref. 57912) in heptane (two rolls in 100 ml of heptane) and spreading a thin layer of this solution onto a clean microscope slide. Once the heptane had evaporated (under a fume hood), the slide was ready for wing mounting. All wing images were acquired as 16-bit images on a Ti2 Eclipse Nikon microscope equipped with a Nikon 10× plan apochromatic lens (numerical aperture, 0.45; Nikon Corporation, Tokyo, Japan) and a pco.edge 5.5 Mpx sCMOS camera (PCO, Kelheim, Germany) under illumination from a Lumencor SOLA SE II light source (Lumencor, Beaverton, OR, USA). Each wing was imaged by tiling and stitching of several z-stacks (z-step, 4 μm) with 50% overlap between tiles. Each image comprises a fluorescent channel (ET-DSRed filter cube, Chroma Technology Corporation, Bellows Falls, VT, USA) and a bright-field channel (acquired using flat field correction from the Nikon NIS-Elements software throughout), the latter being used for later image alignment. To ensure that fluorescence measurements are comparable between imaging sessions, we used identical settings for the fluorescence light source (100% output), light path, and camera (20-ms exposure time, no active shutter) to achieve comparable fluorescence excitation.

### Z-projection

Stitched three-dimensional (3D) stacks were projected to 2D images for subsequent analysis. The local sharpness average of the bright-field channel was computed for each pixel position in each z-slice, and an index of the slice with the maximum sharpness was recorded and smoothed with a Gaussian kernel (sigma = 5 px). Both bright-field and fluorescent 2D images were reconstituted by taking the value of the sharpest slice for each pixel.

### Image alignment

Wing images were aligned using the veins as a reference. Fourteen landmarks placed on vein intersections and end points and 26 sliding landmarks equally spaced along the veins were placed on bright-field images using a semi-automatized pipeline. Landmark coordinates on the image were then used to warp with a deformable model (thin plate spline) bright-field and fluorescent images to match the landmarks of an arbitrarily chosen reference wing by the thin plate spline interpolation (*43*). All wings were then in the same coordinate system, defined by their venation.

### Fluorescent signal description

A transgenic line with an empty reporter vector (ø) was used as a proxy to measure noise and tissue autofluorescence. The median raw fluorescent image was computed across all ø images and used to remove autofluorescence, subtracted from all raw images before the following steps. All variation of fluorescence below the median ø value was discarded. The DsRed reporter signal was mostly localized in the cell nuclei. We measured the local average fluorescent levels by smoothing fluorescence intensity, through a Gaussian filter (sigma = 8 px) on the raw 2D fluorescent signal. The sigma corresponded roughly to two times the distance between the adjacent nuclei. To lower the memory requirement, images were then subsampled by a factor of 2. We used the 89,735 pixels inside the wings as descriptors of the phenotype for all subsequent analyses.

### Average phenotypes, differences, *logRatio* colormaps, and normalization

Average reporter expression phenotypes were computed as the average smoothed fluorescence intensity at every pixel among all individuals in a given group (tens of individuals from the same transgenic line). The difference between groups was computed as the pixel-wise difference between the average of the groups (fig. S3). *logRatio* between two constructs represents the fold change of a phenotype relative to another and is calculated as the pixel-wise logarithm of the ratio between the two phenotypes. Averages, difference, and *logRatio* images were represented using colors equally spaced in CIELAB perceptual color space (*44*). With these colormaps, the perceived difference in colors corresponds to the actual difference in signal. Colormaps were spread between the minimal and maximal signals across all averages for average phenotypes. Difference and *logRatio* spread between minus and plus represent the absolute value of all difference for the phenotype differences, with gray colors indicating that the two compared phenotypes are equal.

### Mutation effect direction and intensity

We proposed to represent the necessity of a stretch of the sequence along the enhancer with the activity levels of mutants of this stretch relative to the wild-type (*[+]*) activity. To summarize the overall effect of mutants (overexpression or underexpression), we measured the average level of activity across each wing relative to that of the reference. The reference level was defined as the average level of activity of all *[+]* individuals. The value at each position corresponds to the average of all individuals that present a sequence that have an effect on this position. The effect of a mutation is not strictly limited to the mutated bases, because they can also modify properties of DNA of flanking positions (*45*). To take this effect into account and produce a more realistic and conservative estimation of necessity measure at each position, we weighted the phenotypic contribution of each mutant line to the measure by the strength of the changes they introduce to the DNA shape descriptors at this position. At each position, the phenotype of constructs not affecting the DNA shape descriptors compared to *[+]* was not considered. When two mutants modify the DNA shape descriptors at one position, typically near the junction of two adjacent mutations, the effect at this position was computed as the weighted average of the effect of the two mutants, where the weight is the extent of the DNA shape modification relative to the *[+]* sequence. DNA shape descriptors were computed by the R package DNAshapeR (*46*).

Downloaded from https://www.science.org on January 26, 2022

Notably, with an average of 11.5 bp, our A-tract mutations are somewhat larger than an average eukaryotic TFBS [~10 bp (*32*)], and each mutation is likely to affect up to two TFBSs. This size represents the limit of regulatory content that we can discriminate in this study.

### PCA and difference significance

The intensity measure is an average of the overall and variable expression across the wing. Hence, mutations causing a different effect on the phenotype can have the same intensity value. To test whether the mutant significantly differs from *[+]*, we used comprehensive and unbiased phenotype descriptors provided by PCA, which removes the correlation between pixel intensities and describes the variation in reporter gene expression. PCA was calculated on the matrix regrouping intensities of all pixels for every individual, of dimensions (n_individuals × n_pixels on the wing). The significance of the difference between two constructs considers the multivariate variation of the phenotypes and is tested using multivariate analysis of variance (MANOVA) on all five first components explaining more than 0.5% of the total variance (data file S3).

### Overall expression intensity and significance

The overall expression level was measured for each individual as the average intensity across the wing. This was used to test the significance of overall increase and decrease in expression levels relative to the wild-type levels.

### DNA rigidity scores

A-tracts are runs of consecutive A/T base pair without a TpA step. Stacking interactions and inter–base pair hydrogen bonds in ApA (TpT) or ApT steps of A-tracts lead to conformational rigidity (*28*). The length of an A-tract directly correlates with increased rigidity (*47*). To parametrize DNA rigidity at nucleotide resolution, we used A-tract length as a metric. For each position in a given DNA sequence, we find the longest consecutive run of the form $A_nT_m$ that contains this position (with the requirement of $n \geq 0$, $m \geq 0$, and $n + m \geq 2$), and score DNA rigidity at that position using the length of this subsequence. For example, the sequence AATCGCAT will map to the scores 3,3,3,0,0,0,2,2 because AAT and AT are A-tracts of lengths 3 and 2 bp, respectively.

## SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/6/49/eabe2955/DC1

View/request a protocol for this paper from *Bio-protocol*.

## REFERENCES AND NOTES

1. D. Shlyueva, G. Stampfel, A. Stark, Transcriptional enhancers: From properties to genome-wide predictions. *Nat. Rev. Genet.* **15**, 272–286 (2014).
2. M. Levine, Transcriptional enhancers in animal development and evolution. *Curr. Biol.* **20**, R754–R763 (2010).
3. I. S. Peter, E. H. Davidson, *Genomic Control Process: Development and Evolution* (Academic Press, ed. 1, 2015), p. 460.
4. S. B. Carroll, From pattern to gene, from gene to pattern. *Int. J. Dev. Biol.* **42**, 305–309 (1998).
5. F. Spitz, E. E. M. Furlong, Transcription factors: From enhancer binding to developmental control. *Nat. Rev. Genet.* **13**, 613–626 (2012).
6. J. Crocker, A. Tsai, D. L. Stern, A fully synthetic transcriptional platform for a multicellular eukaryote. *Cell Rep.* **18**, 287–296 (2017).
7. B. J. Vincent, J. Estrada, A. H. DePace, The appeasement of Doug: A synthetic approach to enhancer biology. *Integr. Biol. (Camb.)* **8**, 475–484 (2016).
8. E. H. Davidson, *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution* (Elsevier, 2010).
9. D. M. King, C. K. Y. Hong, J. L. Shepherdson, D. M. Granas, B. B. Maricque, B. A. Cohen, Synthetic and genomic regulatory elements reveal aspects of *cis*-regulatory grammar in mouse embryonic stem cells. *eLife* **9**, e41279 (2020).
10. E. K. Farley, K. M. Olson, W. Zhang, A. J. Brandt, D. S. Rokhsar, M. S. Levine, Suboptimization of developmental enhancers. *Science* **350**, 325–328 (2015).
11. M. Kircher, C. Xiong, B. Martin, M. Schubach, F. Inoue, R. J. A. Bell, J. F. Costello, J. Shendure, N. Ahituv, Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. *Nat. Commun.* **10**, 3583 (2019).
12. A.-R. Kim, C. Martinez, J. Ionides, A. F. Ramos, M. Z. Ludwig, N. Ogawa, D. H. Sharp, J. Reinitz, Rearrangements of 2.5 kilobases of noncoding DNA from the *Drosophila even-skipped* locus define predictive rules of genomic *cis*-regulatory logic. *PLOS Genet.* **9**, e1003243 (2013).
13. T. Fuqua, J. Jordan, M. E. van Breugel, A. Halavatyi, C. Tischer, P. Polidoro, N. Abe, A. Tsai, R. S. Mann, D. L. Stern, J. Crocker, Dense and pleiotropic regulatory information in a developmental enhancer. *Nature* **587**, 235–239 (2020).
14. J. Dufourt, A. Trullo, J. Hunter, C. Fernandez, J. Lazaro, M. Dejean, L. Morales, S. Nait-Amer, K. N. Schulz, M. M. Harrison, C. Favard, O. Radulescu, M. Lagha, Temporal control of gene expression by the pioneer factor Zelda through transient interactions in hubs. *Nat. Commun.* **9**, 5194 (2018).
15. J. Crocker, D. L. Stern, Functional regulatory evolution outside of the minimal *even-skipped* stripe 2 enhancer. *Development* **144**, 3095–3101 (2017).
16. J. Crocker, N. Abe, L. Rinaldi, A. P. McGregor, N. Frankel, S. Wang, A. Alsawadi, P. Valenti, S. Plaza, F. Payre, R. S. Mann, D. L. Stern, Low affinity binding site clusters confer hox specificity and regulatory robustness. *Cell* **160**, 191–203 (2015).
17. J. Park, J. Estrada, G. Johnson, B. J. Vincent, C. Ricci-Tam, M. D. Bragdon, Y. Shulgina, A. Cha, Z. Wunderlich, J. Gunawardena, A. H. DePace, Dissecting the sharp response of a canonical developmental enhancer reveals multiple sources of cooperativity. *eLife* **8**, e41266 (2019).
18. L. Bentovim, T. T. Harden, A. H. DePace, Transcriptional precision and accuracy in development: From measurements to models and mechanisms. *Development* **144**, 3855–3866 (2017).
19. L. Arnoult, K. F. Y. Su, D. Manoel, C. Minervino, J. Magriña, N. Gompel, B. Prud'homme, Emergence and diversification of fly pigmentation through evolution of a gene regulatory module. *Science* **339**, 1423–1426 (2013).
20. N. Gompel, B. Prud'homme, P. J. Wittkopp, V. A. Kassner, S. B. Carroll, Chance caught on the wing: *Cis*-regulatory evolution and the origin of pigment patterns in *Drosophila*. *Nature* **433**, 481–487 (2005).
21. Y. Xin, Y. Le Poul, L. Ling, M. Museridze, B. Mühling, R. Jaenichen, E. Osipova, N. Gompel, Enhancer evolutionary co-option through shared chromatin accessibility input. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 20636–20644 (2020).
22. S. Neidle, *Principles of Nucleic Acid Structure* (Academic Press, 2010), p. 302.
23. C. I. Swanson, N. C. Evans, S. Barolo, Structural rules and complex regulatory circuitry constrain expression of a Notch- and EGFR-regulated eye enhancer. *Dev. Cell* **18**, 359–370 (2010).
24. E. Z. Kvon, Y. Zhu, G. Kelman, C. S. Novak, I. Plajzer-Frick, M. Kato, T. H. Garvin, Q. Pham, A. N. Harrington, R. D. Hunter, J. Godoy, E. M. Meky, J. A. Akiyama, V. Afzal, S. Tran, F. Escande, B. Gilbert-Dussardier, N. Jean-Marçais, S. Hudaiberdiev, I. Ovcharenko, M. B. Dobbs, C. A. Gurnett, S. Manouvrier-Hanu, F. Petit, A. Visel, D. E. Dickel, L. A. Pennacchio, Comprehensive *in vivo* interrogation reveals phenotypic impact of human enhancer variants. *Cell* **180**, 1262–1271.e15 (2020).
25. M. Slattery, T. Zhou, L. Yang, A. C. Dantas Machado, R. Gordan, R. Rohs, Absence of a simple code: How transcription factors read the genome. *Trends Biochem. Sci.* **39**, 381–399 (2014).
26. N. Abe, I. Dror, L. Yang, M. Slattery, T. Zhou, H. J. Bussemaker, R. Rohs, R. S. Mann, Deconvolving the recognition of DNA shape from sequence. *Cell* **161**, 307–318 (2015).
27. I. Barozzi, M. Simonatto, S. Bonifacio, L. Yang, R. Rohs, S. Ghisletti, G. Natoli, Coregulation of transcription factor binding and nucleosome occupancy through DNA features of mammalian enhancers. *Mol. Cell* **54**, 844–857 (2014).
28. H. C. Nelson, J. T. Finch, B. F. Luisi, A. Klug, The structure of an oligo(dA).oligo(dT) tract and its biological implications. *Nature* **330**, 221–226 (1987).
29. B. Suter, G. Schnappauf, F. Thoma, Poly(dA.dT) sequences exist as rigid DNA structures in nucleosome-free yeast promoters *in vivo*. *Nucleic Acids Res.* **28**, 4083–4089 (2000).
30. M. D. Robinson, D. J. McCarthy, G. K. Smyth, edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2010).
31. L. J. Zhu, R. G. Christensen, M. Kazemian, C. J. Hull, M. S. Enuameh, M. D. Basciotta, J. A. Brasefield, C. Zhu, Y. Asriyan, D. S. Lapointe, S. Sinha, S. A. Wolfe, M. H. Brodsky, FlyFactorSurvey: A database of *Drosophila* transcription factor binding specificities determined using the bacterial one-hybrid system. *Nucleic Acids Res.* **39**, D111–D117 (2011).

Downloaded from https://www.science.org on January 26, 2022

32. A. J. Stewart, S. Hannenhalli, J. B. Plotkin, Why transcription factor binding sites are ten nucleotides long. *Genetics* **192**, 973–985 (2012).
33. D. N. Arnosti, S. Barolo, M. Levine, S. Small, The eve stripe 2 enhancer employs multiple modes of transcriptional synergy. *Development* **122**, 205–214 (1996).
34. E. K. Farley, K. M. Olson, W. Zhang, D. S. Rokhsar, M. S. Levine, Syntax compensates for poor binding sites to encode tissue specificity of developmental enhancers. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 6508–6513 (2016).
35. V. Bertrand, C. Hudson, D. Caillol, C. Popovici, P. Lemaire, Neural tissue in ascidian embryos is induced by FGF9/16/20, acting via a combination of maternal GATA and Ets transcription factors. *Cell* **115**, 615–627 (2003).
36. D. Thanos, T. Maniatis, Virus induction of human IFN beta gene expression requires the assembly of an enhanceosome. *Cell* **83**, 1091–1100 (1995).
37. C. I. Swanson, D. B. Schwimmer, S. Barolo, Rapid evolutionary rewiring of a structurally constrained eye enhancer. *Curr. Biol.* **21**, 1186–1196 (2011).
38. C. C. Fowlkes, C. L. L. Hendriks, S. V. E. Keränen, G. H. Weber, O. Rübel, M.-Y. Huang, S. Chatoor, A. H. De Pace, L. Simirenko, C. Henriquez, A. Beaton, R. Weiszmann, S. Celniker, B. Hamann, D. W. Knowles, M. D. Biggin, M. B. Eisen, J. Malik, A quantitative spatiotemporal atlas of gene expression in the *Drosophila* blastoderm. *Cell* **133**, 364–374 (2008).
39. N. Martínez-Abadías, R. Mateu, M. Niksic, L. Russo, J. Sharpe, Geometric morphometrics on gene expression patterns within phenotypes: A case example on limb development. *Syst. Biol.* **65**, 194–211 (2016).
40. A. C. Groth, M. Fish, R. Nusse, M. P. Calos, Construction of transgenic Drosophila by using the site-specific integrase from phage φC31. *Genetics* **166**, 1775–1782 (2004).
41. K. J. T. Venken, Y. He, R. A. Hoskins, H. J. Bellen, P[acman]: A BAC transgenic platform for targeted insertion of large DNA fragments in *D. melanogaster*. *Science* **314**, 1747–1751 (2006).
42. S. Barolo, B. Castro, J. W. Posakony, New *Drosophila* transgenic reporters: Insulated P-element vectors expressing fast-maturing RFP. *Biotechniques* **36**, 436–442 (2004).
43. M. F. Hutchinson, Interpolating mean rainfall using thin plate smoothing splines. *Int. J. Geogr. Inf. Syst.* **9**, 385–403 (1995).
44. E. C. Carter, J. D. Schanda, R. Hirschler, S. Jost, M. R. Luo, M. Melgosa, Y. Ohno, M. R. Pointer, D. C. Rich, F. Viénot, L. Whitehead, J. H. Wold, *Colorimetry* (CIE Central Bureau, ed. 4, 2018).
45. T. Zhou, L. Yang, Y. Lu, I. Dror, A. C. Dantas Machado, T. Ghane, R. Di Felice, R. Rohs, DNAshape: A method for the high-throughput prediction of DNA structural features on a genomic scale. *Nucleic Acids Res.* **41**, W56–W62 (2013).
46. T.-P. Chiu, F. Comoglio, T. Zhou, L. Yang, R. Paro, R. Rohs, DNAshapeR: An R/Bioconductor package for DNA shape prediction and feature encoding. *Bioinformatics* **32**, 1211–1213 (2016).
47. R. Rohs, S. M. West, A. Sosinsky, P. Liu, R. S. Mann, B. Honig, The role of DNA shape in protein–DNA recognition. *Nature* **461**, 1248–1253 (2009).

## Acknowledgments

**Funding:** This work was supported by funds from the Ludwig Maximilian University of Munich, the Human Frontiers Science Program (program grant RGP0021/2018 to N.G., S.P., and R.R.), the Deutsche Forschungsgemeinschaft (grants INST 86/1783-1 LAGG and GO 2495/5-1 to N.G. and SPP 2202 to H.L. and H.H.), the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013/ERC Grant Agreement no. 615789 to B.P.), and the NIH (grant R35GM130376 to R.R.). Y.X. was supported by a fellowship from the China Scholarship Council (fellowship 201506990003). L.L. was supported by a DFG fellowship through the Graduate School of Quantitative Biosciences Munich (QBM). M.M. and D.D. are recipients of fellowships from the German Academic Exchange Service (DAAD). E.M. was supported by the Amgen Scholar program of the LMU. **Author contributions:** Y.L.P.: conceptualization, methodology, software, validation, formal analysis, data curation, writing—original draft, and visualization; Y.X.: validation, investigation, formal analysis, and data curation; L.L.: investigation and formal analysis; B.M.: investigation; R.J.: investigation; D.H.: software and data curation; D.B.: software and data curation; H.H.: methodology and supervision; H.L.: supervision; Y.W.: methodology, software, and formal analysis; E.O.: investigation; M.M.: investigation and formal analysis; D.D.: investigation and formal analysis; E.M.: investigation and formal analysis; R.R.: methodology, supervision, and funding acquisition; S.P.: software, supervision, and funding acquisition; B.P.: conceptualization, writing—original draft, and funding acquisition; N.G.: conceptualization, validation, writing—original draft, visualization, supervision, project administration, and funding acquisition. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Additional data related to this paper may be requested from the authors.

Submitted 12 August 2020
Accepted 20 October 2020
Published 2 December 2020
10.1126/sciadv.abe2955

**Citation:** Y. Le Poul, Y. Xin, L. Ling, B. Mühling, R. Jaenichen, D. Hörl, D. Bunk, H. Harz, H. Leonhardt, Y. Wang, E. Osipova, M. Museridze, D. Dharmadhikari, E. Murphy, R. Rohs, S. Preibisch, B. Prud'homme, N. Gompel, Regulatory encoding of quantitative variation in spatial activity of a *Drosophila* enhancer. *Sci. Adv.* **6**, eabe2955 (2020).

Downloaded from https://www.science.org on January 26, 2022

Science Advances

# Regulatory encoding of quantitative variation in spatial activity of a *Drosophila* enhancer

Yann Le PoulYaqun XinLiucong LingBettina MühlingRita JaenichenDavid HörlDavid BunkHartmann HarzHeinrich LeonhardtYingfei WangElena OsipovaMariam MuseridzeDeepak DharmadhikariEamonn MurphyRemo RohsStephan PreibischBenjamin Prud'hommeNicolas Gompel

*Sci. Adv.*, 6 (49), eabe2955. • DOI: 10.1126/sciadv.abe2955

**View the article online**
https://www.science.org/doi/10.1126/sciadv.abe2955
**Permissions**
https://www.science.org/help/reprints-and-permissions

Downloaded from https://www.science.org on January 26, 2022

Use of think article is subject to the Terms of service

---

*Science Advances* (ISSN 2375-2548) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Advances* is a registered trademark of AAAS.

Copyright © 2020 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution NonCommercial License 4.0 (CC BY-NC).

### 2.1.2 Supplementary information

Science Advances
AAAS

advances.sciencemag.org/cgi/content/full/6/49/eabe2955/DC1

# Supplementary Materials for

## Regulatory encoding of quantitative variation in spatial activity of a *Drosophila* enhancer

Yann Le Poul, Yaqun Xin, Liucong Ling, Bettina Mühling, Rita Jaenichen, David Hörl, David Bunk, Hartmann Harz, Heinrich Leonhardt, Yingfei Wang, Elena Osipova, Mariam Museridze, Deepak Dharmadhikari, Eamonn Murphy, Remo Rohs, Stephan Preibisch, Benjamin Prud'homme*, Nicolas Gompel*

*Corresponding author. Email: benjamin.prudhomme@univ-amu.fr (B.P.); gompel@bio.lmu.de (N.G.)

Published 2 December 2020, *Sci. Adv.* **6**, eabe2955 (2020)
DOI: 10.1126/sciadv.abe2955

**The PDF file includes:**

Figs. S1 to S4
Tables S1 to S3
Legends for data files S1 to S4
Additional notes on *logRatios*

**Other Supplementary Material for this manuscript includes the following:**

(available at advances.sciencemag.org/cgi/content/full/6/49/eabe2955/DC1)

Data files S1 to S4

## Supplementary Materials

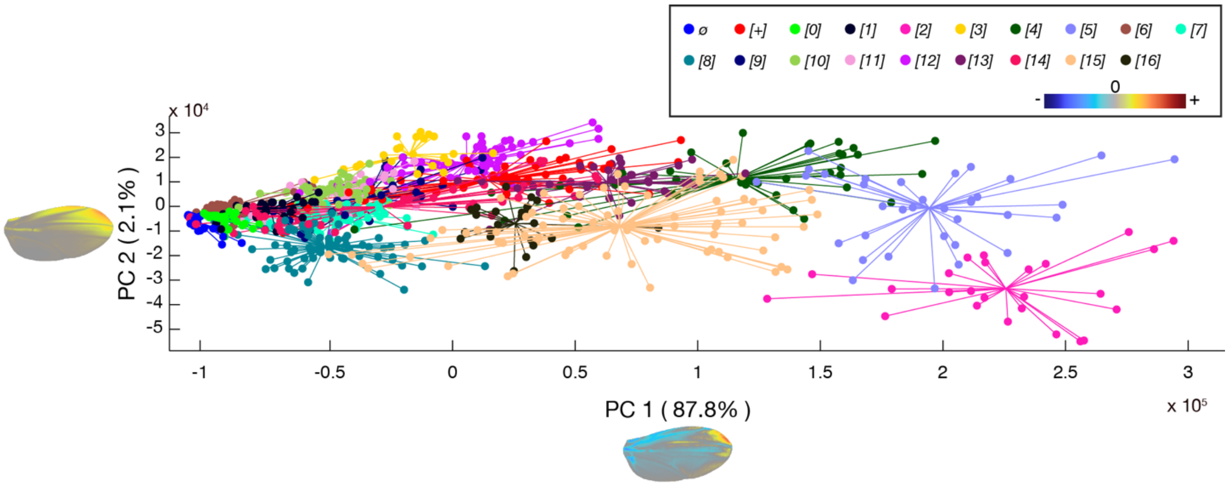


**Figure S1**. **First two axes of variation in a principal component analysis of all individual wings used to generate the average reporter expression of Figure 1**. Each wing is depicted by a colored dot, and each construct by a color. PC1 captures 87.8% of the variation and corresponds to overall changes in the activity of the *spot*$^{196}$ CRE. PC2 captures 2.1% of the variation and appears to represent spatial difference in CRE activity between lines. The direction of variation along each principal component is represented on a wing with a colormap next to each axis.

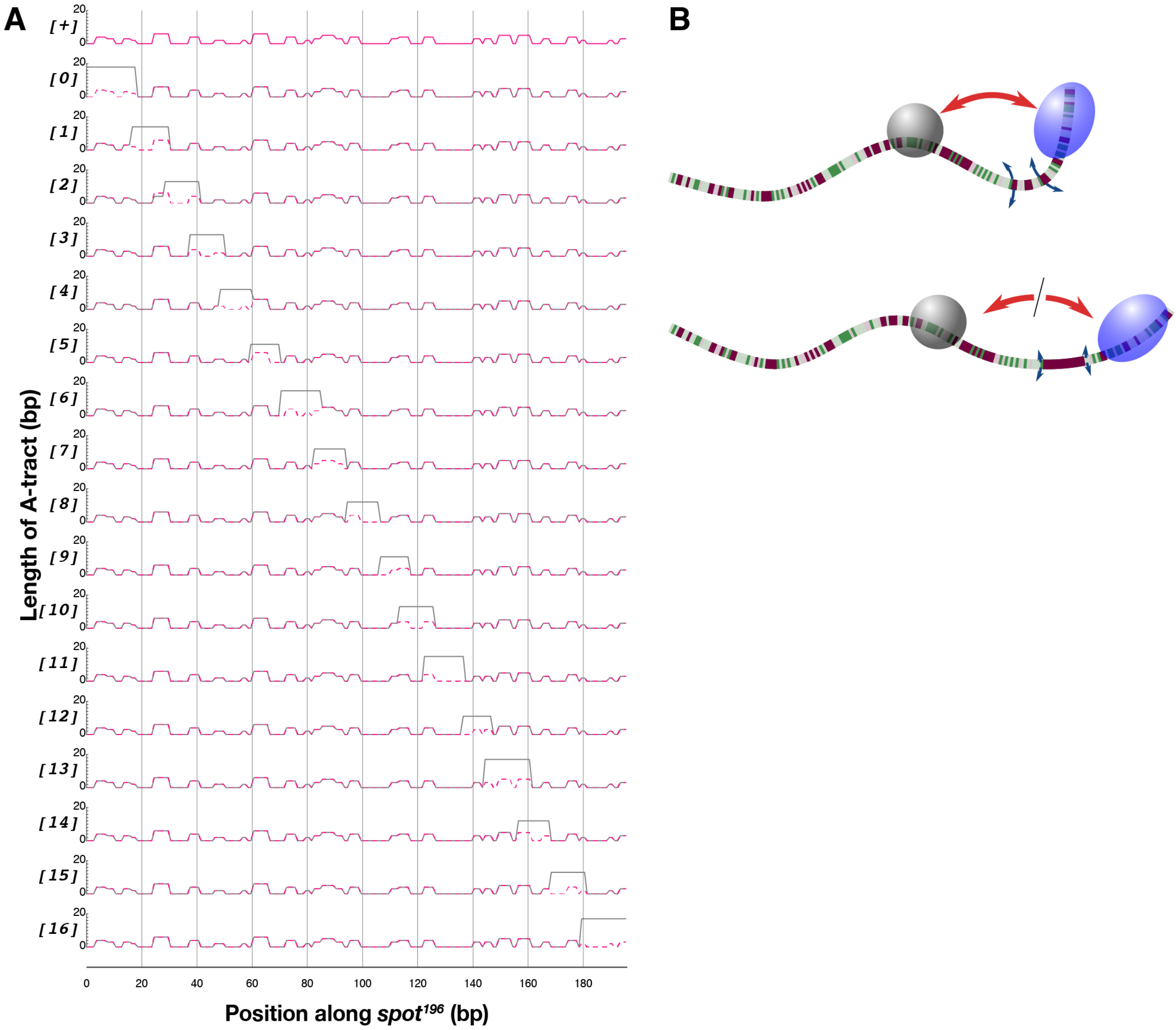


**Figure S2. Local rigidity along the wild-type and mutant *spot*$^{196}$**. (**A**) Each graph is a plot of the length of the longest consecutive $A_nT_n$ sequence that a base pair participates in, a proxy for sequence rigidity at this position. The first graph on top is the wild type (*[+]*) alone. The remaining graphs show plots for each mutant (*[0]*, …, *[16]*) with a solid black line, compared to the wild type represented with a dotted magenta line. (**B**) Schematics illustrating the hypothetical consequence of local DNA rigidity (caused by an A-tract) on TF interactions. A flexible linker between two TFBSs would favor interactions between 2 bound TFs, while a stiffer linker of the same length would limit, or prevent these interactions.

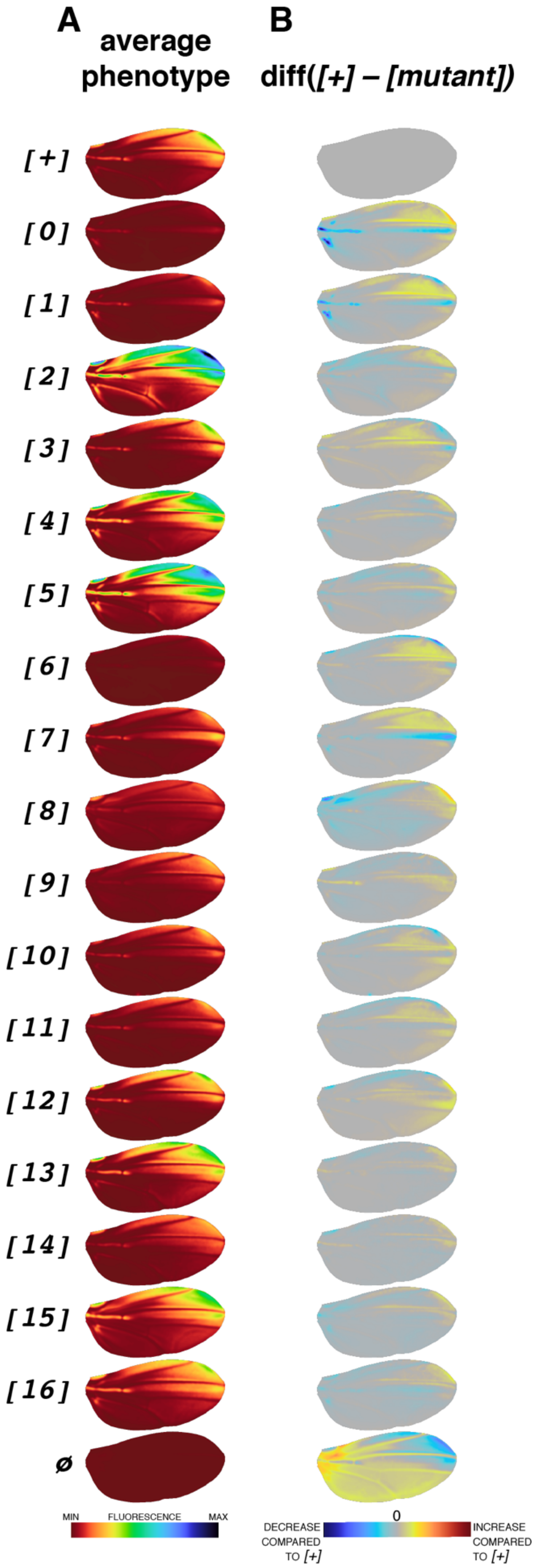

A
average phenotype
B
diff([+] – [mutant])
[+]
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]
[10]
[11]
[12]
[13]
[14]
[15]
[16]
ø
MIN
FLUORESCENCE
MAX
0
DECREASE COMPARED TO [+]
INCREASE COMPARED TO [+]

**Figure S3**. **Pattern changes between wild-type and mutant *spot*$^{196}$ constructs**. (**A**) Average phenotypes reproduced from Figure 1B. (**B**) difference images (*[+] – [mutant]*) for intensity values of each pixel of registered wing images) highlight changes in the distribution of the enhancer activity across the wing. Note that this operation introduces a visual bias towards changes in region of high expression, contrasting with *logRatio* images of Figure 2.

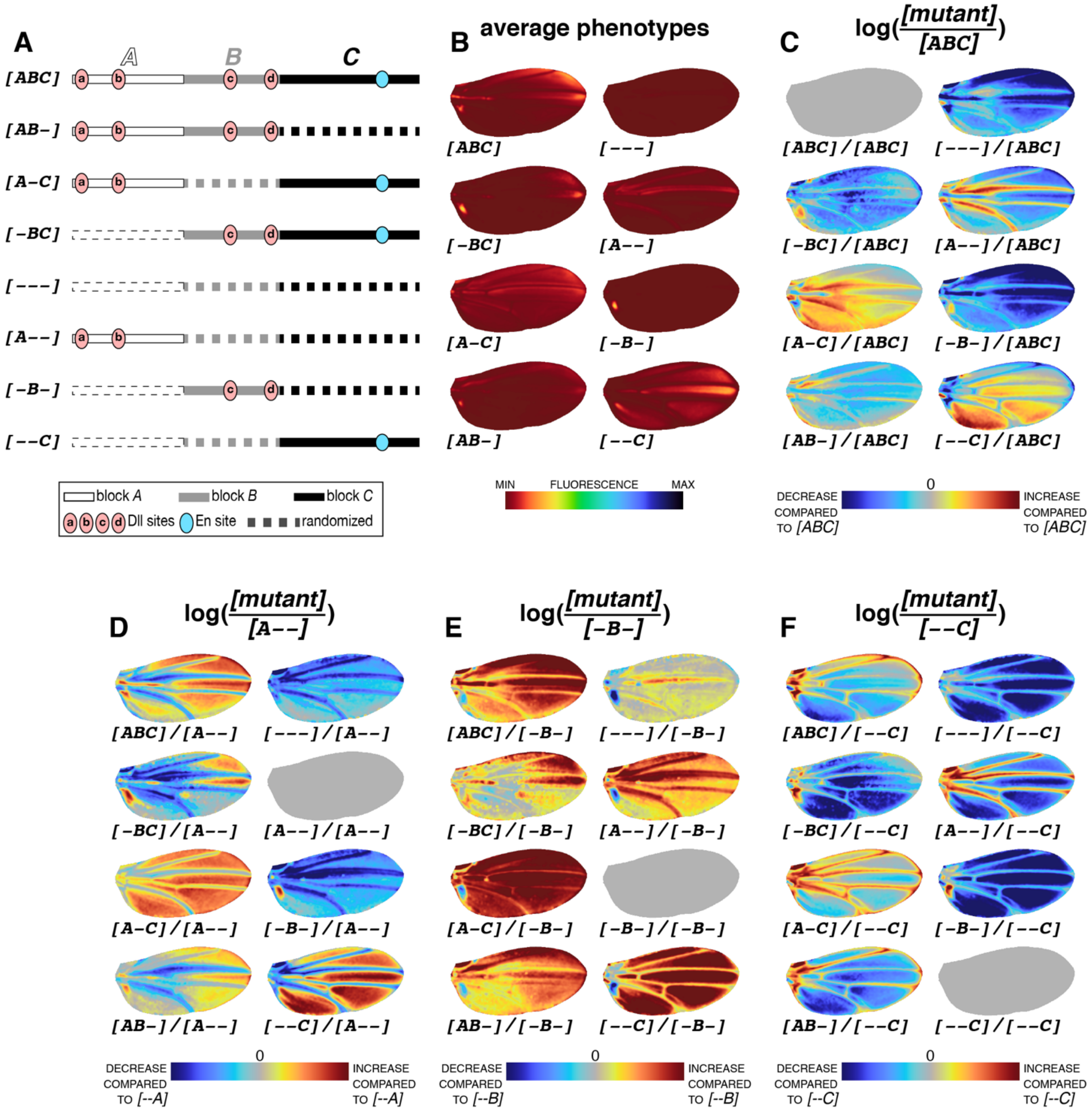


**Figure S4**. ***logRatio* of all block constructs.** (**A**) Schematics of block constructs repeated from Figure 3A for legibility. (**B**) Average phenotypes of constructs shown in (**A**), repeated from Figure 3B for legibility. Colormap of average phenotypes normalized for all constructs of the block series, including block permutations of Figure 4B. (**C**) Average phenotypes in (**B**) compared to the average phenotype of the wild type *[ABC]* (*logRatio*). (**D**) Average phenotypes in (**B**) compared to the average phenotype of *[A--]* (*logRatio*). (**E**) Average phenotypes in (**B**) compared to the average phenotype of *[-B-]* (*logRatio*). (**F**) Average phenotypes in (**B**) compared to the average phenotype of *[--C]* (*logRatio*). Colormaps in (**C**)-(**F**) indicate an increase or a decrease of activity compared to the reference (denominator).

**Table S1. Sequences of $spot^{196}$ enhancer variants.**

- wild type *[+]* or *[ABC]*

>$spot^{196\,[+]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGATGTGCGCCCATGCAAT

- single mutants *[0]* to *[16]*

>$spot^{196\,[0]}$
AAAAAAAAAAAAAAAAAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAG
ATCTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTA
AAACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCG
CCTAATTGATGTGCGCCCATGCAAT

>$spot^{196\,[1]}$
TCTAATTATTCCGTTTAAAAAAAAAAAATTCTGAGCTAAAACTCGCTTATGGAGAGA
TCTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAA
AACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>$spot^{196\,[2]}$
TCTAATTATTCCGTTTAAGGACGCAATTAAAAAAAAAAAAACTCGCTTATGGAGAGA
TCTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAA
AACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>$spot^{196\,[3]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAAAAAAAAAATGGAGAGA
TCTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAA
AACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>$spot^{196\,[4]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTAAAAAAAAAA
ATAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGATGTGCGCCCATGCAAT

>$spot^{196\,[5]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CAAAAAAAAAAAGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAA
AACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>$spot^{196\,[6]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCAAAAAAAAAAAAAATAAATTAATCGAATTCCCCGCTGGCTATTAA
AACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC

TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [7]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAAAAAAAAAAAAGAATTCCCCGCTGGCTATTAA
AACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [8]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCAAAAAAAAAAAAGGCTATTAA
AACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [9]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTAAAAAAAAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [10]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
AAAAAAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [11]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAAAAAAAAAAATCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [12]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGAAAAAAAAAATGTAAATTGCAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [13]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAAAAAAAAAAAAAAATTGCTCAATCCGCC
TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [14]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAAAAAAAAAAACCGCC
TAATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [15]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATAAAAA
AAAAAAATGTGCGCCCATGCAAT

>*spot*$^{196\ [16]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGAAAAAAAAAAAAAAAAA

- Permutations of blocks

> *spot*$^{196\ [ACB]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCC
GCCTAATTGATGTGCGCCCATGCAATTTTCCCCGCTTTTGGCTTGAATAAATTAATCG
AATTCCCCGCTGGCTATTAAAA

>*spot*$^{196\ [BAC]}$
TTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAAATCTA
ATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGATCTAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [BCA]}$
TTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAAACACA
CAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCTAATTG
ATGTGCGCCCATGCAATTCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAA
CTCGCTTATGGAGAGATCTAAA

>*spot*$^{196\ [CBA]}$
CACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCTA
ATTGATGTGCGCCCATGCAATTTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTC
CCCGCTGGCTATTAAAATCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAA
CTCGCTTATGGAGAGATCTAAA

- Randomized blocks

>*spot*$^{196\ [A--]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATCCGAATTTTTTCTTGTCCGACTAGAAACGACTAATTTAGCCGTACCACATGT
TGTCGACTCAGAAACATTATTCCCATTTACGCGTAAGCAAAAAATGCGTCCTTATCGA
ACTTACACTCGCCTGCGTTGGT

>*spot*$^{196\ [-B-]}$
ATAATATTGCATCTCATTGTGGTGCTAGATAATCATCTAGGCTAAATCCAAAACTGTT
GCATGTTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
AGTCGACTCAGAAACATTATTCCCATTTACGCGTAAGCAAAAAATGCGTCCTTATCG
AACTTACACTCGCCTGCGTTGGT

>*spot*$^{196\ [--C]}$
ATAATATTGCATCTCATTGTGGTGCTAGATAATCATCTAGGCTAAATCCAAAACTGTT
GCATGTCCGAATTTTTTCTTGTCCGACTAGAAACGACTAATTTAGCCGTACCACATGT
TCACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [AB-]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
AGTCGACTCAGAAACATTATTCCCATTTACGCGTAAGCAAAAAATGCGTCCTTATCG
AACTTACACTCGCCTGCGTTGGT

>*spot*$^{196\ [A-C]}$
TCTAATTATTCCGTTTAAGGACGCAATTTTCTGAGCTAAAACTCGCTTATGGAGAGAT
CTAAATCCGAATTTTTTCTTGTCCGACTAGAAACGACTAATTTAGCCGTACCACATGT
TCACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [-BC]}$
ATAATATTGCATCTCATTGTGGTGCTAGATAATCATCTAGGCTAAATCCAAAACTGTT
GCATGTTTCCCCGCTTTTGGCTTGAATAAATTAATCGAATTCCCCGCTGGCTATTAAA
ACACACAAAAGGCGCTCTCGTCTGTTTCAATGTAAATTGCAAATTGCTCAATCCGCCT
AATTGATGTGCGCCCATGCAAT

>*spot*$^{196\ [---]}$
ATAATATTGCATCTCATTGTGGTGCTAGATAATCATCTAGGCTAAATCCAAAACTGTT
GCATGTCCGAATTTTTTCTTGTCCGACTAGAAACGACTAATTTAGCCGTACCACATGT
TGTCGACTCAGAAACATTATTCCCATTTACGCGTAAGCAAAAAATGCGTCCTTATCGA
ACTTACACTCGCCTGCGTTGGT

| genotype | number of individuals |
|---|---|
| *ø* | 38 |
| *[+]* | 49 |
| *[0]* | 27 |
| *[1]* | 31 |
| *[2]* | 25 |
| *[3]* | 22 |
| *[4]* | 38 |
| *[5]* | 35 |
| *[6]* | 51 |
| *[7]* | 60 |
| *[8]* | 67 |
| *[9]* | 27 |
| *[10]* | 46 |
| *[11]* | 33 |
| *[12]* | 61 |
| *[13]* | 39 |
| *[14]* | 44 |
| *[15]* | 77 |
| *[16]* | 23 |
| *WT-[ABC]* | 61 |
| *[-BC]* | 32 |
| *[A-C]* | 49 |
| *[AB-]* | 24 |
| *[A--]* | 33 |
| *[-B-]* | 35 |
| *[--C]* | 32 |
| *[---]* | 37 |
| *[ACB]* | 39 |
| *[BAC]* | 34 |
| *[BCA]* | 37 |
| *[CBA]* | 34 |

**Table S2. Number of individuals analyzed for each construct in this study.**

|  | **regulatory potential (sufficiency)** | **necessity** |
|---|---|---|
| ***[A--]*** | A is sufficient for vein expression |  |
| ***[-B-]*** | B is sufficient for alula expression |  |
| ***[--C]*** | C is sufficient for wing blade expression |  |
| ***[AB-]*** |  | C is necessary for high levels in the spot |
| ***[A-C]*** | A is sufficient to repress wing blade expression (outside of spot region) | B is necessary for alula expression<br>B is necessary for full spot levels |
| ***[-BC]*** | B is sufficient to repress wing blade expression (outside of spot region) | A is necessary for full spot levels |

**Table S3. Analysis of necessity and sufficiency of each block.**

**Data file S1. Scores for the PCA shown in Figure S1**.

**Data file S2**. **Significance of difference in activity between pairs of groups, using the first 6 principal components.**

**Data file S3. Significance of the difference in average expression levels among constructs of the first mutant series (*[0]-[16]*).**

**Data file S4. Significance of difference in average expression levels among constructs of the second mutant series (blocks).**

**Additional notes on *logRatios*.**

Using average phenotypes to evaluate the effect of the mutations we introduced is useful but limited. Indeed, the differences we observe are visually driven by changes in regions of the wing with elevated enhancer activity. It is then difficult to appreciate whether a mutation affects enhancer activity locally or uniformly across the wing. Differential gene expression is generally represented using log ratios (see reference *(30)* in main text), which measure the fold changes in expression level of a gene relative to a reference (*e.g.*, the expression of the same gene under different conditions). We applied this principle to our image data to visually compare the activity of different constructs across the wing. Classical log ratio translates here to the log of the pixel-wise ratio between two average phenotypes at every pixel (hereafter noted *logRatio*). *logRatio* images of mutants *vs.* wild type are of particular interest to decipher the regulatory logic, because they reveal in which proportion a mutant affects the enhancer activity across the wing. Compared to absolute difference, *logRatio* are not driven by regions with high levels of expression, but by regions with a large fold change, irrespective of the wild-type activity pattern. In a theoretical case where the enhancer activity depends directly and linearly on a given TF concentration, the *logRatio* image reflects logically the spatial distribution of this particular TF. This is also the case if this integration of this TF information is only modulated by uniformly distributed TFs. The underlying logic is straightforward: in this theoretical case, a sequence mutation breaking the interaction between the DNA and the TF will have a significant effect on the phenotype. The intensity of the local phenotypic effect (relatively to the wild-type levels) will depend on the local intensity of the TF-DNA interaction across the wing, and therefore on the local concentration of the TF. Logically, this interaction is not happening where the TF is absent, with no effect on the phenotype. For any situation departing from these ideal conditions, the resemblance between the *logRatio* and the TF distribution is compromised. For instance, when a TF is locally repressed by another, *logRatio* will correspond to the net loss of spatial information integration, including the loss of this repression. The *logRatio* of a mutant affecting a known TFBS for which the corresponding TF distribution is known therefore informs us on its contribution in the regulatory logic of the enhancer, and how linearly this integration happens. Moreover, even without additional knowledge on the regulatory logic and TF spatial variation, the variety of *logRatio* patterns suggests the action of different spatial inputs integrated by the enhancer.

## 2.2 Cristae-dependent quality control of the mitochondrial genome

### 2.2.1 Main Paper

SCIENCE ADVANCES | RESEARCH ARTICLE

GENETICS

# Cristae-dependent quality control of the mitochondrial genome

**Christopher Jakubke[1,2†], Rodaria Roussou[1,2†], Andreas Maiser[1], Christina Schug[3], Felix Thoma[1,2], David Bunk[1], David Hörl[1], Heinrich Leonhardt[1], Peter Walter[4,5], Till Klecker[3], Christof Osman[1]***

Copyright © 2021
The Authors, some
rights reserved;
exclusive licensee
American Association
for the Advancement
of Science. No claim to
original U.S. Government
Works. Distributed
under a Creative
Commons Attribution
NonCommercial
License 4.0 (CC BY-NC).

**Mitochondrial genomes (mtDNA) encode essential subunits of the mitochondrial respiratory chain. Mutations in mtDNA can cause a shortage in cellular energy supply, which can lead to numerous mitochondrial diseases. How cells secure mtDNA integrity over generations has remained unanswered. Here, we show that the single-celled yeast *Saccharomyces cerevisiae* can intracellularly distinguish between functional and defective mtDNA and promote generation of daughter cells with increasingly healthy mtDNA content. Purifying selection for functional mtDNA occurs in a continuous mitochondrial network and does not require mitochondrial fission but necessitates stable mitochondrial subdomains that depend on intact cristae morphology. Our findings support a model in which cristae-dependent proximity between mtDNA and the proteins it encodes creates a spatial "sphere of influence," which links a lack of functional fitness to clearance of defective mtDNA.**

## INTRODUCTION

Mitochondria contain their own genome, known as mitochondrial DNA (mtDNA), which in most organisms encodes core subunits of the respiratory chain and the adenosine triphosphate (ATP) synthase as well as transfer RNAs (tRNAs) and ribosomal RNAs (rRNAs) required for mitochondrial protein translation. Multiple copies of mtDNA are distributed throughout the mitochondrial network. Mutations in mtDNA can have detrimental consequences for mitochondrial function and can lead to a multitude of mitochondrial diseases, which have a prevalence of ~20 cases per 100,000 individuals (*1*). How cells maintain the integrity of mtDNA over generations, despite high mitochondrial mutation rates, has remained unclear. Studies in *Drosophila melanogaster* and mice have revealed that mutant copies of mtDNA are removed in the female germline in a process known as purifying selection (*2*–*4*). Mitochondrial fission has been proposed to contribute to this process in *D. melanogaster* (*5*, *6*). Generation of small mitochondrial fragments containing only one or a few mitochondrial genomes is believed to separate mtDNA copies from one another to prevent complementation of mutant mtDNA by gene products of intact mtDNA. Subsequently, mitochondrial fragments containing mutated mtDNA are proposed to be detected on the basis of a decreased membrane potential or ATP content and removed by mitophagy. In addition, it has been found that compromised protein import efficiency into damaged mitochondria leads to a decrease in mtDNA replication and hence disfavors propagation of mutated mtDNA copies (*3*, *7*). Despite this progress, many questions remain about the cellular mechanisms that facilitate selection against mutant mtDNA. By exploiting the possibility to genetically manipulate mtDNA in *Saccharomyces cerevisiae*, we establish budding yeast as a model system to study mtDNA quality control.

## RESULTS

### A pedigree analysis reveals mtDNA quality control in *S. cerevisiae*

First, we asked whether the single-celled *S. cerevisiae* intracellularly distinguishes between intact and mutant mtDNA and supports generation of young progeny with a healthy mtDNA content. We devised an approach to genetically follow segregation of wild-type (WT) and mutant mtDNA copies from heteroplasmic single yeast cells. Two yeast strains of opposing mating types were used for this approach. The first strain is of mating type *a* and harbors WT mtDNA and a deletion of the nuclear-encoded *ARG8* gene that encodes the mitochondrially localized Arg8 protein and is required for synthesis of arginine. Therefore, this strain is able to grow on medium containing nonfermentable carbon sources but not on medium lacking arginine. The second strain is of mating type *alpha* and harbors a deletion of the nuclear-encoded *ARG8*, but contains mtDNA in which the *COB* gene is replaced by the *ARG8* gene (Δ*cob::ARG8*) (*8*). This latter strain can thus grow in the absence of arginine but not on nonfermentable carbon sources due to the deletion of the mtDNA-encoded *COB* gene. The open reading frames (ORFs) of *COB* and *ARG8* are 1158 and 1294 base pairs (bp) in size, respectively. Replacement of the *COB* gene, therefore, results in a minor increase of the overall size of the mtDNA of less than 0.2%, which is unlikely to confer a replicative disadvantage on the Δ*cob::ARG8* mtDNA due to larger size. Both strains contained comparable amounts of mtDNA as determined by quantitative real-time polymerase chain reaction (PCR; fig. S1A).

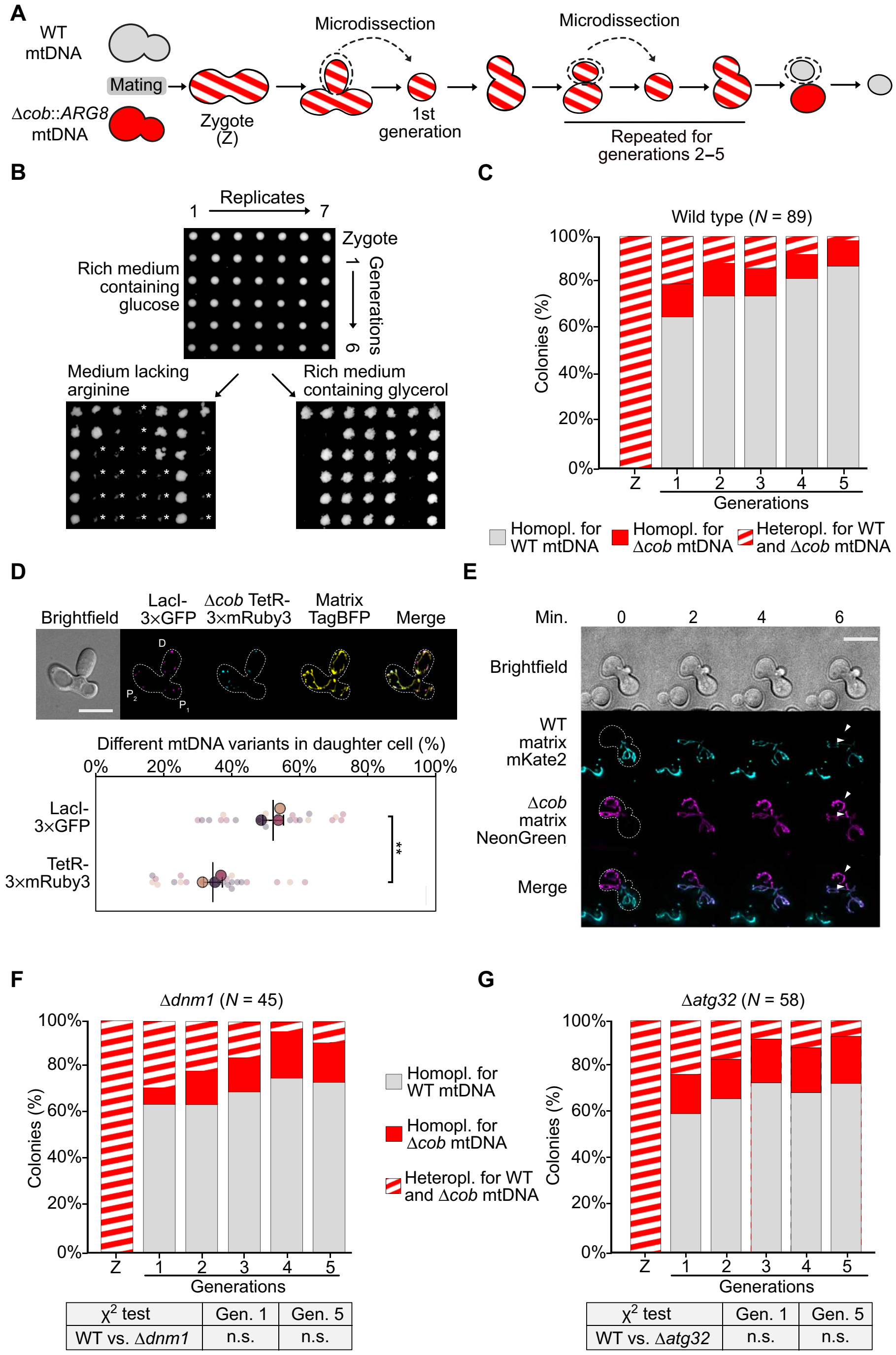
Both yeast strains were mated to obtain heteroplasmic diploid zygotes containing WT and Δ*cob::ARG8* mtDNA. Microdissection was used to transfer single zygotes to a cell-free area on an agar plate containing rich medium and glucose as a fermentable carbon source, which supports growth of cells containing WT, Δ*cob::ARG8*, or both mtDNA species. Once the first daughter cell had budded from the zygote, it was moved to a new location on the agar plate. Growth of this former daughter cell was again monitored, until the second-generation daughter cell had budded, which was again transferred to a new position. This procedure was repeated for up to five generations, and isolated cells were incubated to allow growth of colonies (Fig. 1, A and B). The mtDNA genotype in such colonies was inferred by

Downloaded from https://www.science.org on January 26, 2022

---

[1]Faculty of Biology, Ludwig-Maximilian-Universität München, 82152 Planegg-Martinsried, Germany. [2]Graduate School Life Science Munich, Planegg, Germany. [3]Zellbiologie, Universität Bayreuth, 95440 Bayreuth, Germany. [4]Howard Hughes Medical Institute and Department of Biochemistry and Biophysics, University of California, San Francisco, San Francisco, CA 94143, USA. [5]Department of Physiology, University of California, San Francisco, San Francisco, CA 94143, USA.
*Corresponding author. Email: osman@bio.lmu.de

†These authors contributed equally to this work.



| $\chi^2$ test | Gen. 1 | Gen. 5 |
|---|---|---|
| WT vs. Δ*dnm1* | n.s. | n.s. |

| $\chi^2$ test | Gen. 1 | Gen. 5 |
|---|---|---|
| WT vs. Δ*atg32* | n.s. | n.s. |

**Fig. 1. *S. cerevisiae* cells distinguish between WT and mutant mtDNA.** (**A**) Schematic illustration of the pedigree analysis. Two Δ*arg8* yeast strains harboring WT or Δ*cob::ARG8* mtDNA were mated. Zygotes were isolated, and daughter cells from up to five consecutive generations were separated and placed on free spots on the agar plate by microdissection. (**B**) Growth pattern of the pedigree analysis after microdissection. Colonies were initially grown on rich medium containing glucose. The mtDNA genotype was inferred from the ability of colonies to grow on synthetic arginine lacking medium (Δ*cob::ARG8* mtDNA) or on medium containing the nonfermentable carbon source glycerol (WT mtDNA). Asterisks indicate cell material that was carried over by replica plating and failed to produce obvious colonies upon further incubation. For further illustrative explanation and confirmation of the pedigree analysis, refer to fig. S1 (B to D). (**C**) Pedigree analysis of WT cells. Striped bars indicate percentage of heteroplasmic cells containing WT and Δ*cob::ARG8* mtDNA. Gray or red bars indicate percentage of homoplasmic cells containing WT or Δ*cob::ARG8* mtDNA, respectively. (**D**) Inheritance of either intact or mutated mtDNA. Mating events between two cells either with WT-LacO (GFP, $P_1$) or Δ*cob::ARG8*-TetO (mRuby3, $P_2$) mtDNA. Both cells expressed a nuclear-encoded, matrix-targeted TagBFP. The percentage of either GFP or mRuby3 spots in the daughter cells relative to total number of the respective mtDNA variant have been plotted. Big circles represent the mean values from individual experiments. $^{**}P<0.01$, *t* test. (**E**) Mitochondrial morphology during mating. Mating events between two cells containing either WT or Δ*cob::ARG8* mtDNA. Cells expressed either matrix-targeted mKate2 or mNeonGreen (NG). Mating events were monitored by live-cell microscopy. Time point before mixing of the matrix contents has been defined as T0. (**F** and **G**) Pedigree analysis of Δ*dnm1* or Δ*atg32* cells. Scale bars, 10 μm (D and E).

Downloaded from https://www.science.org on January 26, 2022

their ability to grow on medium containing a nonfermentable carbon source or medium lacking arginine, indicative of the presence of WT or Δ*cob::ARG8* mtDNA, respectively (Fig. 1B and fig. S1B). The presence or absence of mtDNA was further corroborated by PCR analysis of selected lineages (fig. S1C). Of note, this growth-based assay cannot distinguish whether mtDNA variants are entirely absent or present in insufficient amounts to support growth on the respective selective medium. Previous analyses in *S. cerevisiae* have revealed rapid segregation of mtDNA variants containing neutral genetic markers within less than 10 and often within one generation (*9*). We similarly observe rapid segregation; notably, however, we detected a strong bias for the WT mtDNA copy. More than 60% of the colonies derived from the first-generation daughter contained exclusively the WT mtDNA, while only about ~15% of the colonies contained exclusively the Δ*cob::ARG8* mtDNA and ~25% remained in a heteroplasmic state (Fig. 1C). A further segregation toward WT mtDNA was observed in following generations. More than 80% of colonies derived from daughters of the fifth generation contained only the WT mtDNA (Fig. 1C). Only 3% of the colonies of the fifth generation were heteroplasmic, indicating virtually complete segregation of both mtDNA types. Very similar results were obtained in pedigree analyses, where the mating types of the WT and the Δ*cob::ARG8* mtDNA containing strains had been switched, indicating that selection for WT mtDNA over Δ*cob::ARG8* mtDNA is independent of the mating type (fig. S2A). Furthermore, we did not observe selection against mtDNA variants with an *ARG8* gene inserted into a noncoding region of mtDNA, indicating that the *ARG8* gene does not impart a strong selective disadvantage on the mtDNA (fig. S2B).

We developed a microscopic approach to visualize selection of WT over Δ*cob* mtDNA during budding of daughter cells from zygotes. To this end, we used our previously generated strain that harbors tandem LacO repeats in a noncoding region of mtDNA upstream of *COX2* and expresses a nuclear-encoded LacI-3×GFP (green fluorescent protein) fusion protein harboring an N-terminal mitochondrial targeting sequence (*10*). In a separate strain, we replaced the mtDNA-encoded *COB* gene with tandem TetO repeats and introduced a gene encoding a mitochondrially targeted Tet-repressor fused to three repeats of the fluorescent protein mRuby3 into the nuclear genome. This strain contained very similar mtDNA levels compared to a reference strain, indicating that the TetO-TetR system does not lead to major defects in mtDNA maintenance (fig. S3A). In the strains used for this analysis, LacI-3×GFP and TetR-3×mRuby3 bind to the LacO and TetO repeats, respectively. To visualize mitochondria, both strains express the mitochondrially targeted blue fluorescent protein TagBFP. Both strains were mated, and zygotes were analyzed by fluorescence microscopy. We detected mutant mtDNA in the daughter cell. Quantitative image analysis, however, revealed that relative amounts of WT LacO-mtDNA were more abundant in the daughter cell than the Δ*cob* TetO-mtDNA (Fig. 1D and fig. S3B). While we cannot entirely rule out that the LacO-LacI or the TetO-TetR systems affect the mtDNA selection process or lead to biased results in this imaging approach, these data support the conclusion that *S. cerevisiae* cells can distinguish between WT and mutant mtDNA and promote generation of progeny with predominantly healthy mtDNA content.

### Selection against mutant mtDNA occurs in a continuous mitochondrial network

Like in higher eukaryotes, mitochondria form a continuous tubular network in *S. cerevisiae* that is constantly rearranged by fusion and fission events. A simple explanation for rapid segregation of WT and mutant mtDNA could be that dysfunctional mitochondria do not fuse with healthy mitochondria to form a continuous network upon mating. In such a scenario, mutant and WT copies of mtDNA would not mix but be kept in separated mitochondrial compartments, which could facilitate efficient segregation. To examine this possibility, we used live-cell microscopy to monitor mitochondrial fusion during mating of two yeast strains expressing either mNeonGreen (NG) or mKate2 targeted to the mitochondrial matrix. We first examined mating events between two cells, both containing WT mtDNA. In line with previous observations, we observed fusion of mitochondria in newly formed zygotes (*11*–*13*). Fusion was characterized by quick and mostly complete equilibration of mKate2 and NG within 2 min throughout virtually all parts of the mitochondrial network in zygotes (fig. S4A). Similarly, we observed rapid fusion of mitochondrial networks in mating events between cells containing either WT or Δ*cob::ARG8* mtDNA (Fig. 1E, fig. S4B, and movie S1). Detection of both fluorescent proteins throughout the mitochondrial network was the result of fusion rather than ongoing translation, because unfused fragments could be distinguished during the mitochondrial fusion process in zygotes (arrowheads in Fig. 1E). Despite content mixing of mitochondria, our pedigree analysis reveals that yeast zygotes can still distinguish between mutant and WT mtDNA to produce daughter cells that predominantly contain WT mtDNA.

### Mitochondrial fission or Atg32-mediated mitophagy is not required for selection against mutant mtDNA

We asked whether mitochondrial fission could facilitate selection against mutant mtDNA, as has been proposed for *D. melanogaster* (*5*, *6*). We did not observe increased mitochondrial fragmentation after mating between cells containing WT mtDNA or mating between cells containing either WT or Δ*cob::ARG8* mtDNA. Thus, excessive mitochondrial fragmentation does not occur during selection against mutant mtDNA in yeast zygotes (Fig. 1E; fig. S4, A and B; and movie S1).

To further assess the role of mitochondrial fission in selection against mutant mtDNA, we performed pedigree analyses of crosses between cells lacking the fission protein Dnm1 and harboring either WT or mutant Δ*cob::ARG8* mtDNA. Notably, the absence of mitochondrial fission did not compromise rapid selection for WT mtDNA in a colony produced by the first daughter cell of heteroplasmic yeast zygotes (Fig. 1F). Furthermore, live-cell microscopy of mating events between two Δ*dnm1* strains harboring either WT or mutant mtDNA and expressing matrix-targeted mKate2 or NG, respectively, revealed that mitochondria from both parental cells rapidly fused in zygotes (fig. S4C and movie S2). We also tested the importance of the mitophagy receptor Atg32 (*14*, *15*) for selection against mutant mtDNA. Δ*atg32* cells behaved similar to WT cells in pedigree analyses and exhibited complete mitochondrial fusion during the mating process (Fig. 1G, fig. S4D, and movie S3). Thus, neither mitochondrial fission nor Atg32-mediated mitophagy is essential for selection against mutant mtDNA during mating events of *S. cerevisiae* cells. Notably, selection for WT mtDNA in generations 2 to 5 appeared less prominent in pedigree analyses of Δ*dnm1* or Δ*atg32* cells compared to matings between WT cells. This observation could indicate that selection toward WT mtDNA in daughter cells of zygotes is independent of fission and mitophagy, whereas further selection in dividing diploids may be supported by these processes.

Downloaded from https://www.science.org on January 26, 2022

### The mtDNA-encoded protein Atp6 exhibits severely limited diffusion rates

The fact that selection occurs in a continuous network that shares WT and mutant mtDNA presents a conundrum. How can cells distinguish between WT and mutant mtDNA if mitochondrial content equilibrates in a continuous network? We hypothesized that selection can only occur when subdomains are maintained within the mitochondrial network, whose functionality is determined by nearby copies of mtDNA. In such a scenario, it is important that mtDNA-encoded respiratory chain subunits do not exhibit rapid diffusion, because this would allow complementation of dysfunctional subdomains by gene products from WT mtDNA. To examine diffusion of an mtDNA-encoded protein, we modified mtDNA and fused an NG-tag to the C terminus of the mtDNA-encoded *ATP6* gene (fig. S5A). Yeast strains exclusively expressed the NG-tagged variant of Atp6 and did neither show growth defects on medium containing fermentable or nonfermentable carbon sources at 30° or 37°C nor exhibit increased formation of so-called petite cells that are respiratory deficient or display altered mtDNA or Atp6 protein levels (Fig. 2A and fig. S5, B to D). These observations indicate full functionality of the NG-tagged Atp6.

Fluorescence microscopy of diploid cells harboring the *ATP6-NG* mtDNA and expressing a nuclear-encoded matrix-targeted mScarlet revealed that, in contrast to the uniformly distributed mScarlet signal, Atp6-NG exhibited a patchy distribution in the mitochondrial network (fig. S5E). This distribution has also been observed previously

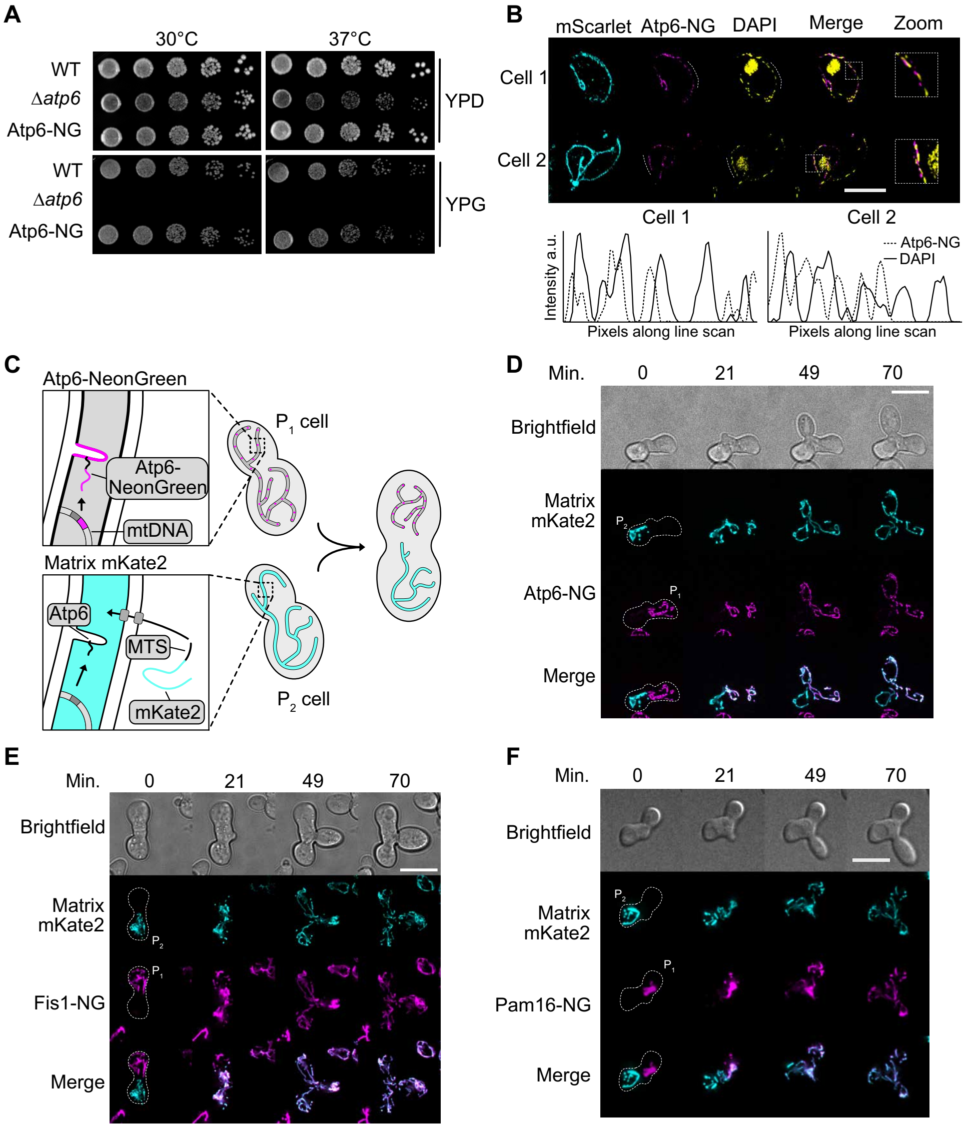


**Fig. 2. mtDNA-encoded Atp6-NG exhibits limited diffusion.** (**A**) Growth analyses of strains expressing NG-tagged Atp6. Serial dilutions of indicated strains were spotted on rich medium containing glucose or glycerol as carbon sources and incubated at 30° or 37°C. (**B**) Structured illumination microscopy of DAPI-stained cells expressing Atp6-NG and matrix-targeted mScarlet. Line graphs of pixel intensities along indicated lines are shown. a.u., arbitrary units. (**C**) Schematic illustration of the mating experiment of the Atp6-NG–tagged strain with a strain expressing the matrix labeling Su9-mKate2. (**D** to **F**) Diffusion of mitochondrial membrane proteins in fused mitochondria of zygotes. Cells expressing matrix-targeted mKate2 were mated with cells expressing mtDNA-encoded Atp6-NG (D), nuclear-encoded Fis1-NG (E), or Pam16-NG (F). Scale bars, 5 μm (B) and 10 μm (D to F).

Downloaded from https://www.science.org on January 26, 2022

using nuclear-encoded GFP-tagged variants of ATP synthase subunits, and it has been proposed that these patches report on mitochondrial cristae, in which the ATP synthase is enriched (*16*). We used structured illumination microscopy of 4′,6-diamidino-2-phenylindole (DAPI)–stained cells to query the relationship between Atp6-NG and mtDNA. Atp6-NG foci were spatially linked to mtDNA and were detected in the immediate surroundings of DAPI spots rather than directly colocalizing with them (Fig. 2B). This finding demonstrates that the ATP synthase localizes close to mtDNA but is largely excluded from areas occupied by mtDNA itself. This observation is in line with recent super-resolution data in HeLa cells that showed a lack of cristae in close proximity to mtDNA (*17*).

To further test whether Atp6-NG remains close to the copy of mtDNA by which it is encoded, we examined diffusion of Atp6-NG in living cells during a mating experiment. Cells harboring mtDNA encoding Atp6-NG were mated with cells expressing a nuclear-encoded matrix-targeted mKate2 and harboring nonmodified mtDNA. Equilibration of both fluorescent signals throughout the mitochondrial network of zygotes was studied over time by live-cell microscopy (Fig. 2C). In line with our previous observation, soluble mKate2 rapidly equilibrated throughout the mitochondrial network upon fusion of parental mitochondria. Notably, the Atp6-NG signal remained localized to mitochondria from the cell of its origin (Fig. 2D, fig. S6A, and movie S4). Only very small Atp6-NG amounts were present in the mitochondrial network of the mating partner ~80 min after mitochondrial fusion, when a daughter bud had already emerged at midpoint between both parental cells and had been invaded by mitochondria. To quantify equilibration of Atp6-NG in late zygotes harboring a large medial bud, we first segmented the mitochondrial network based on the mitochondrial mKate2 signal and assigned mitochondrial parts to either the parental Atp6-NG ($P_1$), the parental mKate2 ($P_2$), or the daughter cell. The Atp6-NG signal was then determined along the mitochondrial network parts that had been assigned to $P_1$, $P_2$, and daughter cell. The Atp6-NG signal was further normalized to the length of the subnetwork localized to the respective cells. This analysis revealed that on average only approximately 20% of the Atp6-NG was detected in the mitochondrial network of the other parental cell (Fig. 3B). In contrast, no difference in the mKate2 signal in both parental cells was apparent in this analysis, indicating complete mitochondrial fusion and full equilibration of this soluble matrix protein (fig. S7A). We cannot distinguish between preexisting and newly synthesized Atp6-NG in this assay but conclude that proteins from neither group efficiently populate the mitochondrial network of the parental cell that did not contain the Atp6-NG mtDNA.

To examine diffusion of another protein component of the oxidative phosphorylation (OXPHOS) complexes, we created a strain expressing an NG-tagged version of the nuclear-encoded protein Cox4, which is a subunit of complex IV. Cells expressing Cox4-NG cells were mated with cells expressing matrix-targeted mKate2. Of note, upon mating of these cells, continued cytosolic synthesis of Cox4-NG will lead to import of Cox4 into all parts of the mitochondrial network at later time points. Despite this complication and similar to our results obtained with Atp6-NG, Cox4-NG exhibited strongly reduced diffusion throughout the mitochondrial network of zygotes compared to the soluble matrix–targeted mKate2 and remained largely restricted to the cell of its origin during the duration of the microscopy experiment (fig. S6B).

Next, we asked whether limited diffusion of Atp6-NG or Cox4-NG was general to transmembrane (TM) proteins of mitochondrial membranes. We mated cells expressing matrix-targeted mKate2 with cells either expressing NG fused to the TM-domain of the outer membrane protein Fis1 or a functional Pam16-NG fusion protein (Fig. 2, E and F, and fig. S6, C and D). Pam16 is a subunit of the translocase of the inner mitochondrial membrane, which localizes predominantly to the inner boundary membrane (*18*). Time-lapse microscopy of such mating events revealed that diffusion of Fis1-TM-NG and Pam16-NG was slightly delayed compared to the soluble matrix protein mKate2, but both proteins equilibrated much faster than Atp6-NG throughout mitochondrial networks and no difference in signal intensity in both parental cells could be observed ~10 min after fusion of mitochondria in zygotes (fig. S6, E and F, and movies S5 and S6). In summary, mtDNA-encoded Atp6-NG and nuclear-encoded Cox4-NG, which are both subunits of respiratory chain complexes consisting of nuclear- and mtDNA-encoded subunits, are severely limited in equilibration throughout mitochondrial tubules compared to proteins of the inner boundary or the outer mitochondrial membranes.

## Components important for cristae morphology affect diffusion rates of Atp6-NG

We hypothesized that components involved in maintenance of cristae morphology could be important for hindering diffusion of OXPHOS subunits and particularly mtDNA-encoded proteins within the mitochondrial network. Mic60 and Mic10 are components of the mitochondrial contact site (MICOS) complex, which stabilizes cristae junctions (*19*–*21*). Lack of Mic60 leads to cristae that are detached from the inner mitochondrial membrane but maintain their sheet-like morphology in the matrix. Atp20 and Atp21 are crucial for dimerization of the ATP synthase, which, in turn, stabilizes strongly bent cristae rims (*22*). Lack of the dimeric ATP synthase has been described to lead to an onion-like morphology of mitochondria, where uncontrolled proliferation of cristae membranes leads to a multilayered appearance (*23*). To examine the role of cristae in limiting equilibration of OXPHOS proteins, we quantified Atp6-NG mobility in our mating assay in Δ*atp20*, Δ*atp21*, Δ*atp20*Δ*atp21*, Δ*mic10*, or Δ*mic60* strains. Both parental cells contained the respective deletion in these crosses. In all mating events, the nuclear-encoded mKate2 equilibrated across the mitochondrial network (fig. S7A). We could observe a slightly less complete equilibration of mKate2 in zygotes lacking the dimeric ATP synthase, which is likely explained by septae that may create separate matrix compartments (*24*) and thereby may decrease efficiency of matrix content exchange. Δ*mic10* and Δ*mic60* cells exhibited increased diffusion of Atp6-NG between mitochondrial networks of parental cells compared to WT cells in our mating assay, which was characterized by a greater amount of Atp6-NG detected in the mating partner harboring the WT mtDNA (~35% compared to ~25% in WT cells; Fig. 3, A and B). Cells lacking Atp20, Atp21, or both proteins showed a strongly increased diffusion of Atp6-NG across the mitochondrial network in zygotes. In many cases, the source cell that contained *ATP6-NG*-mtDNA was hardly discernible from the mKate2 source cell in the zygotes. On average, ~45% of Atp6-NG was detected in the mating partner harboring the WT mtDNA in the absence of Atp20, Atp21, or both proteins (Fig. 3, A and B). In contrast to limited diffusion within the mitochondrial network formed by parental cells, Atp6-NG was readily detected in the budding daughter cells. In daughters from zygotes, the Atp6-NG signal was at ~70% compared to the *ATP6-NG* signal from the *ATP6-NG* parental cell in all mutants,

Downloaded from https://www.science.org on January 26, 2022

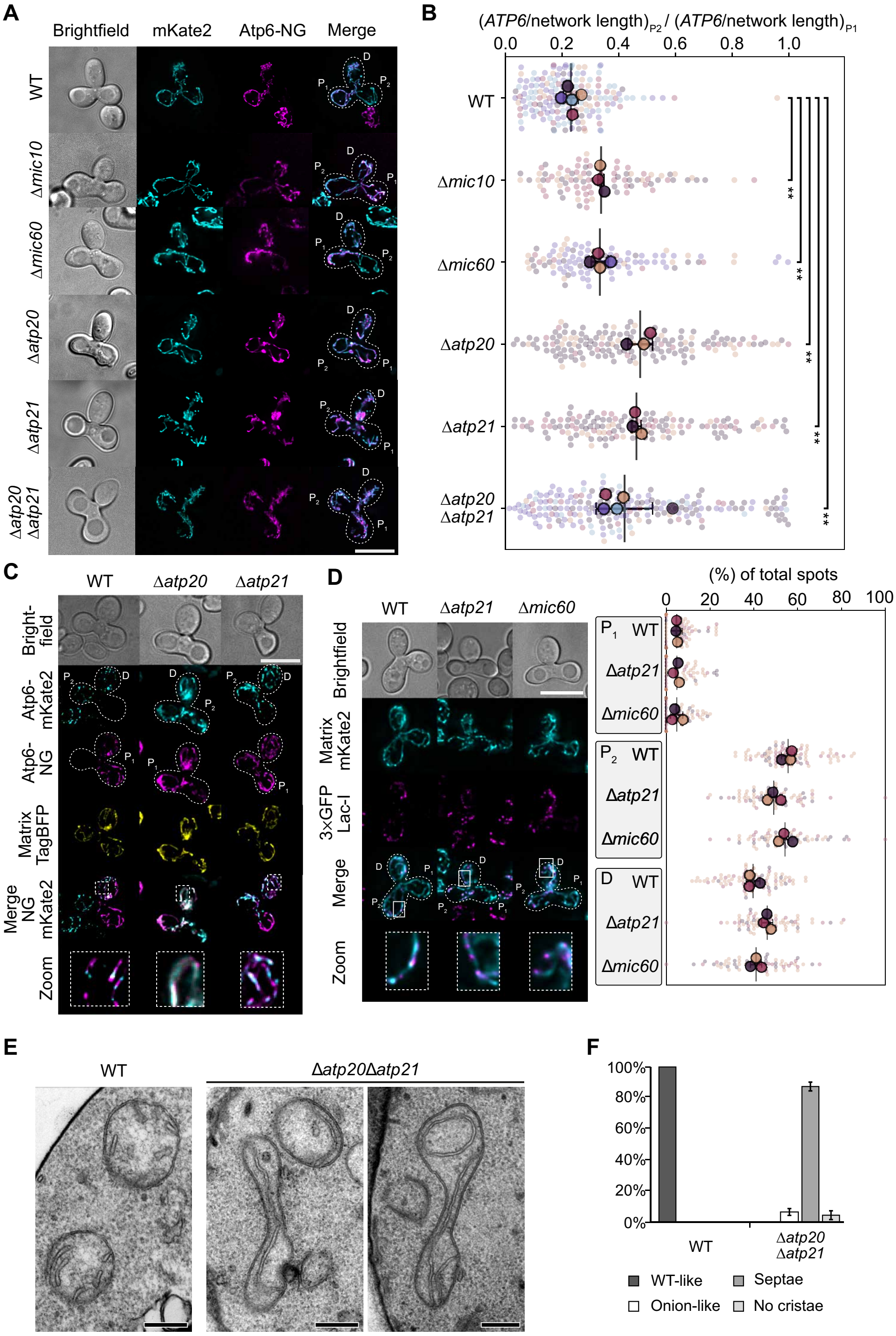


**Fig. 3. Intact cristae morphology limits diffusion of mtDNA-encoded Atp6-NG.** (**A**) Zygotes derived from mating events between WT or indicated mutant cells. In each mating event, one cell contained *ATP6*-NG mtDNA ($P_1$) and the other cell contained WT mtDNA and expressed matrix-targeted mKate2 ($P_2$). D, daughter cell. (**B**) The Atp6-NG signal was quantified in both parental cells of zygotes and normalized to the mitochondrial network length. The ratio of Atp6-NG signals in $P_2$ cells to $P_1$ cells of the same zygote is plotted; each point represents one zygote; $^{**}P < 0.01$, *t* test. (**C**) Mating events between WT, Δ*atp20*, or Δ*atp21* cells. Mating cells harbored either Atp6-NG mtDNA and nuclear-encoded Su9-TagBFP ($P_1$) or Atp6-mKate2 ($P_2$) mtDNA (D, daughter cell). (**D**) Mating events between WT, Δ*atp21*, and Δ*mic60* cells. Parental cell $P_1$ contained WT mtDNA and expressed matrix-targeted mKate2, whereas parental cell $P_2$ contained mtDNA with integrated LacO repeats (D, daughter cell). mtDNA-LacO spots in $P_1$, $P_2$, and D cells were counted and plotted as percentage of total number of spots in the zygote. Each data point in the plot represents one zygote. (**E**) Electron microscopic analysis of mitochondrial ultrastructure in WT and Δ*atp20*Δ*atp21* cells. (**F**) Quantification of electron microscopy data shown in (E). Scale bars, 10 μm (A, C, and D) and 250 nm (E).

Downloaded from https://www.science.org on January 26, 2022

with the exception of Δ*mic60* cells, where Atp6-NG levels only amounted to ~60% (fig. S7B). The reason why less Atp6-NG is found in the daughter cells of matings between Δ*mic60* cells is currently unclear. In summary, we observe increased diffusion of Atp6-NG in mutants lacking intact cristae morphology across existing mitochondrial networks of yeast zygotes. Because Δ*mic10* or Δ*mic60* mutants still have the dimeric form of the ATP synthase (*25*, *26*), our results strongly suggest that Atp6-NG mobility is increased due to compromised cristae morphology rather than conversion of the ATP synthase into the monomeric form in cells lacking Atp20 or Atp21.

We developed an additional assay to test the generation of mtDNA autonomous domains and the role of cristae therein. Cells expressing mtDNA-encoded Atp6-NG and nuclear-encoded matrix-targeted TagBFP were mated with cells expressing mtDNA-encoded Atp6-mKate2, and the patterns produced by the differently tagged Atp6 proteins were examined in zygotes. The Atp6-mKate2 strain grew normally on medium containing fermentable or nonfermentable carbon sources, and no increase in *petite* formation was observed, indicating that the mKate2-tagged variant of Atp6 was functional (fig. S7C). In line with our previous observations, the Atp6 proteins remained localized to their respective origin cells in zygotes in crosses between WT cells (Fig. 3C). Notably, we often observed an alternating, mostly nonoverlapping pattern of green and red signals along the TagBFP-stained mitochondrial network in the daughter cell produced by the zygote (Fig. 3C, zoom, and fig. S7D). Thus, Atp6-NG and Atp6-mKate2 occupy separate domains in the mitochondrial network and exhibit limited mixing. This result indicates that mtDNA copies encoding Atp6-NG or Atp6-mKate2 maintain semi-autonomous mitochondrial subdomains. The alternating pattern in the daughter cell further suggests a complex procedure in which mitochondrial domains from both parental cells are sorted into the daughter cell. In mating events between Δ*atp20* and Δ*atp21* cells expressing either Atp6-NG or Atp6-mKate2, both fluorescently tagged Atp6 variants showed a stronger colocalization in daughter cells and separate domains were hardly discernible (Fig. 3C and fig. S7D). These observations show that cristae are required for maintenance of inner membrane domains and prevent extensive mixing between gene products of different mtDNA copies.

### mtDNA diffusion is limited in WT, Δ*atp21* and Δ*mic60* cells

Cristae could support mtDNA-mediated formation of mitochondrial subdomains via at least two mutually nonexclusive mechanisms: (i) Cristae could limit mobility of mtDNA within the mitochondrial matrix by forming physical barriers and thereby promote local synthesis of mtDNA-encoded proteins. (ii) Cristae could corral mtDNA-encoded proteins by preventing their diffusion through cristae junctions into the inner boundary membrane. To determine the importance of cristae in restricting mtDNA mobility, we used our recently developed mtDNA LacO-LacI-GFP system (*10*) to specifically mark mtDNA in one of the parental cells in a mating experiment ($P_2$ cell). The other parental strain used in the mating assay contained WT mtDNA lacking the LacO repeats and expressed the matrix-targeted mKate2 protein ($P_1$ cell). In mating events between two WT cells, two Δ*atp21*, or Δ*mic60* cells, LacO-marked mtDNA was evident in the daughter cells produced by zygotes, indicating efficient transport of mtDNA into daughter cells. Of note, we observed a slight increase of the percentage of LacO mtDNA spots in the daughter cell in matings between Δ*atp21* cells compared to matings between WT or Δ*mic60* cells (Fig. 3D). The underlying reason for this observation remains to be determined. LacO-marked mtDNA did not accumulate in the $P_1$ parental cell even at late zygotic stages in any of the matings (Fig. 3D). Thus, mobility of mtDNA remains limited in the mitochondrial network of parental cells in matings between Δ*atp21* or Δ*mic60* cells that contain strongly compromised cristae architecture. We conclude that WT-like cristae architecture plays a crucial role in limiting equilibration of the Atp6-NG protein, rather than diffusion of mtDNA.

To find a potential explanation for the rapid equilibration of Atp6-NG in the absence of dimer-specific ATP synthase subunits, we carefully examined WT, Δ*atp20*, Δ*atp21*, and Δ*atp20*Δ*atp21* cells by electron microscopy and also performed serial sectioning to obtain three-dimensional (3D) information about the mitochondrial ultrastructure. Mitochondria of WT cells showed cristae with the typical perpendicular orientation to the mitochondrial tubular axis. In mutant cells lacking dimeric ATP synthase, however, inner membrane structure was markedly altered and virtually no WT-like cristae were apparent. Instead, mitochondria exhibited onion- and balloon-like inner membrane profiles, as previously described. In accordance with previous work, we frequently observed strongly elongated cristae-like structures that traversed the whole mitochondrion (Fig. 3, E and F, and fig. S8) (*22*, *23*,*27*).

These observations provide a possible explanation for the inability of cells lacking dimeric ATP synthase to constrain diffusion of Atp6-NG. While respiratory chain complexes get trapped in cristae in WT cells, they are free to diffuse along the altered cristae membranes in Δ*atp20*, Δ*atp21*, and Δ*atp20*Δ*atp21* mutants.

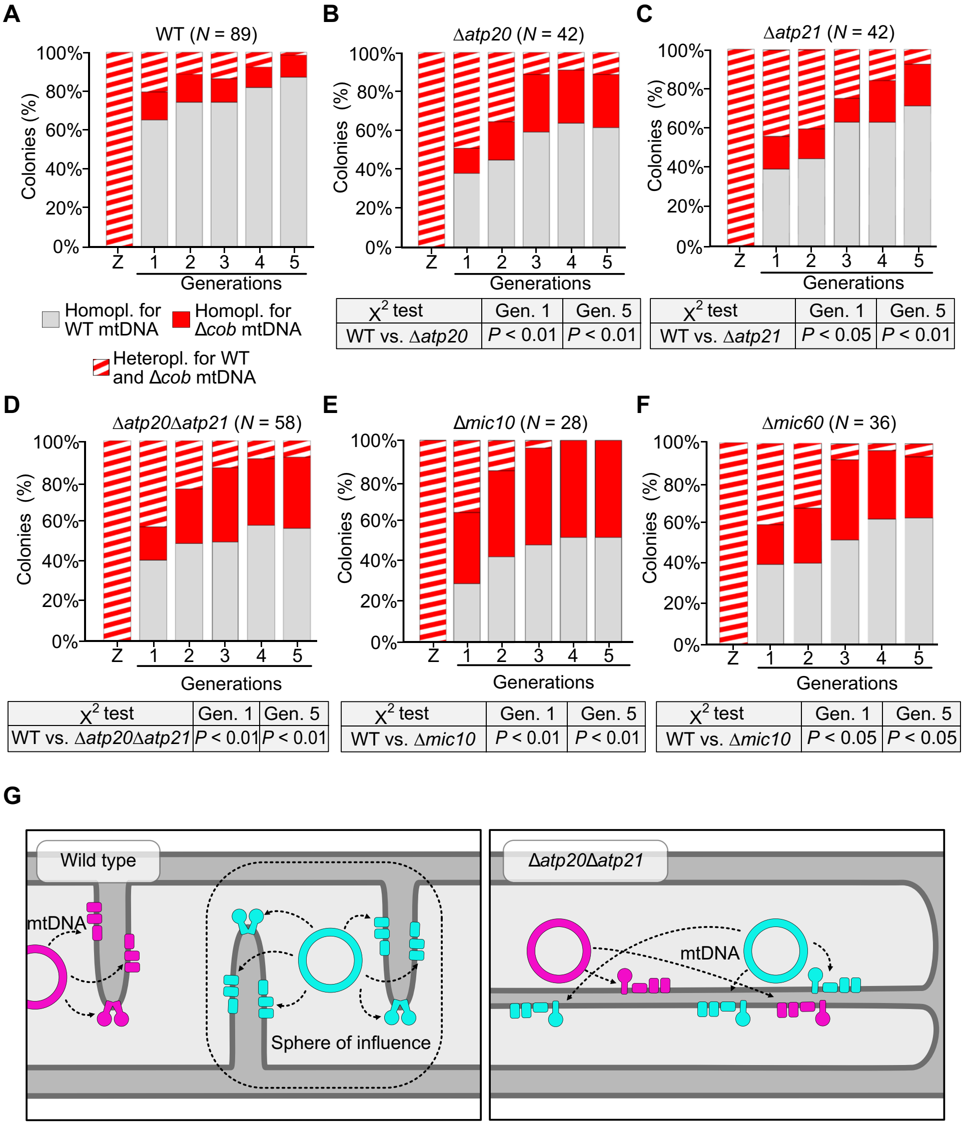
### Cristae morphology is required for mtDNA quality control

Having determined that mtDNA-encoded subunits exhibit a severely reduced mobility in the inner mitochondrial membrane that is dependent on normal cristae biogenesis, we asked whether mutants defective in cristae biogenesis would have problems to distinguish between WT and mutant mtDNA. We applied our pedigree analysis to follow inheritance of WT over Δ*cob::ARG8* mtDNA from heteroplasmic zygotes in the absences of *ATP20*, *ATP21*, *MIC10*, or *MIC60*. Deletion of these genes did not lead to greatly altered mtDNA levels in strains containing WT or Δ*cob::ARG8* mtDNA (fig. S9A). In particular, mtDNA levels of the mating partners of each pedigree mating pair did not differ significantly to one another. Notably, clearance of the mutant mtDNA was significantly delayed in the absence of any of these proteins. While 60% of lineages for the WT were already homoplasmic for WT mtDNA in the first generation, this value dropped to 40% in cristae mutants. Moreover, while only ~20% of cells of the first generation in WT cells still maintained a heteroplasmic state, ~40% of such cells were detected in the first generation of matings between cristae mutants. After five generations, a significantly smaller proportion of the lineages were homoplasmic for WT mtDNA in the cristae mutants (Fig. 4, A to F). Thus, the ability to choose WT over mutant mtDNA is severely compromised in mutants with defective cristae architecture. In line with these pedigree results and in accordance with previous work, Δ*atp20*, Δ*atp21*, Δ*atp20*Δ*atp21*, Δ*mic10*, or Δ*mic60* strains harboring exclusively WT mtDNA exhibit increased formation of *petites*, revealing a reduced ability to maintain a healthy mtDNA population in cells (fig. S9B) (*23*).

## DISCUSSION

Our data reveal that the unicellular *S. cerevisiae* is able to promote generation of progeny with a healthy mtDNA population from

Downloaded from https://www.science.org on January 26, 2022



| $\chi^2$ test | Gen. 1 | Gen. 5 |
|---|---|---|
| WT vs. Δ*atp20* | $P < 0.01$ | $P < 0.01$ |

| $\chi^2$ test | Gen. 1 | Gen. 5 |
|---|---|---|
| WT vs. Δ*atp21* | $P < 0.05$ | $P < 0.01$ |

| $\chi^2$ test | Gen. 1 | Gen. 5 |
|---|---|---|
| WT vs. Δ*atp20*Δ*atp21* | $P < 0.01$ | $P < 0.01$ |

| $\chi^2$ test | Gen. 1 | Gen. 5 |
|---|---|---|
| WT vs. Δ*mic10* | $P < 0.01$ | $P < 0.01$ |

| $\chi^2$ test | Gen. 1 | Gen. 5 |
|---|---|---|
| WT vs. Δ*mic10* | $P < 0.05$ | $P < 0.05$ |

**Fig. 4. mtDNA quality control depends on normal cristae morphology.** (**A** to **F**) Pedigree analyses of strains lacking indicated genes. WT pedigree (A) is identical to Fig. 1C, shown to allow comparison to mutant analyses in (B) to (F). $\chi^2$ test has been applied as statistical hypothesis tests to compare first- and fifth-generation data of mutant strains to the WT. (**G**) Model for quality control of mtDNA in a fused mitochondrial network. mtDNA copies in WT cells supply only the cristae in their immediate surrounding with mtDNA-encoded proteins. Respiratory chain complex containing mtDNA-encoded subunits has limited diffusion capability and is trapped within cristae, which leads to a sphere of influence of mtDNA. Limited diffusion of respiratory chain complexes is compromised in cells with compromised cristae structure.

heteroplasmic zygotes that contain WT and mutant mtDNA. So far, our demonstration of the purifying selection in yeast is limited to a model mutant mtDNA variant that entirely lacks the *COB* gene. It will be interesting to examine in future studies how yeast cells deal with deletions of other mtDNA-encoded genes or milder point mutations. Unexpectedly, we show that the purifying selection occurs in a continuous mitochondrial network and does not require mitochondrial fission. This finding raised the question how cells can detect mutant mtDNA copies among WT copies in a mitochondrial network, where soluble matrix proteins rapidly equilibrate and potentially even out physiological differences and thus conceal mutant copies.

A possible solution for this problem is provided by our finding that diffusion of OXPHOS components, namely, the mtDNA-encoded protein Atp6-NG and the nuclear-encoded Cox4-NG, is severely limited in WT cells compared to proteins localized to the inner boundary membrane, the outer membrane, or the matrix. Of particular interest in this respect is the observation that the mtDNA-encoded Atp6-NG forms subdomains that remain in the vicinity of the mtDNA by which it is encoded. These findings are in line with analyses in HeLa cells that have similarly demonstrated reduced mobility of OXPHOS proteins along the longitudinal axis of mitochondrial tubules (*28*, *29*). A previously proposed plausible explanation for the restricted mobility is that OXPHOS proteins, likely aided through their assembly into complexes and supercomplexes (*30*), get trapped within cristae membranes and are prevented from diffusion past cristae junctions into the inner boundary membrane (*18*, *31*). In strong support of this hypothesis, we find that the mtDNA-encoded Atp6-NG exhibits increased equilibration across the mitochondrial

Downloaded from https://www.science.org on January 26, 2022

network in mutants with defective cristae architecture, while movement of the mtDNA from which it originates remains restricted. Notably, our pedigree analysis revealed that mutants with defective cristae biogenesis are incapable of distinguishing effectively between WT and mutant mtDNA to support progeny with functional mtDNA.

Together, our findings are in line with a previously proposed model in which each mtDNA copy has a "sphere of influence" and remains spatially linked to its gene products (*32*, *33*). Our data support the hypothesis that such a sphere of influence is dependent on normal cristae architecture (Fig. 4G). According to this model, mutations in mtDNA would lead to mitochondrial subdomains with physiological defects that could flag such areas and facilitate purging of the linked mtDNA copy. In line with this idea, it has recently been shown that individual cristae within the same mitochondrial tubule can maintain independent membrane potentials (*34*). Defective cristae biogenesis would lead to defective subdomain formation and a weakened link between mtDNA and its gene products, which, in turn, would blur physiological differences caused by mtDNA mutations (Fig. 4G). As a consequence, detection and purging of mutated mtDNA would be hindered.

For such a system to work, not only the proteins but also their mRNAs need to remain close to the mtDNA copy from which they originate. In matings between two WT cells containing either regular or *ATP6-NG* mtDNA, we observe very low levels of Atp6-NG protein in parts of the mitochondrial network that does not contain the *ATP6-NG* mtDNA, even more than 80 min after mitochondria of both parental cells have fused (fig. S6A). This result indicates that little synthesis of Atp6-NG occurs in these parts of the network within the time frame of this analysis. It is, however, possible that biogenesis of OXPHOS complexes is generally low in the mitochondrial network of parental cells because of little OXPHOS turnover. In this case, we would see little synthesis of Atp6-NG even if its mRNA would be present. Not much is known from the literature about the mobility of mitochondrial mRNAs. It was recently proposed that mitochondrial transcription and translation may occur in a coupled manner in *S. cerevisiae* (*35*). Taking into account that insertion of mtDNA-encoded proteins into the mitochondrial inner membrane occurs cotranslationally (*36*, *37*) and, at least in part, at cristae membranes (*38*), all steps of mitochondrial gene expression from transcription to insertion of proteins into cristae membranes could be spatially linked. Nevertheless, characterization of mRNA mobility in mitochondria awaits further studies.

We demonstrate that yeast cells can distinguish between mutant and WT mtDNA and propose that cristae-dependent subdomains are an important prerequisite for efficient purging of mutant mtDNA. It remains an outstanding and exciting question how cells mechanistically detect and remove mutant mtDNA. Plausible physiological parameters that may serve as signals on dysfunctional mitochondria are a reduced membrane potential (*39*), decreased ATP levels (*5*), or altered redox states (*40*). What processes may remove mutant mtDNA from cells? Our microscopic analyses suggest that mutant mtDNA transfer to daughter cells of heteroplasmic zygotes may be reduced, but not entirely prevented (Fig. 1D). Preferential transport of fit mitochondria into daughter cells has been demonstrated in yeast previously (*40*). In combination with our pedigree analysis, where a large percentage of first-generation colonies appears homoplasmic (Fig. 1C), we suggest that further rounds of selection against mutant mtDNA likely occur in the daughter cells that eventually entirely purge mutant mtDNA. While we find that mitochondrial fission or Atg32-mediated mitophagy is not absolutely essential for selection against mutant mtDNA, especially during production of the first daughter cell of heteroplasmic zygotes, these processes may contribute to efficient clearance during further growth and cell divisions. Furthermore, WT mtDNA could also be preferentially replicated as suggested for *D. melanogaster* (*3*, *7*) or mutant mtDNA could be selectively degraded within mitochondria by nucleases. It was demonstrated in yeast that the mtDNA polymerase Mip1 degrades mtDNA through its exonuclease domain upon prolonged starvation (*41*). It remains to be determined whether such a mechanism could also be involved in clearance of mutant mtDNA. Together, selection against mutant mtDNA could entail a combination of mechanisms that may include selective transport of mitochondria containing healthy mtDNA to daughter cells, mitophagic removal of mitochondrial fragments containing mutant mtDNA, selective replication of healthy mtDNA, or selective degradation of mutant mtDNA by nucleases. The discovery of a purifying selection in *S. cerevisiae* combined with the tools we developed in this study will be of great value in elucidating molecular mechanisms responsible for clearance of mutant mtDNA.

## MATERIALS AND METHODS

### Yeast strains and plasmids

All yeast strains are derived from W303 background. Strain information can be found in table S1. Deletions of genes and C-terminal tagging of nuclear-encoded genes were performed in haploid strains using homologous recombination as described previously (*42*). For the experiment presented in fig. S3A, the mating type of strain yCO354 had to be switched from Mat *alpha* to Mat *a*. Mating type switching was performed by transient (90 min) galactose-induced expression of the HO endonuclease from the plasmid pJH132 (*43*). Subsequently, a Mat *a* strain that had lost the plasmid pJH132 was isolated. Primer and plasmid information can be found in tables S2 and S3, respectively.

### Pedigree analysis

Approximately $1.85 \times 10^7$ cells from early post-log phase [optical density at 600 nm ($OD_{600}$) ~ 1.5] growing cultures of strains with opposing mating types were combined in a 1.5-ml reaction tube, vortexed, centrifuged at 3000 rpm for 3 min, and resuspended in 50 μl of rich medium containing glucose [yeast extract peptone dextrose (YPD)]. Subsequently, the cell suspension was spotted onto a YPD plate and incubated for 2 to 3 hours to allow mating of strains. A small amount of cells was scraped off the plate and was resuspended in 200 μl of YPD medium. From this mated cell suspension, 30 μl was spread as a line on a YPD plate. Individual zygotes were identified on the basis of their characteristic shape and transferred to free areas on the agar plate. Cell growth was monitored, and daughter cells were separated from zygotes or mother cells and transferred to free areas on the agar plate. This procedure was repeated for five generations (Fig. 1, A and B). After incubation for 2 days at 30°C, the grown colonies were replica-plated onto plates containing synthetic defined medium lacking arginine (SD-Arg) and plates containing rich medium and glycerol as a carbon source [yeast extract peptone glycerol (YPG)]. Those plates were again incubated for 24 to 48 hours, and growth was scored. Of note, colonies were scored as "growing" for SD-Arg or YPG even when only parts of the replicated colony displayed growth. Growth rates were not considered in this

Downloaded from https://www.science.org on January 26, 2022

experiment. The presence or absence of growth was verified by restreaking of cell material from selected colonies onto YPG or SD-Arg plates (fig. S1B). In addition, a mating-type test was performed to confirm that the initially picked cells were zygotes. Lineages were only considered when the starting zygotes were able to grow on YPG and SD-Arg plates, indicating a heteroplasmic state. Of note, we sporadically observed that mtDNA species were lost in one generation but reappeared at a later generation (e.g., replicate 2 in Fig. 1B). We attribute this observation to cases in which all copies of one particular mtDNA species are passed on to the daughter cell. In such a scenario, a mother cell lacking this mtDNA species is produced, whereas this species will be present in the daughter cell (fig. S1D).

### Live-cell microscopy

For all live-cell microscopy experiments, yeast cells were imaged in ibidi 8-well μ-Slides (ibidi GmbH, Gräfelfing). For immobilization of yeast cells during microscopy, wells were coated with concanavalin A, a lectin that can bind the cell wall of yeast cells. In brief, wells were filled with 200 μl of concanavalin A (0.5 μg/ml) and incubated for 30 min. Concanavalin A was removed, and wells were dried for 30 min at room temperature. For single-frame images and videos of mating events, 1000 μl from two cultures with strains of opposing mating types was combined in a reaction tube, vortexed thoroughly, and centrifuged for 3 min at 5000 rpm. The pellet was resuspended in 50 μl of YPD medium and spotted onto a YPD plate to allow cells to mate. Plates were incubated for 3 hours for single-frame images and for 1.5 hours for videos. The premated cells were scraped off and resuspended in 400 μl of sterile-filtered 1× phosphate-buffered saline (PBS) buffer, and 200 μl was transferred into a concanavalin A–coated well of ibidi 8-well μ-Slide. The slides were centrifuged for 2 min at 2000 rpm to promote adherence of the cells to the bottom of the well. Wells were washed twice with 400 μl of filtered synthetic complete (SC) medium to remove floating cells and eventually resuspended in SC medium for imaging. Microscopy was performed at 30°C on a Nikon Ti2-Eclipse microscope equipped with a CFI Apochromat TIRF 100×/1.49 numerical aperture (NA) oil objective and a TwinCam LS dual-camera splitter attached to two Photometrics Prime 95B 25-mm cameras. The dual-camera setup enabled simultaneous imaging of red and green fluorophores. Specifications of filters and dichroics can be made available upon request. For time-lapse microscopy, cells were imaged every 2 or 7 min for indicated total periods of time. For quantification of the imaging presented in Fig. 1D, zygotes were only considered when patchy structures were apparent for GFP and mRuby3 signals. Cells harboring the TetO repeats often showed a diffuse red staining of mitochondria, which is likely explained by recombination of the TetO repeats.

### Image processing and analysis

All images, except those obtained by structured illumination microscopy, were postprocessed by deconvolution with the Huygens software (Scientific Volume Imaging). Fluorescent channels acquired simultaneously on two different cameras were aligned using a custom-built Python script. Alignment parameters were obtained from simultaneous imaging in bright-field mode. For quantification of Atp6-NG levels presented in Fig. 3A, mitochondria were first segmented in 3D based on the matrix-mKate2 signal using the mitograph software (*44*). Binary masks for parental and daughter cells were manually created in Fiji by drawing outlines on the bright-field image. Masks were used in a next step to assign coordinates of the mitochondrial network to the respective cells. Then, Atp6-NG intensities were determined and summed up along the mitochondrial network of parental or daughter cells using custom-built Python scripts. To account for differences in the mitochondrial amount in different cells, the Atp6-NG signal was normalized to the mitochondrial network length present in the respective cell.

Quantification of colocalization of Atp6-NG and Atp6-mKate2, presented in fig. S7D, was performed as follows. The mitochondrial network was first segmented in 3D based on the matrix-TagBFP signal using the mitograph software. Manually created binary masks were used to assign parts of the mitochondrial network to parental or daughter cells. Fluorescent intensities of Atp6-NG and Atp6-mKate2 were then determined for pixels along the mitochondrial network. Next, Manders and Pearson correlation coefficients were determined between both signals. For determination of the Manders correlation coefficient, signals along the mitochondrial network were thresholded beforehand with Yen *et al.*'s method (*45*).

### Genetic manipulation of mtDNA

Strains harboring mtDNA in which the *COB* gene was replaced with an arginine marker and a nonrecombinable TetO array were generated as follows. First, a synthesized TetO array, in which 21 TetO repeats are separated by spacers of varying length and sequence, was inserted after the stop codon of the *ARG8* gene in the plasmid pCOB-ST5 (*8*). This cloning step resulted in the plasmid pCO307, which thus contains an insert in which sequences homologous to the up- and downstream regions of the *COB* gene flank the ORF of *ARG8* followed by the TetO array. pCO307 was introduced into the *kar1-1* strain αDFS160 $\rho^0$ by biolistic transformation with the PDS-1000/He particle delivery system (Bio-Rad Laboratories), and transformants were selected by their ability to rescue the *cox2-62* mutation of the strain NB40-3C (*46*). pCO307 was then cytoducted into a Δ*arg8* W303 WT strain, which resulted in the deletion of the *COB* gene by *ARG8*-TetO through homologous recombination. Cells containing the Δ*cob::ARG8-TetO*-mtDNA were selected on the basis of their arginine prototrophy. Last, a construct (pCO407) consisting of the Cup1 promoter driving expression of an ORF in which the Su9 mitochondrial targeting sequence was fused to the TetR gene, which, in turn, was followed by three copies of the red fluorescent protein mRuby3, was chromosomally integrated into the HO locus (strain yCO460).

Strains harboring mtDNA in which the *ATP6* is tagged with either NG or mKate2 were generated as follows. First, synthesized gene fragments encoding NG and mKate2 compatible with the mitochondrial genetic code (Twist Bioscience, San Francisco) were fused to the C terminus of the coding region of *ATP6*, which was amplified from genomic DNA. *ATP6*-NG or *ATP6*-mKate2 was flanked with an 806-bp region homologous to the upstream region of *ATP6* and a 60-bp region homologous to the downstream region of *ATP6*. The entire fragments were then cloned into the Xho I site of the plasmid pPT24 (*47*), which resulted in either plasmid pCO444 or pCJ013 for ATP6-NG and ATP6-mKate2, respectively. pCO444 and pCJ013 were introduced into the *kar1-1* strain αDFS160 $\rho^0$ by biolistic transformation, and transformants were selected by their ability to rescue the cox2 mutation of NB40-3C (*46*). pCO444 and pCJ013 were then cytoducted into the strain MR10 (*48*). Colonies in which *ATP6*-NG (yCJ043) or *ATP6*-mKate2 (yCJ120) had successfully integrated into the *ATP6* locus were identified by their ability to grow on a nonfermentable carbon source. A strain containing *ATP6-NG*-mtDNA

Downloaded from https://www.science.org on January 26, 2022

and mating type *alpha* was obtained by mating yCJ043 with a $\rho^0$ W303 WT strain followed by sporulation and tetrad dissection (yCJ084).

Primer sequences used for cloning and plasmid maps can be made available upon request.

## DAPI staining and structured illumination microscopy

Coverslips (no. 1.5) were coated with 50 µl of concanavalin A (0.5 µg/ml) and air-dried for 1 hour. Residual concanavalin A was removed, and coverslips were dried for 45 min before fixation. $OD_{600}$ = 0.5 of log phase yeast cells were harvested at 5000 rpm for 3 min and washed using 1× PBS + 0.02% Tween 20. Cells were resuspended in 20 µl of 1× PBS + 0.02% Tween 20. Subsequently, all cells were placed on the concanavalin A–coated coverslips and incubated for 45 min to let cells settle on the coverslip. Then, cells were washed for 2 min with filtered SC medium. Cells were fixed using a 4% formaldehyde solution + DAPI (1 µg/ml) in filtered SC medium for 30 min. Cells were washed once more for 2 min using filtered SC medium. Twenty microliters of MOVIOL 4-88 (Roth) was added to the coverslips, and microscope slides were put onto the coverslips. Fixed immobilized cells were used for super-resolution imaging by structured illumination microscopy. The acquisition was performed on a 3D SIM DeltaVision OMX V3 microscope (General Electric) equipped with a 10× 1.4 NA oil immersion objective UPlanSApo (Olympus), 405-, 488-, and 593-nm diode lasers, and Cascade II electron-multiplying charge-coupled device (CCD) cameras (Photometrics). After acquisition with an appropriate refractive index oil, raw data were first reconstructed and corrected for color shifts using the provided software softWoRx 6.0 Beta 19 (unreleased). In a second step, a custom-made macro in Fiji (*49*) finalized the channel alignment and established composite TIFF (tag image file format) stacks, which were used for image analysis.

## Electron microscopy

For electron microscopy, all strains were pregrown in YPG medium to select for functional mtDNA. Subsequently, cells were grown to log phase in YPD or rich medium containing galactose medium, as indicated. Sample preparation for electron microscopy was essentially performed as previously described (*50*) with two minor changes: The fixation with glutaraldehyde was performed for 1 hour, and all centrifugation steps were carried out at 1610*g* for 5 min. Ultrathin sectioning was performed using a Leica Ultracut UCT (Leica Microsystems, Wetzlar, Germany) ultramicrotome and an ultra 35° diamond knife (Diatome, Nidau, Switzerland). Ultrathin 50- to 70-nm sections were placed on Pioloform-coated copper slot grids (Plano, Wetzlar, Germany) and poststained for 15 min with uranyl acetate and 3 min with lead citrate, as previously described (*50*). Electron microscopy was performed using a JEOL JEM-1400 Plus transmission electron microscope (JEOL, Tokyo, Japan) operated at 80 kV. Images were taken with a JEOL Ruby CCD camera (3296 × 2472 pixels) and the TEM Center software Ver.1.7.12.1984 (JEOL, Tokyo, Japan).

## Miscellaneous

Western blot analyses were performed with isolated mitochondria. Fifty micrograms of isolated mitochondria was preheated on 95°C for 5 min in 1× SDS loading buffer and separated on a 12% SDS gel. After transfer to polyvinylidene difluoride membranes, membranes were incubated with the following primary antibodies in 5% milk and tris-buffered saline: mouse anti-NG (1:1000; Chromotek GmbH), rabbit anti-aconitase1 (1:1000), and rabbit anti-Atp6 (1:1000). Quantitative real-time PCR experiments to determine mtDNA levels were performed as described previously (*51*). For the petite analysis, cells were grown overnight at 30°C, then freshly diluted to an $OD_{600}$ = 0.2, and grown for another 3 hours. A total of 200 cells were plated onto YPG plates containing 0.1% glucose. Plates were incubated for 3 days at 30°C. Only cells proficient in respiratory growth are able to continue growth after all glucose has been consumed.

## SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at https://science.org/doi/10.1126/sciadv.abi8886

View/request a protocol for this paper from *Bio-protocol*.

## REFERENCES AND NOTES

1. G. S. Gorman, P. F. Chinnery, S. DiMauro, M. Hirano, Y. Koga, R. McFarland, A. Suomalainen, D. R. Thorburn, M. Zeviani, D. M. Turnbull, Mitochondrial diseases. *Nat. Rev. Dis. Primers* **2**, 16080 (2016).
2. W. Fan, K. G. Waymire, N. Narula, P. Li, C. Rocher, P. E. Coskun, M. A. Vannan, J. Narula, G. R. Mac Gregor, D. C. Wallace, A mouse model of mitochondrial disease reveals germline selection against severe mtDNA mutations. *Science* **319**, 958–962 (2008).
3. J. H. Hill, Z. Chen, H. Xu, Selective propagation of functional mitochondrial DNA during oogenesis restricts the transmission of a deleterious mitochondrial variant. *Nat. Genet.* **46**, 389–392 (2014).
4. J. B. Stewart, C. Freyer, J. L. Elson, A. Wredenberg, Z. Cansu, A. Trifunovic, N.-G. Larsson, Strong purifying selection in transmission of mammalian mitochondrial DNA. *PLOS Biol.* **6**, e10 (2008).
5. T. Lieber, S. P. Jeedigunta, J. M. Palozzi, R. Lehmann, T. R. Hurd, Mitochondrial fragmentation drives selective removal of deleterious mtDNA in the germline. *Nature* **570**, 380–384 (2019).
6. Z. Chen, Z.-H. Wang, G. Zhang, C. K. E. Bleck, D. J. Chung, G. P. Madison, E. Lindberg, C. Combs, R. S. Balaban, H. Xu, Mitochondrial DNA segregation and replication restrict the transmission of detrimental mutation. *J. Cell Biol.* **219**, e201905160 (2020).
7. Y. Zhang, Z. H. Wang, Y. Liu, Y. Chen, N. Sun, M. Gucek, F. Zhang, H. Xu, PINK1 inhibits local protein synthesis to limit transmission of deleterious mitochondrial DNA mutations. *Mol. Cell* **73**, 1127–1137.e5 (2019).
8. S. Gruschke, K. Römpler, M. Hildenbeutel, K. Kehrein, I. Kühl, N. Bonnefoy, M. Ott, The Cbp3–Cbp6 complex coordinates cytochrome ***b*** synthesis with $bc_1$ complex assembly in yeast mitochondria. *J. Cell Biol.* **193**, 1101 (2011).
9. P. S. Philip, C. William Birky, R. L. Strausberg, Segregation of mitochondrial markers in yeast. *Methods Enzymol.* **56**, 139–154 (1979).
10. C. Osman, T. R. Noriega, V. Okreglak, J. C. Fung, P. Walter, Integrity of the yeast mitochondrial genome, but not its distribution and inheritance, relies on mitochondrial fission and fusion. *Proc. Natl. Acad. Sci. U.S.A.* **112**, E947–E956 (2015).
11. J. Nunnari, W. F. Marshall, A. Straight, A. Murray, J. W. Sedat, P. Walter, Mitochondrial transmission during mating in *Saccharomyces cerevisiae* is determined by mitochondrial fusion and fission and the intramitochondrial segregation of mitochondrial DNA. *Mol. Biol. Cell* **8**, 1233–1242 (1997).
12. K. Okamoto, P. S. Perlman, R. A. Butow, The sorting of mitochondrial DNA and mitochondrial proteins in zygotes: Preferential transmission of mitochondrial DNA to the medial bud. *J. Cell Biol.* **142**, 613–623 (1998).
13. R. Azpiroz, R. A. Butow, Patterns of mitochondrial sorting in yeast zygotes. *Mol. Biol. Cell* **4**, 21–36 (1993).
14. K. Okamoto, N. Kondo-Okamoto, Y. Ohsumi, Mitochondria-anchored receptor Atg32 mediates degradation of mitochondria via selective autophagy. *Dev. Cell* **17**, 87–97 (2009).
15. T. Kanki, K. Wang, Y. Cao, M. Baba, D. J. Klionsky, Atg32 is a mitochondrial protein that confers selectivity during mitophagy. *Dev. Cell* **17**, 98–109 (2009).
16. L. Jimenez, D. Laporte, S. Duvezin-Caubet, F. Courtout, I. Sagot, Mitochondrial ATP synthases cluster as discrete domains that reorganize with the cellular demand for oxidative phosphorylation. *J. Cell Sci.* **127**, –719, 726 (2014).
17. T. Stephan, A. Roesch, D. Riedel, S. Jakobs, Live-cell STED nanoscopy of mitochondrial cristae. *Sci. Rep.* **9**, 12419 (2019).
18. F. Vogel, C. Bornhövd, W. Neupert, A. S. Reichert, Dynamic subcompartmentalization of the mitochondrial inner membrane. *J. Cell Biol.* **175**, 237–247 (2006).
19. S. Hoppins, S. R. Collins, A. Cassidy-Stone, E. Hummel, R. M. Devay, L. L. Lackner, B. Westermann, M. Schuldiner, J. S. Weissman, J. Nunnari, A mitochondrial-focused genetic interaction map reveals a scaffold-like complex required for inner membrane organization in mitochondria. *J. Cell Biol.* **195**, 323–340 (2011).

Downloaded from https://www.science.org on January 26, 2022

20. K. von der Malsburg, J. M. Müller, M. Bohnert, S. Oeljeklaus, P. Kwiatkowska, T. Becker, A. Loniewska-Lwowska, S. Wiese, S. Rao, D. Milenkovic, D. P. Hutu, R. M. Zerbes, A. Schulze-Specking, H. E. Meyer, J.-C. Martinou, S. Rospert, P. Rehling, C. Meisinger, M. Veenhuis, B. Warscheid, I. J. van der Klei, N. Pfanner, A. Chacinska, M. van der Laan, Dual role of mitofilin in mitochondrial membrane organization and protein biogenesis. *Dev. Cell* **21**, 694–707 (2011).
21. M. Harner, C. Körner, D. Walther, D. Mokranjac, J. Kaesmacher, U. Welsch, J. Griffith, M. Mann, F. Reggiori, W. Neupert, The mitochondrial contact site complex, a determinant of mitochondrial architecture. *EMBO J.* **30**, 4356–4370 (2011).
22. K. M. Davies, C. Anselmi, I. Wittig, J. D. Faraldo-Gomez, W. Kuhlbrandt, Structure of the yeast F1Fo-ATP synthase dimer and its role in shaping the mitochondrial cristae. *Proc. Natl. Acad. Sci. U.S.A.* **109**, 13602–13607 (2012).
23. P. Paumard, J. Vaillier, B. Coulary, J. Schaeffer, V. Soubannier, D. M. Mueller, D. Brèthes, J. P. di Rago, J. Velours, The ATP synthase is involved in generating mitochondrial cristae morphology. *EMBO J.* **21**, 221–230 (2002).
24. M. E. Harner, A.-K. Unger, W. J. Geerts, M. Mari, T. Izawa, M. Stenger, S. Geimer, F. Reggiori, B. Westermann, W. Neupert, An evidence based hypothesis on the existence of two pathways of mitochondrial crista formation. *eLife* **5**, e18853 (2016).
25. R. Rabl, V. Soubannier, R. Scholz, F. Vogel, N. Mendl, A. Vasiljev-Neumeyer, C. Körner, R. Jagasia, T. Keil, W. Baumeister, M. Cyrklaff, W. Neupert, A. S. Reichert, Formation of cristae and crista junctions in mitochondria depends on antagonism between Fcj1 and Su e/g. *J. Cell Biol.* **185**, 1047–1063 (2009).
26. H. Rampelt, M. Bohnert, R. M. Zerbes, S. E. Horvath, B. Warscheid, N. Pfanner, M. van der Laan, Mic10, a core subunit of the mitochondrial contact site and cristae organizing system, interacts with the dimeric F 1 F o-ATP synthase. *J. Mol. Biol.* **429**, 1162–1170 (2017).
27. J. Velours, A. Dautant, B. Salin, I. Sagot, D. Brèthes, Mitochondrial F1F0-ATP synthase and organellar internal architecture. *Int. J. Biochem. Cell Biol.* **41**, 1783–1789 (2009).
28. V. Wilkens, W. Kohl, K. Busch, Restricted diffusion of OXPHOS complexes in dynamic mitochondria delays their exchange between cristae and engenders a transitory mosaic distribution. *J. Cell Sci.* **126**, 103–116 (2013).
29. T. Appelhans, C. P. Richter, V. Wilkens, S. T. Hess, J. Piehler, K. B. Busch, Nanoscale organization of mitochondrial microcompartments revealed by combining tracking and localization microscopy. *Nano Lett.* **12**, 610–616 (2012).
30. H. Schägger, K. Pfeiffer, Supercomplexes in the respiratory chains of yeast and mammalian mitochondria. *EMBO J.* **19**, 1777–1783 (2000).
31. R. W. Gilkerson, J. M. Selker, R. A. Capaldi, The cristal membrane of mitochondria is the principal site of oxidative phosphorylation. *FEBS Lett.* **546**, 355–358 (2003).
32. A. Kowald, T. B. L. Kirkwood, Evolution of the mitochondrial fusion-fission cycle and its role in aging. *Proc. Natl. Acad. Sci. U.S.A.* **108**, 10237–10242 (2011).
33. K. B. Busch, A. Kowald, J. N. Spelbrink, Quality matters: How does mitochondrial network dynamics and quality control impact on mtDNA integrity? *Philos. Trans. R. Soc. B* **369**, 20130442 (2014).
34. D. M. Wolf, M. Segawa, A. K. Kondadi, R. Anand, S. T. Bailey, A. S. Reichert, A. M. van der Bliek, D. B. Shackelford, M. Liesa, O. S. Shirihai, Individual cristae within the same mitochondrion display different membrane potentials and are functionally independent. *EMBO J.* **38**, e101056 (2019).
35. K. Kehrein, R. Schilling, B. V. Möller-Hergt, C. A. Wurm, S. Jakobs, T. Lamkemeyer, T. Langer, M. Ott, Organization of mitochondrial gene expression in two distinct ribosome-containing assemblies. *Cell Rep.* **10**, 843–853 (2015).
36. Y. Itoh, J. Andréll, A. Choi, U. Richter, P. Maiti, R. B. Best, A. Barrientos, B. J. Battersby, A. Amunts, Mechanism of membrane-tethered mitochondrial protein synthesis. *Science* **371**, 846–849 (2021).
37. G. Szyrach, M. Ott, N. Bonnefoy, W. Neupert, J. M. Herrmann, Ribosome binding to the Oxa1 complex facilitates co-translational protein insertion in mitochondria. *EMBO J.* **22**, 6448–6457 (2003).
38. S. Stoldt, D. Wenzel, K. Kehrein, D. Riedel, M. Ott, S. Jakobs, Spatial orchestration of mitochondrial translation and OXPHOS complex assembly. *Nat. Cell Biol.* **20**, 528–534 (2018).
39. S. M. Jin, M. Lazarou, C. Wang, L. A. Kane, D. P. Narendra, R. J. Youle, Mitochondrial membrane potential regulates PINK1 import and proteolytic destabilization by PARL. *J. Cell Biol.* **191**, 933–942 (2010).
40. J. R. McFaline-Figueroa, J. Vevea, T. C. Swayne, C. Zhou, C. Liu, G. Leung, I. R. Boldogh, L. A. Pon, Mitochondrial quality control during inheritance is associated with lifespan and mother-daughter age asymmetry in budding yeast. *Aging Cell* **10**, 885–895 (2011).
41. T. C. Medeiros, R. L. Thomas, R. Ghillebert, M. Graef, Autophagy balances mtDNA synthesis and degradation by DNA polymerase POLG during starvation. *J. Cell Biol.* **217**, 1601–1611 (2018).
42. C. Janke, M. M. Magiera, N. Rathfelder, C. Taxis, S. Reber, H. Maekawa, A. Moreno-Borchart, G. Doenges, E. Schwob, E. Schiebel, M. Knop, A versatile toolbox for PCR-based tagging of yeast genes: New fluorescent proteins, more markers and promoter substitution cassettes. *Yeast* **21**, 947–962 (2004).
43. J. A. Nickoloff, J. D. Singer, M. F. Hoekstra, F. Heffron, Double-strand breaks stimulate alternative mechanisms of recombination repair. *J. Mol. Biol.* **207**, 527–541 (1989).
44. S. M. Rafelski, M. P. Viana, Y. Zhang, Y.-H. M. Chan, K. S. Thorn, P. Yam, J. C. Fung, H. Li, L. da F Costa, W. F. Marshall, Mitochondrial network size scaling in budding yeast. *Science* **338**, 822–824 (2012).
45. J. C. Yen, F. J. Chang, S. Chang, A new criterion for automatic multilevel thresholding. *IEEE Trans. Image Process.* **4**, 370–378 (1995).
46. D. F. Steele, C. A. Butler, T. D. Fox, Expression of a recoded nuclear gene inserted into yeast mitochondrial DNA is limited by mRNA-specific translational activation. *Proc. Natl. Acad. Sci. U.S.A.* **93**, 5253–5257 (1996).
47. P. E. Thorsness, T. D. Fox, Nuclear mutations in *Saccharomyces cerevisiae* that affect the escape of DNA from mitochondria to the nucleus. *Genetics* **134**, 21–28 (1993).
48. M. Rak, E. Tetaud, F. Godard, I. Sagot, B. Salin, S. Duvezin-Caubet, P. P. Slonimski, J. Rytka, J.-P. di Rago, Yeast cells lacking the mitochondrial gene encoding the ATP synthase subunit 6 exhibit a selective loss of complex IV and unusual mitochondrial morphology. *J. Biol. Chem.* **282**, 10853–10864 (2007).
49. J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, A. Cardona, Fiji: An open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
50. A. K. Unger, S. Geimer, M. Harner, W. Neupert, B. Westermann, Analysis of yeast mitochondria by electron microscopy, in *Methods in Molecular Biology* (Humana Press Inc., 2017), vol. 1567, pp. 293–314.
51. A. Göke, S. Schrott, A. Mizrak, V. Belyy, C. Osman, P. Walter, Mrx6 regulates mitochondrial DNA copy number in *Saccharomyces cerevisiae* by engaging the evolutionarily conserved Lon protease Pim1. *Mol. Biol. Cell* **31**, 511–545 (2019).

**Acknowledgments:** We thank T. Kautzleben and N. Lebedeva for technical assistance, L. Heiderscheid and M. Kroker for help with the experimental work, and S. Geimer for help with electron microscopy. OMX microscopy was performed at the Center for Advanced Light Microscopy (CALM). We thank M. Schuldiner and members of the Osman laboratory for critically reading the manuscript. We are grateful for stimulating discussions within the "Mito-Club" throughout the course of this project. We thank Martin Ott and Jean Velours for providing reagents. **Funding:** C.O., C.J., R.R., and F.T. are supported by a grant from the European Research Council (ERCStG-714739 IlluMitoDNA). T.K. is supported by the Elitenetzwerk Bayern through the" Biological Physics" program and by a grant from the Deutsche Forschungsgemeinschaft (DFG, project number 459304237). A.M., D.B., D.H., and H.L. are supported by the SPP 2202 Priority Program of the Deutsche Forschungsgemeinschaft (DFG, project number 422857584). **Author contributions:** C.O. and C.J. designed the project. C.O., C.J., R.R., A.M., F.T., C.S., and T.K. performed experiments and/or analyzed experimental data. D.B. and D.H. developed a software to automatically detect yeast zygotes in microscopic images. C.O., C.J., R.R., and T.K. wrote the manuscript, with contributions from all coauthors. H.L. supervised super-resolution microscopy experiments and development of zygote detection software. P.W. supervised early stages of the project. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials.

Submitted 7 April 2021
Accepted 8 July 2021
Published 1 September 2021
10.1126/sciadv.abi8886

**Citation:** C. Jakubke, R. Roussou, A. Maiser, C. Schug, F. Thoma, D. Bunk, D. Hörl, H. Leonhardt, P. Walter, T. Klecker, C. Osman, Cristae-dependent quality control of the mitochondrial genome. *Sci. Adv.* **7**, eabi8886 (2021).

Downloaded from https://www.science.org on January 26, 2022

Science Advances

# Cristae-dependent quality control of the mitochondrial genome

Christopher JakubkeRodaria RoussouAndreas MaiserChristina SchugFelix ThomaDavid BunkDavid HörlHeinrich LeonhardtPeter WalterTill KleckerChristof Osman

*Sci. Adv.*, 7 (36), eabi8886. • DOI: 10.1126/sciadv.abi8886

**View the article online**
https://www.science.org/doi/10.1126/sciadv.abi8886
**Permissions**
https://www.science.org/help/reprints-and-permissions

Downloaded from https://www.science.org on January 26, 2022

Use of think article is subject to the Terms of service

---

*Science Advances* (ISSN ) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Advances* is a registered trademark of AAAS.

Copyright © 2021 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution NonCommercial License 4.0 (CC BY-NC).

### 2.2.2 Supporting Information

Science Advances
AAAS

# Supplementary Materials for

## Cristae-dependent quality control of the mitochondrial genome

Christopher Jakubke, Rodaria Roussou, Andreas Maiser, Christina Schug, Felix Thoma, David Bunk, David Hörl,
Heinrich Leonhardt, Peter Walter, Till Klecker, Christof Osman*

*Corresponding author. Email: osman@bio.lmu.de

Published 1 September 2021, *Sci. Adv.* **7**, eabi8886 (2021)
DOI: 10.1126/sciadv.abi8886

**The PDF file includes:**

Figs. S1 to S9
Tables S1 to S3
Legends for movies S1 to S6

**Other Supplementary Material for this manuscript includes the following:**

Movies S1 to S6

# Supplementary materials

Figure legends S1 to S9

Movie legends S1 to S6

Table legends S1 to S3

Supplementary figure S1-S9

Tables S1 to S3

Movie S1 to S6

## Figure S1

**(A)** Quantitative PCR analysis of mtDNA levels of WT or Δ*cob::ARG8* strains, which were used in the pedigree analysis (Fig. 1C). No difference in mtDNA levels between both strains are apparent. The average value is derived from three biological replicates. The value for each biological replicate was derived from three technical replicates; Error bars indicate SD. n. s. - non significant, t-test.

**(B)** Carry-over cell material does not support growth upon restreaking. Replica plating during the pedigree analysis results in carry-over of cell material. While such cell material is easily distinguished from growing cells by direct visual inspection of plates, it is difficult to discern on photos. For this experiment, cell material was restreaked on selective plate to demonstrate the difference between carry-over cell material and cell growth. Asterisks indicate carry-over cell material.

**(C)** PCR analysis corroborates results from the growth-based pedigree assay. Genomic DNA was extracted directly from YPD grown colonies without further incubation from three pedigree lineages and remaining cell material was replica plated onto YPG or SC-ARG (lacking arginine) plates. Presence of the *COB* gene (present in WT mtDNA, which supports growth on YPG) or the *ARG8* gene (present in Δ*cob::ARG8* mtDNA, which supports growth on SC-Arg) was tested by PCR using specific oligos for the respective genes. PCR results correlate perfectly with the growth based analysis.

**(D)** Schematic illustration of the pedigree analysis, explaining the skipping of generations on a respective medium. YPG - rich medium containing the non-fermentable carbon source glycerol, SC-Arg - synthetic defined medium lacking arginine.

## Figure S2

**(A)** Pedigree analysis of WT cells. The experiment is virtually identical to the experiment presented in Fig. 1C with the difference that the starting strains for the pedigree analysis had switched mating types. Specifically, in this experiment the strain containing WT mtDNA had mating type *alpha*, whereas the strain containing Δ*cob::ARG8* had mating type *a*. Striped bars indicate percentage of heteroplasmic cells containing WT and Δ*cob::ARG8* mtDNA. Grey or red bars indicate percentage of homoplasmic cells containing WT or Δ*cob::ARG8* mtDNA, respectively.

**(B)** Pedigree analysis of cells containing *ARG8* inserted neutrally into mtDNA upstream of the *COX2* gene. We cannot distinguish between cells that are heteroplasmic for both mtDNA species and cells where *ARG8* had recombined into WT mtDNA. The shown result, however, indicates that *ARG8* does not confer a strong disadvantage on mtDNA, which could lead to its removal.

## Figure S3

**(A)** Quantitative PCR analysis of mtDNA levels of WT or Δ*cob::ARG8-TetO*-TetR-3xmRuby3 strains, which were used for the microscopy experiments presented in Fig. 1D. A slight increase in the mtDNA levels of the Δ*cob::ARG8-TetO*-TetR-3xmRuby3 is apparent. The average value is derived from three biological replicates. The value for each biological replicate was derived from three technical replicates; Error bars indicate SD. n. s. - non significant, t-test.

**(B)** Additional microscopic images of the inheritance of either LacO-marked intact or TetO-marked Δ*cob::ARG8-TetO* mtDNA. Images complement data shown in Fig. 1D.

Scale bar: D 10 $\mu$m.

## Figure S4

**(A and B)** Time-lapse microscopy of mating events between WT cells expressing either matrix-targeted NG or matrix-targeted mKate2. Mating events are shown, where either both cells contained WT mtDNA (A) or one cell contained WT mtDNA and the other contained $\Delta$*cob::ARG8* mtDNA (B). Selected time-frames for (B) are shown in Fig. 1E.

**(C and D)** Similar to (A), except that cells with deletions of the nuclear-encoded $\Delta$*dnm1* (C) or $\Delta$*atg32* (D) were used.

Scale bars: A-D 10 $\mu$m.

## Figure S5

**(A)** Schematic illustration of the NG-tagged *ATP6* within mtDNA.

**(B)** Quantitative PCR analysis of mtDNA levels of WT or strains expressing the Atp6-NG protein. The average value is derived from three biological replicates. The value for each biological replicate was derived from three technical replicates; Error bars indicate SD. n. s. - non significant, t-test.

**(C)** Petite frequency of the Atp6-NG strain in comparison to WT and the Δ*atp21* strains.

**(D)** Westernblot analysis of strains harbouring either WT or *ATP6-NG* mtDNA. Aconitase (Aco1) was used as a loading control.

**(E)** Widefield fluorescence microscopy images of diploid cells harbouring *ATP6*-NG mtDNA and expressing nuclear-encoded matrix-targeted mScarlet.
Scale bar: E 10 $\mu$m.

## Figure S6

**(A)** Time-lapse microscopy of mating events between WT cells harbouring *ATP6*-NG mtDNA and WT cells harbouring WT mtDNA and expressing nuclear-encoded matrix-targeted Kate2.

**(B)** Time-lapse microscopy of mating events between cells expressing NG-tagged Cox4 and cells expressing nuclear-encoded matrix-targeted mKate2.

**(C)** Growth analysis of strains expressing NG fused to Pam16 or the transmembrane domain of Fis1 in comparison to WT. Strains were used in the time-lapse experiment presented in Fig. 2, E and F and Fig. S4D and E.

**(D)** Westernblot analysis of strains harbouring the NG-tagged version of Pam16. Aconitase was used as a loading control. Note that no signal for free NG is detectable indicating that NG is not cleaved off.

**(E)** Time-lapse microscopy of mating events between WT cells expressing NG fused to the transmembrane domain of Fis1 and WT cells expressing nuclear-encoded matrix-targeted mKate2.

**(F)** Time-lapse microscopy of mating events between cells expressing NG-tagged Pam16 and cells expressing nuclear-encoded matrix-targeted mKate2.

Scale bars: A and D-E 10 $\mu$m.

# Figure S7

**(A)** Quantification of the equilibration of soluble Su9-mKate2 protein in mating experiments shown in Fig. 3A and B; ** $P<0.01$, t-test.

**(B)** Quantification of the Atp6-NG in the daughter cells of zygotes from mating experiments shown in Fig. 3A and B; * $P<0.05$, t-test.

**(C)** Spot test of strains expressing Atp6-NG or Atp6-mKate2 compared to the WT and a strain harbouring Δ*atp6* mtDNA.

**(D)** Quantification of the co-localization of Atp6-NG and Atp6-mKate2 in daughter cells derived from matings between WT, Δ*atp20* or Δ*atp20* cells, in which parental cells contained either *ATP6-NG* or *ATP6-mKate2* mtDNA. The Pearson (PCC) and Manders (MCC) correlation coefficients between Atp6-NG and Atp6-mKate2 signals along the mitochondrial network of daughter cells were determined for multiple cells in three independent experiments; * $P<0.05$, ** $P<0.01$, t-test.

## Figure S8

**(A)** Electron micrographs of mitochondria from WT, Δ*atp20*, Δ*atp21*, and Δ*atp20* Δ*atp21* cells grown in rich medium containing glucose.

**(B)** Quantification of cristae shape from cells grown in rich medium containing glucose. For each strain, mitochondria from 50 cells were scored for mitochondrial ultrastructure and grouped into the indicated categories. Shown is the mean ± standard deviation from three independent experiments. Examples of mitochondria with altered cristae shape are depicted on the right.

**(C)** Electron micrographs of mitochondria from Δ*atp20* Δ*atp21* cells grown in rich medium containing galactose. Shown are two representative cells. For each cell, the same mitochondrion was imaged in consecutive 70 nm ultrathin sections.

Scale bars: A and C 500 nm; B 200 nm.

## Figure S9

**(A)** Quantitative PCR analysis of mtDNA levels of WT, $\Delta$*atp20*, $\Delta$*atp21*, $\Delta$*atp20*$\Delta$*atp21*, $\Delta$*mic10*, or $\Delta$*mic60* strains containing WT or $\Delta$*cob* mtDNA. Strains were used for the pedigree analysis presented in Figure 4A-F. The average value is derived from three biological replicates. The value for each biological replicate was derived from three technical replicates; Error bars indicate SD. n. s. - non significant, t-test.

**(B)** Petite frequency of the indicated deletion strains.

## Movie S1

**Mitochondrial morphology during mating events of WT cells.** Mating events between two cells containing either WT or Δ*cob::ARG8* mtDNA. Cells expressed either matrix-targeted mKate2 (WT mtDNA, cyan) or NG (Δ*cob::ARG8* mtDNA, magenta). Mating events were monitored by live-cell microscopy. Brightfield, single fluorescent channels and a merge from both fluorescent channels are shown.

## Movie S2

**Mitochondrial morphology during mating events of Δ*dnm1* cells.** Mating events between two Δ*dnm1* cells containing either WT or Δ*cob::ARG8* mtDNA. Cells expressed either matrix-targeted mKate2 (WT mtDNA, cyan) or NG (Δ*cob::ARG8* mtDNA, magenta). Mating events were monitored by live-cell microscopy. Brightfield, single fluorescent channels and a merge from both fluorescent channels are shown. Images were taken in 7 min intervals.

## Movie S3

**Mitochondrial morphology during mating events of Δ*atg32* cells.** Mating events between two Δ*atg32* cells containing either WT or Δ*cob::ARG8* mtDNA. Cells expressed either matrix-targeted mKate2 (WT mtDNA, cyan) or NG (Δ*cob::ARG8* mtDNA, magenta). Mating events were monitored by live-cell microscopy. Brightfield, single fluorescent channels and a merge from both fluorescent channels are shown. Images were taken in 7 min intervals.

## Movie S4

**Diffusion of mitochondrial-encoded Atp6-NG throughout the mitochondrial network of zygotes.** Cells expressing matrix-targeted mKate2 (cyan) were mated with cells expressing mtDNA-encoded Atp6-NG (magenta) and were monitored by live cell imaging. Brightfield, single fluorescent channels and a merge from both fluorescent channels are shown. Images were taken in 7 min intervals.

## Movie S5

**Diffusion of Fis-NG throughout the mitochondrial network of zygotes.** Cells expressing matrix-targeted mKate2 (cyan) were mated with cells expressing nuclear-encoded Fis1-NG (magenta) and were monitored by live cell imaging. Brightfield, single fluorescent channels and a merge from both fluorescent channels are shown. Images were taken in 7 min intervals.

## Movie S6

**Diffusion of Pam16-NG throughout the mitochondrial network of zygotes.** Cells expressing matrix-targeted mKate2 (cyan) were mated with cells expressing nuclear-encoded Pam16-NG (magenta) and were monitored by live cell imaging. Brightfield, single fluorescent channels and a merge from both fluorescent channels are shown. Images were taken in 7 min intervals.

## Table S1

**Yeast strains used in this study.**

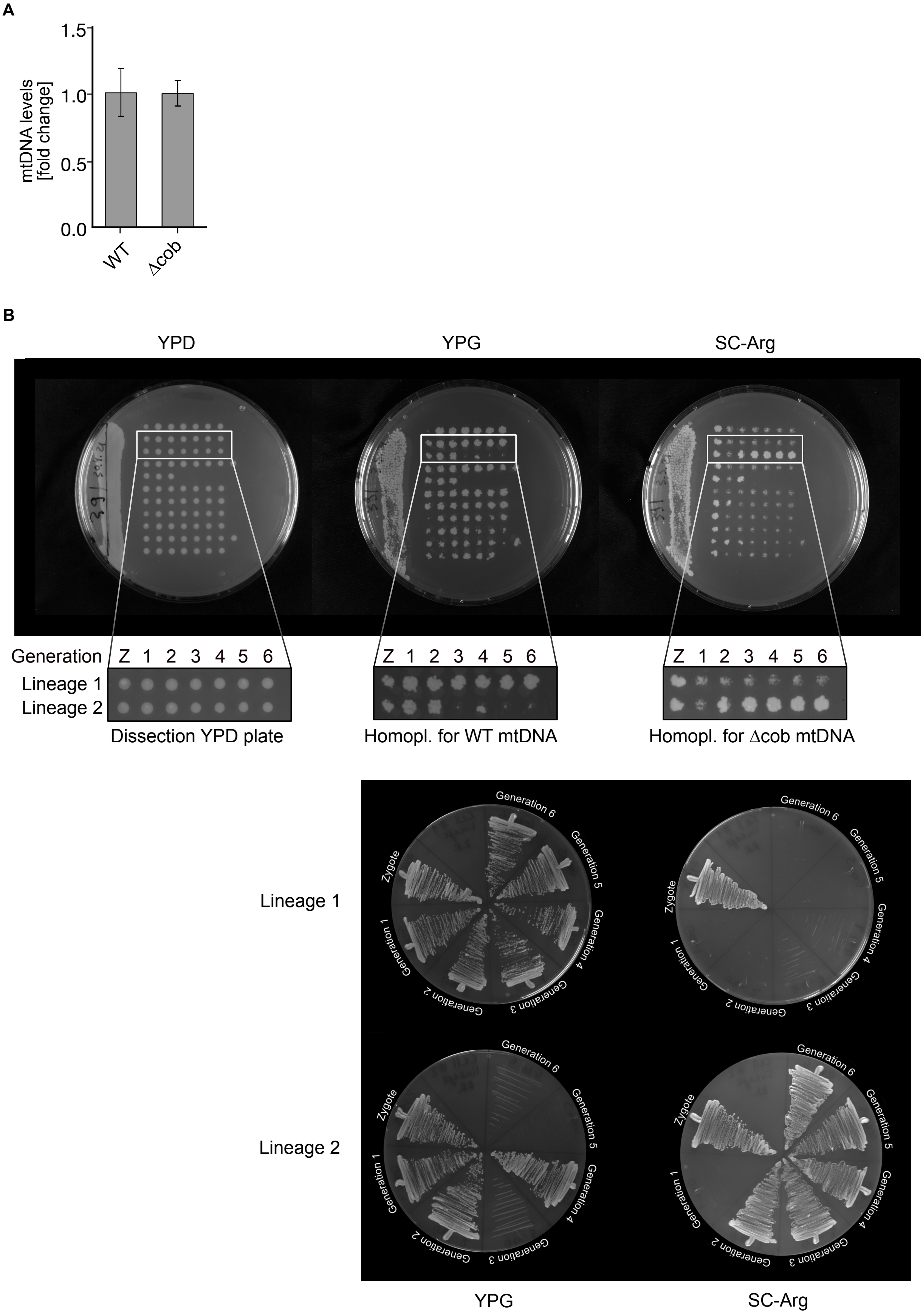
## Table S2

**Primers used in this study.**

## Table S3

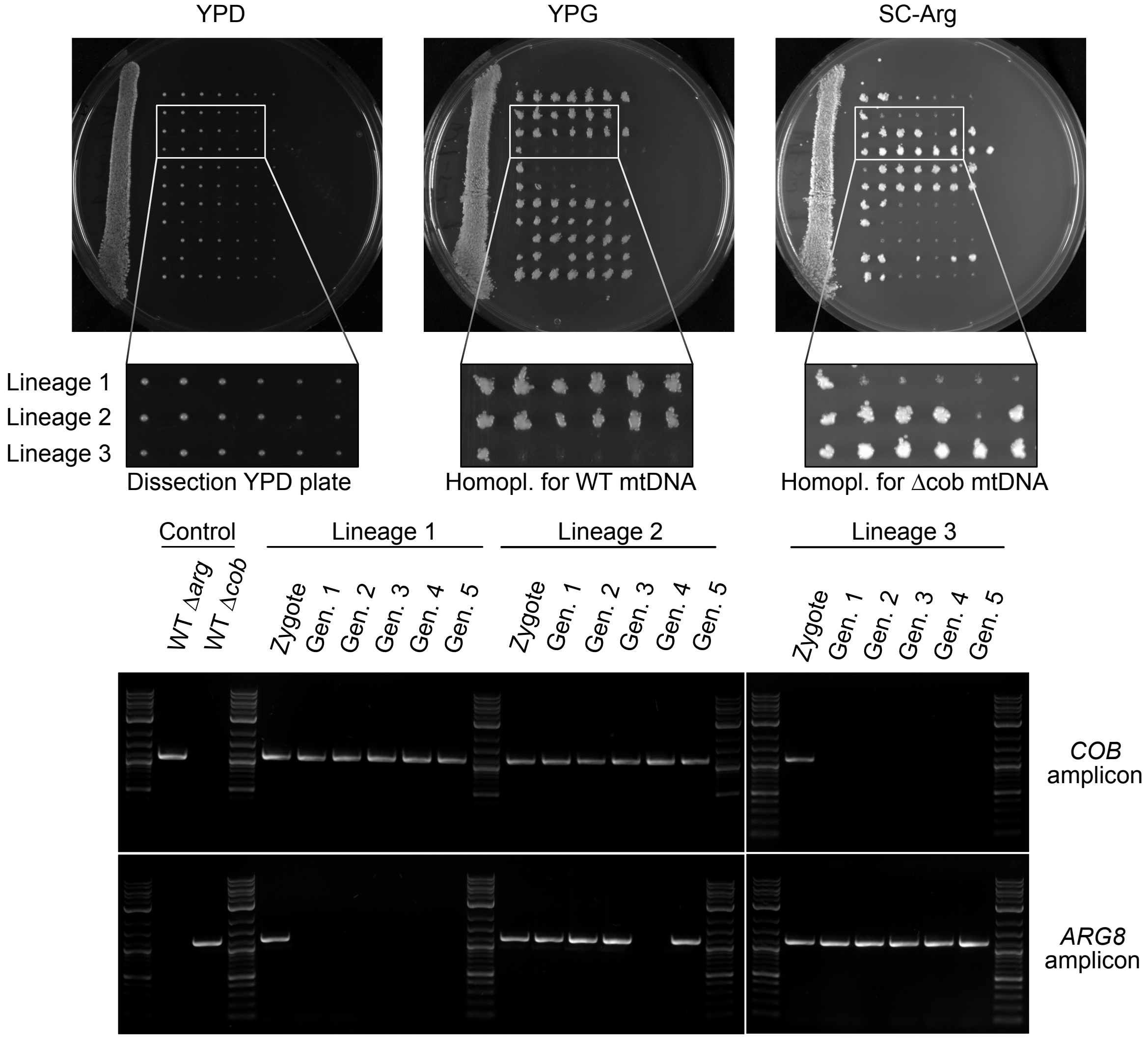
**Plasmids used in this study.**

# Supplemental Figure 1

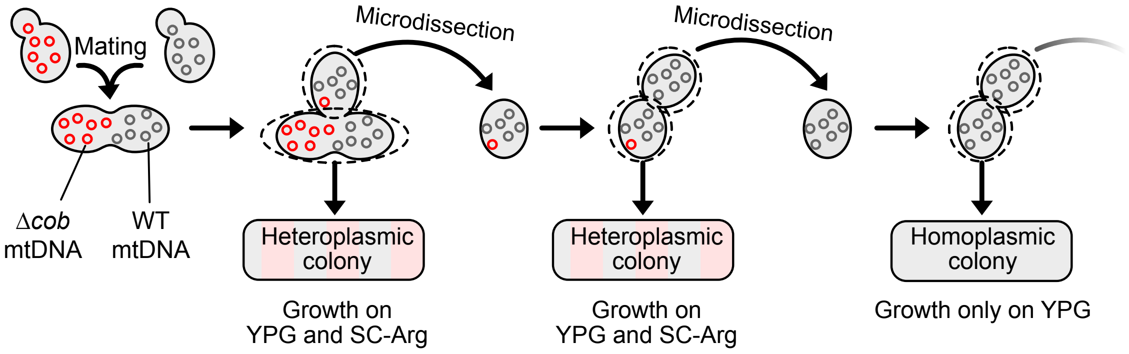


## Supplemental Figure 1

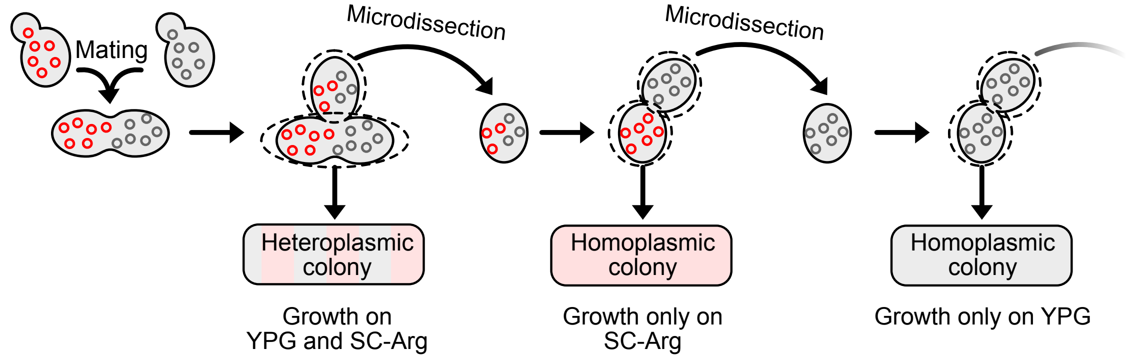
C



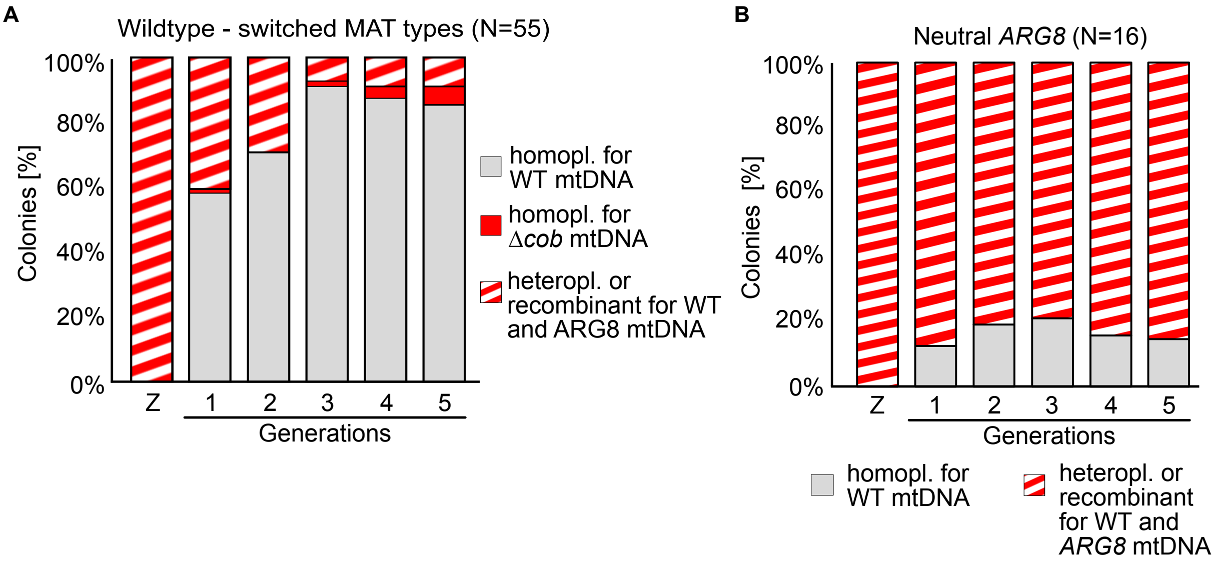
D **Pedigree analysis, most often observed pattern**



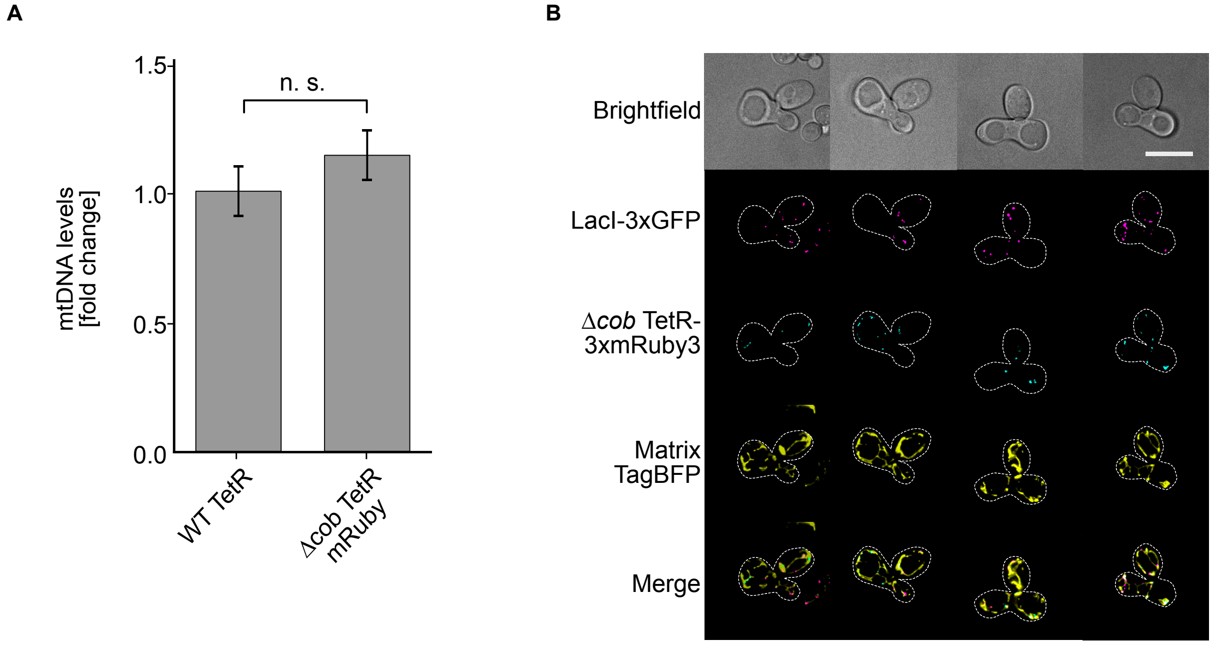
**Pedigree analysis, generation skipping**



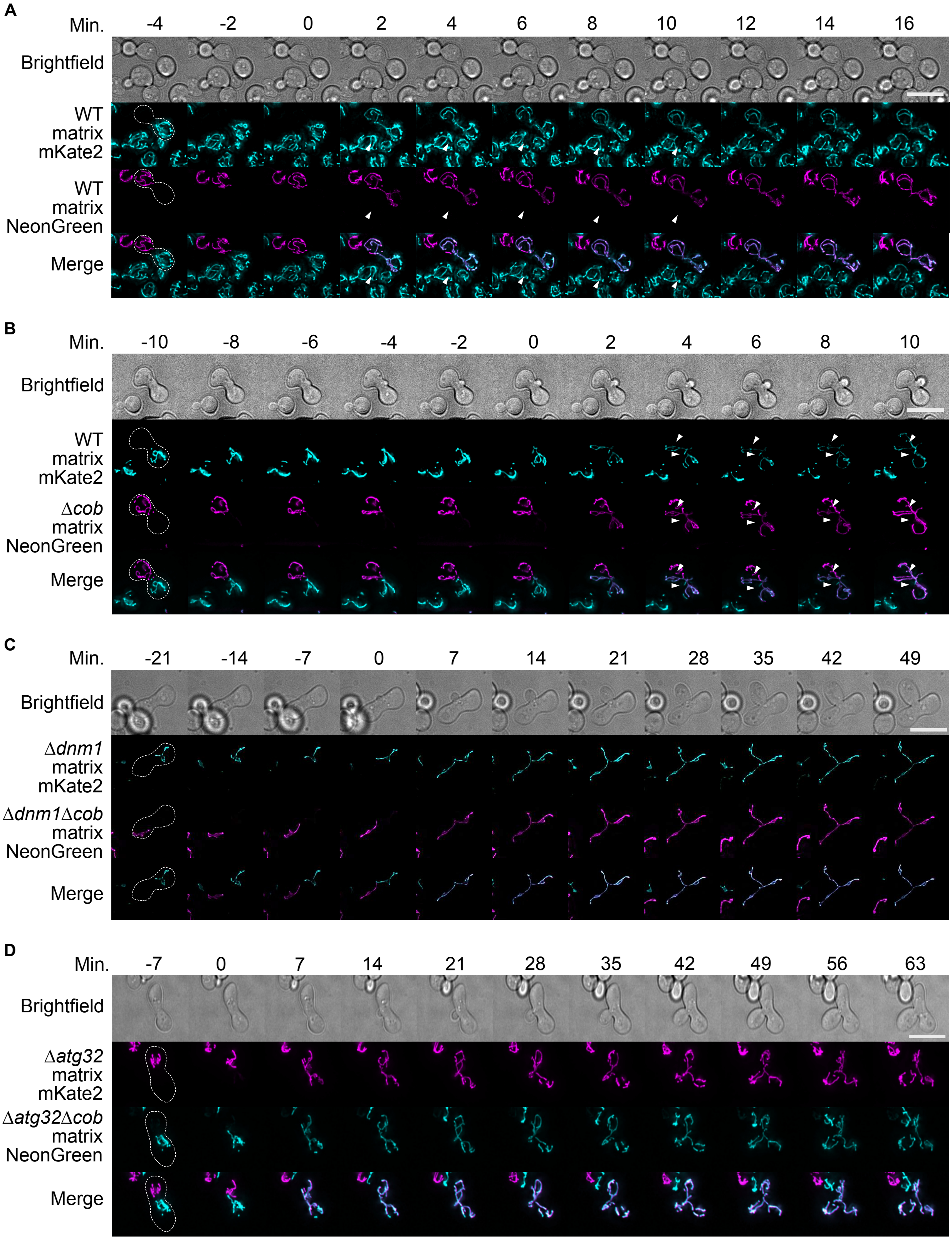
## Supplemental Figure 2



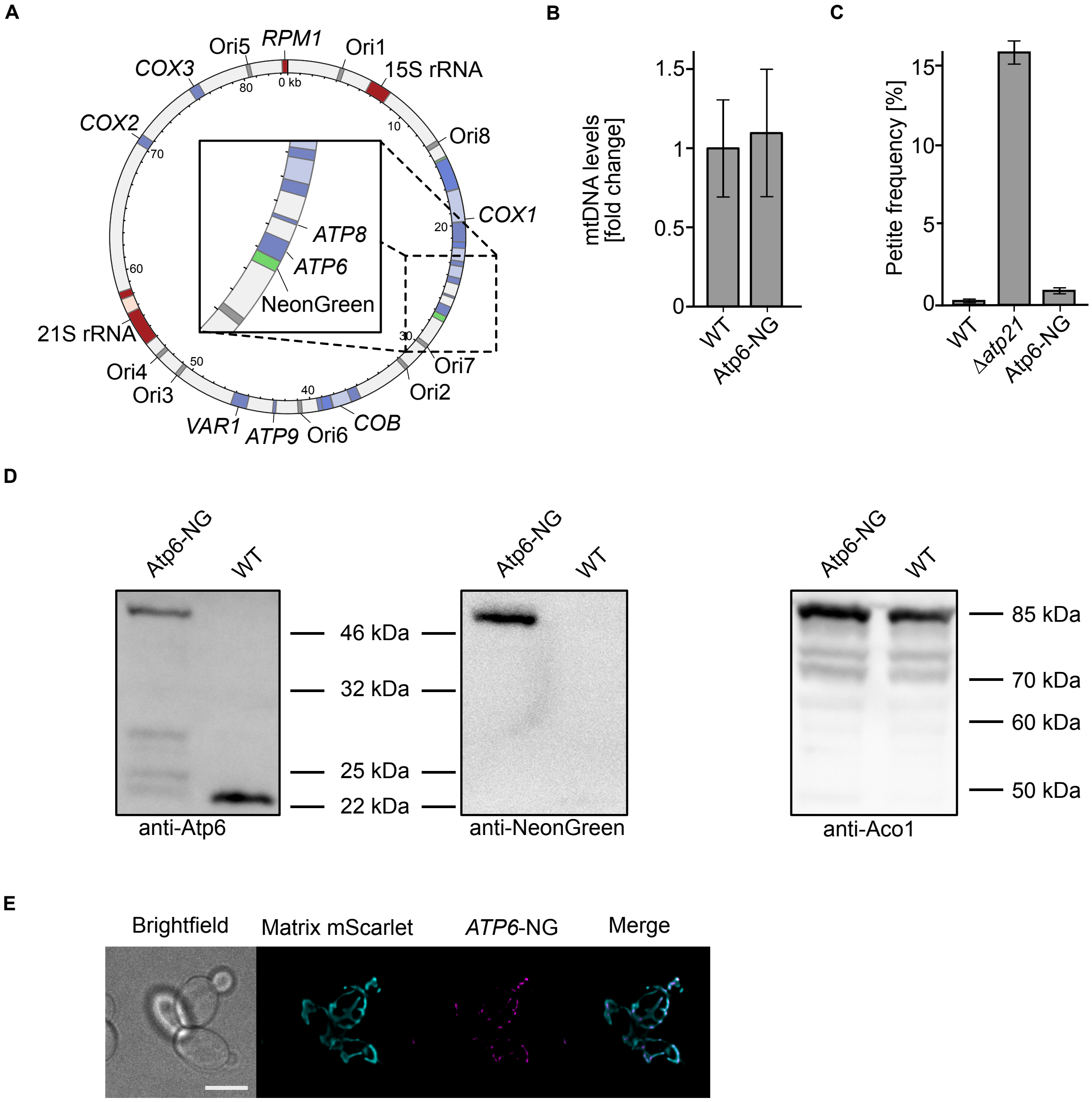
# Supplemental Figure 3



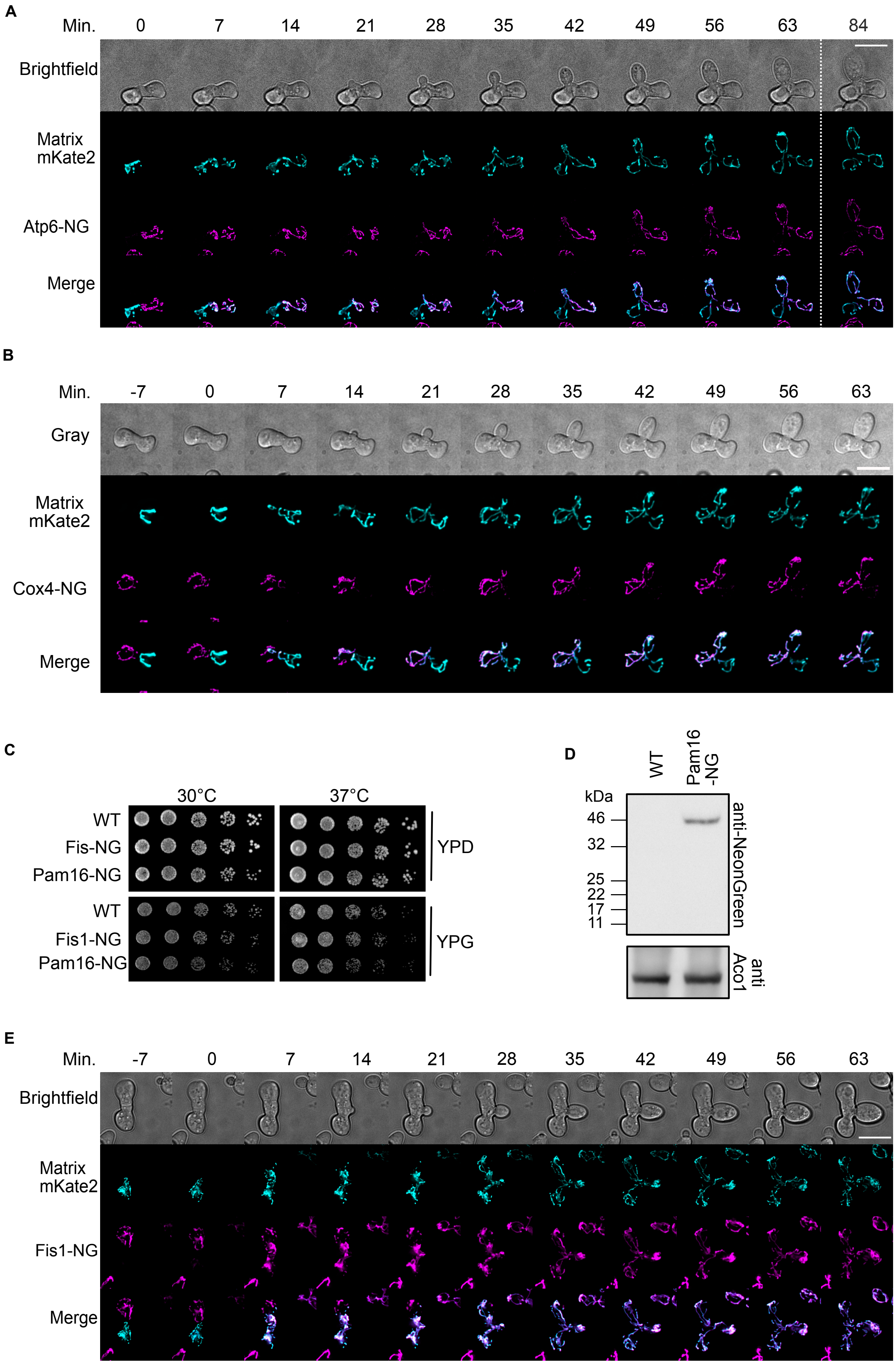
# Supplemental Figure 4



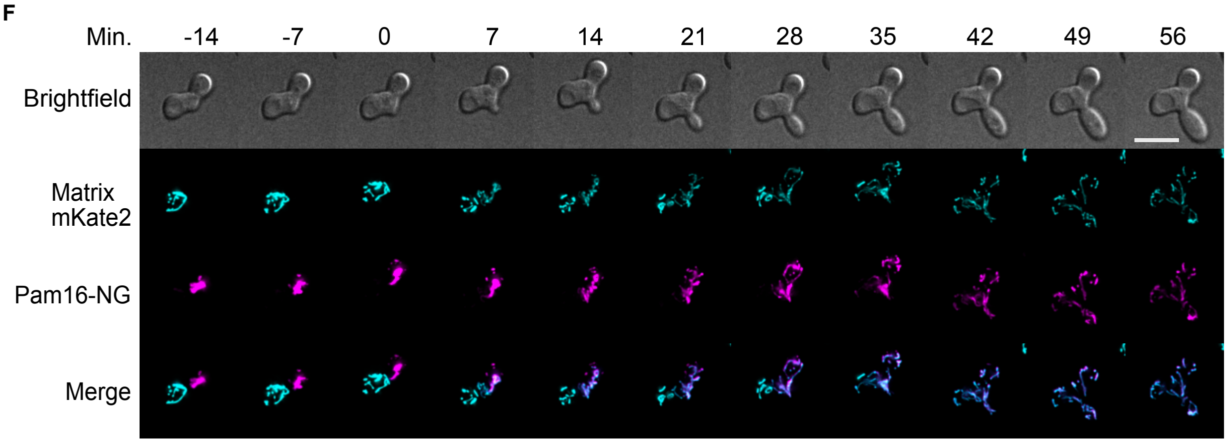
# Supplemental Figure 5



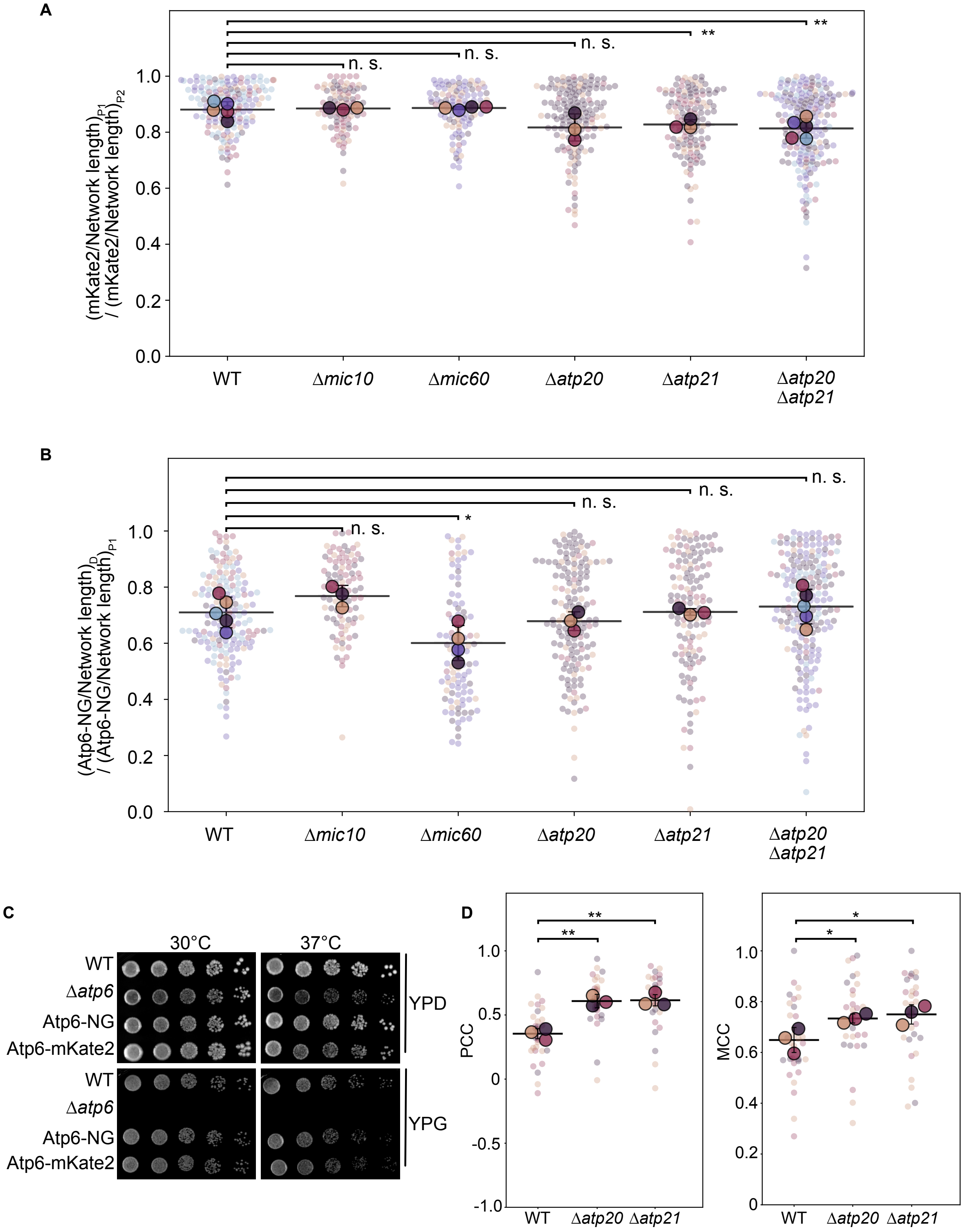
## Supplemental Figure 6



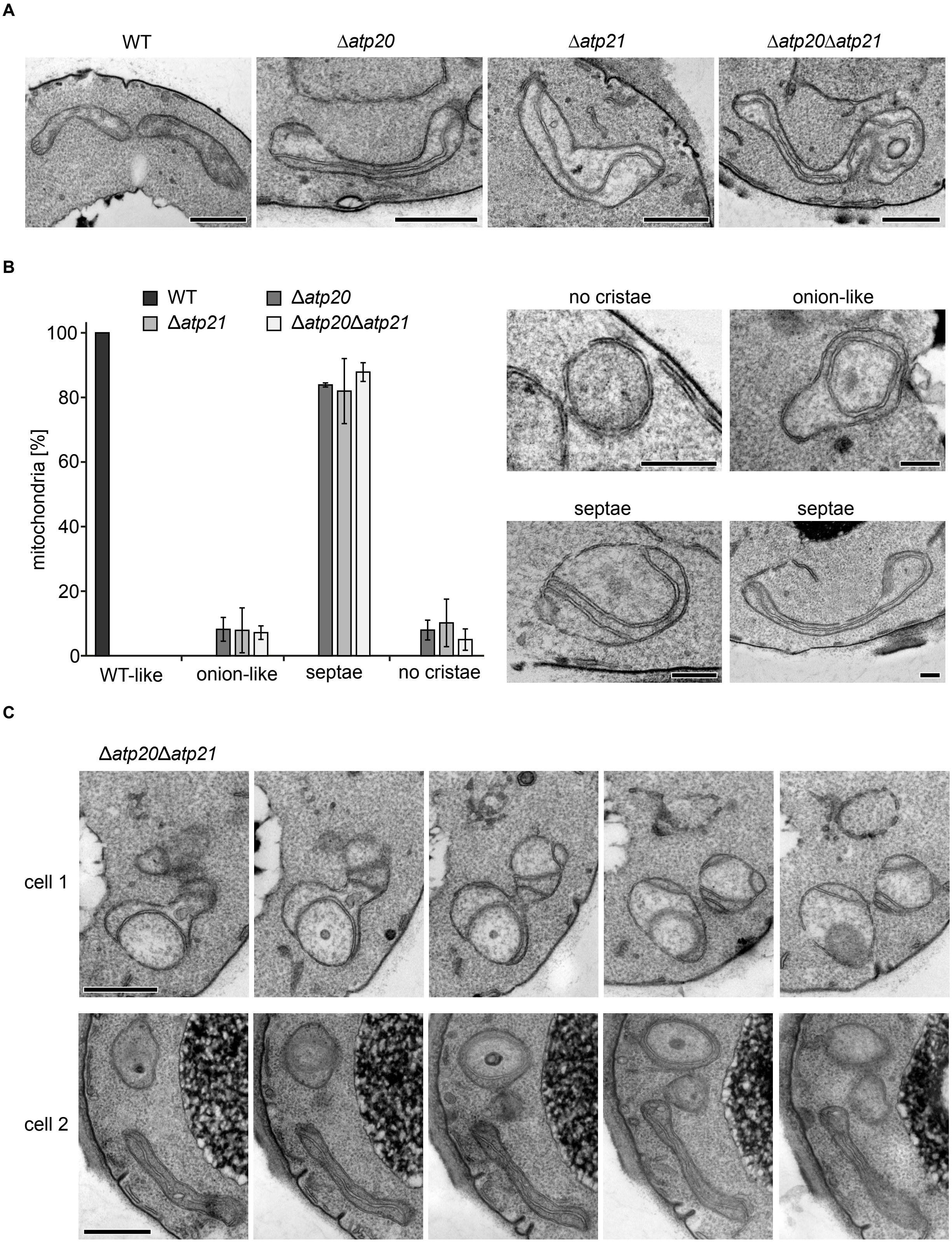
# Supplemental Figure 6



## Supplemental Figure 7



# Supplemental Figure 8



# Supplemental Figure 9

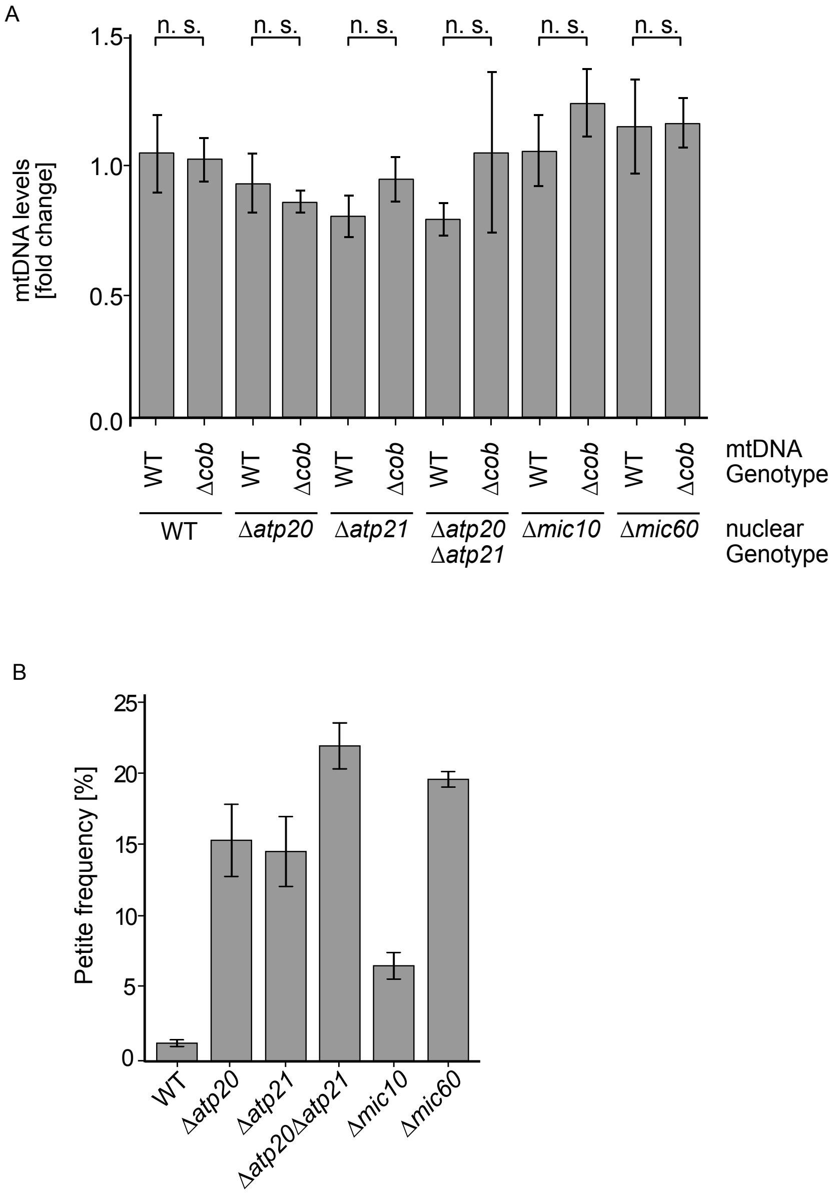


Table 1 - Yeast Strains used in this study

| Name / Alias | Short description | Mating type | Genotype | Source |
|---|---|---|---|---|
| yCO380 | WT | *Mat a* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO* | 1 |
| yCO381 | WT | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO* | 1 |
| yCO391 | Δ*arg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA* | 2 |
| yCO354 | *Δarg8::HIS3*<br>*Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA* | 2 |
| yCJ048 | Neutral ARG8 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 Δarg8::hphNT1 ARG8::mtDNA* | This study |
| yCO392 | Δ*arg8* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA* | This study |
| yCO969 | *Δarg8::HIS3*<br>*Δcob::ARG8M* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA* | This study |
| yCO460 | *Δcob::ARG8*-TetO-TetR-3xRuby | *Mat a* | *leu2-3,112 trp1-1 can1-100 ura3-1 his3-11,15 Δarg8::hphNT pvt100u-mtTagBFP HO-Su9-TetR-3xmRuby3--KanMX4-HO Δcob::ARG8-TetO* | This study |
| yCJ033 | LacO-LacI-3xGFP | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO HO-Su9-TagBFP-Pcup-Su9-3xGFP-LacI::kanMX4* | This study |
| yCJ009 | HO::Su9-mKate2 | *Mat a* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO HO::Su9-mKate2::kanMX6* | This study |

| | | | | |
|---|---|---|---|---|
| yCJ010 | HO::Su9-NG | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO HO::Su9-NG::kanMX6* | This study |
| yCJ078 | *Δarg8::HIS3*<br>Δcob::*ARG8M*<br>HO::Su9-NG | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M HO::Su9-NG-kanMX6 intronless mtDNA* | This study |
| yCJ081 | *Δarg8::HIS3*<br>HO::Su9-mKate2 | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA HO::Su9-mKate2::kanMX6* | This study |
| yCJ002 | *Δdnm1*<br>*Δarg8::HIS3*<br>*Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δdnm1::hphNT1* | This study |
| yCJ004 | *Δdnm1 Δarg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δdnm1::hphNT1* | This study |
| yCJ085 | *Δdnm1*<br>*Δarg8::HIS3*<br>*Δcob::ARG8M*<br>*HO::Su9-NG* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δdnm1::hphNT1 HO::Su9-NG-kanMX6* | This study |
| yCJ086 | *Δdnm1 Δarg8*<br>*HO::Su9-mKate2* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δdnm1::hphNT1 HO::Su9-mKate2-kanMX6* | This study |
| yCJ074 | *Δatg32*<br>*Δarg8::HIS3*<br>*Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δatg32::hphNT1* | This study |
| yCJ075 | *Δatg32 Δarg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δatg32::hphNT1* | This study |
| yCJ076 | *Δatg32*<br>*Δarg8::HIS3*<br>*Δcob::ARG8M*<br>*HO::Su9-NG* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δatg32::hphNT1 HO::Su9-NG-kanMX6* | This study |

| | | | | |
|---|---|---|---|---|
| yCJ077 | *Δatg32 Δarg8 HO::Su9-mKate2* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δatg32::hphNT1 HO::Su9-mKate2-kanMX6* | This study |
| yCO114 | *rho*$^{0}$ | *Mat alpha* | *leu2 ura3-52 ade2-101 arg8::URA3 kar1-1 rho*$^{0}$ | 3 |
| yCO115 | *rho*$^{+}$ *Δcox2* | *Mat a* | *lys2 leu2-3,112 ura3-52 his3HinDIII arg8::hisG rho*$^{+}$ *cox2-62* | 3 |
| yCJ025 | MR6 WT | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3* | 4 |
| yCJ026 | MR10 *Δatp6* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 Δatp6* | 4 |
| yCJ043 | *ATP6*-NG | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-NG* | This study |
| yCJ126 | HO::Su9-mScarlet *ATP6*-NG | *Mat a/ alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-NG HO::Su9-mScarlet-URA3* | This study |
| yCJ045 | HO::Su9-mKate2 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO HO::Su9-mKate2-kanMX6* | This study |
| yCJ019 | *COX*4-NeonGreen | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Cox4-NeonGreen* | This study |
| yCJ072 | Fis1-NG | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA FIS1-NG::kanMX6* | This study |
| yCJ114 | Pam16-NG | *Mata* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO PAM16-NG::kanMX* | This study |
| yCJ051 | *Δatp21 ATP6*-NG | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-NG Δatp21::hphNT1* | This study |
| yCJ052 | *Δatp21* HO::Su9-mKate2 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Mito Δatp21::hphNT1 HO::Su9-mKate2-kanMX6* | This study |

| | | | | |
|---|---|---|---|---|
| yCJ053 | *Δatp20 ATP6*-NG | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-NG Δatp20::hphNT1* | This study |
| yCJ054 | *Δatp20* HO::Su9-mKate2 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Mito Δatp20::hphNT2 HO::Su9-mKate2-kanMX6* | This study |
| yCJ057 | *Δatp20Δatp21 ATP6*-NG | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-NG Δatp21::hphNT1 Δatp20::hphNT2* | This study |
| yCJ058 | *Δatp20Δatp21* HO::Su9-mKate2 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Mito Δatp21::hphNT1 Δatp20::hphNT2 HO::Su9-mKate2-kanMX6* | This study |
| yCO754 | *Δmic10 ATP6*-NG | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-NG Δmic10::hphNT1* | This study |
| yCO755 | *Δmic10* HO::Su9-mKate2 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Mito Δmic10::hphNT1 HO::Su9-mKate2-kanMX6* | This study |
| yCJ066 | *Δmic60 ATP6*-NG | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-NG Δmic60::hphNT1* | This study |
| yCJ067 | *Δmic60* HO::Su9-mKate2 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Mito Δmic60::hphNT1 HO::Su9-mKate2-kanMX6* | This study |
| yCJ120 | *ATP6*-mKate2 | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-mKate2* | This study |
| yCJ084 | *ATP6*-NG | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 ATP6-NG* | This study |
| yCJ123 | ATP6-NG pvt100u-TagBFP | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 ATP6-NG pvt100u-TagBFP* | This study |

| | | | | |
|---|---|---|---|---|
| yCJ124 | ATP6-mKate2 pvt100u-TagBFP | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-mKate2 pvt100u-TagBFP* | This study |
| yCJ127 | ATP6-NG *Δatp20* pvt100u-TagBFP | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 ATP6-NG Δatp20::NatNT2 pvt100u-TagBFP* | This study |
| yCJ128 | ATP6-mKate2 *Δatp20* pvt100u-TagBFP | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-mKate2 Δatp20::NatNT2 pvt100u-TagBFP* | This study |
| yCJ130 | ATP6-NG *Δatp21* pvt100u-TagBFP | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 ATP6-NG Δatp21::hphNTI pvt100u-TagBFP* | This study |
| yCJ132 | ATP6-mKate2 *Δatp21* pvt100u-TagBFP | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 ATP6-mKate2 Δatp21::hphNTI pvt100u-TagBFP* | This study |
| yCJ103 | WT LacO-LacI | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Su9-3xGFP-LacI::kanMX6* | This study |
| yCJ104 | WT matrix-mKate2 | *Mat a* | *leu2-3,112 trp1-1 can1-100 ura3-1 ade2-1 his3-11,15 HO-Su9-mKate2* | This study |
| yCJ105 | LacO-LacI *Δatp21* | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Su9-3xGFP-LacI::kanMX6 Δatp21::NatNT2* | This study |
| yCJ106 | matrix-mKate2 *Δatp21* | *Mat a* | *leu2-3,112 trp1-1 can1-100 ura3-1 ade2-1 his3-11,15 HO-Su9-mKate2 Δatp21::NatNT2* | This study |
| yCJ134 | LacO-LacI Δmic60 | *Mat alpha* | *leu2-3,112 can1-100 ura3-1 his3-11,15 mt-LacO Su9-3xGFP-LacI::kanMX6 Δmic60::hphNT1* | This study |
| yCJ135 | matrix-mKate2 Δmic60 | *Mat a* | *leu2-3,112 trp1-1 can1-100 ura3-1 ade2-1 his3-11,15 HO-Su9-mKate2 Δmic60::hphNT1* | This study |
| yCJ020 | *Δatp20 Δarg8::HIS3 Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δatp20::hphNT1* | This study |

| | | | | |
|---|---|---|---|---|
| yCJ022 | *Δatp20 Δarg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δatp20::hphNT1* | This study |
| yCJ046 | *Δatp20Δatp21 Δarg8::HIS3 Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δatp21::hphNT1 Δatp20:: kanMX6* | This study |
| yCJ047 | *Δatp20Δatp21 Δarg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δatp21::hphNT1 Δatp20::kanMX6* | This study |
| yCJ005 | *Δatp21 Δarg8::HIS3 Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δatp21::hphNT1* | This study |
| yCJ007 | *Δatp21 Δarg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δatp21::hphNT1* | This study |
| yCO756 | *Δmic10 Δarg8::HIS3 Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δmic10::hphNT1* | This study |
| yCO757 | *Δmic10 Δarg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δmic10::hphNT1* | This study |
| yCJ070 | *Δmic60 Δarg8::HIS3 Δcob::ARG8M* | *Mat alpha* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 cob::ARG8M intronless mtDNA Δmic60::hphNT1* | This study |
| yCJ071 | *Δmic60 Δarg8* | *Mat a* | *ade2-1 his3-11,15 trp1-1 leu2-3,112 ura3-1 CAN1 arg8::HIS3 intronless mtDNA Δmic60::hphNT1* | This study |

[1] Osman et al. 2015
[2] Gruschke et al. 2011
[3] Steele et al. 1996
[4] Rak et al. 2007

Tabel 2 - Primers used in this study

| Name / Alias | Sequence |
|---|---|
| CO356<br>Δ*dnm1* S1 | CATTAAGTAGCTACCAGCGAATCTAAATACGACGGATAAAGAATG<br>CGTACGCTGCAGGTCGAC |
| CO357<br>Δ*dnm1* S2 | CGCAATGTTGAAGTAAGATCAAAAATGAGATGAATTATGCAATTA<br>ATCGATGAATTCGAGCTCG |
| CO573<br>Δ*atg32* S1 | TCACAAAAGCAAAAAAAATCTGCCAGGAACAGTAAA<br>CATATGCGTACGCTGCAGGTCGAC |
| CO574<br>Δ*atg32* S2 | GTGAGTAGGAACGTGTATGTTTGTGTATATTGGAA<br>AAAGGTTAATCGATGAATTCGAGCTCG |
| CO888<br>Δ*mic10* S1 | TGCTACGAGAGGGAATAAACACGGAAAAAGACAAAATATACCAT<br>GCGTACGCTGCAGGTCGAC |
| CO889<br>Δ*mic10* S2 | TATTTTTTTTTTTGAATATATATAAAGCATCGTCGCTTAAGACTAAT<br>CGATGAATTCGAGCTCG |
| CO986<br>Δ*atp20* S1 | ACCTGCCGATAAATCATATTTCAGAACGCTAATCAATTCATC<br>ATGCGTACGCTGCAGGTCGAC |
| CO987<br>Δ*atp20* S2 | ACGAATATACAAGGGTTTTGCGAATAGATAGAATTAAAAAGCT<br>TAATCGATGAATTCGAGCTCG |
| CO1104<br>Δ*atp21* S1 | CGGAACATAACGTATATAGGAACTAGCTGAGTGAGTTAAAG<br>GATGCGTACGCTGCAGGTCGAC |
| CO1105<br>Δ*atp21* S2 | TAATGTGCATTTTTAGTATCCTATTTATGTTGAAGCTTCTATTTA<br>ATCGATGAATTCGAGCTCG |
| CO1114<br>qPCR *Cox1* fv | CTACAGATACAGCATTTCCAAGA |
| CO1115<br>qPCR *Cox1* rv | GTGCCTGAATAGATGATAATGGT |
| CO1116<br>qPCR *Act1* fv | CACCCTGTTCTTTTGACTGA |
| CO1117<br>qPCR *Act1* rv | CGTAGAAGGCTGGAACGTTG |
| CO1196<br>Δ*mic60* S1 | GGCATAAGAACGCATTGAAAAGTCTAAAAAACTAATATTCGT<br>ATGCGTACGCTGCAGGTCGAC |

| | |
|---|---|
| CO1197 Δ*mic60* S2 | AGGTGTAATGACGTACATCTCTTTTCTCTTTGTATTATTCTTTCAATCGATGAATTCGAGCTCG |
| CO1268 tagging of *Fis1* fw | GTAATTATCTACTTTTTACAACAAATATAAAACAATGGTCTCAAAAGGGGAGG |
| CO1269 tagging of *Fis1* rv | CCAAGCTTCTTATATAATTCATCCATTCCCATGACA |
| CO1270 backbone for *Fis1*-NG fw | GGATGAATTATATAAGAAGCTTGGTCATGGTACTGA |
| CO1271 backbone for *Fis1*-NG rv | GCAAGCTAAACAGATCTTACCTTCTCTTGTTTCTTAAGAAGAAAC |
| CO1677 tagging of *Pam16* fw | CGAATTCAACAAATTCATCTGGTGCAGATAATAGTGCAAGCAGCAATCAGCGTACGCTGCAGGTCGAC |
| CO1678 tagging of *Pam16* rv | GCTGCATGCTTTCGATAACACTTGTGACGTAATGATGGAGGCTTCCTTGAATCGATGAATTCGAGCTCG |
| CO562 Amplification of *COB* for | AATCAAATGTGTATTTAAGTTTAGTG |
| CO562 Amplification of *COB* rev | TTATTTATTAACTCTACCGATATAGAAT |
| CO891 Amplification of *ARG8* for | TCAAGACCTGAAGATTTATGTATCACAAGAGG |
| CO601 Amplification of *ARG8* rev | TTAAGCATATACAGCTTCGATAGC |
| CO982 S3 tagging of Cox4 | TACAAACTAAACCCTGTTGGTGTTCCAAATGATGACCACCATCACGGTGACGGTGCTGGTTTA |
| CO982 S2 tagging of Cox4 | AAAAAGTAAAAGAGAAACAGAAGGGCAACTTGAATGATAAGATTAATCGATGAATTCGAGCTCG |

Table 3 - Plasmids used in this study

| Name / Alias | Needed for | Source |
|---|---|---|
| pCO021 | Janke deletion cassette G418 resistance | 1 |
| pCO059 | Janke deletion cassette NAT resistance | 1 |
| pCO074 | Janke deletion cassette hygromycin resistance | 1 |
| pCO151 | pvt100u-TagBFP | This study |
| pCO282 | pCOB-ST5 | 2 |
| pCO441 | HO-$P_{PGK1}$-Su9-mKate2-kanMX6-HO | This study |
| pCO442 | HO-$P_{PGK1}$-Su9-NG-kanMX6-HO | This study |
| pCO307 | pCOB/ST5 Arg8+Term+synth-TetO | This study |
| pCO408 | HO-$P_{CUP}$-Su9-3xGFP-LacI—$P_{PGK}$-Su9-TagBFP-KanMX | This study |
| pCO407 | HO-$P_{CUP}$-Su9-TetR-3xRuby3--$P_{PGK1}$-Su9-TagBFP-KanMX4-HO | This study |
| pCO444 | Plasmid for biolistic transformation and generation of *ATP6*-mtNeonGreen | This study |
| pCO494 | C-Terminal tagging of proteins with NeonGreen | This study |
| pCJ013 | Plasmid for biolistic transformation and generation of *ATP6*-mtNeonGreen | This study |

[1] Janke et al. 2004

[2] Gruschke et al. 2011

## 2.3 YeastMate: Neural network-assisted segmentation of mating and budding events in *S. cerevisiae*

### 2.3.1 Main Paper

*Bioinformatics*, 2022, 1–3
https://doi.org/10.1093/bioinformatics/btac107
Advance Access Publication Date: 18 February 2022
Applications Note

Bioimage informatics

# YeastMate: neural network-assisted segmentation of mating and budding events in *Saccharomyces cerevisiae*

**David Bunk , Julian Moriasy, Felix Thoma, Christopher Jakubke, Christof Osman and David Hörl ***

Faculty of Biology, Ludwig-Maximilians-Universität München, 82152 Planegg-Martinsried, Germany

*To whom correspondence should be addressed.

Associate Editor: Jinbo Xu

Received on October 21, 2021; revised on January 14, 2022; editorial decision on February 7, 2022; accepted on February 16, 2022

## Abstract

**Summary:** Here, we introduce *YeastMate*, a user-friendly deep learning-based application for automated detection and segmentation of *Saccharomyces cerevisiae* cells and their mating and budding events in microscopy images. We build upon Mask R-CNN with a custom segmentation head for the subclassification of mother and daughter cells during lifecycle transitions. *YeastMate* can be used directly as a Python library or through a standalone application with a graphical user interface (GUI) and a Fiji plugin as easy-to-use frontends.

**Availability and implementation:** The source code for YeastMate is freely available at https://github.com/hoerlteam/YeastMate under the MIT license. We offer installers for our software stack for Windows, macOS and Linux. A detailed user guide is available at https://yeastmate.readthedocs.io.

**Contact:** hoerl@bio.lmu.de

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

An important experimental approach when working with the budding yeast *Saccharomyces cerevisiae* is to examine the effects of mutations on morphological features or protein localization either by brightfield or fluorescent microscopy (Ohya *et al.*, 2015). Such experiments can also be performed in a systematic manner by combining automated microscopy with strain libraries comprising thousands of yeast strains that carry gene deletions or express fluorescently tagged proteins (Giaever *et al.*, 2002; Huh *et al.*, 2003; Weill *et al.*, 2018), which, however, calls for robust automated image analysis pipelines. In recent years, tools based on convolutional neural networks (CNNs) have become state-of-the-art for many tasks in biomedical image analysis (von Chamier *et al.*, 2019), including segmentation of individual *S.cerevisiae* cells (Dietler *et al.*, 2020; Lu *et al.*, 2019; Salem *et al.*, 2021). Many experimental strategies facilitated by the yeast system also make use of specific transitions in the yeast lifecycle like budding and mating to study things like organelle inheritance and mitochondrial quality control (Jakubke *et al.*, 2021; Rafelski *et al.*, 2012). Detection of matings and buddings is often done by hand or through dedicated postprocessing routines on the output of a single-cell segmentation tool (e.g. tracking of cells in a time series of images), highlighting the need for easy-to-use end-to-end solutions for these more complex tasks.

Here, we present *YeastMate*, a novel deep learning-based tool for end-to-end segmentation of single cells and detection of transitions in the lifecycle of *S.cerevisiae* in single transmitted light images. YeastMate performs three tasks: instance segmentation of single cells, object detection of zygotes and budding events and automatic assignment of mother and daughter cells involved in a mating or budding event. The detection backend is based on Mask R-CNN (He *et al.*, 2017) and is complemented by a user-friendly frontend using modern web technologies as well as a Fiji plugin. In the task of detecting mating and budding events, we achieve accuracies comparable to manual human reannotation and we can also perform single-cell segmentation robustly across various datasets. YeastMate is already being used in ongoing research in our lab (Jakubke *et al.*, 2021). In addition to the software, we also provide a new dataset of images with manually annotated cells and mating and budding events.

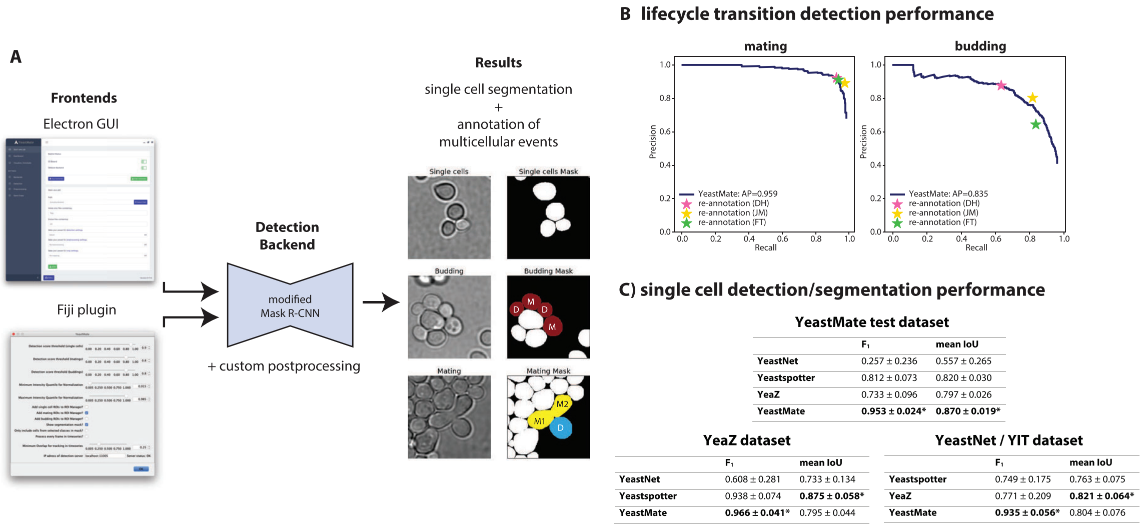
## 2 Materials and methods

*CNN architecture and training*: Our network architecture builds upon Mask R-CNN with a modified mask segmentation head producing multiclass semantic segmentations (Supplementary Fig. S1) for each detected object. Instead of individually detecting mother and daughter cells, we first detect the whole mating and budding events as well as single cells. In a postprocessing step, we resolve the roles of the cells involved in budding or mating events based on the multiclass segmentation masks (Supplementary Fig. S2). We used 80% of our images for training our network and cross-validation and 20% as a hold-out test set for the final performance assessment.

© The Author(s) 2022. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btac107/6531960 by guest on 13 April 2022



**YeastMate test dataset**

| | $F_1$ | mean IoU |
|---|---|---|
| **YeastNet** | 0.257 ± 0.236 | 0.557 ± 0.265 |
| **Yeastspotter** | 0.812 ± 0.073 | 0.820 ± 0.030 |
| **YeaZ** | 0.733 ± 0.096 | 0.797 ± 0.026 |
| **YeastMate** | **0.953 ± 0.024*** | **0.870 ± 0.019*** |

**YeaZ dataset**

| | $F_1$ | mean IoU |
|---|---|---|
| **YeastNet** | 0.608 ± 0.281 | 0.733 ± 0.134 |
| **Yeastspotter** | 0.938 ± 0.074 | **0.875 ± 0.058*** |
| **YeastMate** | **0.966 ± 0.041*** | 0.795 ± 0.044 |

**YeastNet / YIT dataset**

| | $F_1$ | mean IoU |
|---|---|---|
| **Yeastspotter** | 0.749 ± 0.175 | 0.763 ± 0.075 |
| **YeaZ** | 0.771 ± 0.209 | **0.821 ± 0.064*** |
| **YeastMate** | **0.935 ± 0.056*** | 0.804 ± 0.076 |

**Fig. 1.** (**A**) Main components and output of *YeastMate*: We perform instance segmentation of single cells and detection of lifecycle transitions using a modified Mask R-CNN (middle), which can either be used directly from Python code or via two GUI frontends (left) to provide instance segmentation of single cells as well as detection of budding events and mating events with identification of the mother (M) and daughter (D) cells involved in the event (right). (**B**) Precision–recall (PR) curves of YeastMate performance for detection of mating and budding events in our test dataset. Human precision and recall during reannotation by three different authors are plotted as colored stars. (**C**) Single-cell detection performance ($F_1$ score) and segmentation performance (mean intersection-over-union of true positive detections) of YeastNet, YeastSpotter, YeaZ and YeastMate on our own test dataset as well as datasets from Dietler *et al.* (2020) and Salem *et al.* (2021). The values in the table are mean ± standard deviation across all images in a dataset

*Software architecture*: YeastMate is implemented in a modular way: the detection backend can be used as a Python library but also runs as a webservice to provide its capabilities to client applications. As clients, we provide a Fiji plugin that directly interfaces with the server via HTTP requests as well as a standalone GUI Desktop application (Fig. 1 and Supplementary Figs S3–S5). We provide the whole YeastMate stack as a single installable file for use on a local workstation.

For more details, please refer to the Supplementary Methods.

## 3 Results

*Dataset*: For training YeastMate, we collected 147 brightfield and differential interference contrast (DIC) images of *S.cerevisiae* acquired on two different microscopes at various imaging conditions and generated curated single-cell segmentation masks and mating and budding annotations. In total, our data contain 17 058 individual cells, 3615 buddings and 2380 zygotes (Supplementary Table S1).

*Object detection performance*: Our network achieves a mean average precision (mAP) of 0.878 as well as favourable APs for the individual classes of objects when applied to our test set (Fig. 1B and Supplementary Fig. S6). We also assessed interhuman reproducibility by comparing the annotations for mating and budding events of four different annotators (with the most experienced one chosen as the reference). On our test dataset, mean interhuman precision and recall are (0.908, 0.946) for mating events and (0.774, 0.763) for budding events. These values lie close to the PR-curves of our network, indicating that YeastMate can achieve object detection performance comparable to manual human annotation.

*Single-cell segmentation performance*: YeastMate also compares favourably to existing CNN-based solutions in single-cell segmentation, showing consistent performance not only on our own test dataset, but also two publicly available datasets (Fig. 1C, Supplementary Results, Supplementary Fig. S7 and Table S2). Based on the robust Mask R-CNN architecture, YeastMate achieves the highest single-cell detection performance of all tools as well as competitive results in segmentation.

## 4 Conclusion

With YeastMate, we introduce an easy-to-use application to not only perform single-cell segmentation in images of *S.cerevisiae* with high robustness across datasets, but also detect transitions in the cell cycle in single images with accuracies comparable to human annotation. YeastMate is implemented in a modular way and can be run on local workstations but the detection server can also be run on a remote compute server. Additionally, we provide two user-friendly frontends to make the tool available without the need to write code. We can envision YeastMate to be expanded to detect other life cycle states, such as meiotic asci, similar to existing work in *Arabidopsis thaliana* (Lim *et al.*, 2020). YeastMate can provide a considerable improvement to high-throughput studies of yeast, facilitating not only the automated analysis of large image datasets of single cells, but also enabling the study the complex interplay of cellular components during both sexual and asexual reproduction of *S.cerevisiae*.

## Acknowledgements

The authors would like to thank Mikhail Slivinskiy for advice on the React.js frontend and Heinrich Leonhardt and Hartmann Harz (LMU) for comments on the manuscript and support. Some of the images in our dataset were acquired in the lab of Maya Schuldiner (Weizmann Institute) as part of an ongoing collaboration.

## Data availability

The data underlying this article are available in Open Science Framework (OSF), at https://doi.org/10.17605/osf.io/287fr

## Funding

This work was supported by the European Research Council [ERCStG-714739 IlluMitoDNA to C.O.]; and the Deutsche Forschungsgemeinschaft [SPP 2202 422857584].

*Conflict of Interest*: none declared.

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btac107/6531960 by guest on 13 April 2022

## References

Dietler,N. *et al.* (2020) A convolutional neural network segments yeast microscopy images with high accuracy. *Nat. Commun.*, **11**, 1–8.

Giaever,G. *et al.* (2002) Functional profiling of the *Saccharomyces cerevisiae* genome. *Nature*, **418**, 387–391.

He,K. *et al.* (2017) Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Institute of Electrical and Electronics Engineers (IEEE), New York, NY, pp. 2961–2969.

Huh,W.-K. *et al.* (2003) Global analysis of protein localization in budding yeast. *Nature*, **425**, 686–691.

Jakubke,C. *et al.* (2021) Cristae-dependent quality control of the mitochondrial genome. *Sci. Adv.*, **7**, eabi8886.

Lim,E.-C. *et al.* (2020) Deeptetrad: high-throughput image analysis of meiotic tetrads by deep learning in *Arabidopsis thaliana*. *Plant J.*, **101**, 473–483.

Lu,A.X. *et al.* (2019) YeastSpotter: accurate and parameter-free web segmentation for microscopy images of yeast cells. *Bioinformatics*, **35**, 4525–4527.

Ohya,Y. *et al.* (2015) Single-cell phenomics in budding yeast. *Mol. Biol. Cell*, **26**, 3920–3925.

Rafelski,S.M. *et al.* (2012) Mitochondrial network size scaling in budding yeast. *Science*, **338**, 822–824.

Salem,D. *et al.* (2021) YeastNet: deep-learning-enabled accurate segmentation of budding yeast cells in bright-field microscopy. *Appl. Sci.*, **11**, 2692.

von Chamier,L. *et al.* (2019) Artificial intelligence for microscopy: what you should know. *Biochem. Soc. Trans.*, **47**, 1029–1040.

Weill,U. *et al.* (2018) Genome-wide swap-tag yeast libraries for proteome exploration. *Nat. Methods*, **15**, 617–622.

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btac107/6531960 by guest on 13 April 2022

### 2.3.2 Supplementary information

Supplementary Material

# YeastMate: Neural network-assisted segmentation of mating and budding events in *S. cerevisiae*

David Bunk, Julian Moriasy, Felix Thoma, Christopher Jakubke, Christof Osman and David Hörl

# Supplementary Methods

## Data acquisition and cell culture

The images in our dataset are a collection of DIC and brightfield images from multiple acquisitions on either a Nikon Ti2 microscope equipped with a CFI SR HP Apo TIRF 100xH NA 1.49 oil immersion objective (with an optional 1.5x magnification relay lens) and dual Photometrics Prime 95B 25mm cameras or on a fully automated Olympus IX 83 microscope with an UPLFLM 60x NA 0.9 air objective with a Hamamatsu ORCA-Flash4.0 V3 camera. All together, we assembled our dataset from 6 individual experiments on the Nikon system and parts of 2 multi-well experiments imaged at the Olympus system. In all experiments, two haploid *S. cerevisiae* strains were mated. Data from the Nikon system were acquired for (Jakubke *et al.*, 2021), data from the Olympus system are preliminary data for a high-throughput screen. For a complete reference of the individual experiments included in the dataset, refer to Supplementary Table ST1.

Yeast cultivation and sample preparation was performed as described previously (Jakubke *et al.*, 2021). In short, to image the mating process, yeast cells of opposing mating type were grown in separate flasks in YPD medium at 30°C to an OD600 ~1.0. 1000 μl from both cultures were combined in a reaction tube, vortexed thoroughly and centrifuged for 3 minutes at 5.000 rpm. The pellet was resuspended in 50 μl of YPD medium and spotted onto a YPD plate to allow cells to mate. After incubation at 30°C for 3 h, pre-mated cells were scraped off and resuspended in 200 μl PBS. The cell suspension was added into wells of 8-well IBIDI slides that had previously been coated with Concanavalin A. Attachment of cells to the coverslip was supported by centrifuging the IBIDI slide for 60 s at 1000 rpm.

## Generation of ground truth annotations

Single-cell segmentation masks were generated in an iterative, feedback-based process: We initially trained a standard Mask R-CNN on the brightfield dataset provided by YeaZ (Dietler *et al.*, 2020), applied it to our images and manually corrected the generated masks using napari (Sofroniew *et al.*, 2021). For the annotation of cells involved in mating events or buddings, we implemented a custom annotation scheme, again using napari: For annotation of mating events, we manually drew paths in a shape layer going from mother 1 to mother 2 and, optionally, the medial daughter cell for each mating event in our images. Likewise, in another layer, we added paths going from mother to daughter cell in all buddings (Supplementary Figure S4, annotations are visualized in this manner in Supplementary Figure S6B). Our path-based annotations were saved in JSON files and were converted to the bounding box annotations needed by R-CNN-based networks on the fly when training the network.

All annotations in the final ground truth were manually verified and corrected and all networks trained for validation and benchmarking were trained de-novo from these final data. The mating and budding annotations of the 30 images in the test set were re-annotated by four different authors (DB, JM, FT and DH) of this study to allow us to compare the accuracy of YeastMate to inter-human variation (Supplementary Figure S6). The single cell masks were left as-is unless the re-annotator wanted to annotate a mating or budding event where not all cells are present in the segmentation mask (e.g., a small bud), in which case they manually added the missing object to the mask with a new label in napari.

## CNN architecture and training

We use the Detectron2 (Wu *et al.*, 2019) implementation of Mask R-CNN (He *et al.*, 2017) with modifications to the mask head and mask loss to produce a multi-class segmentation (instead of the default binary segmentation) for each detected object.

In short, Mask R-CNN extends Faster R-CNN (Ren *et al.*, 2015), which performs object detection using a 2-step strategy: images are first processed by a *backbone* CNN to extract features, which are then fed into a region proposal network (RPN) that performs binary object/non-object classification for a predefined regular grid of bounding boxes, called *anchors.* The features of proposal boxes with the highest scores from the RPN are then resized to a common shape and fed into box classification and box regression *head* subnetwork, performing a second round of (multiclass) classification of the proposed regions, and refining the coordinates. Mask R-CNN adds a mask head to this architecture that produces a binary segmentation mask for each proposal region.

For YeastMate, we replaced the default mask head with a multiclass mask head: instead of a binary mask for each region, we produce a multichannel semantic segmentation (Supplementary Figure S1). We also do not threshold the segmentation, instead we continue working with the raw probabilities of the different segmentation classes $c_{seg}$. Note that the classes for the classification head $c_{obj}$, classifying the whole region, do not have to equal the classes of the segmentation. In our case, $c_{obj}$ classifies a whole region as single cell, mating or budding, whereas the segmentation mask has classes $c_{seg} \in$ {0: background, 1: single cell, 2: mating mother, 3: mating daughter, 4: budding mother and 5: budding daughter}.

During training, the single cell masks from the ground-truth (GT) data are used as one-hot encoded segmentations of class single cell and for all mating and budding events, we use masks encoding the role of all the individual cells (mother or daughter) during the event as GT. As our mask head performs multiclass classification, we use softmax activation in the last layer of the mask head and cross entropy (CE) loss instead of sigmoid activation and binary CE (BCE) loss. During inference, the intersection-over-union (IoU) threshold for the non-maximum suppression (NMS) of the ROI header output was set to 0.5.

We performed all training on a compute server equipped with 2 Intel(R) Xeon(R) E5-2680v4 CPUs (14 cores/28 threads each), 256GB of RAM and 2 NVIDIA Tesla V100 GPUs with 32GB VRAM each. We did not perform single training runs on multiple GPUs, instead using the two GPUs to process two cross-validation splits in parallel. For optimization of the network weights, we used stochastic gradient with momentum of 0.9 and a learning rate warm-up period of 1000 iterations. During cross-validation, the performance on the validation set stopped increasing after

200,000 iterations, we therefore also trained the final model using all training images for 200,000 iterations. We used a learning rate of 0.005 for the final model, based on performance during our cross-validation. Each training iteration was done on a batch of two random 400x400px image crops. Random augmentation was done each iteration in the form of random flips (50% chance for an upside-down flip and 50% for a left-right flip) and random rotation (a random 0-3 number of 90° rotations).

## Postprocessing of CNN outputs

In a postprocessing step, we resolve the complementary segmentation and object detection outputs of our network to generate a single cell instance segmentation that assigns roles to cells participating a lifecycle transition event (Supplementary Figure S2).

For each detected bounding box of the compound classes mating or budding, we proceed as follows: we collect all detected objects of class single cell whose centers fall inside the compound object. We threshold the segmentation mask (for channel $c_{seg}=$ single cell) of the single cell at 0.5 and calculate the average scores for all classes inside the mask in the segmentation of the compound object.

We consider the assignment of single cell objects to the available roles in a lifecycle transition object (I.e., mother and daughter in buddings, and two mothers and an optional daughter in a mating event) as a linear assignment problem and find the cells-to-roles matching with the highest overall score. For optional roles (e.g., the daughter in matings, as not all zygotes will have produced a medial bud), we discard the assignment of the cell to that role if the score lies under a threshold optional_object_thresh (we currently do not expose this parameter to the front end, instead using a value of 0.15, which was determined to be optimal during cross-validation). In each compound object, we also perform this matching for the other type and if the average score of the assignments is at least parent_override_thresh times higher, we use this assignment instead, thus using the segmentation output as a second line of evidence to catch zygotes erroneously classified as buddings by the object detection pathway, and vice-versa. During cross-validation we did not see strong improvements by using this overriding strategy, so we currently leave the parameter at a high default value of 20. All single cells that are not assigned a role in this procedure remain classified as single cells in the final segmentation.

Note that the procedure is written in a generalized way and could easily be adapted to accommodate, e.g., meiotic asci detection as well, if adequately annotated data were provided to our system.

## Tracking in timeseries

We offer the option to perform basic overlap-based tracking of cells and lifecycle transition events in time series of N frames, i.e., to assign the same label to the same cell/event in subsequent frames. To do so, we process the network output for all adjacent pairs of frames (1,2), (2,3), ..., (N-1, N). For each pair, we first compute the pairwise mask IoU of all detected individual cells in the network output for frame n with all cells in frame n+1 (IoUs below a user-settable threshold are set to zero). We find a bipartite matching that maximizes the sum of all IoUs via linear sum assignment. Cells in frame n+1 that can be matched with nonzero IoU are given the same label as the matched cell in frame n. We then perform the same procedure for mating and budding events, using the bounding boxes of detected events for IoU calculation. We keep track of the maximum label used so far and for all cells and events that could not be matched, we assign new sequential labels (first to individual cells, then to transition events).

## Validation and Benchmarking of object detection performance

For validation of our models, we split our 147-image dataset into 5 parts of 29/30 images each, using a stratified splitting strategy placing a roughly equal number of images from individual experiments (Supplementary Table ST1) in each split. Four of the splits were used as cross-validation data to optimize network architecture and hyperparameters, while the final 30-image split was left aside as test data and only used in a final test run. Performance during cross-validation was assessed by training the network on 3 of the 4 training splits and using the remaining one as a validation set. The final model was trained on all 4 training splits and tested on the test split.

YeastMate provides bounding boxes of all detected objects as well as their scores. For each class of object, we sorted the detections by score (with no minimal score threshold) and generated precision-recall (PR) curves as well as average precision (AP) scores at 0.5 intersection-over-union (IoU) using the Python implementation of common object detection metrics in (Padilla *et al.*, 2021). We not only computed scores and PR-curves for the compound object classes mating and budding, but also for the roles of single cells within them (mating mother, mating daughter/medial bud, budding mother, budding daughter). For this, we assigned the single cell detection the score of the enclosing compound object before calculating performance metrics.

To assess reproducibility in mating and budding detection between different human annotators, we chose the first author (DB) as the reference annotator and compared the annotations of three other authors (JM, FT and DH). We assessed precision and recall for mating events, buddings, as well as the individual mother and daughter cells involved in them. For each class of object, the annotation from the second annotator was considered true positive (TP) if its bounding box overlapped an annotation by DB with an IoU of at least 0.5, otherwise, it was considered a false positive (FP). Annotations by DB that are not present in the re-annotation were considered false negatives (FN). For each class of object and each re-annotator, we calculated precision ($\frac{\#TP}{(\#TP+\#FP)}$) and recall ($\frac{\#TP}{(\#TP+\#FN)}$) and visualized them as points in the PR-curve describing the network performance (Supplementary Figure S6).

## Single cell segmentation performance

To compare the single-cell segmentation performance of YeastMate to existing deep learning-based solutions, we applied our model (trained on our entire training dataset) as well as the pretrained models of YeastSpotter, YeaZ and YeastNet (Lu *et al.*, 2019; Dietler *et al.*, 2020; Salem *et al.*, 2021) to our test dataset, the brightfield images dataset provided by YeaZ as well as the dataset compiled by YeastNet, consisting of two novel annotated timeseries as well as segmentation masks for two of the timeseries provided by the Yeast Imaging Toolkit (YIT, http://yeast-image-toolkit.biosim.eu/pmwiki.php) .

Given a ground truth segmentation mask and a predicted segmentation mask, we matched detected cells to ground truth cells to maximize the total IoU of all GT-prediction pairs. We considered a predicted cell true positive (TP) if it had an IoU of over 0.5 with a ground truth cell and false positive (FP) otherwise. Ground truth cells that were not matched to a predicted cell with IoU > 0.5 were considered false negatives (FN). For each image, we calculated precision, recall and $F_1$-score ($\frac{2*(precision * recall)}{(precision + recall)}$) to measure object detection performance. In addition, we calculated the mean IoU of all true positive detections. If we tested multiple parameters for a tool, we only used the prediction with the highest $F_1$ score for each image.

## YeastMate

We used YeastMate as a Python library to perform the benchmarking, but only varied parameters that are also available to end users via our graphical interfaces. For images with pixel sizes differing considerably from 110nm (our own images with 1.5x magnification and the images from YIT), we rescaled them using bilinear interpolation and then rescaled the mask produced by YeastMate to the original size using nearest-neighbor interpolation with no anti-aliasing. We normalized images based on their 0.015 and 0.985 quantiles before feeding them into our network. For each image, we generated segmentation masks with single cell score thresholds in the range (0.4, 0.5, ..., 0.9).

## YeastSpotter

We set up the standalone Python code of YeastSpotter (https://github.com/alexxijielu/yeast_segmentation) in a conda environment with the minimum required versions of tensorflow (1.10) and keras (2.2.4), as per the authors instructions. The only user-settable parameter in the main options file is a scale factor, but it can only be set to integer downsampling factors and is intended as a performance option to speed up inference. Therefore, we only used the default settings of no downsampling when making predictions with YeastSpotter.

## YeaZ

We set up the standalone version of YeaZ (https://github.com/lpbsscientist/YeaZ-GUI) as per the authors instructions and followed the instructions for segmentation without using the GUI (using the provided weights for brightfield detection), omitting the tracking step. We applied YeaZ with different values for the two main parameters exposed in the GUI, threshold (0.3, 0.4, ..., 0.9) and minimum distance between seeds (2, 5, 10).

### YeastNet

We set up YeastNet (https://github.com/kaernlab/YeastNet) as per the authors instructions and used the main track.py script to process all datasets. No user-settable parameters are available via the command line interface, so we applied YeastNet with default parameters.

## Application Architecture

YeastMate consists not only of the code for CNN-based cell detection and segmentation, but we also offer two separate graphical frontends for it to maximize user-friendliness: 1) standalone frontend written in React.js and packaged using Electron.js and 2) a plugin that can be used from the popular biomedical image analysis program Fiji (Schindelin *et al.*, 2012). An overview of the components of our software stack is shown in Supplementary Figure S3.

### Detection Server and Python library

Our object detection and segmentation model is an extended version of the PyTorch Mask R-CNN implementation provided by Detectron2, combined with custom postprocessing routines to resolve single cells and lifecycle transitions. The detection model is wrapped via a YeastMatePredictor class, which can load pretrained weights and network configurations and provides methods that will accept an image as input and return an integer-valued instance segmentation image as well as object detections in a custom JSON format. Upon installing the YeastMate module using pip, users can use functions from it in their own Python code. Note that the detection network expects intensities in the 0-1 range for the input images. To mitigate effects of individual bright or dark pixels, we normalize not based on minimal and maximal value, but rather a low and high, user-settable, intensity quantile (by default 0.015 and 0.985). Also, we found that for optimal performance, the input images should have a pixel size comparable to our reference pixel size of 110nm. We therefore also offer automatic resizing of images in the main inference method.

For using the YeastMate detector from other applications, such as our two GUI frontends, we wrap the detection pipeline as a webservice using Flask. Our detection server listens for multipart HTTP POST requests containing a 2-dimensional image (encoded as a single-channel 32-bit TIFF image) as well as parameters encoded as JSON (we follow the AnnotatedImageInput adapter API from BentoML, which we used for model serving initially, see https://docs.bentoml.org/en/latest/api/adapters.html#annotatedimageinput for details).

To ensure a reproducible environment with compatible versions of the libraries we use, we provide a Docker image of YeastMate, which can be used for running the detection server or re-training on new data.

### React.js/Electron GUI client

To interface with our detection service, we created a custom GUI frontend based on modern web technologies (React.js) but provided as a desktop application using Electron.js. The frontend serves as a control hub to interface with other processes that perform image IO and visualization.

### *GUI Frontend*

The frontend is written as a JavaScript web application using React.js and packaged as a desktop app using Electron.js (Supplementary Figure S4). It serves as a user interface for specifying all relevant settings of YeastMate and submitting an inference job to separate processes for image IO and the actual detection via HTTP requests. Users can launch the backend processes from the GUI or specify the IP addresses of already running instances (e.g. when the detection backend is running on a remote compute server)

### *Client IO Backend*

The IO backend of our client is written in Python and is responsible for reading images from paths provided by the frontend (and in the case of multidimensional stacks, selecting the 2D plane the uses wishes to perform detection on), sending them to the detection server and handling and saving detection results. It is implemented as a webservice using Flask, with a task queue based on huey2. In addition to loading images and sending them to the detection server, it can also create crops of detected objects and save them to new files. It also provides several preprocessing functions that are helpful when dealing with multidimensional data.

### *Preprocessing of images*

In the IO backend of our Electron client, we also support TIFF input images of higher dimensionality, given that they follow the TZCYX dimension order (the default order used by ImageJ/Fiji). The user can set a specific time point, z-plane, and channel, which will be sent to the detection server. Export operations such as cropping detected zygotes or budding events will be applied to all channels, timepoints and z-planes.

If data are provided in a different format, such as a microscope vendor-specific image format, we offer an optional preprocessing step that will resave the images as TIFF files meeting our requirements. Image IO from vendor specific formats is done via the BioFormats library (Linkert *et al.*, 2010) using the pims-bioformats wrapper. We tested this process extensively using Nikon nd2 files, but BioFormats should allow the reading of many other formats. As many of our own experiments were performed on a dual-camera microscope, during the resaving step we optionally added the option to align the color channels acquired on the two cameras if a reference channel (e.g., brightfield) was recorded on both. We perform an affine alignment using ORB keypoint detection and descriptor generation (Rublee *et al.*, 2011) followed by descriptor matching and outlier-resistant estimation of an affine transformation from matched keypoints using RANSAC (Fischler and Bolles, 1981) and will use the transformation to warp all channels from the second camera to match channels from the first. For these operations, we make use of the OpenCV and scikit-image packages. Note that for time series, we will align each frame individually and for z-stacks, we estimate a single 2D transformation on a maximum intensity projection of the reference stacks and applying the same transformation to all planes – we thus only correct misalignments along the xy-plane and will not correct drift in time series.

### ***Annotation Backend***

To visualize or correct the detections generated by the detection server, our GUI provides a visualization and annotation function. This will start a separate Python process that loads the specified input files, segmentation masks and JSON object annotations, which will be displayed using napari. Users can edit the masks and object annotations to manually curate the network output.

### ***All-in-one package***

We provide all-in-one packages for all major operating systems consisting of the Electron frontend as well as the IO backend and detection server, packaged using PyInstaller. When starting the Electron application and running a detection task, the IO backend and detection backend will be started in separate processes automatically (unless the user specifically points the frontend to the IP address and port of already running instances).

## Fiji Plugin

In addition to our React-based frontend, we also provide a Fiji plugin that interfaces directly with a detection server using HTTP requests. The plugin is implemented as a SciJava Command (Rueden *et al.*, 2017) that acts on the currently selected image (or a hyperstack with a time dimension) in Fiji (Supplementary Figure S5). It will transmit a normalized version of the image as well as detection parameters to the server and show the segmentation result as a new image and optionally add bounding boxes to Fijis ROI Manager. Execution of the plugin can be recorded as a macro for easy batch-application to many images. We implemented pixel-level calculations in the plugin using imglib2 (Pietzsch *et al.*, 2012).

## User Guide

We provide a detailed user guide for YeastMate at https://yeastmate.readthedocs.io/

## Code and data availability

YeastMate is fully open-source and licensed under the MIT license, all code is available on GitHub in the following repositories:

https://github.com/hoerlteam/YeastMate
(detection backend, **location of releases and issue tracking**)
https://github.com/hoerlteam/YeastMateFrontend (React.js/Electron.js GUI Application)
https://github.com/hoerlteam/YeastMateBackend
(Python-based IO and annotation code for the App)
https://github.com/hoerlteam/YeastMateFiji (Fiji plugin)

Pre-packaged binaries of the whole software stack are also available under:
https://github.com/hoerlteam/YeastMate/releases

Our final model weights and all training and testing data and annotations are available under:
https://osf.io/287fr/?view_only=99d1fddb563b4253957f226c19c4113f

# Supplementary Results

## Single cell segmentation benchmark

When comparing YeastMate to other deep learning-based solutions for yeast cell instance segmentation, we achieve very satisfactory results (Supplementary Table ST2), outperforming all other tools on our own test dataset in all metrics considered. For the YeaZ and YeastNet-YIT datasets, we achieve the highest object detection performance ($F_1$) and are only narrowly beat by YeaZ in terms of recall on the YeastNet-YIT dataset. In terms of the mean IoU of true positive detections, we achieve reasonable performance, but do not outperform YeaZ and YeastSpotter.

Looking at example segmentation masks (Supplementary Figure S7), one can see a potential reason for our relatively low IoU: YeastMate tends to produce small segmentation masks compared to the other tools. So, while we capture the overall shape of cells well (except for very elongated mutants, which we likely mistake for zygotes consisting of two "mother cells", see Supplementary Figure S7E, an issue that could be remedied by including the YeaZ dataset into the training data), biases in the ground truth, e.g., whether to include the out-of-focus halo around cells into the mask, affect the performance of all tools.

Overall, the Mask R-CNN-based tools YeastMate and YeastSpotter show a more consistent performance across datasets, a tribute to the robustness of the underlying network architecture (especially considering that YeastSpotter was not even trained on *S. cerevisiae* images), whereas U-Net-based (Ronneberger *et al.*, 2015) solutions seem to be more dependent on postprocessing steps optimized for a specific dataset. For example, YeastNet, which offers no easy way to finetune the watershed-based postprocessing, generalized poorly to data that do not match the imaging conditions of the original training dataset and seems prone to oversegmentation and artifacts, e.g., in the presence of uneven illumination. The two-step strategy of Mask R-CNN that first identifies proposal regions which are likely to contain objects of interest and then performs segmentation within individual regions also likely serves as a strong prior for the shape of the segmentation results in the second step. Interestingly, the authors of YeaZ cite this as a reason for not using Mask R-CNN as they aimed for the highest segmentation accuracy on mutants of unusual shape. The complementary strengths and weaknesses of Mask R-CNN and U-Net have been observed before, e.g. by (Vuola *et al.*, 2019). Aside from properties of the network architecture, the different datasets used to train the various models likely contribute to the differences is generalization. For YeastMate, we deliberately compiled difficult large images at multiple illumination conditions that also contain isolated cells and debris in the background, not just images of single colonies of cells, as in the other datasets.

# Author contributions

Conceptualization: DB, CO, DH

Code: DB, JM, DH

Experiments and image acquisition: FT, CJ

Annotation of images: DB, JM, FT, DH

DB and DH wrote the manuscript with input from all authors. All authors approved the final version.

## Supplementary References

Dietler,N. *et al.* (2020) A convolutional neural network segments yeast microscopy images with high accuracy. *Nat Commun*, **11**, 5723.

Fischler,M.A. and Bolles,R.C. (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, **24**, 381–395.

He,K. *et al.* (2017) Mask r-cnn. In, *Proceedings of the IEEE international conference on computer vision.*, pp. 2961–2969.

Jakubke,C. *et al.* (2021) Cristae-dependent quality control of the mitochondrial genome. *Science Advances*, **in press**.

Linkert,M. *et al.* (2010) Metadata matters: access to image data in the real world. *The Journal of Cell Biology*, **189**, 777–782.

Lu,A.X. *et al.* (2019) YeastSpotter: accurate and parameter-free web segmentation for microscopy images of yeast cells. *Bioinformatics*, **35**, 4525–4527.

Padilla,R. *et al.* (2021) A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, **10**.

Pietzsch,T. *et al.* (2012) ImgLib2—generic image processing in Java. *Bioinformatics*, **28**, 3009–3011.

Ren,S. *et al.* (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In, Cortes,C. *et al.* (eds), *Advances in neural information processing systems 28.* Curran Associates, Inc., pp. 91–99.

Ronneberger,O. *et al.* (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In, Navab,N. *et al.* (eds), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015.* Springer International Publishing, Cham, pp. 234–241.

Rublee,E. *et al.* (2011) ORB: An efficient alternative to SIFT or SURF. In, *2011 International Conference on Computer Vision.* IEEE, Barcelona, Spain, pp. 2564–2571.

Rueden,C.T. *et al.* (2017) ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics*, **18**, 529.

Salem,D. *et al.* (2021) YeastNet: Deep-Learning-Enabled Accurate Segmentation of Budding Yeast Cells in Bright-Field Microscopy. *Applied Sciences*, **11**, 2692.

Schindelin,J. *et al.* (2012) Fiji: an open-source platform for biological-image analysis. *Nat Methods*, **9**, 676–682.

Sofroniew,N. *et al.* (2021) napari/napari: 0.4.11 Zenodo https://zenodo.org/record/5399494.

Vuola,A.O. *et al.* (2019) Mask-RCNN and U-Net Ensembled for Nuclei Segmentation. In, *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019).* IEEE, Venice, Italy, pp. 208–212.

Wu,Y. *et al.* (2019) Detectron2. *https://github.com/facebookresearch/detectron2.*

# Supplementary Tables

Supplementary Table ST1: Dataset and Imaging Conditions

Supplementary Table ST2: Single cell segmentation performance

**Supplementary Table ST1:** Overview of images included in the YeastMate dataset. Unless noted otherwise, we took the middle z-plane from stacks acquired for (Jakubke *et al.*, 2021) to train and test our network.

| *Identifier* | *Cell types* | *Imaging conditions* | *#images* | *#single cells* | *#mating events* | *#budding events* | *Notes* |
|---|---|---|---|---|---|---|---|
| *wt_dic_1001/ wt_dic_1002* | WT ATP6-NG<br>x<br>WT matrix mKate2 | Ti2, 100x NA 1.49, DIC,<br>pixel size: 110nm | 40 | 5338 | 958 | 697 | |
| *dmic60_dic* | *Δmic60* ATP6-NG<br>x<br>*Δmic60* matrix mKate2 | Ti2, 100x NA 1.49, DIC,<br>pixel size: 110nm | 24 | 4125 | 735 | 928 | |
| *wt_bf_1/ wt_bf_2* | WT ATP6-NG<br>x<br>WT matrix mKate2 | Ti2, 100x NA 1.49, brightfield,<br>Low illumination intensity,<br>pixel size: 110nm | 40 | 4217 | 556 | 602 | Preliminary data for (Jakubke *et al.*, 2021) |
| *wt_bf_ magnified_rep1* | WT ATP6-NG matrix TagBFP<br>x<br>WT ATP6-mKate2 matrix TagBFP | Ti2, 100x NA 1.49 + 1.5x magnification,<br>brightfield,<br>pixel size: 73nm | 5 | 339 | 8 | 123 | |
| *wt_bf_ magnified_rep2* | WT ATP6-NG matrix TagBFP<br>x<br>WT ATP6-mKate2 matrix TagBFP | Ti2, 100x NA 1.49 + 1.5x magnification,<br>brightfield,<br>pixel size: 73nm | 4 | 248 | 5 | 100 | |
| *wt_bf_ magnified_rep3* | WT ATP6-NG matrix TagBFP<br>x<br>WT ATP6-mKate2 matrix TagBFP | Ti2, 100x NA 1.49 + 1.5x magnification,<br>brightfield,<br>pixel size: 73nm | 19 | 983 | 64 | 384 | |
| *multiwell_bf_1* | WT ATP6-NG<br>x<br>WT matrix mKate2 | IX83, 60x NA 0.9, brightfield,<br>pixel size: 108nm | 9 | 872 | 52 | 330 | Preliminary data for high-throughput screen |
| *multiwell_bf_2* | WT ATP6-NG<br>x<br>WT matrix mKate2<br>/ *Δsrn2* matrix mKate2 | IX83, 60x NA 0.9, brightfield,<br>pixel size: 108nm | 6 | 936 | 2 | 451 | Preliminary data for high-throughput screen |
| *TOTAL* | | | 147 | 17058 | 2380 | 3615 | |

**Supplementary Table ST2:** Single cell detection (Precision, Recall, $F_1$) and segmentation (mean IoU of true positive detections) performance of YeastMate, YeastSpotter, YeaZ and YeastNet, with a 0.5 IoU threshold, shown as mean ± standard deviation across all images in a dataset. The best-performing tool per dataset and metric is highlighted by **bold** text and an asterisk (*). The results of YeaZ and YeastNet are grayed out and marked with a dagger (†) for their respective datasets, as those images were used to train the models and thus the results do not reflect performance on novel data.

A: YeastMate test dataset

| | PRECISION | RECALL | $F_1$ | MEAN IOU |
|---|---|---|---|---|
| **YEASTNET** | 0.256 ± 0.279 | 0.220 ± 0.248 | 0.257 ± 0.236 | 0.557 ± 0.265 |
| **YEASTSPOTTER** | 0.775 ± 0.099 | 0.860 ± 0.070 | 0.812 ± 0.073 | 0.820 ± 0.030 |
| **YEAZ** | 0.670 ± 0.121 | 0.816 ± 0.072 | 0.733 ± 0.096 | 0.797 ± 0.026 |
| **YEASTMATE** | **0.945 ± 0.032*** | **0.961 ± 0.031*** | **0.953 ± 0.024*** | **0.870 ± 0.019*** |

B: YeaZ dataset

| | PRECISION | RECALL | $F_1$ | MEAN IOU |
|---|---|---|---|---|
| **YEASTNET** | 0.602 ± 0.318 | 0.609 ± 0.280 | 0.608 ± 0.281 | 0.733 ± 0.134 |
| **YEASTSPOTTER** | 0.965 ± 0.087 | 0.915 ± 0.114 | 0.938 ± 0.074 | **0.875 ± 0.058*** |
| **YEAZ** | 0.991 ± 0.041† | 0.986 ± 0.045† | 0.988 ± 0.036† | 0.948 ± 0.025† |
| **YEASTMATE** | **0.967 ± 0.051*** | **0.966 ± 0.044*** | **0.966 ± 0.041*** | 0.795 ± 0.044 |

C: YeastNet / YIT dataset

| | PRECISION | RECALL | $F_1$ | MEAN IOU |
|---|---|---|---|---|
| **YEASTNET** | 0.986 ± 0.039† | 0.890 ± 0.074† | 0.934 ± 0.047† | 0.856 ± 0.053† |
| **YEASTSPOTTER** | 0.724 ± 0.229 | 0.839 ± 0.113 | 0.749 ± 0.175 | 0.763 ± 0.075 |
| **YEAZ** | 0.681 ± 0.236 | **0.969 ± 0.053*** | 0.771 ± 0.209 | **0.821 ± 0.064*** |
| **YEASTMATE** | **0.918 ± 0.075*** | 0.955 ± 0.051 | **0.935 ± 0.056*** | 0.804 ± 0.076 |

# Supplementary Figures

Supplementary Figure S1: Network architecture

Supplementary Figure S2: Postprocessing of network output

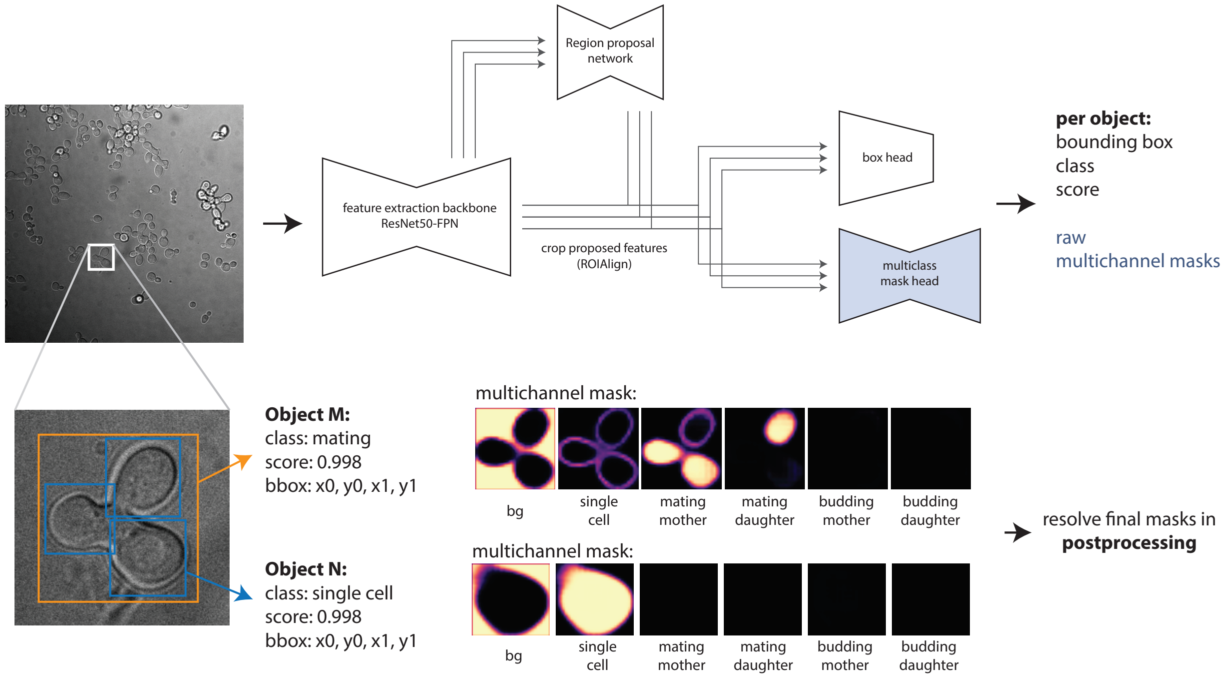
Supplementary Figure S3: Software architecture

Supplementary Figure S4: Screenshots of Electron frontend

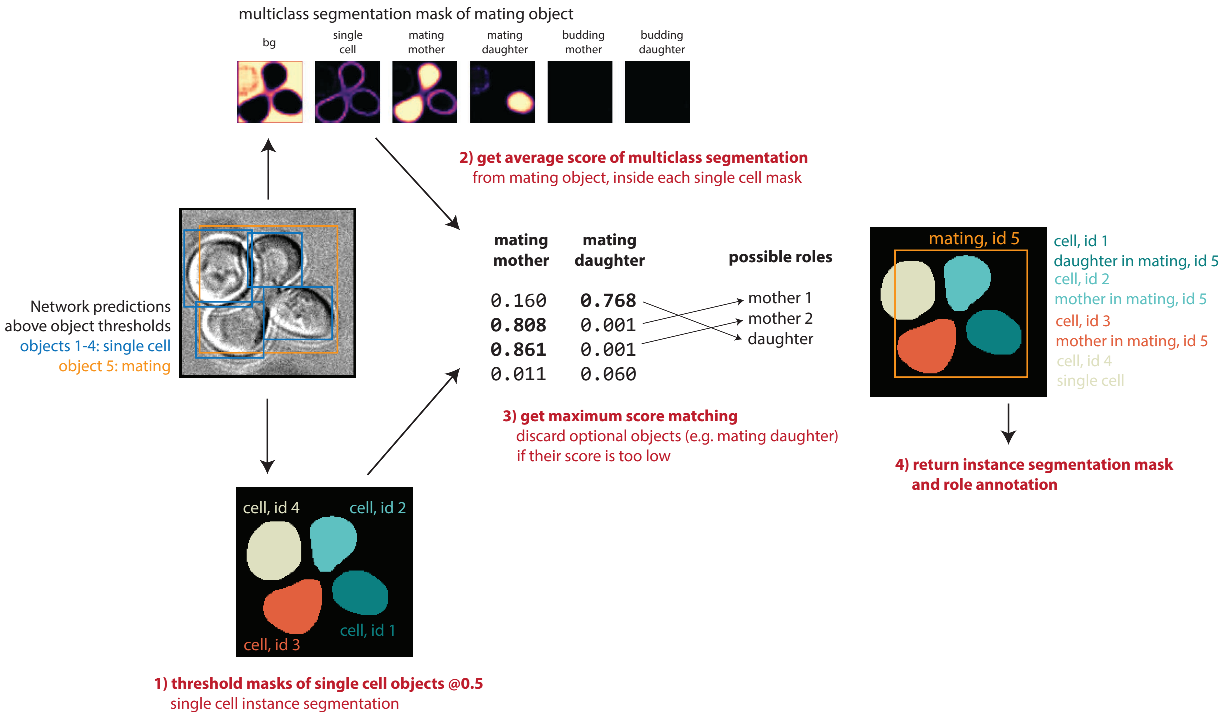
Supplementary Figure S5: Screenshots of Fiji frontend

Supplementary Figure S6: Object detection performance and comparison to human re-annotation

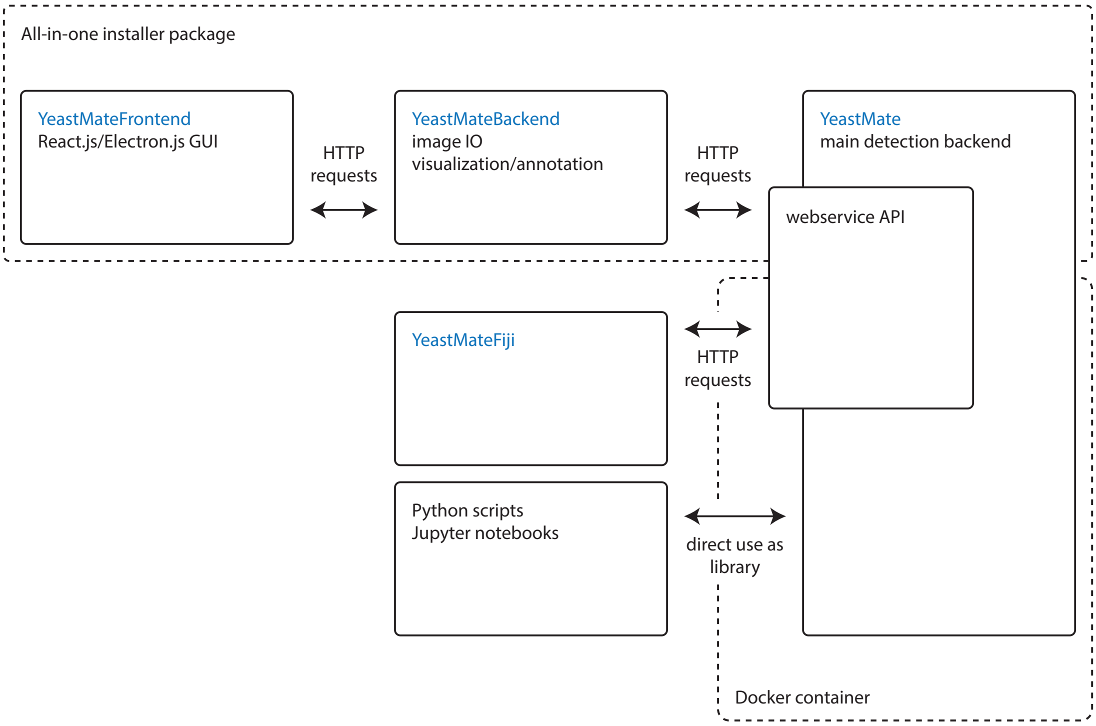
Supplementary Figure S7: Examples of single cell segmentation in comparison to existing tools



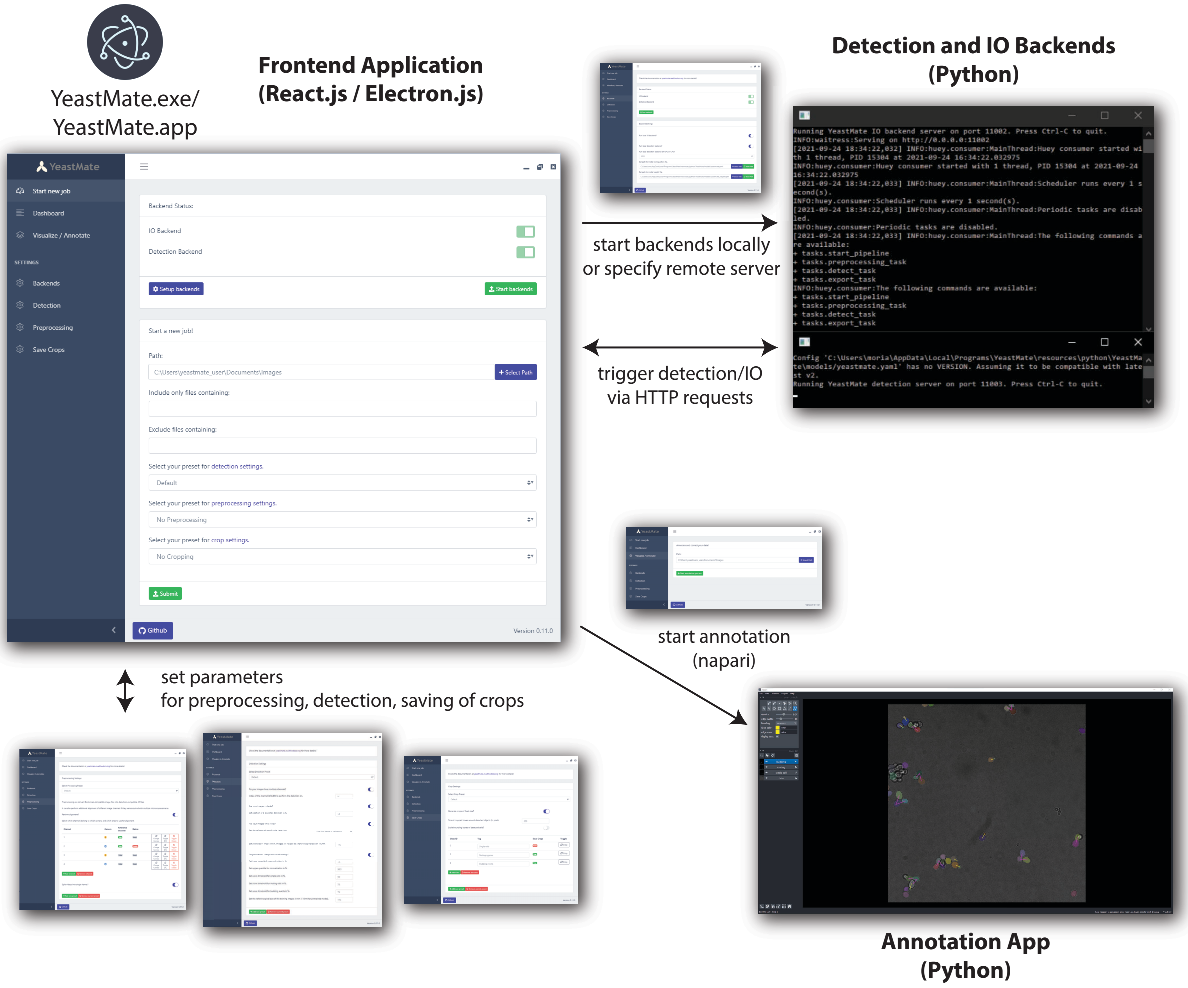
**Supplementary Figure S1:** Schematic Network architecture used in YeastMate: Images are fed through a modified Mask R-CNN with a ResNet50-FPN backbone (top). In addition to object detections with bounding boxes with object classes $c_{obj} \in \{\text{single cell, mating, budding}\}$, our modified mask segmentation head produces multichannel masks with segmentation classes $c_{seg} \in \{\text{background, single cell, mating mother, mating daughter, budding mother, budding daughter}\}$ (bottom).



**Supplementary Figure S2:** YeastMate Postprocessing pipeline: We take the outputs of our modified Mask R-CNN (objects with bounding boxes, scores and classes $\in \{single\ cell, mating, budding\}$, as well as raw segmentation masks with classes $\in \{background, single\ cell, mating\ mother, mating\ daughter, budding\ mother, budding\ daughter\}$ ) and perform the following steps: 1) for all single cell objects (with score above a user-settable threshold) we threshold the mask of the “single cell” class at 0.5, which already gives us an instance segmentation of individual cells 2) for objects of classes mating or budding, we then get the mean score for respective subclasses within the single cell mask of each cell in the bounding box and 3) calculate an optimal assignment to the “roles” in a lifecycle transition event, which gives us 4) the final results consisting of an instance segmentation with additional annotations for transition events.



**Supplementary Figure S3:** Software components of YeastMate (blue components indicate individual repositories available on GitHub): At its core lies the detection backend (YeastMate, right), which contains our modified Mask R-CNN implemented using PyTorch and Detectron2. It can either be installed as a Python package and directly used from custom scripts (examples are provided in the repository). The detection backend can also be served as a webservice, which provides an interface for our two GUI frontends: the Fiji plugin YeastMateFiji (middle) as well as our custom GUI frontend (top). This frontend consists of the actual UI written in React.js and packaged as a desktop application using Electron.js (YeastMateFrontend) which will spawn Python-based subprocesses for Image IO, visualization using napari or communication with the detection server. This functionality is implemented in the YeastMateBackend repository. For convenience, we compile YeastMateFrontend, YeastMateBackend and YeastMate into a single installer package. We also offer a Docker image for the YeastMate package for re-training networks or serving the detection backend from a reproducible environment.



Backend Status:

IO Backend

Detection Backend

Start a new job!

Path: C:\Users\yeastmate_user\Documents\Images

Include only files containing:

Exclude files containing:

Select your preset for detection settings. Default

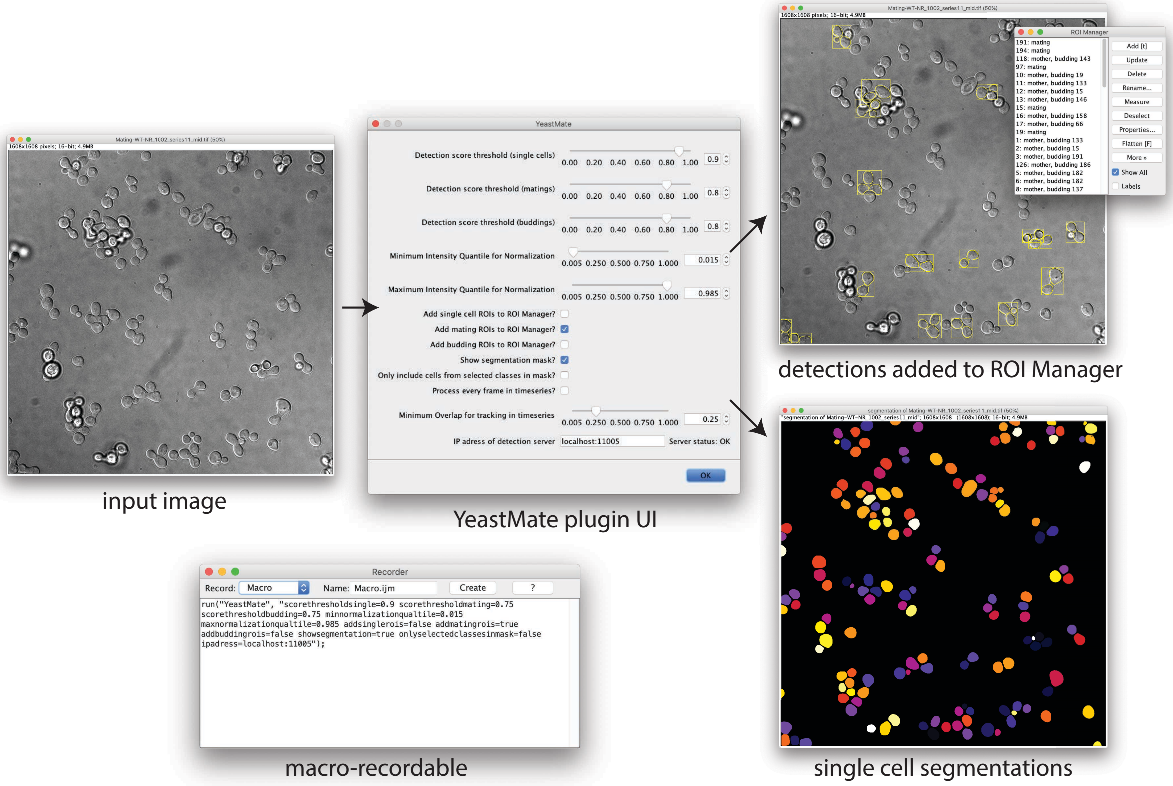
Select your preset for preprocessing settings. No Preprocessing

Select your preset for crop settings. No Cropping

```
Running YeastMate IO backend server on port 11002. Press Ctrl-C to quit.
INFO:waitress:Serving on http://0.0.0.0:11002
[2021-09-24 18:34:22,032] INFO:huey.consumer:MainThread:Huey consumer started wi
th 1 thread, PID 15304 at 2021-09-24 16:34:22.032975
INFO:huey.consumer:Huey consumer started with 1 thread, PID 15304 at 2021-09-24
16:34:22.032975
[2021-09-24 18:34:22,033] INFO:huey.consumer:MainThread:Scheduler runs every 1 s
econd(s).
INFO:huey.consumer:Scheduler runs every 1 second(s).
[2021-09-24 18:34:22,033] INFO:huey.consumer:MainThread:Periodic tasks are disab
led.
INFO:huey.consumer:Periodic tasks are disabled.
[2021-09-24 18:34:22,033] INFO:huey.consumer:MainThread:The following commands a
re available:
+ tasks.start_pipeline
+ tasks.preprocessing_task
+ tasks.detect_task
+ tasks.export_task
INFO:huey.consumer:The following commands are available:
+ tasks.start_pipeline
+ tasks.preprocessing_task
+ tasks.detect_task
+ tasks.export_task
```

```
Config 'C:\Users\moria\AppData\Local\Programs\YeastMate\resources\python\YeastMa
te\models/yeastmate.yaml' has no VERSION. Assuming it to be compatible with late
st v2.
Running YeastMate detection server on port 11003. Press Ctrl-C to quit.
```

**Supplementary Figure S4:** YeastMate standalone application. We package all components of YeastMate in an all-in-one package consisting of a GUI frontend based on React.js and Electron.js. The GUI forms an interface to separate Python-based processes for image IO (including preprocessing of non-TIFF files to meet our format requirements and export of crops around detected objects), detection and annotation. The respective backend processes will be started automatically by the frontend.



Detection score threshold (single cells) 0.9
Detection score threshold (matings) 0.8
Detection score threshold (buddings) 0.8
Minimum Intensity Quantile for Normalization 0.015
Maximum Intensity Quantile for Normalization 0.985
Add single cell ROIs to ROI Manager? ☐
Add mating ROIs to ROI Manager? ☑
Add budding ROIs to ROI Manager? ☐
Show segmentation mask? ☑
Only include cells from selected classes in mask? ☐
Process every frame in timeseries? ☐
Minimum Overlap for tracking in timeseries 0.25
IP adress of detection server localhost:11005 Server status: OK

191: mating
194: mating
118: mother, budding 143
97: mating
10: mother, budding 19
11: mother, budding 133
12: mother, budding 15
13: mother, budding 146
15: mating
16: mother, budding 158
17: mother, budding 66
19: mating
1: mother, budding 133
2: mother, budding 15
3: mother, budding 191
126: mother, budding 186
5: mother, budding 182
6: mother, budding 182
8: mother, budding 137

Record: Macro Name: Macro.ijm

```
run("YeastMate", "scorethresholdsingle=0.9 scorethresholdmating=0.75
scorethresholdbudding=0.75 minnormalizationqualtile=0.015
maxnormalizationqualtile=0.985 addsinglerois=false addmatingrois=true
addbuddingrois=false showsegmentation=true onlyselectedclassesinmask=false
ipadress=localhost:11005");
```

input image

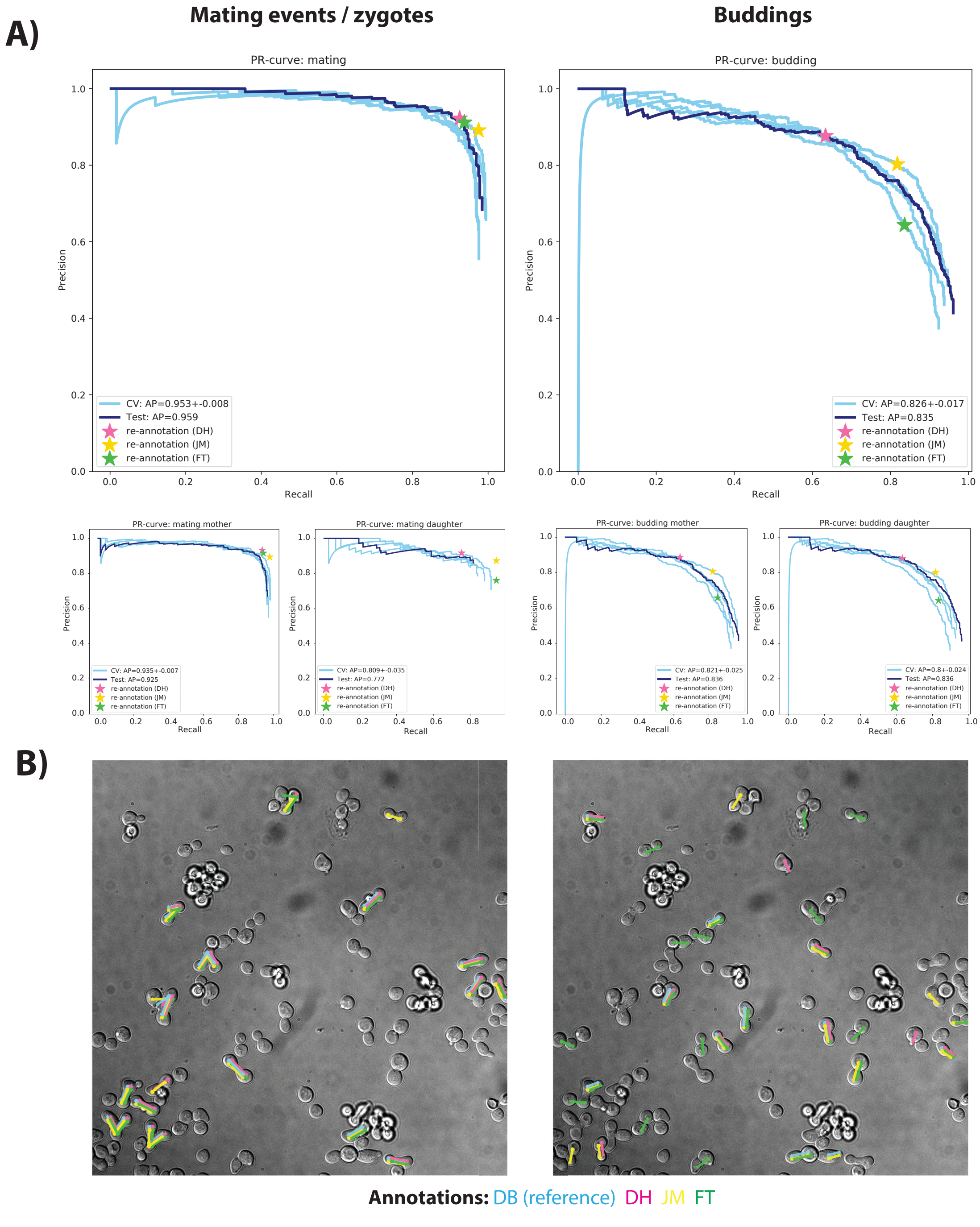
YeastMate plugin UI

detections added to ROI Manager

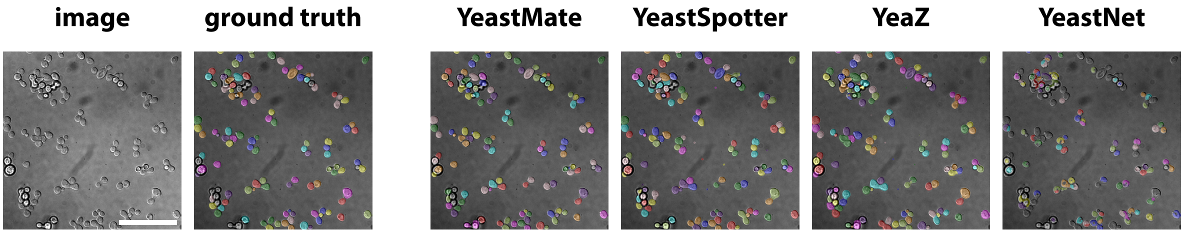
single cell segmentations

macro-recordable

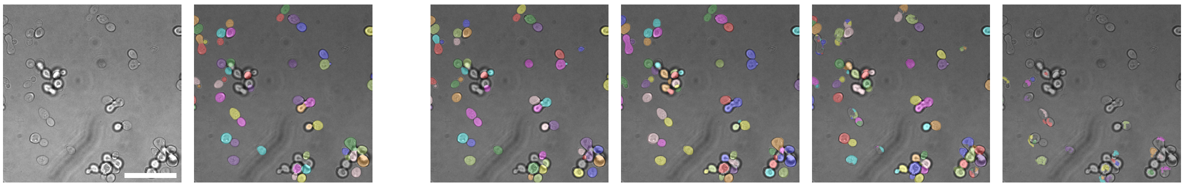
**Supplementary Figure S5:** Fiji-plugin frontend to YeastMate. The plugin can be from the Plugins menu under Plugins->YeastMate. Its GUI consists of a single dialog allowing the user to set score thresholds, normalization quantiles and what kind of output they want to display, as well as the IP of the detection server (a status message in the dialog will indicate if the server is reachable). Upon clicking “OK”, the plugin will connect to the detection server and transmit the currently active image for detection. Once it receives the results, it will add the bounding boxes or outlines of the detected objects (or just objects of specific classes, if desired by the user) to Fijis ROI Manager and display a single cell segmentation mask. If the corresponding options are set, the mask can be limited to cells participating in mating or budding events. The whole process can be recorded using Fijis Macro Recorder and included in macros for, e.g., batch processing of multiple files.



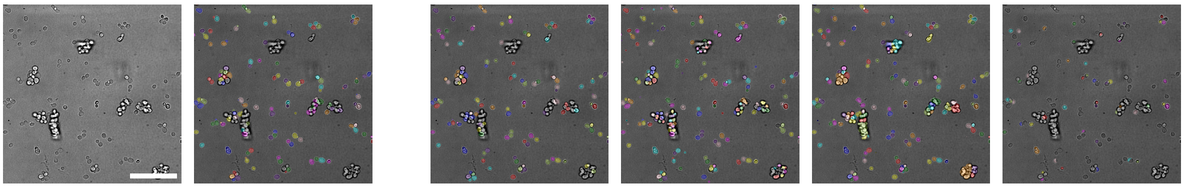
**Supplementary Figure S6:** Object detection performance. A, top) Precision-Recall curves for object detection performance (at 0.5 IoU threshold) for mating and budding events for the 4 cross-validation splits (light blue) and the final inference on the held-out test dataset (dark blue). Human re-annotation precision and recall for 3 re-annotators are shown as colored stars and closely coincide with the network curves. A, middle) Same curves as above, but for the mother and daughter subclasses of mating events and buddings. B): Visualization of the mating and budding annotations of the 4 annotators on an example image from the test set. Mating events are indicated by lines between the mother cells forming the zygote (circles) and, optionally, to a medial daughter bud (star). Buddings are indicated by a line from mother (circle) to daughter (star) cells. Note that shifts between the annotations of the different authors, as well as a random z-ordering of the annotations were introduced on purpose for this figure to make overlapping annotations easier to see.



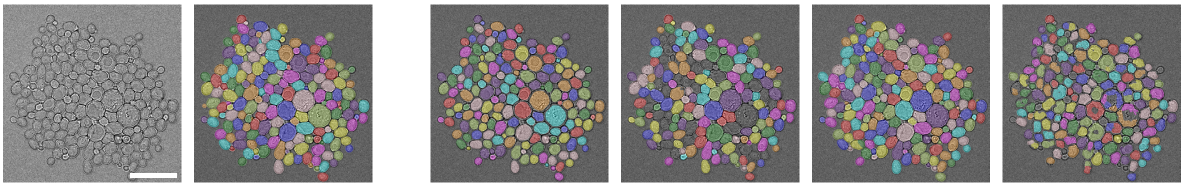
**A)** YeastMate test dataset, WT ATP6-NG x WT matrix mKate, 100x NA 1.49, scalebar: 50 µm



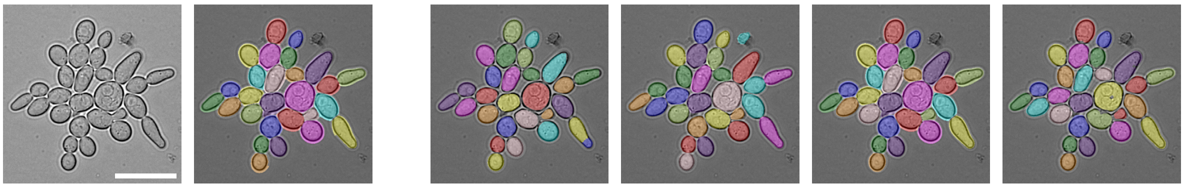
**B)** YeastMate test dataset, WT ATP6-NG matrix TagBFP x WT ATP6-mKate matrix TagBFP, 150x NA 1.49, scalebar: 30 µm



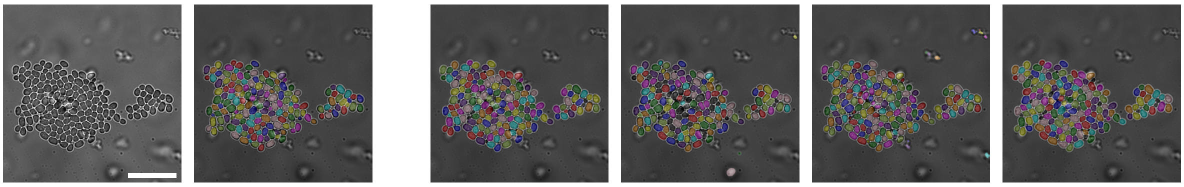
**C)** YeastMate test dataset, WT ATP6-NG x WT matrix mKate, 60x NA 0.9, scalebar: 50 µm



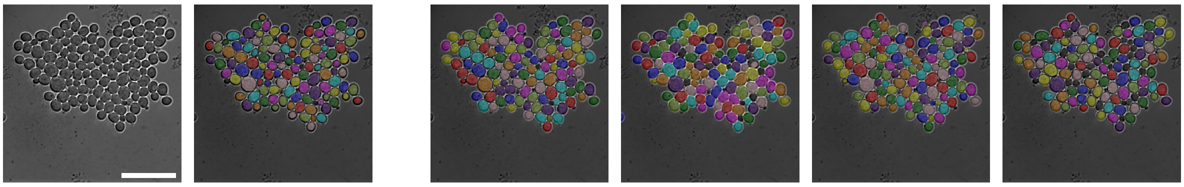
**D)** YeaZ brightfield dataset, WT, 60x magnification, low (1ms) illumination, scalebar: 20 µm



**E)** YeaZ brightfield dataset, filamentous G1/S (clb1-6Δ) arrested cells, 60x magnification, high (10ms) illumination, scalebar: 15 µm



**F)** YeastNet dataset DS2, timepoint 20, 60x NA 1.4, high defocus, scalebar: 30 µm



**G)** YIT dataset 3, timepoint 20, 100x NA 1.4, scalebar: 25 µm

**Supplementary Figure S7:** Single cell segmentation performance and comparison to existing deep learning-based tools for yeast segmentation on example images from our own test dataset (A-C) the YeaZ brightfield dataset (D,E) as well as the YeastNet/YIT dataset (F,G). Masks are overlaid on the original images in random colors using the label2rgb function from scikit-image.

# Chapter 3

# Discussion

DL can be a versatile tool decreasing the workload of researchers if wielded effectively. The last three years, during which this thesis was developed, saw considerable developments in DL and increased adoption of these methods in biomedical research. With the speed of developments seemingly not slowing down, developing practical toolkits becomes paramount to keep up with this progress. I hope that with my contributions, I shifted a few weights in my research areas in the direction of their gradient. In the following section, I will briefly discuss the studies included in this thesis and how my computational tools helped to gain new biological insights.

## 3.1 Encoding of variation in spatial enhancer activity

The study ‘Regulatory encoding of quantitative variation in spatial activity of a Drosophila enhancer’ investigated quantitative variation of spatial gene expression by analysing the spatial activity of the *spot* enhancer from the *yellow* gene in *Drosophila biarmipes* wings (Le Poul et al., 2020). The activity of this enhancer changes patterns in pupal wings which can be seen and analysed in microscopy images. Each wing was imaged in multiple high-resolution tiles, z-planes, and fluorescence channels, thus keeping the researcher waiting at the microscope until the next wing could be imaged.

To speed up the process of imaging the wings in the future, a new high-throughput imaging approach was developed. A *Nikon Ti2 Eclipse* microscope was fully automated, processing up to three inserted sample slides and autonomously imaging the wings in each slide while further passing the images into an end-to-end post-processing and analysis pipeline. This pipeline is fully hands-off after loading the microscope, thus allowing the researcher to spend the time on an imaging run, which might take hours, on other experiments instead. In the first step, the microscope was instructed to take a whole-slide, low-magnification, bright-field overview image. This image was fed into a Faster R-CNN model, similar to the original, unmodified model structure of YeastMate, to detect the coordinates of all wings on the slide. The microscope is then automatically instructed to take the detailed, time-consuming image acquisitions at the provided coordinates; successful acquisition then triggers further downstream analysis processes.

Before the switch to a Faster R-CNN architecture, the first iteration of this project was driven by a U-Net segmentation model (Ronneberger et al., 2015). While performing adequately, it also placed a big initial burden on the project members, as a large number of pixel-wise segmented wings needed to be annotated in the sampled overview images. While analysis tools often require object masks, automation of the microscope only requires object boundary coordinates, called a bounding box, to determine the location of the higher magnification acquisitions. Annotating only the bounding boxes instead of the mask increases annotation throughput drastically; the experience of this project suggests a 10-fold speedup. Besides being inherently faster to draw on an image, bounding boxes are less reliant on pixel-perfect annotation and can be drawn with a higher margin of error. This usually also allows bounding boxes to be annotated in a larger overview without having to zoom into each object. As discussed in section 1.4.1, generation of ground truth is often a limiting factor for deep learning applications, so switching to an object detection model for microscope automation, when possible, can make it easier to deploy and adapt the pipeline.

While less extensive than in YeastMate, a graphical user interface (GUI) integrated into a Jupyter Notebook was developed to make it possible for individual researchers to interface with the microscope without requiring programming language. This GUI, together with the microscope automation and interfacing code was deployed on a workstation directly connected to the microscope with the Nikon NIS-Elements software. Commands to the microscope were translated on-the-fly to the Nikon macro language and fed to NIS-Elements, which would perform the actual acquisition steps. After the acquisition of the overview image, it would be sent via XML-RPC to a dedicated GPU server, on which the deep learning model was deployed. Coordinates of the detected wings would then be sent back from the server to the workstation, which would then translate these coordinates into more acquisition calls for each wing. Individually acquired tiled images of a single wing would then have to be stitched into a single image, which is computationally intensive and had to be done on the dedicated server, after sending the image tiles back again. As the stitching would usually take longer than an image run, the individual stitching tasks were inserted into a task queue, finishing the stitching while still accepting new processing tasks.

This project highlighted some challenges when combining software solutions with complicated hardware. Finding the coordinates of a wing in an image is only the first step, as the coordinates only specify a pixel location within the camera's resolution and not a real-world location within the coordinate system of the microscope. Transforming such virtual observations into real-world instructions proved challenging and showcased the intersectionality that comes with being a bioinformatics researcher, requiring knowledge of both the computational side as well as practical knowledge of the tools and hardware used in biology.

## 3.2 Quality control of the mitochondrial genome

Mutations in the mitochondrial DNA (mtDNA) are connected to a wide range of inherited diseases including metabolic and neurological disorders (Lax et al., 2011). By studying the mechanisms controlling the clearance of mutant mtDNA, novel insights into the pathogenesis of these diseases could be gained (Gorman et al., 2016). Our study 'Cristae-dependent quality control of the mitochondrial genome' showed how cells of the model organism *S. cerevisiae* distinguish between functional and defective copies of mtDNA (Jakubke et al., 2021).

### 3.2.1 Cristae morphology influences mtDNA quality control

This study showed that *S. cerevisiae* cells can promote the generation of progeny with a healthy mtDNA population from heteroplasmic zygotes that contain WT and mutant mtDNA (Jakubke et al., 2021). A central point of the study is that purifying selection does not require mitochondrial fission but instead depends on intact cristae morphology. To support this hypothesis and the other methods presented in this study, an imaging approach was developed in which yeast mating zygotes were analysed with fluorescence microscopy. Because acquiring enough image samples of zygotes is a time-consuming process, we developed a tool facilitating this process, which served as the prototype of our later YeastMate software. We hope to encourage further microscopic studies in this area by streamlining this imaging process.

A limitation of this study was that the purifying selection in yeast was only demonstrated with a model mutant mtDNA variant that lacked the COB gene. Subsequent studies exploring deletions of other mtDNA-encoded genes could yield further insights; this could also be done on a large scale by working with strain libraries comprising thousands of yeast strains that carry gene deletions (Weill et al., 2018). Here, YeastMate, as a tool facilitating high-throughput microscopy, could enable the analysis of a very high amount of different strains. Preliminary tests on new images acquired in a high-throughput fashion from such a gene library yielded satisfying results but also revealed additional obstacles.

When working with large enough datasets, perfect recall in object detection was previously not imperative for our experiments as long as there was no inherent bias in which objects were missed. However, this might paradoxically not be true when working with even bigger datasets if not only the total number of images is increasing, but also the number of variables is growing even steeper. If e.g., 100.000 images are split over 10.000 different conditions, complete extraction of all information in each set of 10 images becomes paramount to reach the statistical significance of the results. This issue is further exacerbated by the inherent randomness of the high-throughput acquisition process, as image coordinates are not chosen based on their contents and quality as they would when manually acquiring them. When dealing with rarer biological occurrences like mating events, this can quickly lead to insufficient captured objects for some conditions. While problematic, it is also a potential avenue of improvement by employing microscope automatisation techniques to choose better image capture locations, similar to my other work described below (Le Poul et al., 2020).

### 3.2.2 YeastMate

Our study 'YeastMate: neural network-assisted segmentation of mating and budding events in *Saccharomyces cerevisiae*' (Bunk et al., 2022) was, on the one hand, an extension of the previous study, showcasing the computational methods used to further automate the connection between wet lab experiments and down-stream analyses. To this end, changes to the DL model underlying this toolkit were made to adapt to the specific requirements of the study. On the other hand, YeastMate was also developed into its own stand-alone application suitable for use in other labs and use cases. Here, the close collaboration between scientific end users and tool developers proved invaluable, as constant feedback about the tool's use in the daily lab routine led through a quick, iterative process to the user interface and experience accommodations now found in YeastMate.

### Detection model

As described in section 1.4.2 in detail, the major innovation of this work is the addition of a multiclass segmentation header to the Faster R-CNN model architecture (Ren et al., 2015). While the development of this tool was firmly rooted within the framework of analysing yeast cells, applications of this model architecture could reach further. Other life cycle states like meiotic asci in *Arabidopsis thaliana* might pose very similar tasks (Lim et al., 2020) that could be solved with YeastMate. Generally, cell divisions are an integral aspect in the life-cycle of all cells (Hunt et al., 2011), and being able to track pedigrees might be relevant to other research prospects. As the user interface of YeastMate includes a custom image annotation toolkit and an interface for retraining the underlying model on new data, new applications can be implemented within a few short steps.

With YeastMate, we wanted to mitigate the issues brought up in section 1.4.1 regarding datasets, benchmarks and reproducibility of AI experiments. Wagner et al. in 2022 showed in a meta-analysis that only approximately 50% of publications using DL models in computational pathology used an independent test set for evaluation, which might also not be publicly available. We could corroborate this observation with the few other available toolkits for yeast detection and segmentation, which either did not provide any public test data or did not have their own dataset in the first place. This makes comparing tools difficult, with the only other public yeast benchmark set being the Yeast Image Toolkit (YIT) (Versari et al., 2017). YeastMate now provides new datasets for training and benchmarking on an independent test set. While the YIT focuses on cell tracking and contains longer time series, YeastMate focuses on single time-point images and exceeds YIT in dataset size when counting independent frames.

To facilitate the generation of a big, annotated dataset, a custom annotation pipeline was developed using the *napari* annotation toolkit (Sofroniew et al., 2022). First, our model was solely pre-trained on single-cell detection and segmentation without the multi-mask header using publicly available yeast images from YeaZ (Dietler et al., 2020). A new dataset on our acquired images was then created by manually correcting predicted segmentations by the previous model. While still time-intensive, because each cell is still inspected individually to ensure annotation correctness, this approach reduced image annotation time by up to 50%. This procedure was performed iteratively on sub-sets of the whole dataset, improving model performance with each step and thus decreasing the time needed to correct the annotations. This highlighted the usefulness of publicly available datasets, as they can lessen the burden of a cold start of a model with no ground truth. Lifecycle transitions could then be annotated on top of the segmentation masks by simply clicking on the cells that belong together, generating a list of coordinates and a class label. These coordinates can be matched to the object IDs in the segmentation mask, out of which a bounding box can be reconstructed by combining the contained objects and returning their cumulative boundaries. This approach was not only necessary to connect the single cells to the annotated lifecycle transitions, as required by the multi-mask model, but also to be very fast, further streamlining the annotation process.

When comparing YeastMate's performance with YeaZ (Dietler et al., 2020), YeastSpotter (Lu et al., 2019) and YeastNet (Salem et al., 2021) on each other datasets, YeastMate is the only tool with consistently high performance across all datasets. In contrast, other tools suffer from worse generalisation performance. While the reasons for this are not very apparent and could be attested to many things, we suspect the composition of its dataset, rather than the model, significantly impacts YeastMate's generalisation performance. On the one hand, YeastMate's dataset consists of images from different microscopes, including brightfield and DIC, and different acquisition parameters like illumination intensity and magnification. This was extended with extensive data augmentation, further increasing the dataset's diversity and thus boosting the model's generalisation capabilities.

On the other hand, annotated images were first picked out from different, unlabeled sets of images. Besides focusing on increasing dataset diversity while doing this, images were picked by prioritising the most challenging images. The difficulty of images was mainly determined by how well previous prototype iterations of YeastMate performed on them. While not following any exact query strategy, this procedure mirrors some active learning techniques. These are commonly used in DL and can select samples from an unlabeled data pool that would most likely increase the model's performance when annotated (Kumar & Gupta, 2020).

An interesting observation arose when comparing generalisation and segmentation performance between model architectures. YeaZ and YeastNet used variations of the U-Net (Ronneberger et al., 2015) architecture, thus putting segmentation as the first step before object detection, as lined out in section 1.2.1. YeastSpotter and YeastMate, on the other hand, used Mask R-CNN (He et al., 2017) as their base model. Models using the U-Net architecture seemed to perform better at the segmentation part, which seems rational and can mainly be explained by Mask R-CNN's inability to segment parts of an object that lie outside the detected bounding box. Curiously, generalisation performance was worse in U-Net models. Unfortunately, as there were other huge differences like dataset sizes and composition, no conclusions can be drawn from this and would have been outside the scope of the study. Still, further insights on when best to use segmentation or object detection frameworks could be gained by studying the generalisation performance of these model architectures.

### Software architecture

When developing YeastMate into its stand-alone tool, we tried to combine the positive user experience of a GUI with flexible interoperability with existing tools and workflows. The GUI was built as a web application with React.js and bundled into a local application with Electron. While React.js was the most popular web development framework in 2022 (Vailshery, 2022), the use of it or web user interfaces, in general, remains rare in biological research. The use of web technologies added complexity to our tool but also added the possibility of deploying YeastMate, as well as further projects building upon its codebase, as a remote web application. A remote server acting as a service provider can help the end-user, as they will not have to worry about any technical requirements, as well as the tool developer, who can directly access the host system to debug and monitor the system. As mentioned in section 1.4.1, sufficient computational resources might not always be available for each researcher, and while YeastMate was explicitly designed to even work on low-grade systems, its use in high-throughput pipelines might require additional computing power. To mitigate this, YeastMate's architecture allows it to be easily set up in cloud environments, such as AWS, Microsoft Azure or Google cloud, following established workflows enabling deployment in a few clicks (Juturu & Yoon, 2018).

This focus can also be seen in the modularity of YeastMate's building blocks (Fig. 3.1). Each block can be independently deployed and can communicate via HTTP requests with internal or external components. At first sight, this might seem like a disproportionate effort for a scientific project, but we hoped two achieve two goals with this operational architecture. On a broader scale, YeastMate might serve as a knowledge source for future projects. As discussed in section 1.4.3, knowledge of

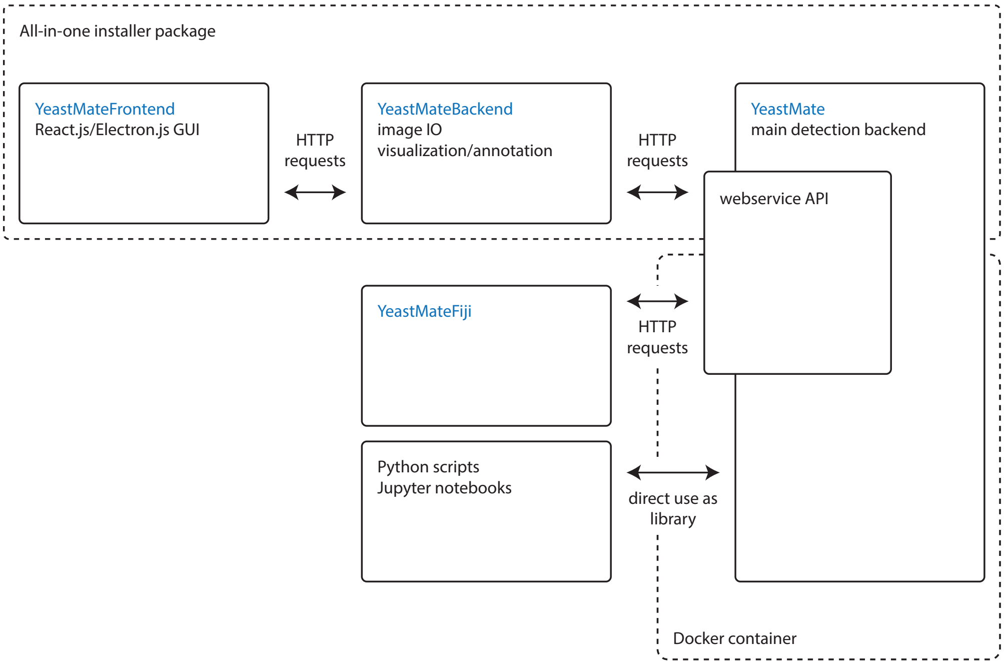


Figure 3.1: Operational architecture of the YeastMate package (Bunk et al., 2022).

best practices in model deployment is still shifting and can be hard to find. While we currently do not provide cookie-cutter templates of our architecture, its modular approach allows other researchers to adapt and incorporate individual components of our system into their projects.

The second goal was interoperability with other systems. While a user interface expands the pool of potential users, many researchers understandably prefer to use new tools as programming scripts or packages or in conjunction with existing tools like Fiji (Schindelin et al., 2012), which is omnipresent in image analysis. While our DL model architecture proved challenging to implement directly in Fiji-compatible ways, we developed a Fiji plugin acting as a client that can communicate with the same analysis backend used by our user interface.

To our knowledge, YeastMate is already used in some labs and has been included in the Cell-ACDC tracking framework (Padovani et al., 2022). Even though cell segmentation has been an afterthought while building YeastMate as a tool for detecting lifecycle transitions, we have received positive feedback and interest regarding whole-image, single-cell segmentation, further attesting to YeastMate's generalisation performance and simplicity in use.

# Appendix A

# Bibliography

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Alla, S., & Adari, S. K. (2021). What is mlops? In *Beginning mlops with mlflow* (pp. 79–124). Springer.

Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* Oxford University Press.

Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C. (2020). On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, *117*(48), 30088–30095.

Araujo, A., Norris, W., & Sim, J. (2019). Computing receptive fields of convolutional neural networks. *Distill*, *4*(11), e21.

Baldi, P., & Brunak, S. (2001). *Bioinformatics: The machine learning approach*. MIT press.

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., & Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, *76*(4), 695–711.

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., et al. (2019). Ilastik: Interactive machine learning for (bio) image analysis. *Nature methods*, *16*(12), 1226–1232.

Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., Bonifacino, J. S., Davidson, M. W., Lippincott-Schwartz, J., & Hess, H. F. (2006). Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, *313*(5793), 1642–1645.

Bock, S., & Weiß, M. (2019). A proof of local convergence for the adam optimizer. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Breker, M., & Schuldiner, M. (2014). The emergence of proteome-wide technologies: Systematic analysis of proteins comes of age. *Nature Reviews Molecular Cell Biology*, *15*(7), 453–464.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Brunelli, R. (2009). *Template matching techniques in computer vision: Theory and practice*. John Wiley & Sons.

Bunk, D., Moriasy, J., Thoma, F., Jakubke, C., Osman, C., & Hörl, D. (2022). Yeastmate: Neural network-assisted segmentation of mating and budding events in saccharomyces cerevisiae. *Bioinformatics*, *38*(9), 2667–2669.

Caicedo, J. C., Goodman, A., Karhohs, K. W., Cimini, B. A., Ackerman, J., Haghighi, M., Heng, C., Becker, T., Doan, M., McQuin, C., et al. (2019). Nucleus segmentation across imaging experiments: The 2018 data science bowl. *Nature methods*, *16*(12), 1247–1253.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *European conference on computer vision*, 213–229.

Casadevall, A., & Fang, F. C. (2014). Specialized science. *Infection and immunity*, *82*(4), 1355–1360.

Chalmers, A. F. (2013). *What is this thing called science?* Hackett Publishing.

Chen, H., Li, C., Wang, G., Li, X., Rahaman, M. M., Sun, H., Hu, W., Li, Y., Liu, W., Sun, C., et al. (2022). Gashis-transformer: A multi-scale visual transformer approach for gastric histopathological image detection. *Pattern Recognition*, *130*, 108827.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, *15*(141), 20170387.

Comer, M. L., & Delp III, E. J. (1999). Morphological operations for color image processing. *Journal of electronic imaging*, *8*(3), 279–289.

Cortes, C., & Vapnik, V. (1995). Support vector machine. *Machine learning*, *20*(3), 273–297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

Crick, F. (1989). The recent excitement about neural networks. *Nature*, *337*(6203), 129–132.

Davidson, S., & Furber, S. B. (2021). Comparison of artificial and spiking neural networks on digital hardware. *Frontiers in Neuroscience*, *15*, 651141.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255.

Dietler, N., Minder, M., Gligorovski, V., Economou, A. M., Joly, D. A. H. L., Sadeghi, A., Chan, C. H. M., Koziński, M., Weigert, M., Bitbol, A.-F., et al. (2020). Yeaz: A convolutional neural network for highly accurate, label-free segmentation of yeast microscopy images. *bioRxiv*.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, *12*(7).

Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, *2*(1), 1–12.

Floridi, L., & Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, *30*(4), 681–694.

Frangi, A. F., Niessen, W. J., Vincken, K. L., & Viergever, M. A. (1998). Multiscale vessel enhancement filtering. *International conference on medical image computing and computer-assisted intervention*, 130–137.

Gauch Jr, H. G., & Gauch, H. G. (2003). *Scientific method in practice*. Cambridge University Press.

Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Godfrey-Smith, P. (2009). Theory and reality. In *Theory and reality*. University of Chicago Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gorman, G. S., Chinnery, P. F., DiMauro, S., Hirano, M., Koga, Y., McFarland, R., Suomalainen, A., Thorburn, D. R., Zeviani, M., & Turnbull, D. M. (2016). Mitochondrial diseases. *Nature reviews Disease primers*, *2*(1), 1–22.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, *11*(1), 23–63.

Gulli, A., & Pal, S. (2017). *Deep learning with keras*. Packt Publishing Ltd.

Gustafsson, M. G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of microscopy*, *198*(2), 82–87.

Hacking, I. (1985). Do we see through a microscope? *Images of science*, 132–52.

Haralick, R. M., Sternberg, S. R., & Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4), 532–550.

Hassabis, D., Kumaran, D., Summerfield, C., & Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, *95*(2), 245–258.

Hasson, U., Nastase, S. A., & Goldstein, A. (2020). Direct fit to nature: An evolutionary perspective on biological and artificial neural networks. *Neuron*, *105*(3), 416–434.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hell, S. W., & Wichmann, J. (1994). Breaking the diffraction resolution limit by stimulated emission: Stimulated-emission-depletion fluorescence microscopy. *Optics letters*, *19*(11), 780–782.

Hey, T., & Trefethen, A. (2003). The data deluge: An e-science perspective. *Grid computing: Making the global infrastructure a reality*, *72*, 809–824.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd international conference on document analysis and recognition*, *1*, 278–282.

Hörl, D., Rojas Rusak, F., Preusser, F., Tillberg, P., Randel, N., Chhetri, R. K., Cardona, A., Keller, P. J., Harz, H., Leonhardt, H., et al. (2019). Bigstitcher: Reconstructing high-resolution image datasets of cleared and expanded samples. *Nature methods*, *16*(9), 870–874.

Huang, G.-B., & Babri, H. A. (1998). Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE transactions on neural networks*, *9*(1), 224–229.

Hunt, T., Nasmyth, K., & Novák, B. (2011). The cell cycle. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *366*(1584), 3494–3497. https://doi.org/10.1098/rstb.2011.0274

Ioannidis, J. P., Fanelli, D., Dunne, D. D., & Goodman, S. N. (2015). Meta-research: Evaluation and improvement of research methods and practices. *PLoS biology*, *13*(10), e1002264.

Jakubke, C., Roussou, R., Maiser, A., Schug, C., Thoma, F., Bunk, D., Hörl, D., Leonhardt, H., Walter, P., Klecker, T., et al. (2021). Cristae-dependent quality control of the mitochondrial genome. *Science advances*, *7*(36), eabi8886.

Juturu, A., & Yoon, D. (2018). Designing and deploying a cloud application using azure. *Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE)*, 110–115.

Kaas, J. H. (2013). The evolution of brains from early mammals to humans. *Wiley Interdisciplinary Reviews: Cognitive Science*, *4*(1), 33–45.

Kanopoulos, N., Vasanthavada, N., & Baker, R. L. (1988). Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, *23*(2), 358–367.

Kayhan, O. S., & Gemert, J. C. v. (2020). On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14274–14285.

Keskar, N. S., & Socher, R. (2018). Improving generalization performance by switching from adam to sgd. *International Conference on Learning Representations (ICLR).*

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR).*

Kitchin, R. (2014). *The data revolution: Big data, open data, data infrastructures and their consequences*. Sage.

Krenker, A., Bešter, J., & Kos, A. (2011). Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*, 1–18.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks (F. Pereira, C. Burges, L. Bottou & K. Weinberger, Eds.). *25*.

Krull, A., Buchholz, T.-O., & Jug, F. (2019). Noise2void-learning denoising from single noisy images. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2129–2137.

Kumar, P., & Gupta, A. (2020). Active learning query strategies for classification, regression, and clustering: A survey. *Journal of Computer Science and Technology*, *35*(4), 913–945. https://doi.org/10.1007/s11390-020-9487-4

Lax, N. Z., Turnbull, D. M., & Reeve, A. K. (2011). Mitochondrial mutations: Newly discovered players in neuronal degeneration. *The Neuroscientist*, *17*(6), 645–658.

Le, K. H., Tran, T. V., Pham, H. H., Nguyen, H. T., Le, T. T., & Nguyen, H. Q. (2022). Learning from multiple expert annotators for enhancing anomaly detection in medical image analysis. *arXiv preprint arXiv:2203.10611*.

Le Poul, Y., Xin, Y., Ling, L., Mühling, B., Jaenichen, R., Hörl, D., Bunk, D., Harz, H., Leonhardt, H., Wang, Y., et al. (2020). Regulatory encoding of quantitative variation in spatial activity of a drosophila enhancer. *Science advances*, *6*(49), eabe2955.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.

Leonelli, S. (2014). What difference does quantity make? on the epistemology of big data in biology. *Big data & society*, *1*(1), 2053951714534395.

Lichtman, J. W., & Conchello, J.-A. (2005). Fluorescence microscopy. *Nature methods*, *2*(12), 910–919.

Lillicrap, T. P., Cownden, D., Tweed, D. B., & Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, *7*(1), 1–10.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, *21*(6), 335–346.

Lim, E.-C., Kim, J., Park, J., Kim, E.-J., Kim, J., Park, Y. M., Cho, H. S., Byun, D., Henderson, I. R., Copenhaver, G. P., et al. (2020). Deeptetrad: High-throughput image analysis of meiotic tetrads by deep learning in arabidopsis thaliana. *The Plant Journal*, *101*(2), 473–483.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision*, 740–755.

Linnainmaa, S. (1970). *The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors* (Master's thesis). Univ. Helsinki.

Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, *21*(3), 105–117.

Liu, M.-Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, *30*.

Ljosa, V., Sokolnicki, K. L., & Carpenter, A. E. (2012). Annotated high-throughput microscopy image sets for validation. *Nature methods*, *9*(7), 637–637.

Longchamp, J.-N., Rauschenbach, S., Abb, S., Escher, C., Latychevskaia, T., Kern, K., & Fink, H.-W. (2017). Imaging proteins at the single-molecule level. *Proceedings of the National Academy of Sciences*, *114*(7), 1474–1479.

Lu, A. X., Zarin, T., Hsu, I. S., & Moses, A. M. (2019). YeastSpotter: Accurate and parameter-free web segmentation for microscopy images of yeast cells (R. Murphy, Ed.). *Bioinformatics*, *35*(21), 4525–4527. https://doi.org/10.1093/bioinformatics/btz402

Macpherson, T., Churchland, A., Sejnowski, T., DiCarlo, J., Kamitani, Y., Takahashi, H., & Hikida, T. (2021). Natural and artificial intelligence: A brief introduction to the interplay between ai and neuroscience research. *Neural Networks*, *144*, 603–613.

MacQueen, J. (1967). Classification and analysis of multivariate observations. *5th Berkeley Symp. Math. Statist. Probability*, 281–297.

Magidson, V., & Khodjakov, A. (2013). Circumventing photodamage in live-cell microscopy. *Methods in cell biology*, *114*, 545–560.

Maqsood, M., Nazir, F., Khan, U., Aadil, F., Jamal, H., Mehmood, I., & Song, O.-y. (2019). Transfer learning assisted classification and detection of alzheimer's disease stages using 3d mri scans. *Sensors*, *19*(11), 2645.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information.* MIT Press.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, *21*(9), 1281–1289.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Minsky, M., & Papert, S. A. (1969). *Perceptrons: An introduction to computational geometry*. MIT press.

Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., & Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. *2011 IEEE international conference on signal and image processing applications (ICSIPA)*, 342–347.

Narayan, S. (1997). The generalized sigmoid activation function: Competitive supervised learning. *Information sciences*, *99*(1-2), 69–82.

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62–66.

Padovani, F., Mairhörmann, B., Falter-Braun, P., Lengefeld, J., & Schmoller, K. M. (2022). Segmentation, tracking and cell cycle analysis of live-cell imaging data with cell-ACDC. *BMC Biology*, *20*(1). https://doi.org/10.1186/s12915-022-01372-6

Pang, Y., Lin, J., Qin, T., & Chen, Z. (2021). Image-to-image translation: Methods and applications. *IEEE Transactions on Multimedia*.

Papert, S. A. (1966). The summer vision project.

Pegoraro, G., & Misteli, T. (2017). High-throughput imaging for the discovery of cellular mechanisms of disease. *Trends in Genetics*, *33*(9), 604–615.

Pepperkok, R., & Ellenberg, J. (2006). High-throughput fluorescence microscopy for systems biology. *Nature reviews Molecular cell biology*, *7*(9), 690–696.

Pietzsch, T., Saalfeld, S., Preibisch, S., & Tomancak, P. (2015). Bigdataviewer: Visualization and processing for large image data sets. *Nature methods*, *12*(6), 481–483.

Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., & Zhang, C. (2021). Defrcn: Decoupled faster r-cnn for few-shot object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8681–8690.

Raykar, V. C., & Saha, A. (2015). Data split strategiesfor evolving predictive models. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 3–19.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, *22*(11), 1761–1770.

Rivenson, Y., Liu, T., Wei, Z., Zhang, Y., de Haan, K., & Ozcan, A. (2019). Phasestain: The digital staining of label-free quantitative phase microscopy images using deep learning. *Light: Science & Applications*, *8*(1), 1–11.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Roerdink, J. B., & Meijster, A. (2000). The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, *41*(1-2), 187–228.

Rokach, L., & Maimon, O. (2005). Decision trees. In *Data mining and knowledge discovery handbook* (pp. 165–192). Springer.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, 234–241.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rust, M. J., Bates, M., & Zhuang, X. (2006). Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm). *Nature methods*, *3*(10), 793–796.

Salamov, A. A., & Solovyev, V. V. (1995). Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *Journal of molecular biology*, *247*(1), 11–15.

Salem, D., Li, Y., Xi, P., Phenix, H., Cuperlovic-Culf, M., & Kærn, M. (2021). YeastNet: Deep-learning-enabled accurate segmentation of budding yeast cells in bright-field microscopy. *Applied Sciences*, *11*(6), 2692. https://doi.org/10.3390/app11062692

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210–229.

Sapoval, N., Aghazadeh, A., Nute, M. G., Antunes, D. A., Balaji, A., Baraniuk, R., Barberan, C., Dannenfelser, R., Dun, C., Edrisi, M., et al. (2022). Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, *13*(1), 1–12.

Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., & Kikinis, R. (1998). Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical image analysis*, *2*(2), 143–168.

Scellier, B., & Bengio, Y. (2017). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, *11*, 24.

Scherf, N., & Huisken, J. (2015). The smart and gentle microscope. *Nature biotechnology*, *33*(8), 815–818.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: An open-source platform for biological-image analysis. *Nature methods*, *9*(7), 676–682.

Schmidt, U., Weigert, M., Broaddus, C., & Myers, G. (2018). Cell detection with star-convex polygons. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 265–273.

Schweikert, G., Rätsch, G., Widmer, C., & Schölkopf, B. (2008). An empirical analysis of domain adaptation algorithms for genomic sequence analysis. *Advances in neural information processing systems*, *21*.

Shapiro, L. G., Stockman, G. C., et al. (2001). *Computer vision* (Vol. 3). Prentice Hall New Jersey.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*.

Sintorn, I.-M., Homman-Loudiyi, M., Söderberg-Nauclér, C., & Borgefors, G. (2004). A refined circular template matching method for classification of human cytomegalovirus capsids in tem images. *Computer methods and programs in biomedicine*, *76*(2), 95–102.

Sofroniew, N., Lambert, T., Evans, K., Nunez-Iglesias, J., Bokota, G., Winston, P., Peña-Castellanos, G., Yamauchi, K., Bussonnier, M., Doncila Pop, D., Can Solak, A., Liu, Z., Wadhwa, P., Burt, A., Buckley, G., Sweet, A., Migas, L., Hilsenstein, V., Gaifas, L., . . . McGovern, A. (2022). Napari: A multi-dimensional image viewer for python. https://doi.org/10.5281/ZENODO.3555620

Stanton, J. M. (2001). Galton, pearson, and the peas: A brief history of linear regression for statistics instructors. *Journal of Statistics Education*, *9*(3).

Stelzer, E. H. (2002). Beyond the diffraction limit? *Nature*, *417*(6891), 806–807.

Stigler, S. M. (1981). Gauss and the invention of least squares. *the Annals of Statistics*, 465–474.

Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: A generalist algorithm for cellular segmentation. *Nature methods*, *18*(1), 100–106.

Szeliski, R. (2022). *Computer vision: Algorithms and applications*. Springer Nature.

Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the turing test* (pp. 23–65). Springer.

Vailshery, L. S. (2022). Most popular web frameworks among developers worldwide 2022. Retrieved December 12, 2022, from https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/

Versari, C., Stoma, S., Batmanov, K., Llamosi, A., Mroz, F., Kaczmarek, A., Deyell, M., Lhoussaine, C., Hersen, P., & Batt, G. (2017). Long-term tracking of budding yeast cells in brightfield microscopy: CellStar and the evaluation platform. *Journal of The Royal Society Interface*, *14*(127), 20160705.

von Chamier, L., Laine, R. F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., Lerche, M., Hernández-Pérez, S., Mattila, P. K., Karinou, E., et al. (2021). Democratising deep learning for microscopy with zerocostdl4mic. *Nature communications*, *12*(1), 1–18.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*.

Wagner, S. J., Matek, C., Boushehri, S. S., Boxberg, M., Lamm, L., Sadafi, A., Waibel, D. J., Marr, C., & Peng, T. (2022). Built to last? reproducibility and reusability of deep learning algorithms in computational pathology. *medRxiv*.

Wang, Y. E., Wei, G.-Y., & Brooks, D. (2019). Benchmarking tpu, gpu, and cpu platforms for deep learning. *arXiv preprint arXiv:1907.10701*.

Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., et al. (2018). Content-aware image restoration: Pushing the limits of fluorescence microscopy. *Nature methods*, *15*(12), 1090–1097.

Weill, U., Yofe, I., Sass, E., Stynen, B., Davidi, D., Natarajan, J., Ben-Menachem, R., Avihou, Z., Goldman, O., Harpaz, N., Chuartzman, S., Kniazev, K., Knoblach, B., Laborenz, J., Boos, F., Kowarzyk, J., Ben-Dor, S., Zalckvar, E., Herrmann, J. M.,

. . . Schuldiner, M. (2018). Genome-wide SWAp-tag yeast libraries for proteome exploration. *Nature Methods*, *15*(8), 617–622. https://doi.org/10.1038/s41592-018-0044-9

Xu, Y., Liu, X., Cao, X., Huang, C., Liu, E., Qian, S., Liu, X., Wu, Y., Dong, F., Qiu, C.-W., et al. (2021). Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, *2*(4), 100179.

Yee, A. A., Savchenko, A., Ignachenko, A., Lukin, J., Xu, X., Skarina, T., Evdokimova, E., Liu, C. S., Semesi, A., Guido, V., et al. (2005). Nmr and x-ray crystallography, complementary tools in structural proteomics of small proteins. *Journal of the American Chemical Society*, *127*(47), 16512–16517.

Zhou, Z.-H. (2021). *Machine learning*. Springer Nature.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.

# Appendix B

# Abbreviations

**AI** artificial intelligence

**ANN** artificial neural network

**BCE** binary cross-entropy

**CE** categorical cross-entropy

**CNN** convolutional neural network

**CV** Computer Vision

**DL** deep learning

**GUI** graphical user interface

**HCS** high content screening

**HTM** high-throughput microscopy

**MAE** mean absolute error

**ML** machine learning

**MSE** mean squared error

**mtDNA** mitochondrial DNA

**ReLU** rectified linear unit

**RPN** region proposal network

**SGD** stochastic gradient descent

**SNR** signal-to-noise ratio

**SVM** support vector machine

**tanh** hyperbolic tangent

**YIT** Yeast Image Toolkit

# Appendix C

# List of Publications

Le Poul, Y., Xin, Y., Ling, L., Mühling, B., Jaenichen, R., Hörl, D., Bunk, D., Harz, H., Leonhardt, H., Wang, Y., et al. Regulatory encoding of quantitative variation in spatial activity of a drosophila enhancer. *Science advances*, *6*(49)

Jakubke, C., Roussou, R., Maiser, A., Schug, C., Thoma, F., Bunk, D., Hörl, D., Leonhardt, H., Walter, P., Klecker, T., et al. Cristae-dependent quality control of the mitochondrial genome. *Science advances*, *7*(36)

Bunk, D., Moriasy, J., Thoma, F., Jakubke, C., Osman, C., & Hörl, D. Yeastmate: Neural network-assisted segmentation of mating and budding events in saccharomyces cerevisiae. *Bioinformatics*, *38*(9)

# Appendix D

# Curriculum Vitae

David Bunk

**Education**

**2019 -** doctorate, Fakultät für Biologie, LMU München

**2014-2019** studies of biology, LMU München

**2006-2014** high school (Gymnasium), HLG Landshut

**Academic titles**

**28.9.2019** M.Sc. in biology, LMU München

**17.9.2017** B.Sc. in biology, LMU München