Bachelor's Thesis

Improving Automated Identification of Trypanosoma Brucei Cell Cycle Stages through Enhanced Features and Selection Techniques

submitted by

Edgar Wolfert
born 30.07.2000 in Dresden

Technische Universität Dresden

Faculty of Computer Science
Institute of Artificial Intelligence
Chair of Scientific Computing for Systems Biology

Supervisors:
Dr. Nandu Gopan

Professor:
Prof. Dr. Ivo F. Sbalzarini

Submitted April 11, 2024

Confirmation

I confirm that I independently prepared this thesis with the title *Improving Automated Identification of Trypanosoma Brucei Cell Cycle Stages through Enhanced Features and Selection Techniques* and that I used only the references and auxiliary means indicated in the thesis.

Dresden, April 11, 2024

Edgar Wolfert

Digital unterschrieben von Edgar Wolfert
Datum: 2024.04.11 15:31:15 +02'00'

Edgar Wolfert

Wolfert

Contents

1 Introduction

1.1 Background and Significance

Machine Learning presents an opportunity for computers to discover and discern patterns in large amounts of data, which is partly why it is becoming increasingly important. It is invaluable for science because it offers a way out of a fundamental bottleneck in the scientific method: data analysis. It accelerates data analysis, is reproducible and can reduce bias. It has revolutionised many areas of science and has greatly affected the public.

The breakthrough in Image Recognition of AlexNet in 2012 sparked a new era of Machine Learning research and led to many new inventions [KSH12]. One of the most relevant inventions was AlphaFold, which has radically transformed the field of protein folding, making predictions of protein structures accurate [Jum+21]. Furthermore, the Transformer architecture revolutionised all kinds of sequence modelling. This includes better vision models like Vision Transformers (ViTs), audio models and, most importantly, Large Language Models (LLMs) like GPT-4 [Vas+17; Dos+21; Ach+23]. LLMs showcase the power of Machine Learning in processing and understanding human language. These models can digest vast quantities of information, draw connections, and generate insights. Thus demonstrating potential applications in literature reviews, hypothesis generation, and even interactive learning in scientific domains, which could eventually lead to greater understanding in all areas of science. But behind the scenes, simpler and more traditional machine learning methods, like Principal Component Analysis (PCA) and K-Means Clustering are still used. For example, in Large Language Models, Word Embeddings capture the meaning of a word as a vector. PCA is a possible solution to make sense of these high-dimensional vectors. This shows that machine learning is much more than just the current state-of-the-art methods and that traditional methods have their place.

Using machine learning, researchers can quickly analyse gigantic datasets, making sense of patterns and details that would take humans much longer to figure out. One specific area where this can be applied is the study of Trypanosoma brucei - a species of single-celled parasitic eukaryotes. Scientifically, Trypanosomes are very interesting as model organisms for the analysis of the flagellum and cytoskeleton, which provide insight into eukaryotic cell biology [Whe+13]. In addition to their scientific significance, Trypanosomes are important because many species are carriers of human and animal diseases, like T. brucei and T. cruzi, which are responsible for the African sleeping sickness and the Chagas disease. Both diseases can be fatal if untreated. Studying Trypanosoma will not only shed light on cell biology but could also help in finding better treatments for those diseases. An essential part of Trypanosome research is the analysis of different cell cycle stages, which can be done by analysing the cell's morphology - especially the nucleus, kinetoplast and flagellum, which are the three most important organelles for telling the stages apart [Sha+08; Min+11]. One resource for studying T. brucei is the TrypTag database. It aims to localise every protein encoded in the Trypanosoma brucei genome. For this, it uses three imaging techniques: high-resolution phase contrast imaging, fluorescent protein tagging with

mNeonGreen (mNG), and DNA staining with Hoechst-33342. Each image has expert localisation annotations and automatically generated masks for the phase contrast channel [DSW17]. Those are also publicly available on the TrypTag website, as seen in figure 1.1.

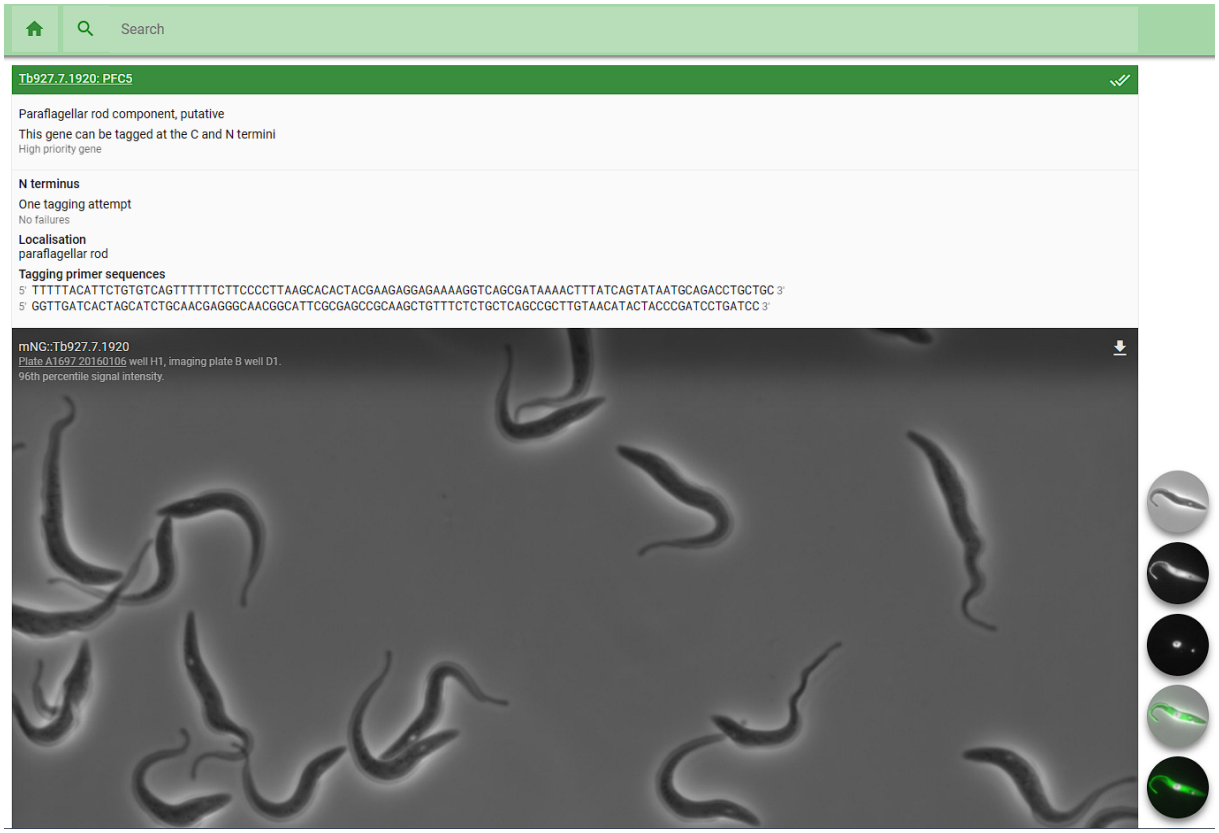

Figure 1.1 – Screenshot of the TrypTag website showing information about the gene TB827.7.1920:PFC5

Richard J. Wheeler, one of the authors of the TrypTag paper [DSW17] provides the TrypTag repository on GitHub for fetching, loading, and analysing the dataset. The paper "Detailed interrogation of Trypanosome cell biology via differential organelle staining and automated image analysis" [WGG12] describes some of the techniques and processes used in TrypTag. Like the image pre-processing, the skeletonisation process and various cell measurements. The TrypTag dataset, this paper and the functions provided by the TrypTag repository provide the basis for the work presented in this thesis. Currently, there is no solution based on TrypTag data that can automatically classify all six cell cycle stages from microscopy images. Although the TrypTag repository contains functionality to perform morphological analysis, such as measurements for cell length, cell area, posterior and anterior points of the cell, kinetoplast and nuclei counts, additional analysis is required for unambiguous classification.

An automated system that can assist in annotating the cell cycle stages would be advantageous for research purposes. Because even human experts can struggle to differentiate between similar stages within the cell cycle. Previous studies have relied heavily on hand-annotating microscopy images to distinguish between different organelles or cell cycle stages, greatly limiting the number of samples. An automated system can improve the sample sizes of future studies, leading to more conclusive results with less uncertainty. Furthermore, it can reduce the bias introduced by human

experts and free up time for researchers to focus on idea generation, analysis, and interpretation.

Such systems have already been applied to organisms like Caenorhabditis elegans (C. elegans), which face comparable problems due to their visual similarities to T. brucei, like their worm-like appearance [DB23; Heb+21]. Which is why they will be mentioned in the literature review. As this area of research might bring new solutions to Trypanosome research.

In this thesis, I aim to provide a brief overview of the life cycle stages and morphology of Trypanosoma brucei, as well as how they change during their cell cycle. I will review research papers both within and outside the field of Trypanosoma brucei, including literature from C. elegans research, which is commonly studied and may offer transferable scientific literature and methods for Trypanosoma research. In addition, this thesis will explore the TrypTag dataset and the TrypTag GitHub repository, providing details on creating a flagella dataset. To use the flagella dataset, I will use K-Means clustering as an adaptive thresholding method, to efficiently create masks for each image channel. I will also examine cell features with multivariate scatterplots and plots of the principal components of the feature vectors from a pre-trained ResNet-50 to find clusters, new correlations and patterns in the image data. After that, I will try to build and evaluate a tree classifier, as trees are easy to understand and more interpretable than other machine-learning methods. They also provide a structure that allows for respecting the limits set by nature. I aim to build on top of and enhance previously developed methods and use them to extract features for classifying the cell stages by making use of the large amounts of data available.

1.2 Problem Statement and Research Objectives

The research objective is the classification of procyclic Trypanosoma brucei cell cycle stages from microscopy images provided by TrypTag. However, because no labelled data exists, an unsupervised algorithm or a manually constructed classifier must be used.

Initially, the research objective was to build a classifier. However, as the title suggests, this main objective changed due to a missing flagella dataset. The latter objective was to develop, refine, and select features that can be used to classify the cell cycle stages. For this initial goal, the following six objectives were originally defined:

First, the literature review must establish a solid understanding of the cell cycle stages. In addition, features and methods already used for classifying cell cycle progress must be found. Furthermore, gaps in the current approaches and solutions need to be identified. Second, good segmentation masks are required to extract the identified features. If the segmentation masks provided with TrypTag are insufficient to extract the morphological features. I must create the segmentation masks myself. The segmentation masks must be evaluated. Third, if current approaches cannot automatically capture features and the morphology of the cells, such as extracting their length or area, then I need to find ways to do so. I can use existing solutions to do this, but I also want to find new ways to capture their morphology. Fourth, features or combinations of features that are most relevant for cell cycle classification need to be identified and selected. Ideally, this happens unsupervised or semi-unsupervised because there is no labelled data. Fifth, a reliable classifier that uses existing and newly found features has to be built. Lastly, the classification model must be evaluated to assess how well it generalises and predicts the cell cycle stages. But first, such an evaluation scheme has to be devised. Either a human expert has to label images or other ways to evaluate the model need to be found.

1.3 Scope and Limitations

Because this thesis is part of a bachelor's degree in computer science, the amount of biological or chemical processes involved in the cell cycle stages of Trypanosoma brucei will be kept to a minimum. The aim is to demonstrate how computers and different algorithms can be applied to biology without the necessity of deeper biological knowledge. Additionally, the goal is not to apply or develop new state-of-the-art solutions but to find simple, reasonable and effective approaches for solving this problem. Furthermore, this thesis should be intelligible to all the people working in this specific field, but also to computer scientists.

To further clarify this, I have set 3 goals that each method used that bring me closer towards classifying cell cycle stages needs to fulfil. Every technique must be efficient, explainable, reproducible and reliable. This is for the following reasons:

Efficiency: It needs to be fast and accessible because I am trying to alleviate the bottleneck of data analysis, which occurs because exponentially more data is being generated. The used machine learning models should be able to run on almost every computer in a reasonable time and should not require specific hardware like workstation GPUs.

Explainability: Most importantly, I want an explainable model. That includes making the model itself easy to understand, even for someone who is not a computer scientist. But also that the predictions of the model are comprehensible. The model should not be a black box.

Reliability and Reproducibility: The methods need to be reliable because otherwise, further research based on the presented methods will not be conclusive because of possible errors in the analysis. The key findings of this thesis should be reproducible. Therefore, randomness should be avoided. Any randomness introduced to reduce bias should only minimally affect the final result.

Lastly, some limitations. This thesis is limited by the datasets that were accessible to me and by the time I had available. Additionally, as it builds on top of other solutions, it will be limited by the accuracy of those existing solutions. The results of this study might not be applicable in a more general sense, but only to this specific research area of Trypanosoma brucei or the TrypTag dataset.

2 Theoretical Foundations and Literature Review

Since this thesis aims to provide a method for classifying cell cycle stages of procyclic Trypanosoma brucei based on images, it is necessary to understand their cell cycle stages and morphology. Furthermore, a comprehensive review of previous research on the topic is required. For that, I will provide an overview of Trypanosoma brucei, covering its life cycle, procyclic cell cycle stages, morphology, and changes throughout the cell cycle. I will also cover what machine learning is and why it is helpful for this problem. After that, I will review select relevant literature for this thesis that comes from within the field of Trypanosoma research and from other areas that study organisms like C. elegans or Jurkat cells.

2.1 Overview of Trypanosoma Brucei

2.1.1 General knowledge

Trypanosoma brucei is a unicellular, parasitic, flagellated species causative of African trypanosomiasis. This disease affects animals and humans and can be fatal if untreated. It is a member of the Trypanosomatidae family, which contains many other pathogenic species like Leishmania spp. and Trypanosoma cruzi. Therefore, it is worth studying T. brucei from a humanitarian perspective to find better preventions, diagnoses, and treatments. In addition to that, it is also exciting from a scientific perspective because it represents an early branch of eukaryotic cell biology, which preserved many organelles. Furthermore, it has a highly complex life cycle, undergoing a series of morphological changes to adapt to different hosts and environments. Of particular interest is the procyclic form, which is found in the midgut of the tsetse fly. This form develops from the mammalian blood-stream form after the tsetse fly takes a blood meal from an infected animal.

2.1.2 Structure

Procyclic T. brucei have a wormlike appearance with a slender, elongated cell body. The cytoskeleton comprises an array of subpellicular microtubules that define the cell's shape. Like all eukaryotes, T. brucei has a nucleus containing DNA, which will be called nuclear DNA (nDNA). In addition, it possesses a kinetoplast. Kinetoplasts are unique to the genus Kinetoplastea, which Trypanosomatids are a part of. The kinetoplast is oval and smaller than the more circular nucleus. Akin to the nucleus, this organelle contains DNA (kDNA). Another striking feature is the laterally attached single flagellum that overhangs the cell and is used for locomotion. Many other organelles exist, for example, the paraflagellar rod, the basal body (BB), the flagellum attachment zone (FAZ), the flagellar pocket (FP), the mitochondrion or the Golgi apparatus. These organelles are well mapped out thanks to electron and fluorescence microscopy.

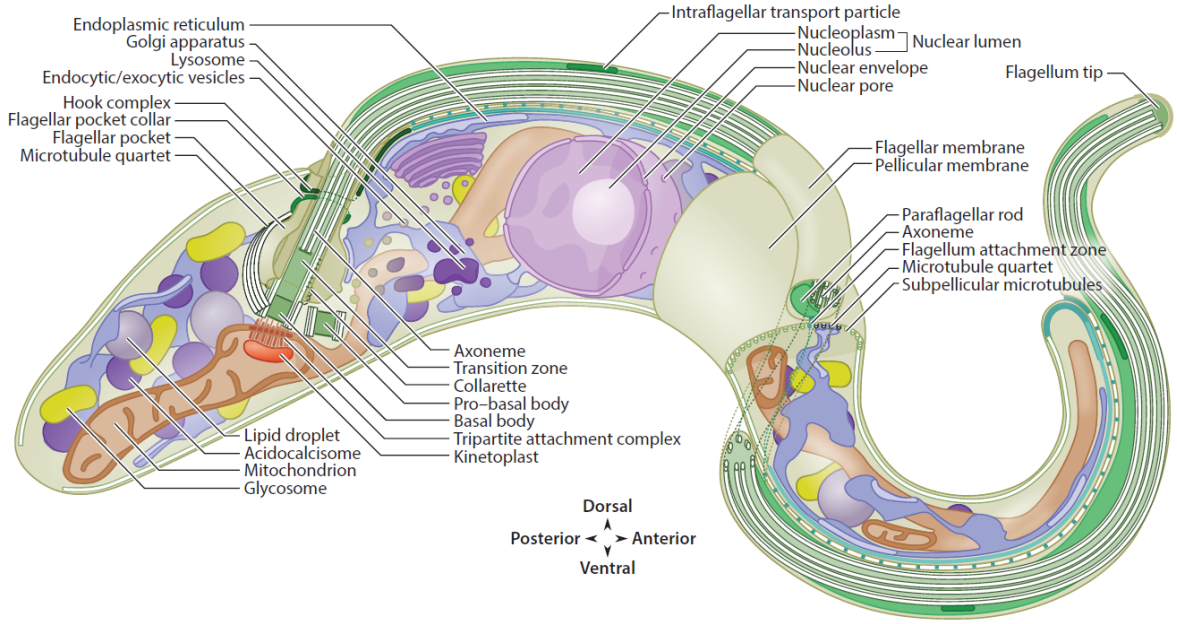

Figure 2.1 – A stylised visualisation of the morphology of a procyclic Trypanosoma brucei, as depicted in Figure 2 of Wheeler et al. 2019 [WGS19].

Figure 2.1 shows a map of all the organelles and their position in procyclic T. brucei. The relative position of the organelles within the cell body is described by the posterior-anterior and dorsal-ventral axes, as shown in the figure [WGS19]. Knowing the nucleus, kinetoplast, and flagellum for this thesis is sufficient, as they are enough to describe the different morphologies found in T. brucei [HW66]. However, the paraflagellar rod also plays an important role, as it will be used as a proxy for the flagellum and other flagellum measurements.

2.1.3 Life Cycle Stages and Procyclic Cell Cycle Stages

I will use "Coordination of the Cell Cycle in Trypanosomes" by Wheeler et al. (2019) for the description of the life cycle stages and the procyclic cell stages.

The procyclic form is just one of many forms in the life cycle of Trypanosoma brucei, as seen in figure 2.2.

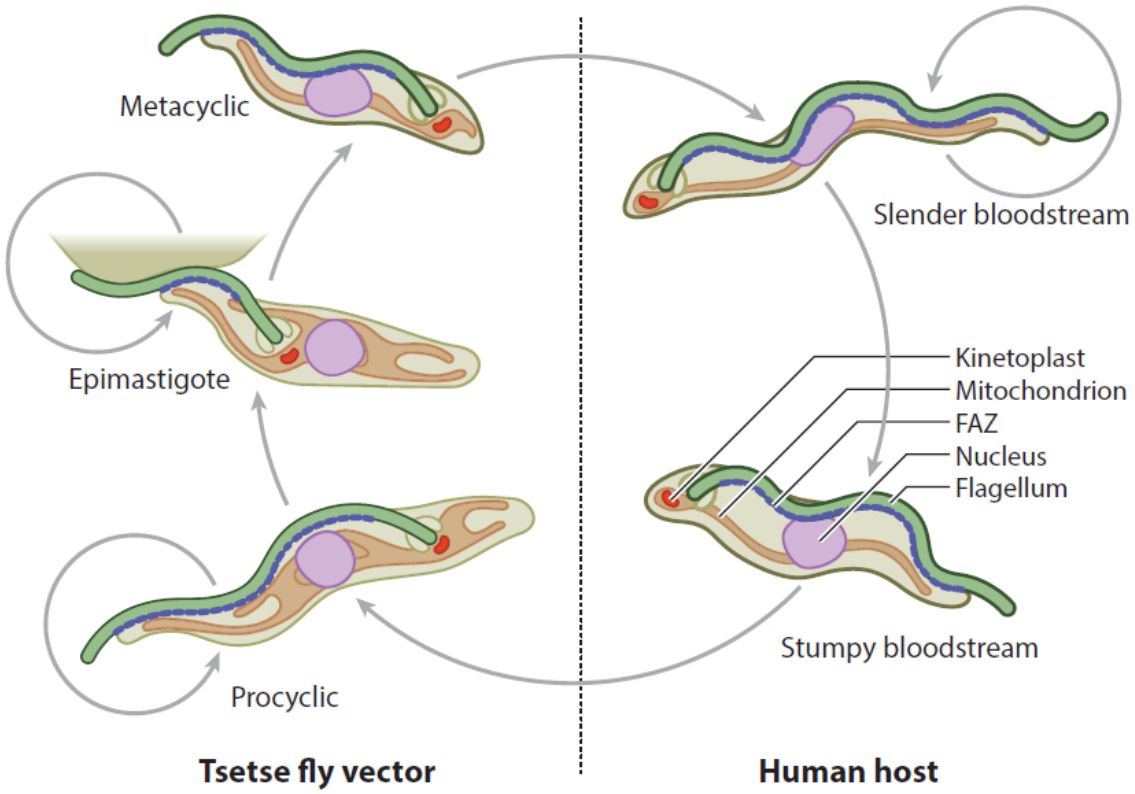

Figure 2.2 – A stylised visualisation of the various cell cycle stages of Trypanosoma brucei, as shown in Figure 1 of Wheeler et al. 2019 [WGS19].

These single-celled organisms are ingested by a tsetse fly upon taking a blood meal from an infected mammal. The trypomastigotes become procyclic in the fly's midgut and multiply via binary fission. When they leave the midgut, they differentiate into epimastigotes, which exhibit a unique morphology with a shortened flagellum attachment zone and anterior kinetoplast positioning. Upon reaching the salivary gland, they start multiplying again via binary fission. These epimastigotes transition into nonproliferative metacyclic trypomastigotes, which are transmitted to humans upon a tsetse fly bite. Inside the human host, they transform into slender bloodstream forms. As part of a transmission adaptation, these slender forms become nonproliferative stumpy bloodstream forms. Notably, all stages in both the tsetse fly and human are extracellular [WGS19].

The procyclic form in the midgut of the tsetse has a complex cell cycle comprising three subcycles. Namely the nuclear subcycle, the kinetoplast subcycle and the cytoskeletal subcycle. The subcycles are the basis of the six cell cycle stages, which can be identified from morphology [WG90; WGS19]. The stages and subcycles can be seen in figure 2.3. A common way of describing them is by their kinetoplast and nucleus count, where the following configurations are expected

during the cell cycle in the procyclic form: 1K1N, 2K1N and 2K2N. However, mutations or disruptions can cause other configurations. For example, the so-called zoids that 1K0N describes. As I want to use the flagellum as an additional source of information, I will use KNF-configuration instead of KN-configurations. The expected configurations are then: 1K1N1F, 1K1N2F, 2K1N2F, and 2K2N2F.

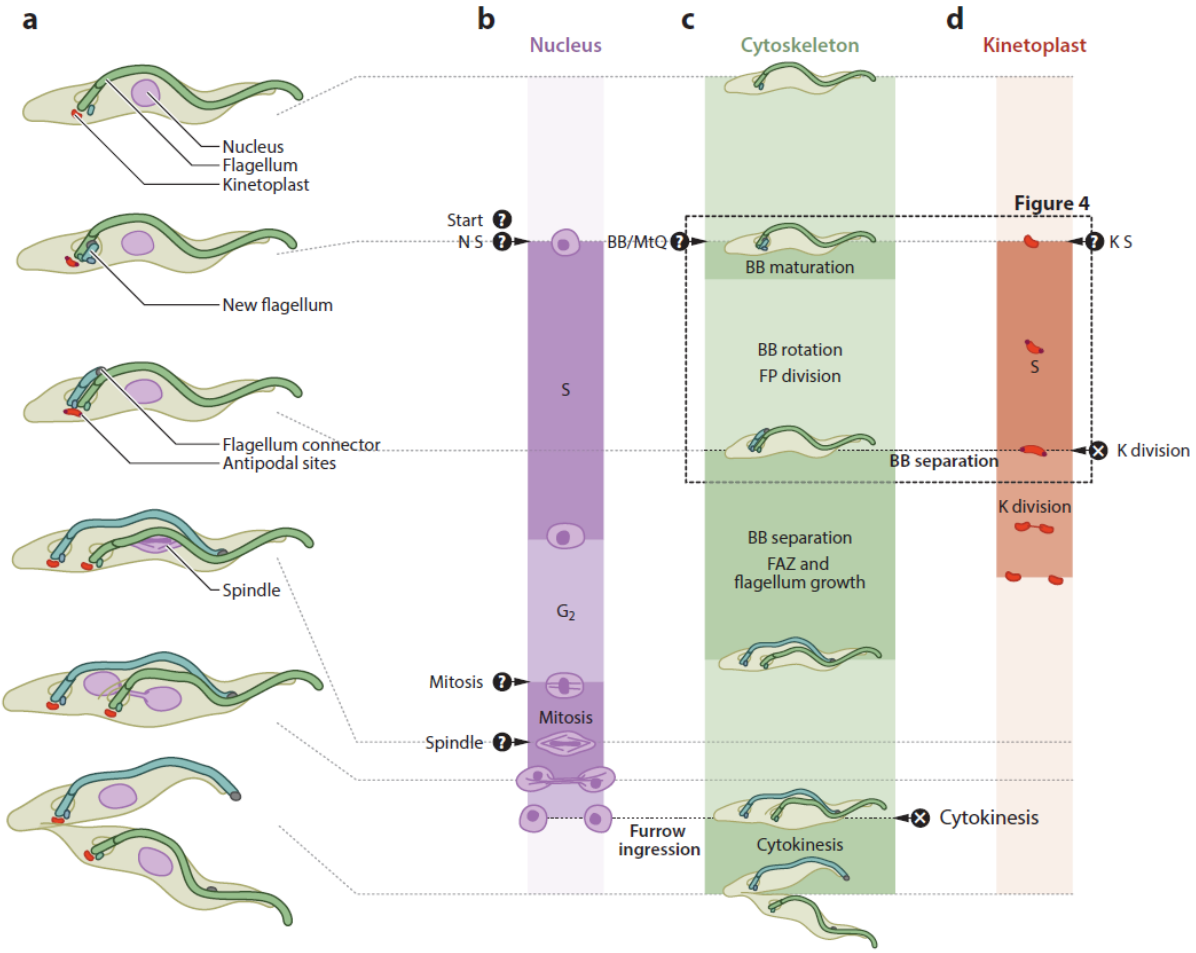

Figure 2.3 – A stylised visualisation of the various cell cycle stages and subcycles in procyclic Trypanosoma brucei taken from Figure 3 of Wheeler et al. 2019 [WGS19].

The cell cycle of Trypanosoma brucei can be divided into six distinct stages, as depicted in figure 2.3, where each row is one stage. The first stage is characterised by the presence of one kinetoplast, one nucleus, and one flagellum, which is denoted as 1K1N1F. In the second stage, a new flagellum emerges, and the kinetoplast grows. This stage is described as 1K1N2F. The third stage is marked by the segregation of the growing kinetoplast and the growth of the nucleus, still retaining the 1K1N2F configuration. During the fourth stage, a new kinetoplast is formed, and the migration of the kinetoplast and the new flagellum begins while the nucleus is still growing resulting in a 2K1N2F configuration. In the fifth stage, the nucleus mitosis is almost complete, therefore the cell is still in the configuration 2K1N2F. Finally, in the sixth stage, the cell divides. This phase is denoted with 2K2N2F. After cytokinesis, two daughter cells emerge, each with one

kinetoplast, one nucleus, and one flagellum.

2.2 Related Literature

To understand the big picture of research, we need to look more closely at what studies have been done. By knowing what's out there, we can see what's missing and where new ideas could come from. To make things clear and organized, I'm going to break this review of studies into two main parts: the first part will look at studies about Trypanosoma brucei itself, including its growth stages, shape, and how it's classified. The second part will look at machine learning practices and studies from other areas to adopt existing ideas which the T. brucei research community might have overlooked.

2.2.1 Trypanosome research

The most related and relevant paper for this thesis is "Detailed interrogation of trypanosome cell biology via differential organelle staining and automated image analysis." By Wheeler, Gull and Gluenz (2012) [WGG12]. In this paper, the authors describe a technique that identifies and differentiates kinetoplast and nuclear DNA. They achieve this by using two fluorescent stains with different binding preferences for AT and GC-rich DNA compositions. They used the minor groove binding (MGB) stains DAPI and Hoechst 33258, and the base pair intercalating (BPI) stains PI and SYBR green on the cells. After staining, they separated the signals through colour deconvolution—the colour deconvolution results in two pictures. One is for the kinetoplasts, and the other is for the nuclei, which makes telling them apart trivial. This is useful because commonly used DNA staining methods like Hoechst 33342 bind to any DNA; therefore, telling the organelles apart must be based on morphological features, requiring an additional processing step.

They tested their method on T. brucei, L. mexicana and C. fasciculata. For all species and stain combinations, they found that the kinetoplast was brighter with the MGB stains and the nucleus with the BPI stains. They computed the histogram of the log base 2 of the ratio of MGB stain intensity to BPI stain intensity. This histogram showed a clear bimodal distribution. They used one-dimensional K-means clustering to assign each point to one cluster representing either the nucleus or kinetoplast.

After confirming their findings by manually checking their 897 data points, where they only found two incorrectly labelled points, they combined this method with morphological analysis to extract features like the kinetoplast and nuclei DNA content, area, length, width, and location within the cell measured either from the anterior or posterior of the cell. For segmentation, they used phase contrast images, applied a rolling ball background filter to reduce the strength of the halo, smoothed the photos and applied a threshold. After that, they removed outliers and excluded cells touching the edge or containing no nucleus or kinetoplast. To extract features like the cell's length and width, they used the medial axis transform (MAT), which computes a midline that also gives the distance to the edges of the mask. They acknowledge that this method only works for typical morphologies. For example, their approach does not work for rolled-up Trypanosomes or dividing Trypanosomes with branched skeletons, which they excluded.

The results of their automated system were consistent with previous descriptions. Specifically, they found that the average measured cell length and standard deviation were 16.3 ± 3.1 µm (1K1N), 19.6 ± 3.0 µm (2K1N) and 21.0 ± 3.3 µm (2K2N). The study they referenced measured 18.9 ± 0.3

µm for 1F1N cells and 24.9 ± 0.3 µm for 2F2N cells, a 6 µm absolute increase and 32% relative increase. Furthermore, the cited study mentions a constant 22.3 ± 0.1 µm length for the flagellum throughout all cell cycle stages [Rob+]. After that, Wheeler et al. manually measured 50 1K1N cells which resulted in a mean length of 16.8 ± 3.0 µm. That is an approximately 3% difference in manual and automated mean cell length, which can be attributed to several reasons, like a small sample size or manual measuring errors. However, they did not provide an explanation because it is well within the margin of error. A statistical test like the two-sample t-test confirms this. However, one possible reason their technique reported slightly lower values than manual measurements could be that the MAT skeleton does not span the whole cell. This is because, at the anterior and posterior ends of the cell, the skeleton generated by the MAT stops precisely at half the cell's width at that point. So, for calculating the length of the cell, the value of the first and last point on the MAT that encodes half the cell's width should be added to the calculated length based on the MAT. This method cannot differentiate between cells and flagellum, as seen on their masks, which indicates that the MAT length includes the flagellum's overhang at the anterior end of the cell.

In their discussion and conclusion, they summarise their findings and reiterate their method's advantages and disadvantages. They claim that their approach "allows the analysis of complex trypanosomatid phenotypes without the risk of bias" from manual identification, that it works on out-of-focus kinetoplasts and nuclei, is perfectly replicable, simplifies analysis, and is low-cost and accessible. In contrast, they acknowledge their method's limitations and call for improvement in the following areas. Their method doesn't allow for branched skeletons, clusters of cells in contact with each other and out-of-focus cells. To improve the image, they propose to use focus stacking. They also mention that this method could be expanded to the flagellum if fluorescently tagged. The same methods could then be applied to measure the flagellum length, width and curvature. The MAT analysis could be adapted to measure the distribution of diffuse fluorescent stains, and the deconvolution could be improved by using zoids (only kDNA) and dyskinetoplastic cells (only nDNA) to get better reference values.

The same authors applied the idea of using the flagellum to classify the cell cycle one year before that. In their previous paper, "The cell cycle of Leishmania: morphogenetic events and their implications for parasite biology" [WGG11] they used immunofluorescence staining of the anti-paraflagellar rod (PFR) to identify the flagella in addition to a DNA staining method to determine the kinetoplasts and nuclei. In addition to other features like the cell's length, width, flagellum length, kinetoplast and nuclei DNA content, they created a scatterplot to show the cell cycle progress in Leishmania. They found correlations between cell cycle stage and cell length, kinetoplast DNA and nuclei DNA, but also flagellum position and size. However, they highlight that the flagellum length is much less correlated with the cell cycle process in Leishmania [WGG11]. Since Leishmania and Trypanosoma are part of the same family, they are fundamentally similar, and their methodology could be applied to Trypanosoma brucei to validate their findings and see if they hold for a larger part of the family or are specific to the species.

Other papers like "Analysis of the Trypanosoma brucei cell cycle by quantitative DAPI imaging" by Siegel, Hekstra and Cross [SHC08] also applied DNA staining and morphological analysis to determine the cell-cycle stages of procyclic T. brucei. Similar to Wheeler et al. 2011 [WGG11], this study incorporated an improvement mentioned in Wheeler et al. 2012 [WGG12]. They utilised three-dimensional deconvolution microscopy to enhance the image and reconstruct out-of-focus

objects. However, they did not utilise the flagellum as additional information. To verify their findings, they used flow cytometry and centrifugal elutriation. Flow cytometry, based on optical properties and centrifugal elutriation, based on physical properties, are common methods to classify cell cycle stages [Ben+17; Cro+18; SHC08].

A common theme among all those papers is the use of Scatterplots to show the relationship between different simple morphological features, like length and area, and kDNA and nDNA content. Some improvements, like z-stacking and using the flagellum for classification, were already implemented. But there are still many challenges, like systematic errors in measuring cell lengths with the MAT, identification of individual cells within clusters of cells, curled-up cells or cells with branched skeletons. I couldn't find any solutions to those hurdles in Trypanosome research. That is one of the reasons why I will also look at existing solutions from other areas and how the machine-learning community approaches segmentation and classification in a more general sense.

2.2.2 Machine Learning and C. elegans research

To provide a more comprehensive overview of existing solutions, I will cover findings and techniques for segmentation and classification used for C. elegans and other organisms and general common machine learning approaches for segmentation and unsupervised classification. This is useful because Trypanosome research covers only a tiny part of cell biology and machine learning.

One of the big problems in classifying T. brucei cells is that it is hard to tell some of the stages apart when cells have the same configuration of nuclei, kinetoplasts and flagella and only differ by the size or relative position of these organelles. A general approach to machine learning in cell biology has been described in the paper "Machine learning in cell Biology - teaching computers to recognise Phenotypes" by Sommer and Gerlich 2013 [SG13]. In this paper, they explain why machine-learning approaches are superior for problems like this, where the cell stages can not be differentiated easily by a few parameters. Additionally, they state that using unsupervised machine-learning methods can reveal unknown phenotypes. Furthermore, they describe the typical pipeline for analysis of microscopy data: image pre-processing, object detection and feature extraction.

First, image pre-processing removes artefacts like uneven illumination, which can be fixed with flat-field correction. Similarly, Noise should be reduced by using smoothing filters.

Second, object detection aims to extract the areas of interest from the background. This is either based on region properties like brightness or based on contours, which can be respectively resolved with thresholding and gradient methods. However, they also mention the use of fluorescent markers, which make segmentation trivial.

Last, important features are derived from the data before building a classifier. As previously mentioned, commonly used features in Trypanosome research are the count of Nuclei, Kinetoplasts and Flagella, kDNA and nDNA content, and morphological features like the cell's size and length. In addition, derived pixel-based features like the mean intensity or standard deviation and measures of roughness or circularity could be used. They also mention the use of dimensionality reduction methods like principal component analysis, to determine essential features. As more features are not always beneficial. This is done by ranking the principal components by the variance they cover and discarding the low-ranking ones. This technique is commonly used to visualise multi-dimensional data.

Other research fields, like the one studying Caenorhabditis elegans (C. elegans), use more recent neural network-based segmentation and classification techniques, like convolutional neural networks (CNNs) and vision transformers (ViTs). For example, WormPose uses a CNN for pose estimation and pose tracking and also provides methods for synthetic image generation [Heb+21]. WormSwin, another paper, uses a variant of the Transformer architecture that can be used for image classification, detection and segmentation. In this case, they trained it for instance segmentation on C. elegans [DB23]. Because the various approaches taken only require images, and TrypTag provides almost unlimited image data, it is reasonable to assume that they will also work for Trypanosomes.

The problem of skeletonising overlapping cells could be solved using a simple algorithm described in the paper "A CNN Framework based on Line Annotations for Detecting Nematodes in Microscopic Images" by Chen, Strach, Daub et al. 2020 [Che+20], in which they use U-Nets for segmentation and skeletonisation of C. elegans. The algorithm only requires to know the endpoints of the organism. It works by cutting the intersection of two overlapping organism skeletons and then using the steering angles to determine which parts likely belong together. The assumption is that C. elegans is mostly smooth and contains no harsh curves [Che+20].

One of the most relevant papers from the field of C. elegans research is "Segmentation and classification of two-channel C. elegans nucleus-labelled fluorescence images" by Zhao, An, Li et al. 2017 [Zha+17], as it goes in-depth about the typical pipeline outlined in Sommer et al. 2013 of pre-processing steps, segmentation, feature extraction and classification. Moreover, they use K-means clustering for segmentation and the distance transform, which is closely related to the medial axis transform. Of particular interest are the features used for classification. After their pre-processing and segmentation steps, they construct a set of 51 geometric, intensity and texture features for classification. Geometric features describe the shape and size of the cell, like area, perimeter, circularity, ellipticity and statistics about curvature. Intensity features are statistical measures extracted from each nucleus's histogram, and texture features describe the grey-level co-occurrence matrix (GLCM). After carefully selecting features, different classifiers are built using various techniques such as support vector machines, random forests, k-nearest neighbours, decision trees, and neural networks. However, their feature selection technique requires labelled data. The paper also provides an overview of their classifier's performance. The decision tree is the worst-performing one, while the random forest is the best-performing classifier. Although decision trees performed the worst in this particular test, the results are still comparable to the other methods, and they are the most straightforward and explainable models.

Other papers outside of Trypanosome and C. elegans research also provide promising methods useful for this thesis. For example, the following paper: "Reconstructing cell cycle and disease progression using deep learning" from Eulenberg, Köhler, Blasi et al. 2017 [Eul+17] describes how features from CNNs can be used and plotted with dimensionality reduction to visualise the cell cycle stages of Jurkat cells. For this, they modified a network based on the Inception architecture (2014) and used t-SNE dimensionality reduction to visualise the activations of the penultimate layer of the neural network. For every input image, the network produces a vector of activations. When the network is well-trained, this vector serves as a representation of the input image. Consequently, similar images should yield similar vectors. Because these vectors are very large and can contain repetitive information, dimensionality reduction is used to visualise them. This can be seen in figure 2.4.

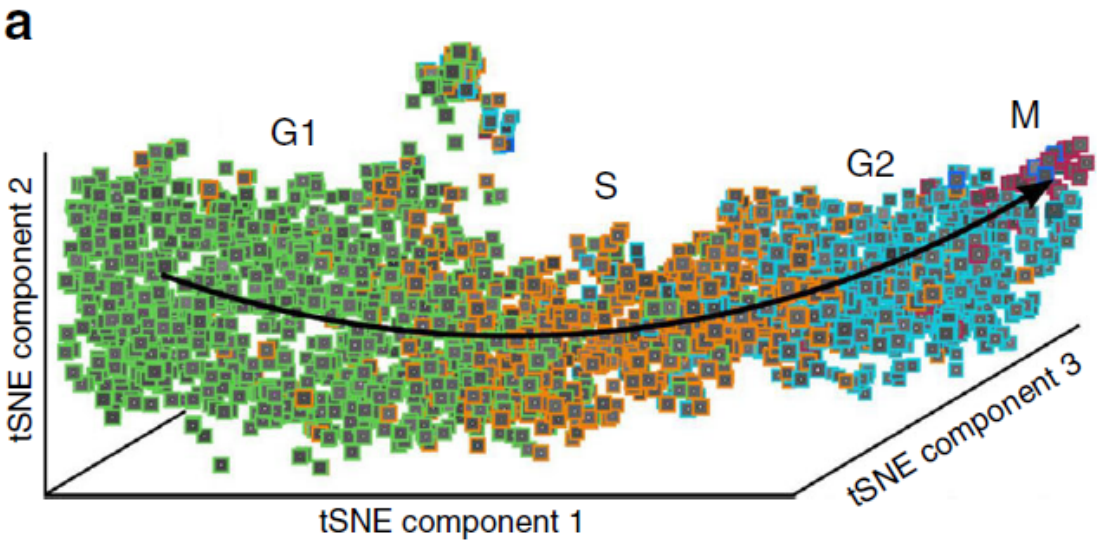

Figure 2.4 – A three-dimensional colored scatter plot, taken from Eulenberg et al. 2017 [Eul+17], displaying the three most important t-SNE components of a feature vector.

The image displays a three-dimensional plot of the three t-SNE components employed to condense the feature vector from the CNN. Each cell cycle stage is uniquely colour-coded. The plot shows that distinct cell cycle stages correspond to different representations, highlighting their varied visual features. This approach of using a pre-trained neural network, extracting features and using dimensionality reduction to find patterns in the image data of Trypanosoma brucei could be helpful.

To summarize, I have covered basic knowledge of Trypanosomes and their life and cell cycle stages, Trypanosome research for classifying cell cycle stages, general machine learning practices, C. elegans research for skeletonisation and segmentation approaches and other fields of research. The key takeaways from Trypanosome research are as follows: First, staining for simple segmentation and nucleus and kinetoplast classification using K-Means clustering. Second, several known features can be used for classifying cell cycle stages: the count of kinetoplasts, nuclei, and flagella, the kDNA and nDNA content, the cell length, width, and area, the relative position of kinetoplasts and nuclei within the cell body and flagellum position and size. Third, measurements of the cell length using the MAT and a potential systemic error in calculating it. Lastly, the limitations in current techniques, namely the inability to discern clusters of cells, curled-up cells and cells with branched skeletons. The key takeaways from general machine learning and C. elegans research are the use of a three-step pipeline of image pre-processing, object detection and feature extraction, the use of texture, intensity and geometric features, the use of dimensionality reduction techniques like PCA and t-SNE for visualising and extracting features and the use of neural networks like CNNs and ViTs for segmentation and skeletonisation, which could be useful to resolve problems faced in Trypanosome research. The literature review highlights the work done to classify cell cycle stages of trypanosomes and other organisms, how to approach this machine-learning task and how to solve some of the limitations of current techniques.

3 Methodology

This chapter contains clear and detailed descriptions and justifications for how I conducted my research. The purpose of this is to make my research reproducible and to prove its reliability and validity. Specifically, it aims to describe how I approached my research objectives and the individual research questions. The primary goal of this research was to classify procyclic Trypanosoma brucei cell cycle stages from microscopy images. The individual research objectives were covered in section 1.2. Each of the following contributions addresses at least one of those research objectives. The main contribution to classifying cell cycle stages is using the flagellum as an additional source of information. This required building a flagella dataset, segmentation masks and combining the information with existing techniques. Secondly, fitting ellipses to the masks of kinetoplasts and nuclei to capture the sub-cycle progress of the cells. Thirdly, improving MAT-based cell length measurements to eliminate the 3% difference in manually and automatically observed cell lengths. Penultimately, trying to identify clusters for each cell cycle stage within the multivariate scatterplots and scatterplots of the principal components of a ResNet-50 feature vector. Finally, building a binary decision tree on top of those features and improvements for classification. Before diving into the details of how those ideas were implemented and tested, I address the general approach of my research. Whether I took a qualitative or quantitative and inductive or deductive approach, which tools I used and more. Then, I will recapitulate my initial thoughts, exploration of the data and how I came up with a plan for my thesis. After that, a description of the exact step-by-step process, from the initial idea to the data collection, the experiments and the data analysis, is provided for each contribution. Next, the problems I encountered, the limitations of my methodology, and how I dealt with them are mentioned. Lastly, a summary of this chapter.

3.1 General approach

Throughout my research, no general approach was applied to all research questions. Some required a quantitative deductive approach, and others required a mixed approach. In a quantitative deductive approach, a hypothesis is tested by gathering, processing and analysing numerical data to identify relations and accept or reject the hypotheses. For example, to test whether my improvement of the MAT-based cell measurement could explain the 3% difference in manually and automatically observed cell lengths. On the other hand, creating the flagella data-set, particularly the gene selection, required a mixed approach. First, I needed to filter the vast amounts of genes using the signal intensities percentile scores provided by TrypTag. However, after that, I had to manually inspect and select the genes because a high signal intensity does not translate to clear signals and noise-free images. In this example, I used both a quantitative and qualitative research approach. Another example of a mixed approach would be the explorative nature of finding new features in scatter-plots. Although I had a hypothesis for why the scatter-plots might show clusters representing the different cell cycle stages, which is deductive, I still had to find clusters by

visually inspecting them. I could have found other unexpected clusters, which would require inductive reasoning. Additionally, the visual inspection adds a qualitative aspect to the analysis. As demonstrated, no straightforward approach can be applied to all problems. I would describe the general approach as data-driven and explorative, which contains quantitative, qualitative, deductive and inductive elements. In the coming sections, where I present the methodology for solving individual research questions, I will mention which approach was taken. But before diving into that, I want to mention more background information on the tools used to conduct this research.

The foundation for this thesis is the TrypTag GitHub repository, which implements many of the necessary methods to retrieve and analyse Trypanosome data; the TrypTag website, which provides a visual interface to the dataset, alongside information about the genes and images; the paper "Detailed interrogation of trypanosome cell biology via differential organelle staining and automated image analysis" by Wheeler et al. 2012 [WGG12], which has already been covered in the literature review; and lastly the paper "Genome-wide subcellular protein map for the flagellate parasite Trypanosoma brucei" by Billington et al. 2023 [Bil+23] and the Zenodo Deposition, both of which I am required to cite.

All the experiments were conducted on my personal computer. I will only mention the components relevant to the program run times and download times: I used an Intel Core i5 12400f 6-Core CPU, 16GB of DDR4 3200MHz memory, and a 2TB Western Digital Black SN770 for storage. No Nvidia or AMD GPU accelerators were used. The data was downloaded from within my personal network at a download speed of around 170 MBit/s. The experiments were implemented in Python, as the TrypTag GitHub repository is a Python package and my personal experience with it. A virtual environment running Python 3.10 inside the PyCharm IDE was used to develop the code. To replicate my findings, the code I developed can be found on my personal GitHub repository. The repository contains instructions on executing the code and what each file does; basic Python and Git knowledge is required. Furthermore, some Python files contain additional hints on how to use them and their run time. The repository also contains some code duplicates from the TrypTag GitHub repository, as I did not want to clone and create a branch of their repository. Those functions should be highlighted as such, and I do not take credit for them. To accelerate the programming part of this thesis, I used autocomplete features integrated within PyCharm and GPT-4, as there are roughly 1500 lines of code without counting the duplicates from the TrypTag repository. Those tools were mainly used for generating boilerplate code, such as plotting with Matplotlib or Plotly.

3.2 Initial thoughts

This section presents the initial steps on how I approached the thesis topic of classifying procyclic Trypanosoma brucei cell cycle stages from images. It also includes my initial attempts to familiarise myself with the TrypTag dataset and GitHub repository, my initial testing of different techniques to gauge whether they are worth exploring, and how I devised a plan to achieve my research objective.

Before I began working on the thesis, there was only a general idea of using morphological features to classify cell cycle stages. At this point, the focus was on building and evaluating a classifier. I had no knowledge of Trypanosomes before working on this thesis. However, due to the correspondence with my supervisor, Dr. Nandu Gopan, I had a rough idea of what the dataset

and the images looked like. Furthermore, I knew I had to use unsupervised methods like clustering and dimensionality reduction, as the data from TrypTag is not labelled. At this time, there were no plans to use the flagella as an additional source of information and therefore no plans of creating a flagella dataset existed. After the literature review and a further meeting with my supervisor, the idea became more precise, as it revealed gaps and limitations of current approaches. For example, it highlighted the complexity of the classification process due to the difficulty in differentiating stages 2 and 3, as well as stages 4 and 5. The most obvious gap was that current approaches only used KN configurations instead of KNF configurations for classification. So, this became a key research objective. The addition of the flagella made this thesis more complex because now I needed a flagella dataset, which I thought would be provided by my supervisor. However, this was not the case, which magnified the problem. The reason for that is that I had to build a dataset on my own, segment the flagella and process the data. During the literature review, I also noticed a systematic error in the MAT-based cell length measurement that, in my mind, could be easily improved. An improvement of the cell length measurement, one of the relevant features for deciding between cell cycle stages, meant an improvement in accuracy in the classification process. Because classifiers are only as good as the features on which they are trained.

Figure 3.1 – Multivariate Scatterplot with misclassified KN configurations with the expected KN configurations greyed out.

My initial experimentation started with the TrypTag Python package and the methods discussed in Wheeler et al. 2012 [WGG12]. The TrypTag Python package simplified my research, as some of the prerequisites for classifying T. brucei are already met. Through it, I had access to the cell images and cell masks provided by the TrypTag database. In the TrypTag repository the cell images and cell masks are used to do KN-configuration analysis, which returns how many kinetoplasts and how many nuclei are present in each cell. It also contains the functionality to extract the nuclei and kinetoplast area, their relative position in the cell and their DNA content. Additionally, it uses the medial axis transform to skeletonise the cells and determine their length, from which

other measures are derived. Initially, I tried using the provided methods for feature extraction, but they proved to be inaccurate in finding clusters for the cell cycle stages in the multivariate plots or the plots of the principal components of the ResNet-50 feature extraction. I encountered many unexpected or impossible cell cycle configurations. To name a few, 1K0N, 1K2N, 1K3N, 3K3N, 3K2N and 4K3N. This can be seen in the legend of figure 3.1, one of my early experiments for multivariate scatterplots, which was based on randomly selected data and not on the later developed flagella dataset.

A specific example of a misclassified 1K3N cell can be seen in Figure 3.2. This type of plot with the highlighted flagellum in red was only later developed but is used for illustration purposes.

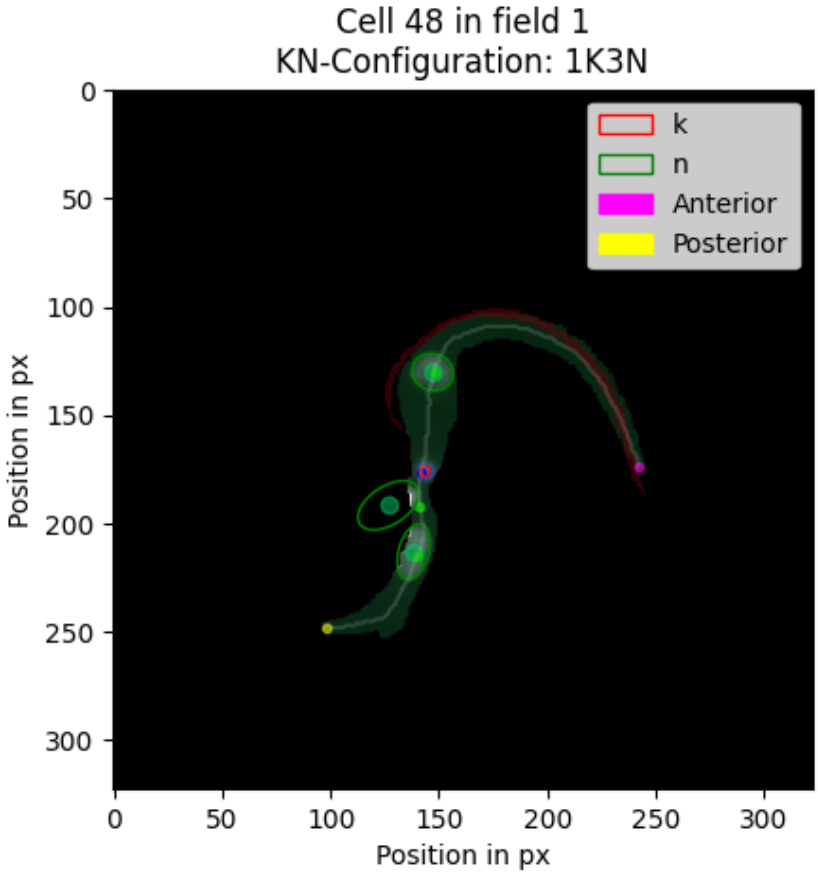

Figure 3.2 – A cell falsely classified as 1K3N by the TrypTag repository. This is likely due to overlapping or very close cells.

Upon visually inspecting these images, I found that the classification errors were due to imprecise masks. Although this initial testing did not contain information about the flagellum, it was useful for gaining intuition about the existing methods. It helped build a foundation of methods that I could use later. After realising that, I decided to implement some of the segmentation and data analysis methods myself, which are needed for better masks, additional flagella information, and ultimately classification. Then I also tried different traditional segmentation methods that tackled the main challenge of removing the halo around the cells. The halo is a bright border around the cells that happens due to the imaging technique used and can be seen in Figure 1.1.

Some of the tested approaches were histogram equalisation, blurring and binarisation for segmentation; rolling-ball filter to remove uneven background noise; and gradient-based methods to detect edges in combination with watershed and morphological operations like erosion, dilation, opening and closing. But I did not make any meaningful progress. This was part of the reason for coming up with a plan for moving forward. Another reason was that I was not provided with the flagella dataset and had to build it on my own. Here is the plan I came up with:

Data Acquisition and Understanding: This involves exploring the TrypTag database in-depth, ensuring that relevant data is correctly sourced and properly organised. Additionally, I need a flagella dataset of the best genes that localise the paraflagellar rod, a part of the flagellum. Because there were no genes that directly localised the flagellum. This is done because the flagellum is necessary for classification.

Pre-processing and Segmentation: Given the nature of image data, pre-processing and segmenting the images to isolate specific features and structures is essential. TrypTag already provides pre-processed images and masks, but those need to be verified. Additionally, I need to create masks for the flagella dataset using simple yet effective methods. As I could not afford to spend even more time on building segmentation methods.

Improvement of previous techniques: As outlined in the literature review, the MAT-based cell length measurement might introduce small inaccuracies. This calculation must be improved to get more accurate cell and flagella length measurements.

Feature Extraction and Selection: Before feeding the data into a classifier, extracting relevant morphological features is vital to ensure dimensionality is managed effectively. My Python implementations and existing implementations, like those provided in the TrypTag GitHub repository, will achieve feature extraction. Use commonly known features to predict the cell cycle stages, such as the count of nuclei and kinetoplast, DNA content, or cell length, and devise new ones. For example, ellipses can grasp the shape of the kinetoplast and nuclei. This would be useful for identifying the difficult stages in which these organelles grow. In addition, more experimental features will be selected and extracted by examining multivariate scatterplots and plots of the principal components from a pre-trained ResNet-50 neural network used for image detection.

Classification: Use a binary decision tree for classification purposes. The primary reason for this choice is the simplicity and interpretability in the context of biological data. They are ideal for achieving the set goals in Section 1.3. Because they are easy to understand, extremely fast, and have no random components.

Finally, validating the classification model and ensuring its accuracy and robustness is crucial. Explanation for the choices made will be provided in the following sections for each method used. With the combination of the TrypTag database and repository, the advancements proposed based on Wheeler et al. (2012), and the systematic approach outlined above, the hope was to improve cell cycle classification in procyclic Trypanosoma brucei.

3.3 Building and Evaluating a Flagella Dataset

In this section, I will first describe the structure and nature of the publicly available TrypTag data and how to fetch it. But more importantly, next I will explain my rationale for needing a flagella dataset and how I created it. Building this image dataset required the following steps: identifying the need for it, finding a source for the data, selecting the correct data, processing the data, which includes pre-processing and segmentation, and lastly, integrating it into existing methods to make use of it.

Let's start by describing the TrypTag database, what it does, how it works and how it can be used. The TrypTag database aims to localise every protein in the trypanosome genome. The genome acts like an instruction manual. It dictates how an organism is built and how it functions. It contains a huge number of genes, each containing information on how to assemble a specific protein. The proteins can then act as structural elements, akin to building blocks for the cell, or regulate how they work. Protein tagging can then be used to attach the instructions for a fluorescent stain to a specific gene. This is useful because when the gene is expressed as a protein, we can use fluorescence microscopy to see where exactly that protein is used. This can, in turn, be used to localise specific cell organelles or signal when a process is happening. In Trypanosomes, this has been very useful for studying their morphology and development.

The TrypTag database uses this technique to create phase contrast images and fluorescence microscopy images of procyclic Trypanosoma brucei. Unique to this resource, every protein within the images is fluorescently tagged to determine its location. More specifically, it provides 3-channel 16-bit TIFF files. The first channel is a phase-contrast image and shows the whole cell; the second channel shows the expressed protein, which was tagged with mNeonGreen(mNG), and the third channel highlights the Kinetoplast and Nuclei that have been stained with Hoechst-33342, a DNA stain. This publicly accessible dataset organises images either by gene accession numbers, like "Tb927.7.1920", or by the specific localisation of proteins such as "paraflagellar rod", "kinetoplast", or "cytoplasm". A direct search using a gene accession number yields images related to that gene, while localisation terms present a list of genes associated with that organelle. Moreover, tryptag.org, where these images are integrated, provides additional information. Including the tagging primer sequence, localisation terms, percentile scores of signal intensity, the terminus employed for fluorescent tagging, and details like the date and plate number. This can be seen in the previously provided figure 1.1. Comprehensive details on the database and methodology can be found in Billington et al. (2023), Halliday et al. (2017), and Dean et al. (2017) [Bil+23; Hal+19; DSW17].

The data can be fetched by using the TrypTag GitHub repository. There are also other ways to fetch the data. Still, this thesis exclusively uses the TrypTag repository to download data because it contains other useful functionality for downloading, searching, organising and analysing the data. Next, I will explain why it was necessary to create a flagella dataset and how I built it.

3.3.1 Gene Selection

Within this subsection, I will explain how and why I selected the genes for my dataset. For this, I used a mixed research approach. The reason for that is, that the gene selection was based on numerical features like the percentile score of the signal intensity but also my subjective evaluation of the images. But first, I want to provide the reasoning behind my decision to build a flagella

dataset.

There are two basic reasons for this. First, as explained in subsection 2.1.3, the cell cycle of procyclic Trypanosoma brucei is divided into six stages that are dictated by the nuclear, kinetoplast and cytoskeletal subcycle, which is visualised in figure 2.3. Additionally those subcycle stages can be identified from morphology. However, the only solution I am aware of for analysing T. brucei data is the TrypTag repository, which only provides KN configurations. However, KN configurations are not enough to distinguish the six stages. This is why KNF configurations exist. They add the count of flagella to determine between stage one, with configuration 1K1N1F and the other stages, which all have two flagella. Notice how the second and third stages both have configuration 1K1N2F. In this case, the simpler KN configurations summarise the first three stages with 1K1N. The KNF configurations, however, are more precise. They allow the distinction between stage one and all the other stages by simply counting the number of flagella. Furthermore, this could also allow for a distinction between stages two and three because the new flagellum is beginning to grow. Additionally, the position of the new flagellum changes in the later stages, which could be useful for further specification of the cell cycle progress. If it were possible to not only count the number of flagella but also determine the lengths and positions, this would greatly enhance the ability of researchers to quantify and classify the cell cycle stages and progress.

To achieve this, several things are necessary: The flagella need to be extracted from the image data, either by using the phase-contrast images from channel one or by using tagged genes located in the flagellum through the mNG channel. After the flagellum is segmented from the cell body, several measurements, like its relative position in the cell body, its area, and, most importantly, its length, need to be calculated. To associate and count the flagella, a simple check of overlap between the flagella masks and the cell masks is enough. However, there is a potential roadblock. If the segmentation of the flagella within one cell only returns a single mask instead of two, we encounter a problem. It would be similar to the overlapping cell problem, which is one of the limitations current studies had. In this case, more advanced methods are needed, like the ones mentioned in the literature review for C. elegans. However, I decided this would go beyond the scope of this thesis because building the flagella dataset on my own was already unplanned and somewhat out of scope. In case my simple segmentation approach fails, I could still provide useful information and a good starting point for further studies.

First, I had to decide whether to try to extract the flagella from the phase-contrast images or use the mNG channel. Extracting the flagella from the raw phase-contrast images is much harder, but it has the upside that future studies could use this method for every possible gene in the TrypTag database. On the other hand, using the mNG channel is much simpler. It only requires finding suitable genes and using much simpler segmentation techniques to extract the masks. When taking a look at 1.1, one can see the flagella are slightly brighter than the cells themselves but not as bright as the halo surrounding them. They have a similar color to the background. For me, that meant segmenting the flagella from phase-contrast images is at least as difficult as segmenting the cells themselves. Additionally, even if that approach were to be successful, one would probably only pick up new flagella that are above a certain length. With this approach, it would probably be impossible to detect the early growth stages of the flagellum, where the basal body rotates, and the flagellar pocket divides, as shown in figure 2.3. Those reasons, in addition to the time constraints and my previous failed attempts at removing the halo effect, led me to conclude that it would be best to find suitable genes and use the mNG-channel approach.

For this, I used the localisation search from the TrypTag GitHub repository. As for the query

term, I used "paraflagellar rod" because it has been used in a previous study to locate the flagellum [WGG11]. This, however, might not be the ideal choice since the paraflagellar rod "[...] does not extend into the flagellar pocket, and the signal fades towards the distal end of the flagellum." [Hal+19]. Alternatively, the gene ontology described by Halliday et al. (2019) also defines the flagellar cytoplasm, which would be a more accurate signal for the flagellum. However, the authors have not found a protein that marks the flagellar cytoplasm. When I searched directly on TrypTag.org for "flagellar cytoplasm", no matches were found. So I tried searching for flagellar cytoplasm with the TrypTag Python package, using the tryptag.localisation_search function. The function returned a few possible genes with flagellar cytoplasm in the localisation field. However, those were not usable for my purpose for the following reasons. For example, here are the genes Tb927.11.7100 and Tb927.7.3020, as seen in figure 3.3 and 3.4. By comparison, Tb927.7.3020 is the better candidate because there are clear brightness differences between the parts of the flagellum, likely the axoneme or paraflagellar rod and the cell cytoplasm. Additionally, figure 3.3 seems to contain more noise and artefacts. Unfortunately, these genes do not only highlight the flagella but also seem localised in the cytoplasm and are, therefore, not ideal for a simple segmentation. This made me conclude that although the paraflagellar rod does not cover the whole flagellum, it is probably better than using the results for flagellar cytoplasm due to the bad and hard-to-segment signals. Moreover, previous studies have already used the paraflagellar rod to determine KNF configurations.

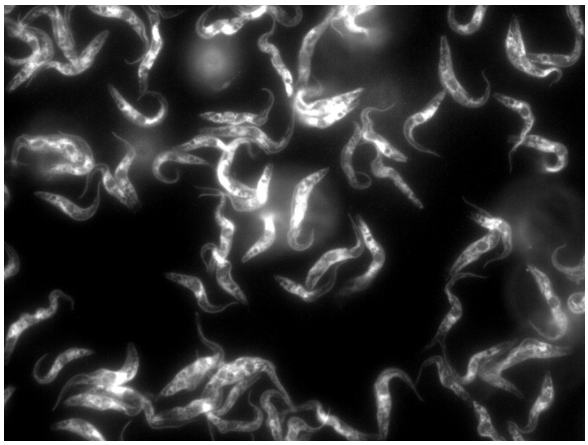

Figure 3.3 – The mNG channel of the tagged gene Tb927.11.7100, tagged on the c terminus and localised in the cytoplasm(patchy) and the flagellar cytoplasm.

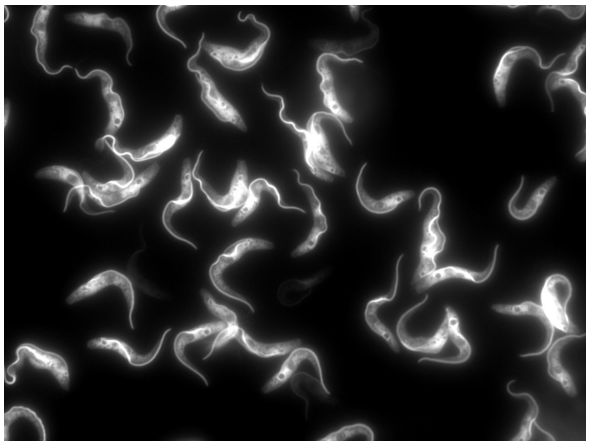

Figure 3.4 – The mNG channel of the tagged gene Tb927.7.3020, tagged on the n terminus and localised in the cytoplasm and the flagellar cytoplasm.

After deciding to use the paraflagellar rod, I wrote a script to retrieve all the genes that match the localisation search for "paraflagellar rod". Next, this script would iterate over all the genes found, thanks to the TrypTag Python package, and call the tryptag.org webpage. From this webpage of a particular gene, it automatically parses the signal intensity percentile, which is a measure for the sensitivity to small bright structures [Bil+23]. After this is repeated for every gene, this script prints out a list of genes ranked by their percentile scores. To achieve this, the script uses Selenium, a browser automation library. It can be found in my personal GitHub repository under the name "gene_selection.py". I used Selenium to scrape the percentile score information

from the webpage because I could not find a way to do the same thing with the TrypTag Python package.

Next, I needed to verify that the genes selected were good candidates for segmentation, so only genes with a percentile signal intensity above 80 were selected to limit the time needed for manual inspection. The manual inspection included reviewing 54 potential candidates with high percentile scores. For each one of them, I decided to use them or decided against using them based on the following criterion. The paraflagellar rod should be brightly highlighted, and ideally, the cell body should not be visible, unlike the genes found for the flagellar cytoplasm, where the cell body was visible. The background should be uniform without any artefacts. This way, a thresholding algorithm can easily segment the area of interest. It has to be noted that this manual selection represents by no means an ideal solution. Because it introduces human bias by someone who is not an expert on the subject. However, this manual selection process resulted in a list of ten genes visible in table A.1.

The exact table, as displayed in this thesis, was modelled as a dictionary in Python and put into a file called ”constants.py” so the list of genes could be reused in different scripts. After that, I needed to download each of the genes. For that, I wrote another script called ”get_flagella_dataset.py”. In this script, each gene is downloaded using functionality from the TrypTag GitHub repository. The used function ”tryptag.fetch_data” downloads compressed files. The uncompressed files contain the image files, image masks and text files which are then saved in the TrypTag cache. Unfortunately, this script also downloads data for a lot of other genes. This is because each gene is part of a cell line and the cell lines include other genes. This is also the part that will take the longest to replicate, as the complete uncompressed TrypTag cache is about 139 GB in size. I did not measure the exact time it took to download, but with my internet connection, it took 2-3 hours. After downloading all the relevant files, my script moves all the TIF files of the selected genes from the TrypTag cache into a folder I named dataset. It contains all TIF files, including the pre-processed and raw images. Each gene in this list has its own subdirectory within the dataset directory to simplify segmentation and processing. This resulted in 10 sub-directories totalling 114 TIF files that are 2.34 GB in size.

The resulting images of different populations vary in brightness and population density. For example, images from Tb927.8.5000 are comparatively bright and contain very few cells. Meanwhile, images from Tb926.10.9570 are dark and contain hundreds of cells. Those large differences in cell density and brightness make the dataset challenging. This is because a dataset with very low cell density might not have many overlapping cells, making instance segmentation easier. A dataset with very high cell density, however, might require more sophisticated segmentation techniques and even untangling methods to avoid throwing out overlapping cells and, therefore, ignoring a big portion of the dataset. Although the manual selection process introduces a selection bias for clear and noise-free images, my approach resulted in a diverse and challenging dataset with differences in image composition. The reason for that is that the brightness or cell density was not a criterion by which I picked the images. However, the variations between images will be important for the following image pre-processing steps and segmentation. My approach highlights the potential of the TrypTag GitHub repository and the TrypTag dataset but also underscores the importance of expert opinions to create similar datasets.

3.3.2 Image preprocessing

After creating a diverse flagella dataset, pre-processing and segmentation are necessary to get accurate measurements of organelles like the nucleus, kinetoplast, flagellum and the cell itself. The precision of segmentation masks is vital for distinguishing between cell cycle stages, as slight variations in cell morphology may be the only indicators of different stages. For example, the second and third stages and the fourth and fifth stages have the same KNF configuration. They may only differ in morphological features, which can even be difficult for human experts to detect.

The images provided by TrypTag are already pre-processed and come with segmentation masks. All images were subjected to the same corrections, as described in "Genome-wide subcellular protein map for the flagellate parasite Trypanosoma brucei" [Bil+23]. Here is the short version of what has been done: The first correction fixes the variations in brightness that could come from the camera's amplifier. The second correction is a flatfield correction, which involves using the median of all images captured on a particular day to correct for any inconsistencies or variations in the background illumination. Additionally, the images were scaled to the same size and the green channel was normalised by their exposure time. The images were also inspected manually for quality.

Since the images are already pre-processed and the images were selected based on their visual fidelity I took no further pre-processing steps to simplify the segmentation process.

3.3.3 Segmentation with K-Means Clustering

This subsection will be a bit different from the others. It will not only describe how and why I segmented the cell images the way I did but also include the results of my segmentation approach. Usually, the results are presented separately from the methodology. But in this case, incorporating the results of the segmentation approach into the methodology is reasonable, because the segmentation masks are the foundation for further experiments. Before further experiments and approaches are shown, the reliability and validity of the segmentation approach are proven. A quantitative research approach with a mainly inductive reasoning approach was employed for this experiment. It was mainly inductive because, first, images were observed. Then, a conclusion was drawn from the observations. After that, K-means clustering was applied to the images, which was justified by the observation. Lastly, the Elbow Method and Silhouette-Scores were applied to verify the initial observation and to find the optimal settings for the K-means clustering approach. From the results of these statistical methods, another conclusion about which number of clusters is best to use for each case and whether my initial assessment was correct was drawn. Parts of this also included deductive reasoning as a hypothesis was verified. This outlines the order in which the methodology will be presented. However, after the verification of the initial observation, another claim has to be proven before moving on to the next experiment. This is due to how the segmentation was implemented. The approach introduced randomness, which theoretically makes this segmentation approach not repeatable and, therefore, not suited for scientific research. However, I claim that even though randomness was introduced, the results will only be minimally impacted. Furthermore, by introducing randomness, this method avoids introducing a selection bias.

First and foremost, let me explain why segmentation masks are needed. Segmentation masks are the basis for feature extraction and further image data processing. A segmentation mask is a

binary image, highlighting which pixels are part of an object or area of interest and which are not. To classify procyclic T. brucei cells from images, it is necessary to know which part of the image is a cell, a nucleus, a kinetoplast or a flagellum, as those are the important organelles that are used to differentiate between cell cycle stages. For humans, segmentation is trivial, but computers do not understand what an image is. All the computer sees is an array of numerical values, but there is no understanding of what they represent. This is why most segmentation tasks require human supervision, and images of T. brucei are no exception. In this case, segmentation masks are primarily required for segmenting the fluorescent stains that highlight the paraflagellar rod on the mNG channel. The paraflagellar rod stains will be used as proxies for the development stage of the flagella to refine cell cycle classifications. The exact stages where this knowledge of the flagellum is important were covered in a previous subsection about gene selection. However, the segmentation approach was also applied to the other channels to test how well it works and compares to the existing masks. Although TrypTag provides segmentation masks for the phase-contrast and DNA channels, which are generally reliable, they can struggle with certain edge cases, such as clusters of cells or out-of-focus cells. However, this is not really a problem caused by the segmentation approach, as post-processing could fix overlapping cells. Therefore, I did not expect my segmentation approach to fix that.

I utilised K-Means clustering as an adaptive thresholding technique to segment the cells, nuclei, kinetoplasts, and flagella. The following paragraphs will explain, why K-Means clustering was used.

Upon first visual inspection of the phase-contrast channel, I noticed that the images have a simple structure with a uniform grey background, likely due to the flatfield correction pre-processing step applied by TrypTag. The cells appear dark and are surrounded by a bright halo. This is where the idea came to me to try K-means clustering on these images. The hypothesis is that the K-means algorithm will partition the histogram into exactly three parts that represent the dark cells, the grey background, and the bright halos around the cells. Using K-Means clustering has the following advantages: It is very simple to understand and use, there are methods to optimise the number of clusters, and therefore, the segmentation results. No tedious manual experimentation with different filters and image transformations are needed. However, the trained K-means classifier is very rigid and might be problematic when applied to the flagella dataset. As mentioned, some of the images are brighter, and some are darker. Additionally, there is the question of how to know which cluster represents the objects that are wanted. As an example, how do we know which cluster represents the segmented cells when we have three groups? But this could also be an advantage. TrypTag uses a rolling ball filter to minimise the impact of the bright halos. This pre-processing step is unnecessary when using K-Means clustering because the bright colour will be assigned to a different cluster than the dark cells. As demonstrated, there are a few problems with this approach, but I will explain how these were solved later. There are similar, very common approaches, like Otsu thresholding. What made me decide to use K-means instead? During my literature review, I discovered that K-means clustering and the Otsu thresholding method optimise the same objective function. However, K-Means finds a local optimum, while Otsu finds a global one. Additionally, this makes K-means more efficient, especially when more than two clusters are used [LY09]. Furthermore, K-means has also been used for segmentation in other related papers like Zhao et al. (2017) [Zha+17]. This made me pick K-means over Otsu thresholding. With that, I had a hypothesis that K-means clustering with 3 clusters can effectively and efficiently segment the phase contrast channel. The same process of manually analysing the mNG and DNA

channel was used to come up with similar hypotheses but with different numbers of expected clusters.

Before continuing further, I want to explain how and why K-means clustering works on images. The clustering algorithm operates through a series of iterations. First, k clusters are identified, and their centres are randomly initialised. Then, each data point is assigned to the closest centre, and the centres are recalculated based on the mean of the assigned points. This process is repeated until the centres converge beyond a certain threshold [SG13]. On images, it works in the following way: First, it looks at the image's pixel values. For a greyscale 8-bit image, each pixel can have a value between 0 and 255. Given the dataset's image size of 2560x2160 pixels, many pixel values will repeat. Additionally, given the observation that the image consists of dark cells, a uniform grey background and bright halos. One can imagine how this creates a histogram with three peaks. Then, K-Means divides this histogram into "k" slices. Initially, those slices are random, but over time, the algorithm will converge by reducing the variance within each slice. To reduce the variance within our histogram with three peaks, which represent a large number of pixels with similar values, the three randomly initialised centres need to each occupy one such peak.

To check if the K-Means clustering method is working well and if the right number of clusters were chosen, we can use two tools: the Elbow Method and the Silhouette Score [Tho53; Rou87]. The Elbow Method looks at how much adding more clusters reduces the sum of squared distances from a point to its assigned centre. We plot the number of clusters against this error, and the point where the error starts to flatten out looks like an "elbow". That's usually a good number of clusters to use. On the other hand, the Silhouette Score tells us how similar items in the same cluster are compared to items in other clusters. A higher score implies items are well-clustered. Together, these methods help us understand if we are clustering the data in the best way.

As a starting point for my experiments on cell segmentation, I used three clusters based on the previously mentioned observation that the image consists of three classes: the background, the halo, and the cell. This observation was then tested with the Silhouette-Score and the Elbow-Method.

The same technique was used for segmenting the kinetoplasts, nuclei, and flagella. The DNA stain used greatly simplifies the segmentation of kinetoplasts and nuclei, producing a strong signal for DNA. The assumption was that only the background should be black, and the kinetoplasts and nuclei should appear bright. Therefore, I assumed two clusters would suffice, which was used as a baseline and tested with the previously mentioned methods. The only rare edge case that could occur is that of ruptured dead cells. In that case, the signal could be smeared out, and consecutive measurements of the nucleus or kinetoplast area would be inaccurate.

The segmentation of the flagella was done similarly. Due to the specific rules I set for the visual inspection of the candidate genes during dataset creation, the segmentation should be straightforward, and two clusters were used as a baseline for the mNG channel and were tested with the Elbow method and Silhouette-Score.

Before creating the segmentation masks for all three channels. I had to verify whether the experiments would match the expectations. For this, a Python script, namely "clustering_metrics.py", was written that uses the sci-kit-learn Python library to compute the K-means segmentation of a single image, the Silhouette Score and the Elbow Method. The Silhouette Score is computationally expensive compared to the K-Means Clustering algorithm and the Elbow Method. It has a quadratic runtime complexity, while K-Means and the Elbow Method have linear runtime. For this reason the runtimes are exploding when using large images. Especially for the high-

resolution images with 2560 by 2160 pixels. Because of that, I used a cropped-out 512 by 512 pixel region in the middle of the image, which took around 15 minutes to compute on my hardware. Computing this score for the whole image and all the different cluster numbers (K) would probably take multiple hours, even though I already used multiple cores to distribute the load. This was achieved by using the Python library Ray. Specifically, the script takes a single image and the base assumptions of how many clusters are expected for each channel. The program begins by calculating the statistics for the first channel. First, the image is cropped, but it can also be resized in case the 15-minute runtime is too long to wait. After cropping, the program distributes the different test cases with different numbers of clusters to the CPU. So one core might calculate the statistics for K = 3, while another core calculates it for K = 4 and so on. On each core, the 2-dimensional image is first flattened to one dimension. After that it can be passed to the K-means classifier. Next, with another sci-kit-learn function, the Silhouette Score and the values for the Elbow Method are calculated and the segmented image is returned. Once all the cores are done calculating the different statistics, the data is saved and later visualised with Matplotlib. My implementations in Python used OpenCV as the primary graphics processing library instead of the alternative Skimage. OpenCV is natively implemented in C++, while only some parts of Skimage are implemented in a lower-level language like C++. This makes OpenCV more performant. In this script, OpenCV can only be applied to resizing the image. But overall I was very careful in deciding which libraries to use. A further example of this: The computation of K-Means clustering, and the Elbow Method was very quick out of the box. Further enhancing this speed was the Intel Extension for Scikit-learn, which only works on Intel hardware. To change the code to also work on AMD CPUs only one single line needs to be changed in the imports. Based on their benchmarks, this extension can accelerate K-Means clustering training by a factor ranging from 2.8 to 49.8 and boost inference speeds by 5.9 to 72.3 times, depending on the dataset size. I could not find a similar implementation for the Silhouette Score to speed it up. For this reason, I used the 512 by 512-pixel images to compute the plot for the Elbow Method and the Silhouette Score, which is also handy for showing them in this thesis. Furthermore, I set the range for the number of clusters. The minimum was the baseline of expected clusters I had previously determined, and the maximum was three clusters above the expected amount. Setting the maximum number of clusters to three more than the baseline was an arbitrary choice. Initial experiments showed that more clusters only pick up noise and do not correctly segment the objects. One drawback of my approach is that I would draw conclusions about the optimal number of clusters based on a single image. But the assumption was that if the segmentation works on one of the images, it would also work on all the other images. As they were captured with the same hardware. Although there are differences in the brightness of the images between different cell lines, the brightness should only shift the histogram to the left and the right and, therefore, not affect the clustering.

The images for showcasing this method were created based on the image Tb927.3.4290_4_N_1.tif.

Figure 3.5 shows the results of using different K in K-Means Clustering applied to a cropped image of the phase-contrast channel, channel 0. The first image on the left is the original without assigned clusters. To the right is the same image four times with different K, ranging from 3 to 6. Each clustered image to the right of the original image should be made up of exactly K colours representing the different clusters. The second image, where K equals three, has three colours: black, green, and grey. As previously explained in my intuition about the lower bound, the image should comprise three components: the background, the cell, and the bright halo. Using three clusters reveals exactly this expected structure, where the black colour represents the

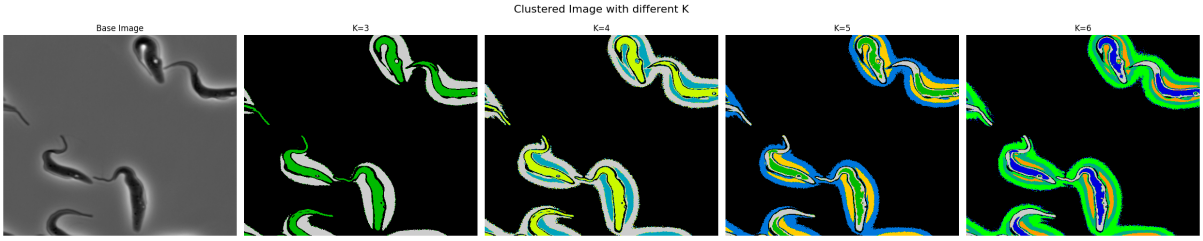

Figure 3.5 – Comparison of the K-Means Segmentation with different numbers of clusters K on the phase contrast channel of a cropped Image.

background, grey represents the halo, and green represents the cell. However, the segmentation is imperfect since the green-coloured cells are surrounded by a black border, which makes sense but wasn't obvious before seeing it. The black border exists because the clustering splits the histogram into three parts: the brightest part is the halo, the darkest part is the cell, and the grey background lies between those two parts. So, if there were a smooth gradient between the halo and the cell, it would have to pass through the region in the histogram that encodes the background colour, therefore colouring it black. There has to be such a gradient because we see the black border. I attribute the existence of the gradient to two reasons. The most important reason is that the cells are not uniform in thickness. This indicates that in the middle of the cell, we are looking through more of the cell than at the edges, which makes the middle appear darker. This is probably a result of the imaging technique. The second reason is the applied pre-processing. Assuming a sudden brightness jump between the bright halo and the dark cell, the smoothing used during pre-processing would create a smoother transition between those two zones. Potentially creating intermediate pixel values that the clustering algorithm assigns to the background colour. Besides that, there are two more downsides to this clustering. First, circular holes in the cell cluster are assigned the halo cluster. This bright circular structure is located near the kinetoplast or nucleus. I excluded those two organelles as the cause of this, because I compared the phase contrast images with the images from the DNA channel, and they were not overlapping, so it can not be the kinetoplast or nucleus. The basal body is also unlikely just because of its appearance. Although it is irrelevant to this thesis, I determined it is likely a lipid droplet due to its circular appearance compared to figure 2.1. Secondly, the flagellum is not entirely picked up in the same cluster as the cell in any of the different K's. It is mostly classified as background, which was somewhat expected, as mentioned in the subsection for the gene selection. The third image, with K equals four, has a similar structure but adds one cluster, which divides the halo again. K equals five, which adds one cluster within the cell, and K equals six, which seems to add another cluster in the halo again. From that, I concluded that more than three clusters add more noise instead of segmenting the image more precisely.

In addition to this visual inspection of the clustering, I used two tools to evaluate the clustering: the Elbow Method and the Silhouette Score. Figure 3.6 shows the plot for the Elbow Method for the phase contrast channel. The y-axis shows the so-called inertia, the sum of the squared distances of each sample to its cluster centre. The x-axis represents the number of clusters or the parameter K. With the Elbow method we expect the inertia to drop sharply in the beginning and flat out as more clusters are added. We are searching for a K where this flattening happens, and additional clusters do not significantly decrease the sum of squared distances within each cluster. For this data, it seems to be at K equals four.

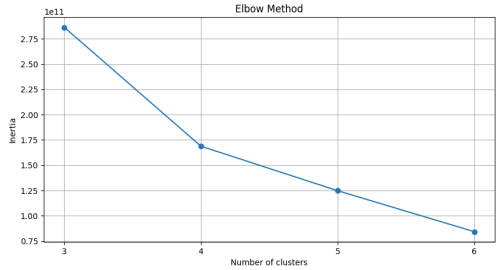

Figure 3.6 – Elbow Method applied to the clustering of the phase-contrast channel

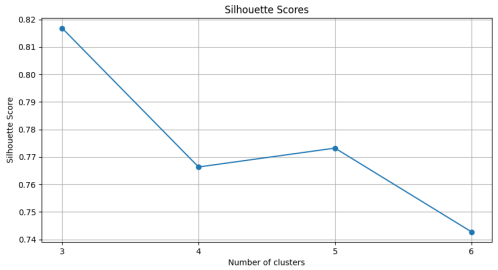

Figure 3.7 – Silhouette-Scores for the clustering of the phase-contrast channel

The plot in Figure 3.7 displays the Silhouette Scores for different values of K in the first channel. The y-axis represents the Silhouette Score, used to evaluate the clustering quality within and across clusters. The Silhouette Score is computed for each sample and ranges from -1 to 1. A score closer to one implies that the point is very close to other points within its cluster and far away from points in other clusters. A score of 0 means that this point lies directly on the separation border between two clusters, while a score of -1 indicates that this point was misclassified. According to the plot in Figure 3.7, for channel 0, the highest Silhouette Score occurs when K equals three, which is around 0.82. Furthermore, for four or more clusters the Silhouette Score drops off to around 0.74 to 0.77. Based on this heuristic three should be chosen as K.

Based on my visual inspection and the Silhouette Scores, I was expecting that three clusters would be the optimal choice. However, after analysing the Elbow Method, it was suggested that four clusters would be a better option. To verify this, I experimented with the data by trying out both three and four clusters and found that four clusters were more consistent. Using four clusters allows the cell to remain as one class, as it is very dark compared to the rest of the image, and adds separation to the halo. The additional cluster might also help with the separation of other parts. In the introduction, two drawbacks of the segmentation with K-means were mentioned. First, its rigidity might be problematic for the diverse dataset and second, the selection of the correct cluster when there are more than two clusters. The first problem was solved by simply training a K-means classifier for each gene, as the images from a gene should come from the same experiment and, therefore, have almost identical image characteristics. Now that we have determined that four clusters are optimal for segmenting the phase-contrast channel, the second problem has to be addressed, as this should be an automatic system. Luckily, TrypTag provides phase-contrast masks and DNA masks. To solve the second problem, the TrypTag masks and the newly generated masks are overlayed, and the cluster with the highest overlap is chosen. In case there are only two clusters, a heuristic might be used. For example, inspect the images and determine the class that covers more area in the image. This class will be labelled as background, and the other will be used as the segmentation mask.

The Elbow Method and Silhouette Scores were also used to analyse clustering for the other two channels. For Channel 1, the mNG channel that highlights the paraflagellar rod, it was anticipated that two clusters would be the optimum since the genes were chosen in a way that simple thresholding would suffice. Figure 3.8 displays a cropped clustering with different K values ranging from 2 to 5. The range differs from channel 0 since the expected base case is different. Clustering with two clusters produces a good mask containing all objects. However, the tagged paraflagellar rods

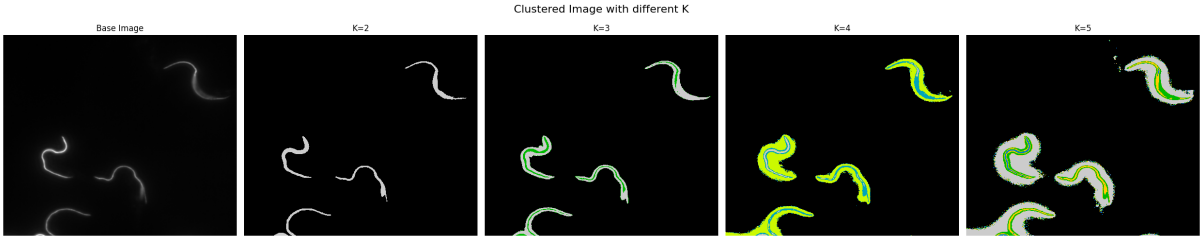

Figure 3.8 – Comparison of the K-Means Segmentation with different numbers of clusters K on the mNG channel of a cropped Image.

in the original image are smeared and have a halo around them, making the masks with two clusters a bit bigger than the original object. Clustering with three clusters improves the separation, and the green class almost perfectly masks the objects. However, some of the objects are not detected entirely, or they are spotty. Such as the one in the top right corner of the image with K equals three, which suddenly ends. Or the object in the bottom left corner, which is divided into multiple green connected components. The same trend of incomplete or divided objects continues when adding more clusters. Another issue is the cluster selection with more than two clusters because there are no obvious heuristic or pre-computed masks that can be used since such masks do not exist. Alternatively, hand-labeling the correct cluster goes against the goal of making this process fast and unsupervised.

Charts for the Elbow Method and the Silhouette Score were created, but they were not as clear as for the phase contrast channel. Figure 3.9 displays the plot for the Elbow Method for channel one. The elbow is not as clearly visible as for channel zero because the curve is smoother. Based on the Elbow Method, three clusters are arguably the best choice. Figure 3.10 shows the Silhouette Scores plotted against the number of clusters. There is no clear peak in the plot like in the previous plot, but all the scores are very high. Specifically, two clusters have the highest score.

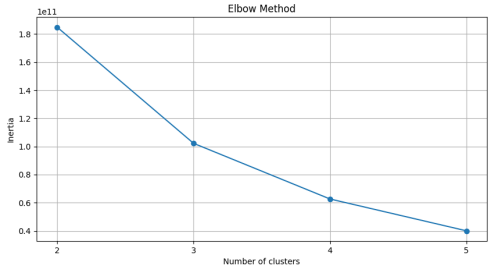

Figure 3.9 – Elbow Method applied to the clustering of mNG channel

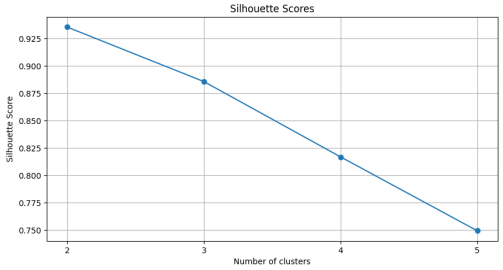

Figure 3.10 – Silhouette-Scores for the clustering of the mNG channel

For the mNG channel that displays the paraflagellar rod, the initial estimate of two clusters was chosen for the following reasons. First, two clusters produce the only images where most objects are masked completely and as one connected component. Second, it produces the highest Silhouette Score. Lastly, the selection of the correct class with more than two clusters is difficult as TrypTag does not provide segmentation masks for the mNG channel.

The same method used for the first channel was also used for the second channel, which is the DNA channel. Two clusters were the base case. Figure 3.11 shows the cropped original image,

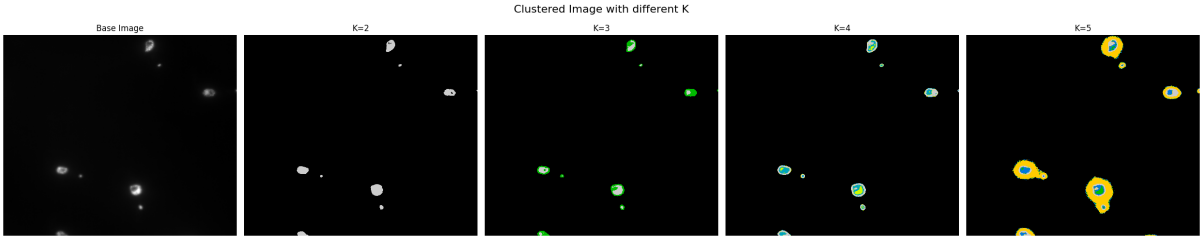

Figure 3.11 – Comparison of the K-Means Segmentation with different numbers of clusters K on the DNA channel of a cropped Image.

ranging from 2 to 5 clusters. When there are two clusters, all the objects are correctly masked with only minor flaws, such as small holes. These holes can be filled using morphological operations. When there are three clusters, the brightness variations within each DNA signal are picked up, which are not needed for the masks. With four clusters, the objects from the original image are divided into more classes, but the overall shape is still maintained. However, with five clusters, the object's shape is no longer intact, and the additional class picks up the noise around the objects. Following a visual inspection, it was determined that selecting K equals two would provide more meaningful information, while selecting more clusters would only add noise. It was noted that the same problem of cluster selection existed as for the mNG Channel. Therefore, it was concluded that two clusters were the only suitable option. For completeness, I also made the Silhouette Score and the Elbow Method plot and analysed them. The plot in figure 3.12 for the Elbow Method is similar to the first one from channel 0. It shows a clear elbow at K equals three. Figure 3.13 shows the Silhouette Score plot, which has high Silhouette Scores for each K. However, only two clusters should be chosen as they have the highest score. This plot also matches the visual description, where 5 clusters significantly reduce the quality of the masks, resulting in a lower score.

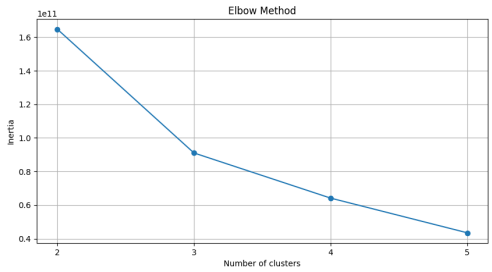

Figure 3.12 – Elbow Method applied to the clustering of the DNA channel

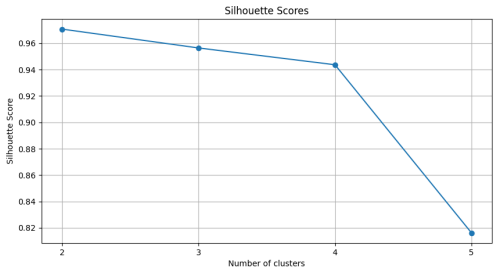

Figure 3.13 – Silhouette-Scores for the clustering of the DNA channel

To segment the DNA channel with K-Means clustering, I decided to use two clusters for the following reasons: The visual inspection of the different clusterings shows that two clusters produce the best results, which the Silhouette Score also confirms. More clusters, like three, are difficult to use since the cluster selection problem arises again.

To train a classifier for each of the ten genes in my dataset, the determined number of clusters was used. This is because the histograms can be very different between genes. For each of the three channels, images from one gene were stacked instead of using just a single example from one gene to train the classifier. This approach was preferred because it allowed the classifier

to see more pixels and hopefully generalise better. It also did not slow down the whole image processing pipeline.

After clustering the images, I still faced the challenge of selecting the correct cluster. The process was straightforward for the mNG and DNA channels since the clustering yielded only two classes: one representing the background and the other representing areas of interest. Given that most of any given image is typically background, I assigned the larger class as the background and the smaller one as the areas of interest. However, segmenting the cell was more complex. Due to variations in brightness and density in the images from each gene, I considered training a K-Means clustering classifier on all images associated with a specific gene. This way, the classifier would recognise the distribution of different classes unique to that gene. Once the cell class was identified for one gene, this classification could be applied to all images of that gene. This approach would be repeated for each of the ten genes in the dataset. The approach I chose was using the pre-computed masks delivered by TrypTag and computing the overlap. This only works with the correct number of clusters and accurate pre-computed masks.

This section provides a detailed description of how I created the necessary segmentation masks for further analysis. I explain how K-Means clustering can be used as a faster alternative to Otsu-Thresholding and demonstrate how to identify the correct cluster using either a simple handcrafted rule, such as selecting the smaller class, or by using existing masks to calculate and select the class with the most overlap. To do this, the correct number of clusters must first be determined. This was done using a visual inspection, the Elbow Method and the Silhouette Score. The final selection of the optimal number of clusters for each channel is as follows: Four clusters for the phase contrast images for creating cell masks, two clusters for the mNG Channel that highlights the paraflagellar rod, which was used as the mask for the flagella, and two clusters for the DNA Channel.

3.3.4 Incorporating Flagella Data

To accurately classify Trypanosoma brucei from microscopy images, the flagella must be considered an additional information source. However, even with the knowledge of KNF configurations, it is challenging to distinguish the different cell cycle stages, particularly the 2nd and 3rd stages, which both have the 1K1N2F configuration and the 4th and 5th stages, which have the 2K1N2F configuration. To differentiate between these stages, measuring the size of the nucleus, kinetoplast, and the new flagellum is necessary. Unfortunately, counting the flagella was impossible due to the unclear signatures in the mNG channel.

After finding the optimum parameters for the K-means-based segmentation approach, the insights had to be incorporated into the dataset to create masks for all of the different genes. For this, the Python script "compute_segmentation_masks_and_dataset.py" was written. This script is responsible for generating all the segmentation masks and adding them to the dataset directory but it can also be used to look at the complete dataset. The dataset is represented as a TrypTag-Dataset class object and saves information about where the dataset is stored, which genes it contains and a list of fields. The fields are another class object that is initialised upon loading the dataset. The Field class has an index, a terminus, a gene name, an image path, a mask path, a base path and a list of cells. It provides functionality for returning the different images and masks but is also responsible for creating the masks for each channel. It also solves the problem of using the correct class as the mask by either using the class with the largest overlap with existing masks or

simply choosing the class with the smaller area, as the larger area is defined as the background. The Cell class only saves an index, different statistics of the cells and the base path. It also provides functionality to return the images and masks for the three channels. This is done by using the statistics that save the cell's position within the Field and cropping the Field images and masks. This is done because it avoids saving the images and masks for thousands of cells when they are already saved for the Fields. The added overhead of cropping the image is minimal and probably less than loading the image from the file. The masks are created in the following way: First, the train_kmeans function is run, which is provided with a dictionary that contains information about which channel uses what number of clusters and a directory of a specific gene. The function then picks a maximum of five random field images from the gene directory to keep the memory in check. The random sampling also serves the purpose of not introducing any bias. However, this makes the program non-deterministic. The randomly sampled field images are then used to train a K-means classifier, saved in the kmeans_models directory. The file name of the classifiers contains information about the gene, the channel and the cluster it is used for. After the models are trained for a specific gene it can be loaded in the TrypTagDataset class. Once loaded, the Field class will automatically use the correct model and image files to create the masks. First, the image is clustered with the model, and the correct class is extracted as the mask. Then, a hole-filling algorithm from the Python library Scipy is used, which internally uses the morphological dilation operation. This is done to prevent the holes that occurred and was attributed to organelles or lipid droplets within the cell. After that, the individual connected components are filtered by area. This prevents noise, destroyed cells, or even the edges of the well plates from being falsely classified as the object of interest. Additionally, this script also provides plotting functions to visualise the K-means clustering process or the area filtering to tune the parameters further. However, the main purpose of this script is to train the different K-means models and to create the masks. All other functionalities exist for data exploration purposes and are a by-product of the development process but have not been applied to solve other research objectives. The resulting masks for the field images for each channel can be seen in A.6, A.5 and A.4. Those images are the results of applying the K-means clustering with the determined optimal number of clusters to gene Tb927.6.4140_4_C_1. However, the further processing steps of hole-filling and filtering connected components by size have not been applied to these images. In image A.6, one can see the mentioned edge of the plate well, the small white areas which are noise or out-of-focus cells, and upon closer inspection, even the small circular cutouts in the cells, where I could not exactly determine which organelle they represent. But all these errors, are largely fixed by applying the size filter and hole-filling. Next, the resulting mNG mask is shown in image A.5. It clearly picks up all the flagella. However, there are a few errors. Close to the bottom, there are a few spotty segmentations. This indicates that the signal of the flagellum was not picked up as a whole but was divided into multiple smaller masks. Right above that are two overlapping segmentation masks, which are not usable. On the other hand there are a few examples of cells with a developing second flagella. For example, the second bottom-most segmentation mask shows two flagella in one mask. This cell is likely in the last cell cycle stage. Another example where the growth of a second flagella is visible is the mask in the top-left corner. The second flagella has not grown as much. But judging by the DNA and phase mask in figures A.4 and A.6, it is visible that this cell has two kinetoplasts and the nucleus is no longer circular. So, this cell is likely in stage four or five. Unfortunately, it is very hard to find examples of the important early stages, where the second flagellum begins to grow. One possible example could be the mask in the top right corner.

Left next to it is a smaller c-shaped flagellum signal. When looking at the raw mNG-channel image, one can see a second flagellum-like signal strand. In the mask, however, this is not directly visible. Only the upper two-thirds of the flagellum signal appears thicker than the lower third. Looking at the DNA mask for this cell, one kinetoplast and one nucleus are visible. But the kinetoplast does not appear oval, like it is dividing. From that, two conclusions are possible. Either the cell is in late-stage two or early-stage three with a 1K1N2F configuration, or the signal from the paraflagellar rod only appears as two strands. It is hard to tell from the raw mNG channel image but almost impossible from the mask alone. Lastly, the segmentation of the DNA masks came out very well. The nuclei and the kinetoplasts are easy to tell apart by their size. There also seems to be no noise; each of the individual masks seems to have another mask next to them. Indicating a pair of kinetoplast and nucleus. Additionally, there are no holes in the individual masks like in the phase-contrast masks for the cells.

Evidently, the segmentation using K-means clustering of the mNG channel using the paraflagellar rod was not accurate enough to differentiate between the important early stages one to three. The masks might only be useful for detecting the second flagellum once it has grown enough, which could be useful for telling the later stages apart. However, this would require skeletonising the masks so that two branches, one for each flagellum, are present. This introduces new challenges like splitting one mask into two likely flagellum masks. This is a challenge that might be solvable by using existing skeletonisation techniques like MAT, representing the skeletonisation as a graph, and using graph algorithms to process it. Unfortunately, this was out of the scope of this thesis.

Although the initial goal of detecting the early growth of the second flagellum failed. The mNG masks could be used to create visualisations and statistics about the flagella lengths. One of the following sections will explain how and why this was done.

Lastly, as highlighted in the introduction, I want to address the randomness involved in the process. As explained, a maximum of 5 images are chosen randomly for each gene to train the K-means clustering model. This will make the following statistics almost impossible to reproduce exactly like in this thesis. Because the masks change slightly, depending on the selection of images. However, this was done to eliminate a sampling bias. Nonetheless, I had to prove that the randomness introduced does not change the results significantly. To prove this, I ran one of my experiments that measured cell and flagellum length for each of the expected KN configurations repeatedly. This experiment will be further explained in the section on MAT improvement. For it, I wrote yet another Python script with the name length_histogram.py. Further code was added to this script to prove my claim. This would code, run the experiment 10 times and record the resulting data. The resulting data of each run, are the raw lengths of flagella and cells but also the mean and standard deviation. Before each run, the K-Means models were trained. Each time they are trained, they introduce randomness by randomly selecting the files to sample the data points. This results in each of the ten runs being unique. At the end, the recorded data was plotted using Matplotlib. A boxplot for each of the expected KN configurations, which are 1K1N, 2K1N and 2K2N, was generated. With that, I want to show three things. First, the results for a single gene across the ten runs do not change significantly. Second, the mean of the lengths across the ten runs for each gene does not change significantly. Lastly, the standard deviation of the lengths across the ten runs for each gene is not impacted significantly. This resulted in a total of nine boxplots. I will only show three of the nine plots for the 1K1N configuration for demonstration

purposes. The resulting plots for 2K1N cells and 2K2N cells can be found in the appendix under figures A.7, A.8, A.9, A.10, A.11 and A.12. This is because the 1K1N configuration is the most common and, therefore, has the largest sample size. The other two configurations have very small sample sizes. Especially in the first plot, which shows the variation in cell and flagellum lengths for a single gene across ten runs, this is important. Only data from a single gene is present, limiting the sample size. I randomly selected the gene Tb927.10.11300 for this. This can be seen in figure 3.14. It consists of two subplots. On the left is the data for the cell lengths, and on the right is the data for the flagella lengths. The y-axis shows the length in micrometres, and the x-axis shows the number of the run. The mean and standard deviation are depicted as blue error bars. The red line shows the median value and highlighted in golden, the boxes for the values that are between the 25th to 75th percentile. The black error bars show the range of the data. Additionally, the black circles are outliers. The outliers are automatically identified by Matplotlib and are classified as such if the value is 1.5 times the inter-quartile range below the first quartile or 1.5 times the inter-quartile range above the third quartile. In both subplots, the individual runs are almost indistinguishable. Furthermore, even the outliers for the cell lengths are the same after randomly changing the training set of the K-Means segmentation model for each run. The only difference I can barely see in the left subplot is the slightly changing medians. In the right subplot, I can tell slight differences in the 25th quartiles.

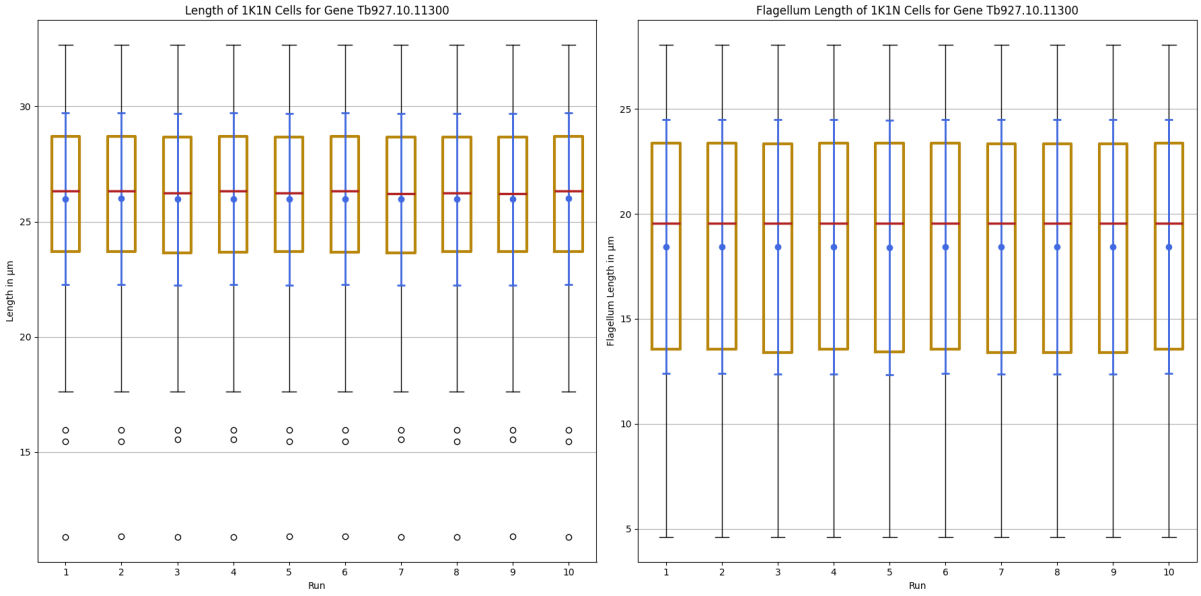

Figure 3.14 – Boxplot comparing the results of the cell and flagellum length measurements across ten runs for 1K1N cells of the gene Tb927.10.11300. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

To further strengthen my point that the randomness introduced does not significantly change the results and that this is true not just for this single gene but for all genes, plots for the mean and standard deviation were created that summarise the information of all runs. To do this, figures 3.15 and 3.16 were created. In the previous figure, boxplots were shown for the individual runs. However, in the following two figures, the information for the ten runs is summarised as a boxplot. The first figure summarises the mean cell and flagellum lengths across all runs, and the

second figure summarises the standard deviation of cell and flagellum lengths. Therefore, in the following two plots, each individual boxplot represents either the variability of the ten mean values computed from the ten runs or the variability of the ten standard deviations computed from the ten runs. This should highlight that the mean and standard deviation for each gene do not change significantly across the ten runs. For this to be true, one would expect very small boxplots. In the ideal case these boxplots would simply be lines, because all the values would be the same. This would mean that my claim is true. However, if very large boxes are observed, the mean or standard deviations across runs differ significantly.

Let's look at the first figure 3.15. Again, the image displays two subplots. The mean cell lengths are on the left side, and the mean flagellum lengths are on the right. The y-axis shows the mean of the lengths in micrometres, and the x-axis shows the gene for which the data was summarised in a boxplot. The very first thing that stands out is that the mean lengths differ significantly between the different genes. This is true for both flagellum and cell lengths. For example, for Tb927.8.4970, the mean cell length in all ten runs was below 24 micrometres. In contrast, Tb927.8.1550, right next to it, where all recorded mean cell lengths were above 28 micrometres. The same is true for the flagellum lengths but with different genes and lengths. Secondly, it is very noticeable that the individual boxes do not look at all like in a typical boxplot. For most of the genes, the median, mean, and standard deviation lines, as well as the boxes, overlap and are almost lines. For the mean cell lengths, only Tb927.10.9570 and Tb927.11.14880 seem to have larger boxes than the others. For the mean flagella lengths, only Tb927.8.4970 and Tb927.3.4290 were larger boxes compared to the other genes. This implies that for those genes, the different K-Means models have a slightly larger effect on the mean value compared to the other genes. However, most of the boxes are, as expected, almost lines. Because all the recorded mean values from the ten different runs lie very close together. This confirms that the observation made for figure 3.14 holds true for all genes and not just the one shown in this plot. The variation between genes could be explained because they are different populations and have not been timed to have the same cell cycle stage.

Next, the standard deviations could still be different across runs with different models. To test whether the randomness introduced in the model training influences the standard deviation in the length measurements, a plot that is very similar to the previous one was created. Again, small, almost line-like boxes are expected. If that is the case, we can conclude that all the relevant statistics, namely mean and standard deviation, are not influenced by randomness.

The structure of figure 3.16 is the same as the previous one, except the plot now shows the standard deviations. Therefore, the y-axis now shows the standard deviation of the length in micrometres. Each box for an individual gene represents the ten standard deviations measured for each of the ten runs. As before, initially, we can see a wide spread between genes. For example, for gene Tb827.8.5000, all of the ten recorded standard deviations of the lengths are below 3 micrometres. In contrast, Tb927.8.1550, where all standard deviations of the cell lengths are about 5 micrometres. Again, the same is true for the standard deviations of the flagella lengths across the genes. This time the boxes for the individual genes are almost all line-like for the left subplot, with the exception of Tb927.11.14880 and Tb927.11.15100, which both have one outlier. For the right subplot, we see for the first time much larger boxes compared to the others in Tb927.8.4970 and Tb927.3.4290. However, from those observations, I would still conclude that the variation in the standard deviations of measure cell and flagellum lengths is minimal. This is because, for the majority of genes, the observation holds true.

After this conclusion, I was still interested in why these larger boxes were occurring. Especially

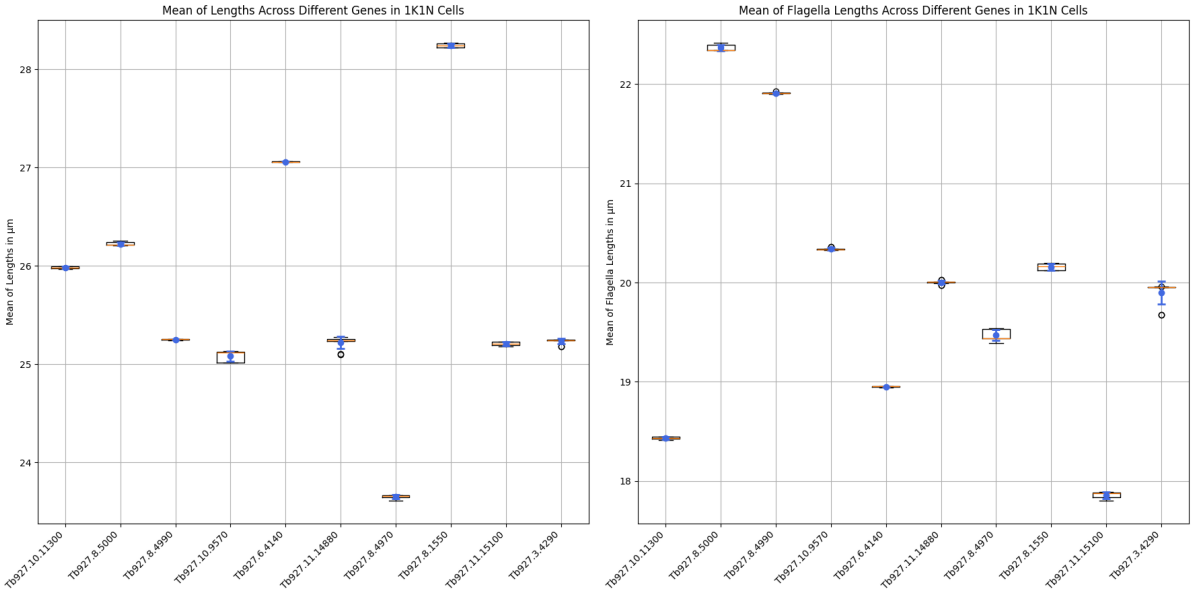

Figure 3.15 – Boxplot comparing the mean of the cell and flagellum length measurements across ten runs for 1K1N cells for all genes. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

in Tb927.11.14880 and Tb927.11.15100. After inspecting the raw images from those genes I had a possible explanation. Those two genes, Tb927.8.4970 and Tb927.10.9570, have much higher cell densities than the others. I used the TrypTag repository and counted 734 cells for Tb927.11.14880, 675 cells for Tb927.11.15100, 725 cells for Tb927.8.4970 and 614 cells for Tb927.10.9570. Each of these genes has five images. Other genes have 237, 82, 152, 296, 150, and 347 cells, respectively, within 4 to 10 images. The fact that the critical cases all have five images makes this even more interesting. In this case, my sampling strategy chooses exactly five files, and no file can be picked twice. Therefore, all images are used for training, and the randomness is not introduced by my sampling methodology but by the K-Means algorithm itself or the TrypTag repository. I have set a random state for the K-Means algorithm, which should produce the same results with the same input data. Therefore, the randomness must come from another source for all the genes with five images. After checking the TrypTag repository and the functions used, I have concluded that the randomness for genes with five or less than five images is not introduced by the segmentation approach but by using the MAT implementation from the Sklearn Python library. There are two sources of randomness for genes with more than five images. First, the random sampling and, secondly, the medial axis transform which is used to determine the cell and flagellum lengths. The medial axis transform method in the Sklearn library provides the ”rng” argument. This argument can be used to set a seed for the random number generator. This is good to know and would improve more reproducible results. However, I decided against implementing this change due to time constraints and because I have already shown that all the randomness that is introduced does not have a significant impact on the experimentation results.

To conclude, this subsection explained how I implemented and used the before-gained information about the optimal number of clusters for the segmentation and how it resulted in segmentation masks for all three channels. Unfortunately, it was also discovered that the approach

of using the paraflagellar rod as a proxy for the flagellum is not viable. It requires further research and novel approaches to extract the flagellum from the mNG masks. Only the later stages can be differentiated from the current approach. This discovery will greatly limit the ability to classify all six stages. It was also highlighted that my approach introduced some randomness, which later uncovered a further source of randomness. However, it was shown that all sources of randomness do not significantly influence the mean and standard deviation of the results. This means that the same approach can be taken for further experimentation. Overall, this demonstrates how K-Means clustering can be used to efficiently segment cell images.

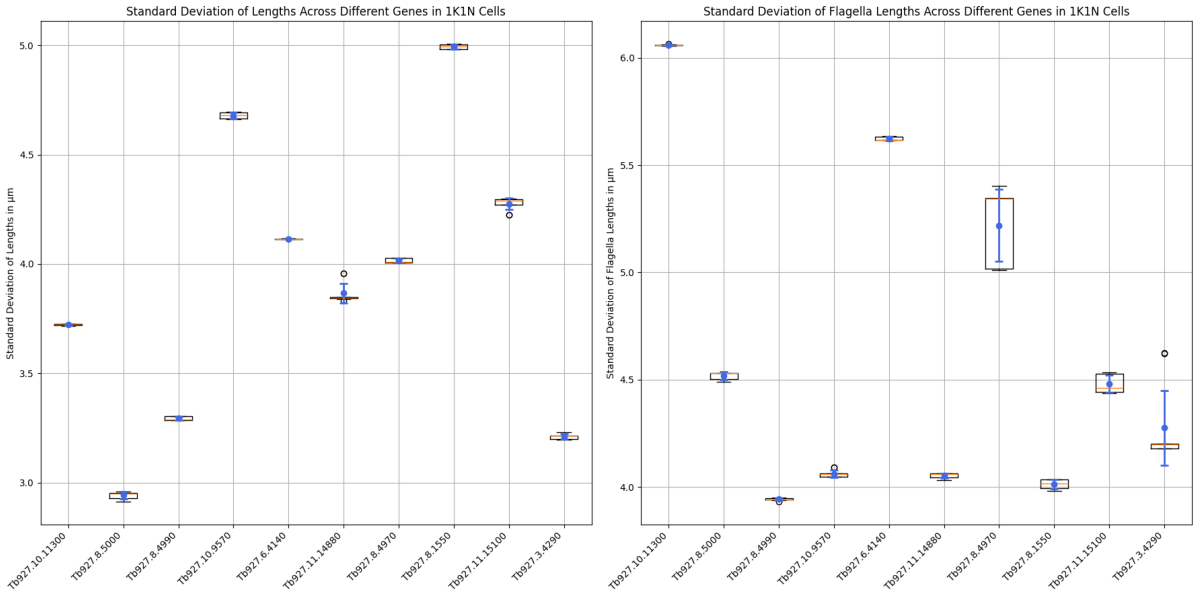

Figure 3.16 – Boxplot comparing the standard deviation of the cell and flagellum length measurements across ten runs for 1K1N cells for all genes. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

3.4 Feature Extraction and Selection techniques

Accurately classifying Trypanosoma cell cycle stages requires careful feature extraction and selection. These techniques help identify the most relevant information from raw data while discarding redundant or irrelevant features to enable more robust and accurate classification models. Traditional features such as kinetoplast, nucleus and flagellum count and position, cell length, cell width, flagella length, and DNA content have been found to be good priors for cell cycle classification. However, the accurate measurement of these features is crucial. This thesis proposes a more accurate way of calculating the cell length based on the medial axis transform. It also proposes new features, namely the minor and major axes of the nucleus and kinetoplast, which are obtained through elliptical fitting to refine the classification process further. ResNet-50, a well-known deep learning model, was used in combination with dimensionality reduction methods in an attempt to find visual differences that might hint at different cell cycle stages. Additionally, multivariate scatterplots are utilised to examine the relationships between the aforementioned features and their KN configurations. Together, these techniques provide a comprehensive ap-

proach towards improving cell cycle classification. The following sections use a quantitative and predominantly deductive approach. However, the analysis of scatterplots uses a mixed reasoning approach.

3.4.1 Improvement in MAT-Based Length Measurements

Improving the accuracy of any measure will ultimately improve classification performance. As outlined in section 2.2.1, the pruned skeleton generated from the MAT does not span the entirety of the cell. This is also visible in Figure 3.17, which uses a mask provided by TrypTag and the skeletonisation method used in Wheeler et al. (2012). To improve this measurement, I will add a corrective term that reduces the error produced by the current method.

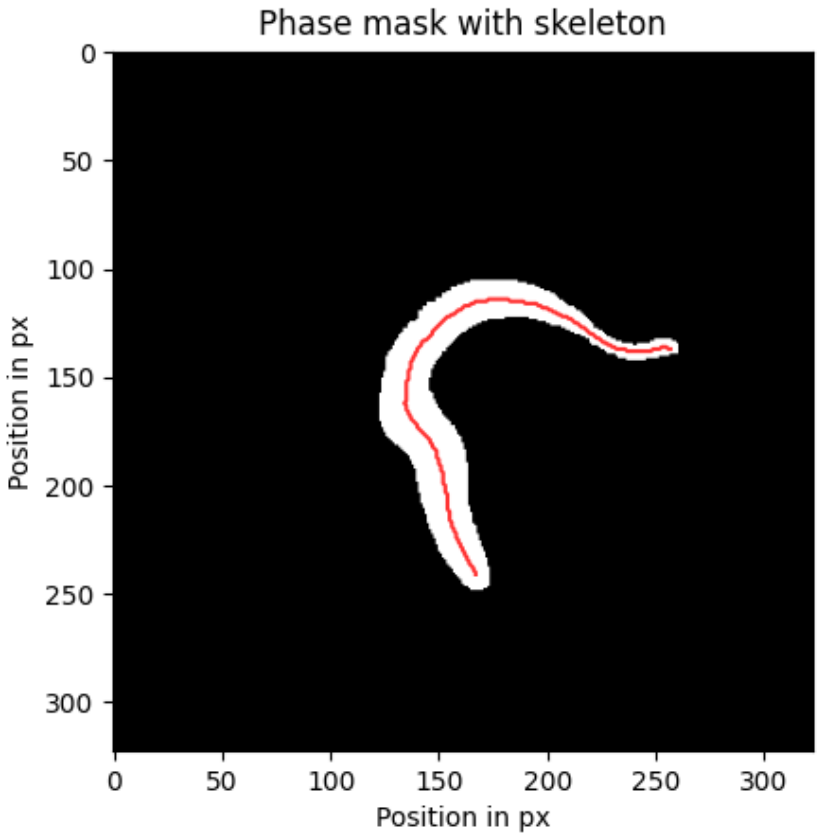

Figure 3.17 – A segmentation mask of the phase contrast channel in white with the skeletonisation line in red.

The current method, which is implemented in the TrypTag GitHub repository and mentioned in "Detailed interrogation of trypanosome cell biology via differential organelle staining and automated image analysis" [WGG12] uses the pre-computed masks provided by TrypTag and applies the medial axis transform to skeletonise the cell. However, this skeleton may contain many branches due to irregularities in the cell masks. A pruning algorithm is applied. that returns only the midline of the cell mask. The length is then calculated by stepping along this midline. Since

the size of each pixel in terms of micrometres is known, one can calculate the distance from one endpoint to another using simple arithmetic and the Pythagorean theorem in case we step diagonally.

They found a three per cent difference in manually measured cell length compared to the automatically measured length. Although three per cent are within the margin of error, this systematic error can and should be fixed by introducing a corrective term. The MAT is made up of the centre points of all maximally inscribed circles, and the value at this position is the radius of this circle. Because of that, the first and the last points along the medial axis do not touch the anterior or posterior end of the cell. To correct the measurement based on the MAT, simply add the following term: Given that the old length of the cell is represented by L_{old}, which is calculated using the previous approach, and the values of the medial axis transform at the first and last index are represented by $\text{MAT}_{\text{first}}$ and MAT_{last} respectively. These values from the MAT contain the radius of the circle that is maximally inscribed in the mask. The corrected length L_{new} can be computed as:

$$L_{\text{new}} = L_{\text{old}} + \text{MAT}_{\text{first}} + \text{MAT}_{\text{last}}$$

This change was implemented in Python using existing methods from the TrypTag GitHub repository and applied to the previously built dataset. The script with the name length_histogram.py has been mentioned before while proving that the randomness does not affect the results. The measurements were performed for each KN configuration that occurs during the cell cycle. Outliers and mutations like 0K1N, 1K0N, 3K2N and others were discarded since those are mostly the result of errors in the methodology. This method of measuring length was also applied to the flagellum. Because I could not match all flagella to a cell but wanted to have the same sample size, I only used cells where I could also match the flagellum. Any errors in my script or the existing scripts were caught, which could result in an overall lower sample size. For each analysis of cell and flagellum measurements, a plot was created that shows the distribution of lengths which can be seen in the appendix under A.13, A.14, A.15, A.16, A.17, and A.18. In summary, this script generates histograms for the cell lengths, corrected cell lengths, flagellum lengths and corrected flagellum lengths for 1K1N, 2K1N and 2K2N cells. Additionally, it saves an image for each individual cell in the directory flagella. Within this directory are subdirectories for each of the relevant KN configurations. These images show the cell mask, the flagellum mask, as well as the kinetoplasts and nuclei. It is built on top of the plot_morphology_analysis from the TrypTag repository.

Here is how the script works in detail. First, the genes were processed separately on different CPU cores to speed it up. This included iterating over all the fields within a gene, and all the cells within each field. Then the cell images from TrypTag and the model for segmenting the specific gene and the mNG channel are loaded. The mNG channel is segmented to extract the paraflagellar rod's individual signals. After that, the phase-contrast mask and the mNG channel mask are overlayed, and the largest overlapping connected mNG component is used as the signal for the flagellum. This was done to better match the flagellum to a cell, when two cells are close to each other. And to avoid picking up noise from the mNG channel mask. Next, Tryptag is used to extract the KN configuration of the cell, as well as its length and anterior and posterior points. The anterior and posterior points are later used to apply the above-defined formula to compute the adjusted cell length. The same is done for the flagellum. We defined the flagellum as the largest

overlapping component. This largest overlapping component is converted to a mask and analysed with the same functionality as the cell mask to extract its length and anterior and posterior points. The corrected flagellum length is calculated in the same way. Lastly, the cell and flagellum length are converted from pixels to micrometres. This is done by multiplying the length in pixels with the conversion factor for micrometres per pixel, which is 6.5/63. The values for the cell length, corrected cell length, flagellum length and corrected flagellum length are then saved in a Python dictionary. The key to this dictionary is the KN configuration. After this process is repeated for each gene, field and cell, the histograms are created with Matplotlib. However, before a histogram for each KN configuration is created, the mean and standard deviations of the cell and flagellum lengths have to be calculated. This resulted in the six plots mentioned in the paragraph before. These plots contain useful information about each KN configuration for cells and flagella. They can be used to compare them to existing measurements. Furthermore, the statistics for the corrected length measurements can be used to determine whether the 3% difference can be explained by modifying the formula for the length calculation.

In summary, this section explains how previous measurements were conducted and provides an explanation for why previous automatic measurements differed from manual measurements. This explanation was used to improve the calculation by introducing a corrective term for measuring cell and flagella length. The new formula was then implemented and applied to measure cell length and flagellum lengths for all KN configurations.

3.4.2 Ellipse Fitting on Kinetoplast and Nuclei Segmentation Masks

As mentioned in earlier sections, a simple KNF count is insufficient to cover the entire cell cycle. This is because the 2nd and 3rd, and 4th and 5th stages can only be differentiated by measuring the growth of the kinetoplast, nucleus, and flagellum. To capture the dimensions of the kinetoplast and nucleus, ellipses were fitted to approximate the size and shape of the organelles. This was partly inspired by Zhao et al. 2017 [Zha+17], where the ellipticity was briefly mentioned. The idea behind that is that the ellipticity of the nucleus and kinetoplast can inform us about the nuclear and kinetoplast subcycle progress and might be a better predictor than simply using the nucleus and kinetoplast areas or their DNA content, which were used in previous studies.

Ellipses are measured using the major and minor axes. The main reason for fitting ellipses to the kinetoplasts and nuclei is to see whether there is a difference in the major axis distributions between the 1K1N, 2K1N and 2K2N configurations. Moreover, it has to be tested whether this approach works at all. Ideally, there should be a bimodal distribution in the plots for the major axis length in 1K1N and 2K1N cells. Within those configurations are the stages that can only be classified by morphological features and not by kinetoplast, nucleus or flagellum count.

To look at this relationship, two new Python scripts were created, which have almost the same functionality. They both iterate over the complete dataset and extract information like the minor and major axis length, the orientation, the intensity mean and max and the area of kinetoplasts and nuclei. The data is then plotted with Matplotlib and additionally returned as a table. The script kinetoplast_nuclei_properties_all_configurations.py looks at all of the data without distinguishing between the different KN configurations. The script kinetoplast_nuclei_properties_individual_configurations.py, however, differentiates between the different KN configurations. I implemented this in the following way. I used the cell_morphology_analysis function

from the TrypTag repository, which internally calls the cell_kn_analysis function. This function usually returns information about the position, DNA content, and count of the kinetoplast and nucleus. For this, it uses the DNA masks and the regionprops_table function from Skimage, which computes various image properties. I modified the variable dna_props_table inside the cell_kn_analysis function which stores the return value of the regionprops_table function. The TrypTag implementation measures intensity mean, intensity max, and the centroid. I added the following arguments to the properties dictionary that is passed to this function: orientation, minor_axis_length and major_axis_length. Then I had to make some minor changes so it would also return this data. Ultimately, the modified function from TrypTag outputs the necessary information, the x and y coordinates, the minor and major axis length and the orientation, to reconstruct an ellipse for kinetoplast and nucleus. Those properties are then captured in a Python dictionary and later plotted with Matplotlib. The only difference between the two programs is how they store and present the information. The kinetoplast_nuclei_properties_individual_configurations.py script additionally extracts the KN configuration from the cell_morphology_analysis function. This information is later used to create different plots for the KN configurations. The resulting plots, one of which is shown in figure 3.18 consist of six subplots displaying the minor and major axis lengths, orientation, intensity mean and maximum, and area for kinetoplasts and nuclei.

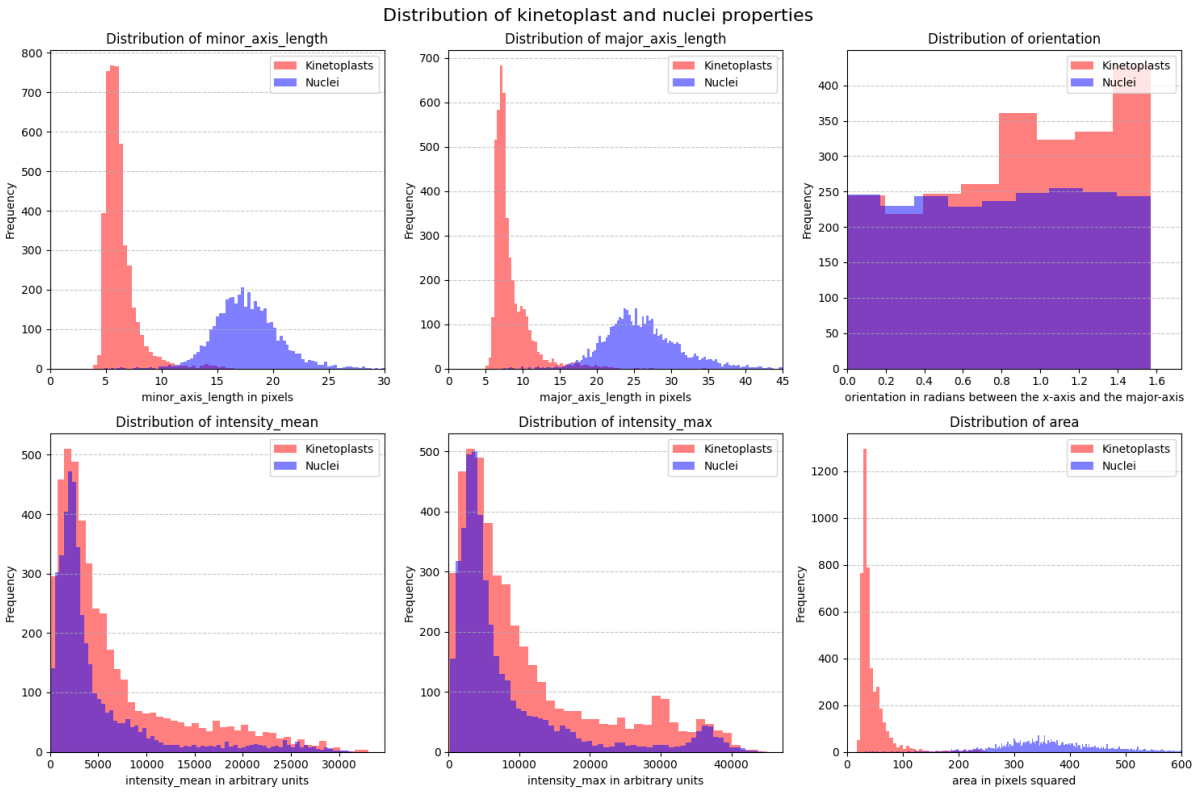

Figure 3.18 – A plot with six subplots, each showing two histograms of measured properties. One for nuclei (in blue) and one for kinetoplasts (in red). From top left to bottom right: minor-axis length, major-axis length, orientation, intensity mean, intensity max and area.

In this subsection, I described how ellipses could potentially be used to capture the subcycle

progress in kinetoplasts and nuclei, which is useful for classifying cell cycle stages. To test and verify this idea, scripts were written that produced various histograms that can later be analysed to search for bimodal distributions in the nucleus and flagellum major axis length.

3.4.3 Multivariate Scatterplots

To further analyse the cell cycle stage, commonly known good priors such as kinetoplast, nucleus and flagellum count, position and relative position and size, as well as their DNA content, were analysed using multivariate scatterplots. The aim was to identify clusters within the data. Similar to the plots from the previous section (3.4.2), where bimodal distributions or distribution shifts could be identified due to morphological changes in the subcycles. These plots could uncover correlations between these variables and enhance the understanding of the cell cycle stages. Furthermore, the features shown in the scatter matrix were standardised and fed into different dimensionality reduction algorithms like PCA, t-SNE and UMAP. This way, all of the features can be displayed in a two-dimensional scatterplot.

As a result of the limited time due to having to construct the flagella dataset, not all features discussed were compared in the scatterplots. Especially features like geometric, intensity, or texture features mentioned by Zhao et al. 2017 [Zha+17] were not tested. Ideally, the multivariate scatterplot would display all variables and enable selection by their KNF Configuration. This would allow the interactive activation of only specific KNF configurations, such as 1K1N2F or 2K1N2F, which are critical for determining the exact cell cycle stage. Each individual plot could then be analysed to detect two groups arising from different stages. However, this was impossible since the KNF configuration was not identifiable from the generated mNG masks with the methods used as described in 3.3.3. The idea was only implemented for the KN configurations. Of particular interest was 1K1N and 2K1N. The research approach applied to this problem is a quantitative inductive approach. This is because it builds on top of numerical data with the aim of identifying clusters or differences in distributions or correlations. Next, based on those observations, a hypothesis could be generated explaining a relationship between two variables.

Plotly, a widely used Python library for interactive visualisation, was used instead of Matplotlib to implement my idea. Because Plotly makes creating interactive charts a lot easier than Matplotlib. To accomplish this, I leveraged pre-existing code from the TrypTag repository and reused code from my previous experiments that calculated the cell and flagellum lengths. Again, I iterated over all the cells and computed the morphological measurements, specifically the area of the cell, the length of the cell, the length of the flagellum, the area of the kinetoplast and nucleus, the DNA content of the kinetoplast and nucleus, the normalised position of the kinetoplast and nucleus along the medial axis and the major axis length of both kinetoplast and nucleus. Each of these variables is plotted against each other, and the individual data points can be selected based on the KN configuration, which was again determined with the TrypTag repository. One might notice that 2K1N and 2K2N have more than one kinetoplast or nucleus. To simplify the data processing, I decided that only the first nucleus or first kinetoplast would be taken. The first one is the one that is first returned by the TrypTag cell_morphology_analysis function, which internally sorts the kinetoplasts and nuclei based on their midline index. Specifically, it returns the kinetoplast or nucleus that is closest to the anterior end of the cell. This decision was based on the fact that this experiment focuses on the 1K1N and 2K1N configurations, as those are the difficult stages where either the kinetoplast or the nucleus grows. However, it might also be interesting to

look at their relative positions for future research. This would require a different data processing approach. Additionally, the program prints out the Pearson correlation coefficients for the unique pairs of features. This is done to capture linear relationships within the data. The coefficients are computed using the Python library Scipy for each pair of features for 1K1N, 2K1N and 2K2N cells.

Different dimensionality reduction techniques, such as PCA, t-SNE, and UMAP, were used to show the data in two-dimensional scatterplots to visualise the data further. For this, all of the features were taken and standardised. This is necessary, as PCA looks at the variance, which is influenced by the scale of the features. Then the three different models are fitted with the standardised data. After that, an interactive plot is created for each of them. Additionally the code returns the percentage contribution of each feature to both principal components. This was done only for the PCA components since they are directly interpretable, while the t-SNE and UMAP embeddings are not.

The code can be found in the file multivariate_scatterplot.py. The generated scatterplot matrices, can be seen in figures 4.4, A.20, A.21, A.22 and A.23. The plots of the dimensionality reduction techniques are figures A.24, A.25 and A.26. The table showing the percentage contribution of each feature for both principal components is also attached in the appendix in tables A.6 and A.7.

Before running the program, the expectation was that some of the features would be correlated. For example, the cell length and cell area should be positively correlated. Similarly, I also expected a positive correlation between the DNA content and the area of the nucleus and kinetoplast. Furthermore, I expected differences between the KN configurations for the kinetoplast and nucleus area. However, there are many more expected correlations, as there are 11 features, which result in 121 plots, from which 66 are unique. Moreover, the hope was that the different KN configurations would show clusters. In the individual KN configurations but also in between the stages. Especially clusters within one KN configuration would be interesting for the classification process. The scatterplots for the dimensionality reduction techniques might reveal higher-dimensional relationships between the features that are not visible on the two-dimensional scatterplots.

Future work should focus on incorporating the additional measurements mentioned in the introduction and expanding the dataset to refine the analysis further. The reason a larger dataset would be useful is that the majority of the data is 1K1N cells and the sample sizes decrease rapidly as the cell cycle progresses. This creates a large class imbalance. Furthermore, the KN configurations need to be improved, as some of the cells are misclassified, as mentioned in the initial thoughts section 3.2.

3.4.4 Principal Component Analysis of ResNet-50 feature vectors

A further idea to extract information about the cell cycle stages from cell images was to use a pre-trained ResNet-50 network, a commonly used convolutional neural network. This network is trained on common objects and, through that, has some understanding of colors, sizes, textures and image composition. The idea is to use this network and extract an intermediate representation of the image, use principal component analysis on this representation and search for clusters. This method has already been used in a previous paper in a similar setting [Eul+17], and how it works was described in the literature review in section 2.2.2. The approach to this idea is also a quantitative inductive approach.

Once again, a Python script that implements this idea and uses Plotly was written. For perform-

ing neural network inference and feature vector extraction, I relied on the Keras Python library, which includes the model and the Imagenet weights. I used principal component analysis from the Sklearn library to reduce the dimensionality. However, before passing the images through ResNet-50, I had to crop and reshape all cell images from the dataset into 224 by 224 pixels and convert them to 8-bit images. This is necessary as the ResNet-50 only accepts this specific image size. Additionally, because the TrypTag phase contrast images only have one color channel, this channel is repeated three times as a replacement for the RGB channels used by ResNet-50. To improve the accuracy of feature extraction, I utilised TrypTag's pre-existing cell masks to eliminate any background variations by making the background a uniform colour. The reasoning behind that is that the network will solely focus on the cell's shape, colour, size and other features rather than any other cells or variations in the background colour of the image. After reshaping the images, they are fed to the network and the intermediate or latent representation of the image is extracted from the penultimate layer of the network. Next, principal component analysis is applied to extract two principal components. The principal components are then visualised in a scatterplot. The scatterplot is made interactive. When hovering over a specific point, it shows the underlying image from which it represents. This was done in case the scatterplot would show multiple clusters. If clusters were present, one could hover over the different points of different clusters and come up with a theory for why the clusters exist. If clusters are present, it would be interesting to see if they relate to the cell cycle stages or KN configurations or are just due to visual differences of the cells, like different cell lengths or curled up and straight cells. The generated plots can be seen in the figures 4.5, A.27, A.28, A.30, A.29 and A.31.

The use of a pre-trained ResNet-50 network for feature extraction from cell images is a promising approach as described in [Eul+17]. This experiment provides insights into using neural networks for feature extraction from cell images. Similar approaches of using neural networks that have learned meaningful representation of images could greatly reduce the time spent on manually investigating the images. A downside of this approach is that it is uncertain whether it works. This is because it uses Imagenet weights, which is problematic, as Imagenet is a dataset of everyday objects instead of microscopy images. However, the application of neural networks to Trypanosome research has not been deeply explored but could prove fruitful. Especially with more modern model architectures like Vision Transformers. Nonetheless, this approach is a first step towards applying more neural networks in this research field. The approach could be further improved by making use of the mNG and DNA channel instead of only using the phase contrast channel.

3.5 Classification with Binary Decision Trees

This section covers how the newly introduced features can be used to classify procyclic Trypanosoma brucei from microscopy images. For this, the following steps were necessary: The creation of the flagella dataset, improved MAT length measurements, a measurement of the subcycle progress with ellipses, and different feature extraction techniques. With those improvements to previous techniques, the classification process is greatly simplified and can be solved by a simple binary decision tree. This allows for efficient computation and interpretable results. In this section, a quantitative deductive reasoning approach is used because the cell cycle stages are known, and a method has to be developed based on that. After constructing a classification scheme, it

needs to be evaluated on a labelled dataset, which is a sign of a quantitative approach.

First, let me provide an explanation for why a binary decision tree was planned to be used. Binary decision trees are very simple models that can be used for classification. They are based on binary trees, a specific type of graph. This type of graph has a root node and two child nodes, which are called left and right subtrees and are trees themselves. So it can be recursively defined. Binary trees can be visualised in layers. These layers define the depth of the tree, where the root node has a depth of 0. A binary decision tree makes use of this structure in the following way. The input for the data is the root node. At each node, a binary decision is made, which is a simple yes-no question. An ideal binary decision tree has yes-no questions that divide the data into two equally sized groups. This would result in the greatest information gain of 1 bit. In this case, the data is divided into six distinct cell cycle stages. Therefore, the minimum number of binary questions that are needed is three. Hence, two is the minimum depth of the binary tree, as at each node, a binary decision is made. Not only are decision trees very easy to understand and computationally efficient, but they are also ideal for the problem at hand due to their rigidity. Since no labelled data is given for this classification problem, a deductive approach has to be taken. This is also the reason why more complex models can not be used. The cell cycle stages in Trypanosoma brucei are well known. As identified in the literature review, the KNF configurations provide a good starting point for this deductive chain of reasoning. This classification can uniquely identify the first and last cell cycle stages, which are denoted by 1K1N1F and 2K2N2F. However, further information is needed to decide between stages 2 and 3, as well as stages 4 and 5. For that, tracking of the subcycle progress and feature selection techniques were introduced. With that, a three-stage binary decision tree can be constructed. However, it has to be noted that this is not the only approach to quantifying the cell cycle stages of T. brucei. Since biological processes are observed that usually happen in a continuous fashion. Such as the growth of the kinetoplast and nucleus before division or the growth of the new flagellum. It might be more useful to use regression instead of classification by tracking the three subcycles. Regression is used for predicting continuous variables, in contrast to classification where classes are predicted. For example, instead of predicting that a cell is in stage two or three, which are very similar and hard to tell apart. One could predict that the cell is at stage 2,7 or the end of stage 2. Between these two stages, the basal body rotates, and the flagellar pocket divides. As far as I know, those are also two continuous processes and do not happen in an instant. By predicting floating point numbers instead of integers, the cell cycle stages are modelled in a more granular and close-to-life way. However, this has a completely new set of challenges.

One such binary decision tree is visualised in figure 3.19. The visualisation was created using the free online tool smartdraw. First, the tree checks if the cell has one or two kinetoplasts. This initial check divides the cells into two groups: stages 1 to 3 with one kinetoplast and stages 4 to 6 with two kinetoplasts. The tree next asks about the number of flagella for cells with one kinetoplast. If there's only one flagellum, the cell is classified as stage 1. If there are more, the tree needs to decide between stages 2 and 3. To do this, it looks at the length of the kinetoplast's major axis and the length of the new flagellum. The cell is assigned to the appropriate stage if the length is above a certain value. For cells with two kinetoplasts, the tree asks about the number of nuclei. If the count matches that of stage 6, the cell is classified accordingly. Otherwise, the tree distinguishes between stages 4 and 5 by looking at the position of the flagella and potentially the kinetoplasts along the medial axis and the size of the nucleus. These features change as the basal body moves

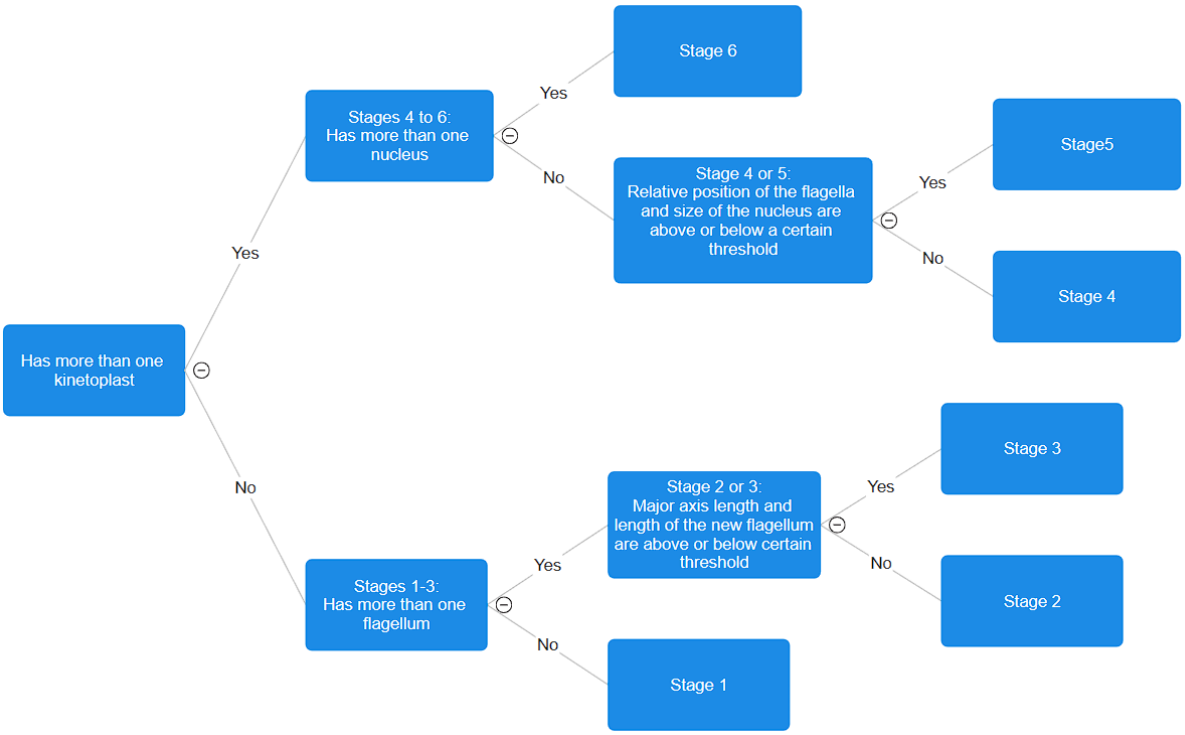

Figure 3.19 – A possible binary decision tree for deciding between all cell cycle stages.

apart and the nucleus grows, helping to place the cell in either stage 4 or 5. The specific features mentioned are not necessary to differentiate between stages 2 and 3, and 4 and 5. However, they are the most obvious features, judging by the literature's description of cell cycles. More detailed analyses of the scatterplot matrix and more refined feature selection approaches could discover other features that could work. To determine the thresholds for the decisions between 2 and 3 and 4 and 5, one needs to find good priors that are ideally linearly separable into two clusters. After that, an algorithm like K-means clustering can be used to assign each point to one of the two clusters.

After building such a classifier, it would need to be evaluated. The evaluation part is a bit tricky for cell cycle classification, as even human experts can struggle to identify the correct cell cycle stages. In previous stages, specific markers were used to determine the cell cycle stages. Either this approach could be used to evaluate the model, human experts could verify classifications or unsupervised methods can be employed. In subsection 3.3.3, the use of the Silhouette Score was introduced; this metric or similar metrics could be used to tune the exact decision boundaries so that the Silhouette Score is maximised. This would result in distinct clusters or classes that are as different as possible from each other. And where data points of the same class are as similar to each other as possible. Usually, a train, test, and validation set is required to evaluate labelled datasets to avoid overfitting and to evaluate the model's performance better. However, no training set is required with this deductive binary decision tree approach. One could apply the model to a set of images and use human annotators to verify that the model classified the cell cycle stage correctly.

This section outlines constructing a binary decision tree for classifying different cell stages

based on their KNF configurations and other features. Due to this requirement for reliable KNF configuration classification, which was ruled out while constructing the flagella dataset and evaluating the masks, the binary decision tree is incomplete and can not be evaluated. This binary decision tree divides cells into two groups: stages 1 to 3 with one kinetoplast and stages 4 to 6 with two kinetoplasts. To differentiate within these groups, the classifier considers features such as the number of flagella, kinetoplast's major axis length, nuclei count, flagella position along the medial axis, and nucleus size. However, the exact features used for distinguishing between stages 2 and 3, and stages 4 and 5, are not strictly defined, suggesting alternative features might offer improved classification. To finalise these features and their respective thresholds for differentiation, a comprehensive analysis and refined feature selection are necessary. Importantly, the classifier needs to be validated and its performance assessed. However, this proves to be challenging without reliably labelled data.

3.6 Summary and Limitations

This section summarises the key methods used to approach the goal of building a classifier for procyclic T. brucei cell cycle stages using microscopy images. Additionally, I want to highlight problems that were encountered and the limitations of my methodology.

In general, a quantitative approach with mixed reasoning approaches was applied. Python was chosen as the primary tool for the implementation and testing of all these quantitative methods that were developed and applied in this thesis. This is because the TrypTag repository, which provides crucial functionality for analysing T. brucei cells, is also written in Python. This way, previous work could be leveraged to accelerate the research process. Additionally, Python was chosen as it provides many useful libraries for image processing and machine learning techniques, such as Sklearn, OpenCV, Scipy, Numpy, Keras and TensorFlow. It also provides useful plotting libraries like Plotly and Matplotlib, which greatly simplify the exploration process detailed in section 3.2, but also the presentation of results. Key findings from my initial exploration described in this section were that the most straightforward way to classify cells is to build on top of the existing KN and KNF configurations. However, the TrypTag Python package only provides KN configurations that are sometimes incorrect. Therefore the information on the flagella had to be added. Additionally, it was identified that the KNF configurations are not enough to classify all cell cycle stages. For this reason, more features are needed that describe the nuclear, cytoskeletal and kinetoplast subcycle progress. First, the flagella dataset had to be built. For this, genes that locate the paraflagellar rod were selected. Second, segmentation masks had to be created, which was done using K-Means clustering. The segmentation masks were additionally evaluated using the Elbow Method, Silhouette Scores and manual inspection. Due to randomness involved in the process, it had to be proven that it did not affect the segmentation results, which was done thereafter. Next, a method for improving the MAT-based length measurement was introduced by adding a corrective term. After that, a new feature, the major axis length of kinetoplasts and nuclei, was developed to gauge their subcycle process using elliptical fitting. Following that, multivariate scatterplots and scatterplots of the principal components of the feature vector of a neural network were implemented to uncover clusters and new features in the data, that could be used for classifying the cell cycle progress. Finally, the concept of utilising binary decision trees for solving this problem was proposed. All the mentioned methods were developed and im-

plemented with the design goals of efficiency, explainability, reliability and reproducibility from section 1.3 in mind. For example, for efficiency, Ray was used to distribute the workload to all CPU cores, and OpenCV was largely used instead of Scikit-image because of faster processing times. Additionally, for explainability, simple models like binary decision trees and interactive visualisations were used. Lastly, for reliability and reproducibility, randomness was largely eliminated, and all of the code is provided on my personal GitHub repository.

Although adhering to those design principles helps develop new techniques, the methods have limitations. First, the paraflagellar rod had to be chosen instead of the flagellar cytoplasm. As a result, the segmentation masks of the mNG channel might not represent the entire area of the flagella. Next, the segmentation masks revealed that it is difficult to extract the second flagellum, especially in the early stages. Both of these problems could be solved by finding different genes. For example, a gene that is expressed in the basal body could be useful or one that covers the whole area of the flagellum. Furthermore, the gene extraction process could be improved by using human experts to judge whether a gene is a good fit for a domain-specific purpose. Akin to previous research, the segmentation masks do contain overlapping cells and were only addressed by filtering the objects by area. It could be explored whether further image pre-processing improves results for segmentation. Especially for the mNG masks, some flagella masks were spotty and broken into multiple individual masks. For the segmentation process itself, a larger sample size could be used to determine the optimal number of clusters. The methods for improving the MAT and the elliptical fitting are only limited by the segmentation masks, by the sample sizes for later stages and potentially by using a very simple approach of taking the largest overlapping component to match the masks to a cell. The multivariate scatterplots are limited because not all features developed in this thesis could be displayed, and again by the low sample sizes for later stages. However, introducing more variables would potentially blow up the amount of individual plots that have to be analysed. Additionally, all methodologies that use the KN configurations are limited by the accuracy of the implementation of the TrypTag repository. Moreover, the multivariate scatterplots do not show all the organelle metrics for 2K1N and 2K2N cells, but only the first organelle of each type is returned by TrypTag. Lastly, the PCA of the ResNet-50 feature vector is limited by using Imagenet weights. Instead, weights that were trained on a cell dataset should be used. It might also be useful to see the different KN configurations highlighted in this plot. The classification with binary decision trees could not implemented due to the problems mentioned for the gene selection and segmentation. Therefore, there is no experiment that could have any methodological limitations.

This was only one of the problems I encountered. The first was that I had no access to labelled data, which complicated the process and required a more explorative approach. The next problem was that no flagella dataset was provided to me. I solved this by creating my own flagella dataset. However, I had to invest a lot of time in building it, which resulted in some of the other limitations of my methods. If the flagella dataset had been provided, it would have probably had better segmentation masks, which might have allowed the classification of the KNF configurations. But this was not the case. After that, I had to circumnavigate the fact that the classification had failed. This meant the classification model's typical evaluation and tuning process could no longer be done. In order to provide scientific value, the decision was made to explain why it failed and slightly shift the focus of this thesis from classifying the cell cycle stages towards improving existing features, creating new ones, and searching and selecting them. This way, the thesis provides novel ideas and insights on approaching future studies.

4 Results and Discussion

This final chapter presents the results of the experiments, analyses the results, and discusses the impact of those results. Finally, the thesis will be concluded, where key findings and contributions are mentioned. Additionally, an outlook for future work will be given. The results are presented in the same order as in the methodology section.

4.1 Research Results

This section presents the results obtained using the described methods in chapter 3. Since each method solves a particular problem and improves upon an existing technique, I will mention the problem again and present the results, which include a wide variety of plots and statistics. The statistics will often be presented in the format: mean ± standard deviation unit. Those problems and improvements are also part of the earlier section 1.2 defined research objectives. They will be covered in the order of which research objective they address. As a reminder, the research objectives were the following: Firstly, creating an understanding of the various cell cycle stages of T. brucei and prior research in this area. Secondly, creating accurate segmentation masks for the cells, kinetoplasts, nuclei, and flagella. Thirdly, developing novel methods to capture the cell and organelles morphology. Fourthly, using unsupervised learning methods and neural networks to identify distinctive features that represent different stages of the cell cycle. Fifthly, developing a classifier using the identified features. Finally, evaluating the performance of the classifier.

The first objective was addressed in chapter 2. The following important facts were gathered from the literature review. There are six cell cycle stages. But usually, only the KN and KNF configurations are used in studies. The expected KN configurations for T. brucei are 1K1N, 2K1N and 2K2N and the expected KNF configurations are 1K1N1F, 1K1N2F, 2K1N2F and 2K2N2F. There are difficult stages, namely, the 2nd and 3rd stages within the 1K1N2F configuration and the 4th and 5th stages with a 2K1N2F configuration. Furthermore, it is known that the cell cycle stages can be identified from morphology. For T. brucei, the following cell length measurements were observed in one of the studies: 16.3 ± 3.1 µm for 1K1N, 19.6 ± 3.0 µm for 2K1N and 21.0 ± 3.3 µm for 2K2N cells. The lengths were measured using the medial axis transform. The same study additionally made 50 manual measurements of 1K1N cells, which resulted in an average length of 16.8 ± 3.0 µm. Furthermore, they did not explain the difference in cell lengths. From this and other studies, it is also known that the cell length, area, kDNA and nDNA content and the flagellum position and size are correlated with the cell cycle stages.

In section 3.3.3, the second research question is addressed, which is whether TrypTag provides all necessary masks and, if not, whether it is possible to use a simple method like K-Means clustering to create good masks that can tell two flagella within one cell apart. This research question was answered within the section because the segmentation was seen as a prerequisite for further

experiments. During the analysis, it was found that masks were available only for cells and DNA but not for the mNG Channel. The analysis of the created segmentation masks indicated that for the phase contrast channel, four clusters worked best, while two clusters were optimal for segmentation in the case of DNA and mNG channels. It was also discovered that the segmentation masks for the cells were not quite as good as the ones provided by TrypTag, mainly because the K-Means segmentation picked up finer details, such as an organelle within the cell, which created holes in the masks. Additionally, the segmentation approach had downsides similar to those of the previous techniques. Which are clustered or overlapping and curled up cells. It was also found, that the created masks for the mNG channel were patchy, meaning that the signals of the paraflagellar rods were separated into multiple parts. Furthermore, cells with multiple flagella were observed. However, the segmentation results and potentially the signals themselves were not enough to differentiate the first three stages by the existence of a second flagellum. Only in later stages were the flagella masks good enough to pick up two distinct flagella signals. The segmentation of the DNA channel with K-Means clustering produced the best results between the channels. In the masks, kinetoplasts and nuclei were clearly segmented and easily differentiable through their size. The resulting unfiltered images can be seen in figures A.4, A.5 and A.6.

A further result of the flagella segmentation and the elliptical fitting were the following plots: A.1, A.2 and A.3. They visualise the ellipses, the flagellum segmentation and overlay the midline on top of an existing visualisation from the TrypTag repository, which highlights the anterior and posterior points, as well as the midline index of the kinetoplasts and the nuclei. Additionally, the background was removed, so only the cell masks that were used to make the decision to which KN configuration the cell belongs are visible. Previously, the same plot type was used to visualise a misclassified cell, which can be seen in figure 3.2.

The third research objective was to find new ways of capturing the cell morphology or enhancing previously used techniques. This covers multiple smaller research questions. The first question was whether a corrective term in the formula could explain the 3% difference in the MAT-based cell length measurement and whether the MAT-based length measurement could be applied to the flagellum. This was covered in subsection 3.4.1. The second question, covered in subsection 3.4.2 was whether ellipses could be used to measure the subcycle progress of the kinetoplast and nucleus. Lastly, the third question was whether multivariate scatterplots can reveal new relationships between selected variables and whether a pre-trained ResNet-50 captures the structure, colour or size of cells. Those two questions were addressed in subsections 3.4.3 and 3.4.4.

In subsection 3.4.1, the first question of these smaller questions was addressed. The approach of applying the MAT-based cell length measurement to the flagellum worked. The methodology generated the following statistics: For the 1K1N configuration without the correction, a mean cell length of 25.39 ± 4.23 µm was measured with a sample size of 1615. The correction lifted this value by 2.72% to 26.08 ± 4.21 µm. For 2K1N cells, a difference of 2.57% was observed, increasing the measured mean cell length from 28.45 ± 4.84 µm to 29.18 ± 4.84 µm with a sample size of 70. For 2K2N cells, an increase of 2.06% was measured, increasing cell length from 31.55 ± 6.88 µm to 32.20 ± 6.83 µm with a sample size of 19.

The same analysis was conducted for the flagellum with the previously generated flagellum masks. The sample sizes for each group were the same. This resulted in a mean uncorrected flagella length of 19.48 ± 4.91 µm for 1K1N cells, which increased by 1.44% to 19.76 ± 4.93 µm with

the corrective term. The 2K1N configuration saw an increase of 1.38% from 21.06 ± 5.66 µm to 21.35 ± 5.68 µm, and the 2K2N configuration saw an increase in mean flagella length of 1.64% from 20.78 ± 5.34 µm to 21.12 ± 5.30 µm. The last observation for the flagella histograms, as seen in the figures in the appendix A.16, A.17 and A.17, is that they look left-skewed. The histograms for the cell lengths did not show similar skews.

The second question within the larger third research objective was covered in subsection 3.4.2. It asks whether ellipses can be used to measure the subcycle process of the kinetoplast and nucleus. To answer this question, a hypothesis was formulated that predicts bimodal distributions during 1K1N for the major axis length of the kinetoplasts and for nuclei during 2K1N. The ellipses were fitted to each kinetoplast and nuclei to capture their dimension. A total of 4521 kinetoplast data points and 4271 Nuclei data points were analysed and plotted with one plot for each KN configuration. This resulted in the following figures: 4.1, 4.2, A.19 and 3.18. Additionally, the data is also available in tables A.4 and A.5. The plots contain additional information about the minor axis length, orientation, DNA intensity mean, DNA intensity maximum and area of kinetoplasts and nuclei. The minor and major axis lengths are given in pixels, as this is how the analysis function returns them. Additionally, converting them into micrometres would result in very small numbers, but can be done by multiplying the result in pixels by 6.5/63.

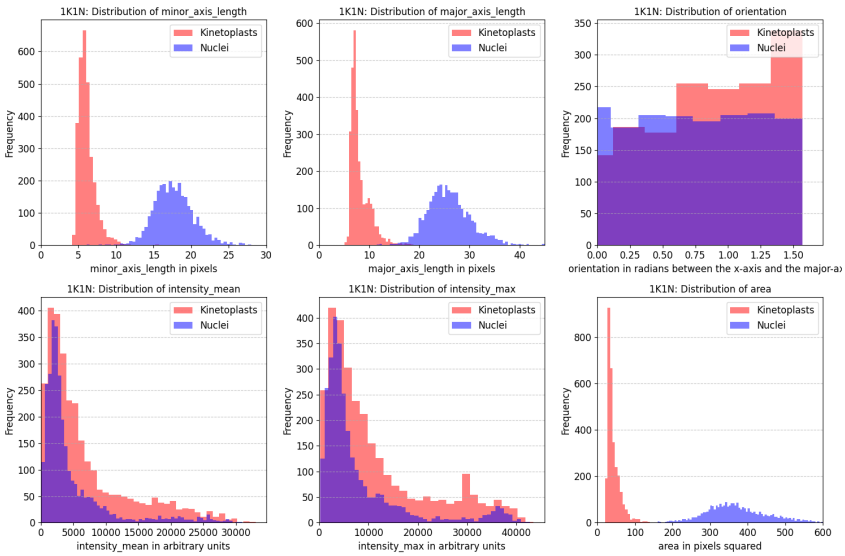

Figure 4.1 – A plot with six subplots comparing different measurements of nuclei and kinetoplasts in 1K1N cells.

In figure 4.1, the results for 1K1N cells are displayed. The mean major axis length of the nuclei is 26.12 ± 5.00 pixels, while that of the kinetoplasts is 8.17 ± 2.11 pixels. The distribution of the kinetoplasts' major axis length only hints at a bimodal distribution with two peaks at around 7 and 10 pixels, where the first peak is much larger than the second one. Another interesting observation is that the kinetoplast distributions look log-normal distributed except for the orientation. However, the nucleus distributions for the minor and major axis length and the area look normally distributed.

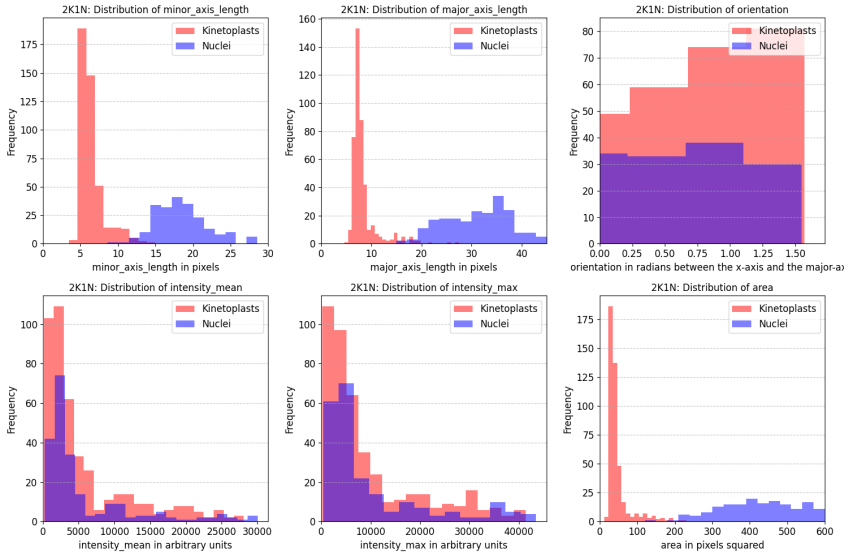

Figure 4.2 – A plot with six subplots comparing different measurements of nuclei and kinetoplasts in 2K1N cells.

In figure 4.2, the same information as in the previous image is displayed, but now for the 2K1N configuration. This configuration corresponds to the cell division processes during ambiguous stages 4 and 5. In this case, a bimodal distribution in the major axis length of the nuclei was expected. The major axis length of nuclei in the 2K1N configuration is 32.34 ± 8.81 pixels. In contrast, the major axis length of kinetoplasts is 8.37 ± 2.81 pixels. The kinetoplast distribution seems to have a thicker right tail than in the 1K1N, which skews the mean higher than in the 1K1N configuration. There still is a smaller peak right at 10 pixels. However, it is harder to tell with certainty because the sample size went from 2977 to only 438. The mean nuclei major axis length grew from 26.12 pixels to 32.34 pixels. Similarly, the standard deviation went from 5 pixels to 8.81 pixels. Overall the distribution is wider, and it looks like there could be a peak at around 26 pixels and another at around 36 pixels. Hinting at a bimodal distribution in the major axis length of nuclei for 2K1N cells.

For completeness, the resulting plot for 2K2N cells is also shown in figure A.19. However, for this stage, no predictions were made, as it has no value for this research question. Furthermore, the sample size is now only 30. There is not much left that could be called a histogram. The major axis length of nuclei in the 2K2N configuration is 20.91 ± 4.51 pixels, and for kinetoplasts, 7.42 ± 1.02 pixels. For all three configurations, a table for kinetoplasts A.2 and nuclei A.3 is attached with the sample sizes and the mean and standard deviation of each measurement. Notice that the number of samples for those tables does not add up to the number of samples for all configurations. This is because the program does not recognise the configuration for each cell and sometimes outputs wrong configurations with 0, 3 or 4 kinetoplasts or nuclei. But those are mostly errors; if they are not, they are irrelevant because I am interested in the non-mutated cells. Therefore they were discarded.

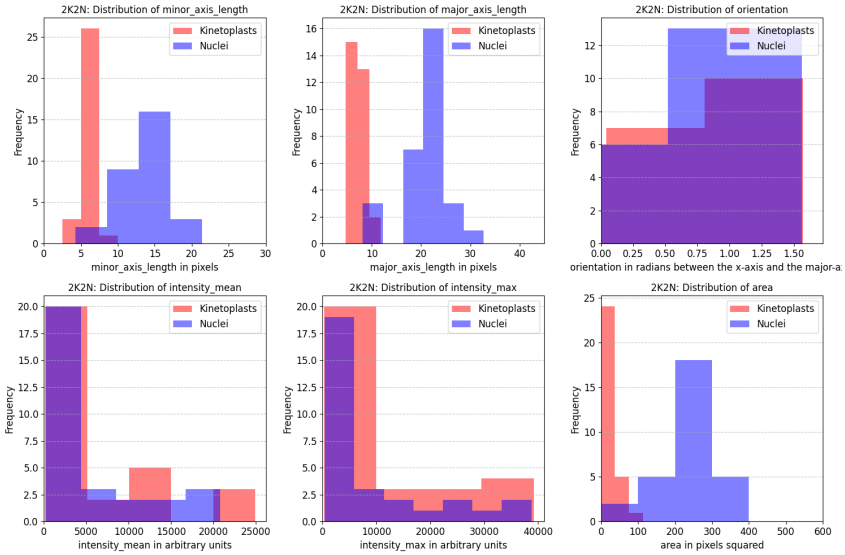

Figure 4.3 – A plot with six subplots comparing different measurements of nuclei and kinetoplasts in 2K2N cells.

The fourth research question was if it is possible to uncover new features, clusters and correlations that could potentially be good priors for deciding between difficult cell cycle stages using unsupervised learning and neural networks. This is split into two parts: first, using Multivariate Scatterplots to find unknown correlations and clusters, and second, using principal component analysis and ResNet-50 to uncover patterns within the image data. The methodology for the first subquestion was explained in subsection 3.4.3. As for the other questions, several plots were created that need to be described and analysed.

The scatter matrix shown in figure 4.4 resulted from the analysis. There are four additional plots. One for each configuration and a further one with 1K1N and 2K1N visible. Namely, figures A.20, A.21, A.22 and A.23. Only the correct, non-mutated configurations are visible in each figure, namely 1K1N in blue, 2K1N in red, and 2K2N in green. This research question uses an inductive approach as we want to uncover so far unknown correlations. Although some predictions were made prior to what we might see on these plots, they are irrelevant to the research objective. Analysing each individual chart for each configuration is difficult because of the size of the scatter matrix. There are 81 plots in total, 45 of which are unique. The diagonal splits the matrix into two halves that contain the same information but with swapped axes. Analysing the important stages, namely 1K1N and 2K1N, would require analysing 90 plots. The analysis was thought to be simplified by the possible existence of clusters, which could be quickly spotted by glancing over all the subplots. Although a complete analysis of every subplot is not feasible, there might be some interesting observations.

The first observation was the large class imbalance, as there are many more blue dots, which represent cell measurements from 1K1N cells, than there are red or green dots, which represent 2K1N and 2K2N cells. Moreover, in all plots, the 1K1N measurements seem to be spread out far more than the 2K1N and 2K2N measurements. There seems to be some localisation of the 2K1N and 2K2N KN configurations. However, there are no clear clusters visible on any of the subplots. In a plot with clear clustering, there should be three groups for the KN configurations that only

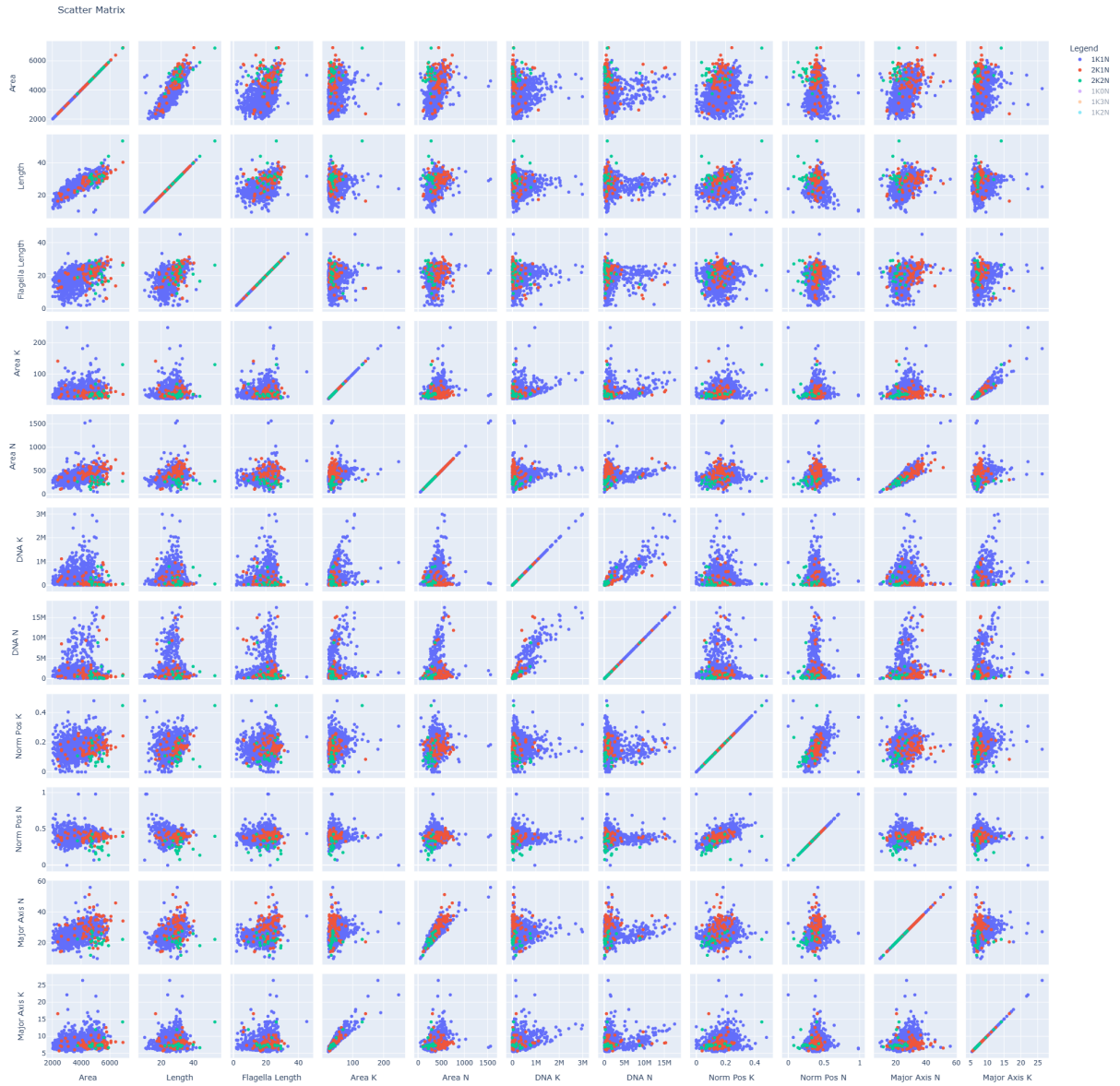

Figure 4.4 – A 2-dimensional Scatter Matrix comparing different properties of Trypanosoma brucei colored by their KN-configuration. Properties from left to right or top to bottom: Area of the cell, length of the cell, length of the flagella, area of the kinetoplast, area of the nucleus, DNA content of the kinetoplast, DNA content of the nucleus, a normalised position of the kinetoplast within the cell, a normalised position of the nucleus within the cell, the major axis length of the nucleus and the major axis length of the kinetoplast.

minimally overlap. However, this is not the case in those plots. Further observations that stood out: The clearest correlations are between cell area and corrected cell length and between the major axis lengths and the areas of the kinetoplasts and nuclei. This is visualised by the data following an almost straight line, similar to the diagonal, where each feature is compared to itself.

The final sub-question is whether a pre-trained ResNet-50 captures the structure, colour or size of cells within its feature vector. It was visualised with a two-dimensional scatterplot using

principal component analysis. The methodology for this was explained in subsection 3.4.4. It resulted in multiple figures, including figure 4.5, which shows the two principal components that are derived from the ResNet-50 feature vector. The same inductive approach was taken for this question.

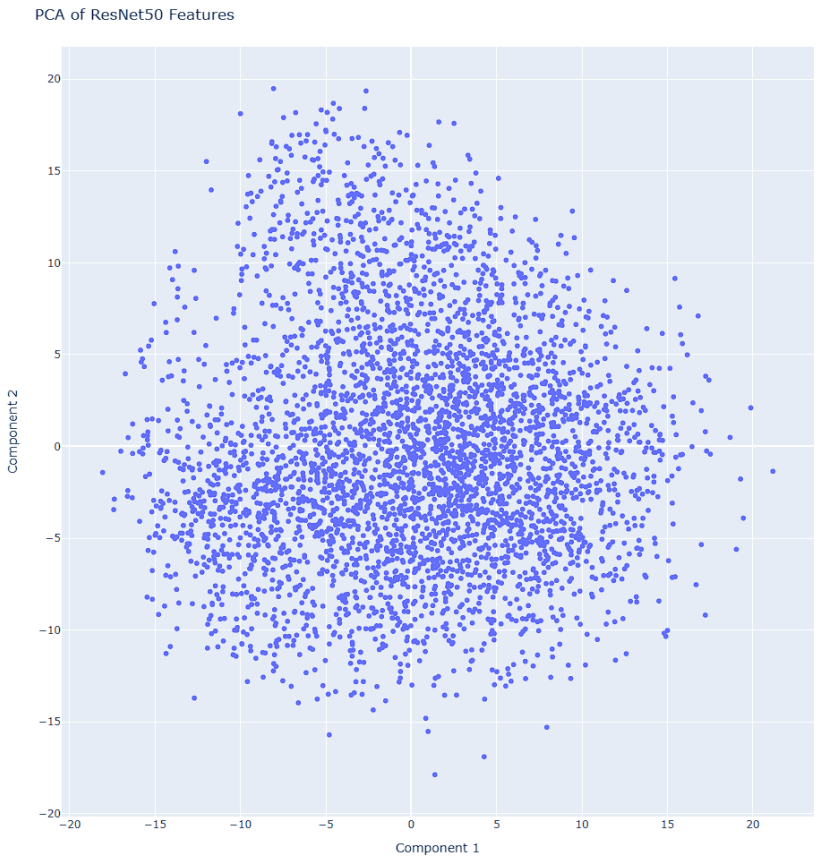

Figure 4.5 – A 2-dimensional Scatter Plot displaying the two most important PCA components of the ResNet-50 Feature Vector. Each individual point represents an image in the dataset.

Similar to the results of the multivariate scatterplots, no clusters or outliers are visible in the visualised principal components. The overall shape of the resulting scatterplot is circular. Additionally, the point density seems lower in the top-left quadrant than in other quadrants, with the overall point density being highest around the origin. No further structural features could be observed that would indicate different visual features in the images. When hovering over the data points, the underlying images are shown. I looked at many images with the following approach. First, I checked the images in the middle, then at the top, right, bottom and left. I made a circular clockwise motion around the origin to see whether any particular direction indicates something about the images. To illustrate the differences, I added one example for each direction to the appendix. (A.27, A.28, A.29, A.30, A.31). While doing this, I found that cells in the top-left corner are comparatively small and curled up. In general, there was a trend that cells in the top half with a positive component two were curlier, whereas cells with a negative principal component two were longer and straighter. I could not observe any other visual differences in the images.

The fifth and sixth research objectives were to build a binary decision tree classifier with the

newly found features from the previous experiments and to evaluate it. Such a classifier was designed, seen in 3.19, but could not be implemented for several reasons that will be further discussed in the next section. Therefore there are no statistics about the evaluation metrics of this classifier that could be presented.

4.2 Analysis of Results

In the previous section, results were presented without any interpretation. However, it is important to the scientific process to analyse and interpret the results. This is done to understand the underlying processes and to prove or disprove the initial hypotheses. If the initial hypothesis is rejected, new hypotheses can be formed with the newly gained information. The analyses will be presented in the same order as before.

The first research objective, the literature review to understand the cell cycle of T. brucei and list known priors for the cell cycle stages, does not require further analysis, as it does not produce any new interpretable results. However, findings made by Wheeler et al. (2012) about cell lengths will be compared to findings from this thesis. Additionally, they proposed that their technique could be applied to the flagellum, which was also tested in this thesis. Both results will be analysed under the MAT improvement.

As previously explained, the second research objective analysis was already covered in section 3.3.3. But to briefly summarise, K-Means clustering was used to create segmentation masks. This was necessary to capture the morphology of the flagellum. In response to the research question, it can be said that the cell masks were not as good as expected, and therefore, the original TrypTag masks should be used instead for future research. The DNA masks had a very good segmentation quality and were comparable to the masks provided by TrypTag, but the additional implementation overhead of this method makes them unnecessary. If, for some reason, TrypTag's masks can not be used in future studies, applying K-means with 2 clusters and filtering outliers by area could be a good and efficient alternative for segmenting the DNA channel. The flagella masks, the main reason for this segmentation, were good enough to differentiate the two flagella in stages 3 to 6 but were not sufficient for earlier stages. Furthermore, It was shown that the masks could be used to measure flagellum lengths in the early stages. For the later stages an improved methodology is needed because the two flagella are segmented as one connected component. For this, methods described in the literature review for C. elegans could be useful. It is unlikely that improvements in segmentation masks with genes localising the paraflagellar rod can be used to identify a second growing flagellum in the early stages 1 to 3. While K-Means clustering is an effective and simple approach to segmentation, it can not compete against specialised manually constructed segmentation approaches. Overall, this research objective of creating and evaluating segmentation masks for the cells, kinetoplasts and flagella was fulfilled as I created masks that were good enough to provide new insight while sticking to my goals of using fast and explainable algorithms.

For the third research objective, let's start with the first part, the improvement of the length measurement that is based on the MAT: As mentioned in the presentation of the results, the flagella lengths have a big left tail. (A.16) The left tail in the flagella length histogram might be because the mNG masks were not filtered by area introducing noise through patchy signals. Because the

segmentation approach is not optimal, some of the flagella are broken down into several smaller parts, resulting in shorter connected components marked as flagellum. Smoothing the masks could be a possible solution, but this would make it even harder to identify multiple flagella in one cell. The improvement of the MAT-based length measurement for cells and flagella resulted in several statistics. The statistics show that during the cell cycle, the mean corrected cell length increases from 26.08 µm (1K1N) to 29.18 µm (2K1N) to 32.20 µm (2K2N). The uncorrected measurements showed an increase in cell length from 25.39 µm to 28.45 µm to 31.55 µm. The increase in cell length throughout the cell cycle is consistent with previous studies. For example, Wheeler et al. (2012) found the cell length increases from 16.3 µm to 19.6 µm to 21.0 µm in 1K1N, 2K1N and 2K2N cells, respectively. My statistics show a total corrected length increase of 6.12 µm or a relative increase of 23.47% from 1K1N to 2K2N. The uncorrected lengths increased by 6.16 µm in absolute terms and 24.26% in relative terms. While they found an absolute increase of 4.7 µm and a relative one of 28.83%. However, the large differences in mean cell length between the same stages are odd since similar methods were used. However, in section 3.3.4, specifically in figures 3.15 and 3.16, it was shown that the mean and standard deviation of cell lengths can differ greatly in between genes. However, not a single of the selected genes from my dataset had a mean length that was close to the 16.3 µm they observed. The smallest mean length was observed in gene Tb927.8.4970 with slightly under 24 µm for the same KN configuration. The only difference in methodology seems to be in the used population. Wheeler et al. used wild-type cells, cells without a modified genome. However, my study used cells with a modified genome, as the genes are tagged with mNG. As far as I know, adding mNG to a specific gene should not affect the original function of the gene. It should only cause the mNG protein to be expressed wherever the original gene is expressed. To answer why this discrepancy in cell length exists, a more in-depth understanding is required. The most logical explanation from my limited domain-specific knowledge is that it is simply a result of the variability in cell lengths between genes.

Next, the corrective term in the MAT-based length measurement increases the cell length by 2-3%. Specifically, increases of 2.72% (1K1N), 2.57% (2K1N) and 2.06% (2K2N). The increase due to the corrective term comes very close to the 3% difference in cell length observed in Wheeler et al. (2012) and could explain the gap between manual and automated measurements. However, it is unclear whether this 3% in their study was actually due to the systematic error in the MAT-based length measurement and not caused by random error, but it would certainly explain it. The variance between the percentage increases might be attributed to the very small sample sizes, which is also visible in the increase in the standard deviations. During experimentation, another issue was identified in the methodology caused by pruning the skeleton. As the MAT has many branches, some need to be pruned. However, some branches that should be part of the skeleton are discarded during the pruning process, further shortening the skeleton. Finding a solution to this problem is not an easy task and would potentially require a rewrite of the pruning algorithm. Although it was not tested in this research due to time constraints, it should be considered in future studies. The length measurements are also affected by the selection of the termini in the pruned skeletons in TrypTag. Although the skeletons are pruned, some still contain cycles or longer branches. My techniques allowed for further measurements of the flagella in different cell cycle stages. The following corrected flagella lengths were observed: 19.76 ± 4.93 µm (1K1N), 21.35 ± 5.68 µm (2K1N) and 21.12 ± 5.30 µm (2K2N). The uncorrected flagella lengths were: 19.48 ± 4.91 µm (1K1N), 21.06 ± 5.66 (2K1N) and 20.78 ± 5.34 µm. Those are relative increases of 6.88% and 6.67% and absolute increases of 1.36 µm and 1.30 µm. The statistics indicate that the flagella

might get slightly longer during the cell cycle. However, this is not conclusive, due to the large standard deviations and low sample sizes, especially for later stages. The results fall a bit short of previous results with constant flagella length measurements of 22.3 ± 0.1 µm [Rob+]. This could be a consequence of the used gene which locates the paraflagellar rod, which does not cover the whole flagellum. Or could again simply be attributed to random error, large standard deviations and low sample sizes.

To summarise, this subquestion of the third research objective was to determine whether a corrective term could explain the 3% difference in the MAT-based cell length measurement in the formula. Additionally, it aimed to determine whether this MAT measurement could be used to measure the length of the flagellum. The left tail in flagella length histograms was attributed to segmentation challenges, including incomplete gene channel coverage and fragmented flagella. The correction applied to cell length measurements reduced the gap between manual and automatic methods by 2 to 3% and could potentially explain the 3% difference. Furthermore, this study enabled flagella length measurements across cell cycle stages. This revealed a weak trend of slightly increasing flagella lengths between 1K1N and 1K1N of around 1.36 µm to 1.30 µm. However, caution is warranted in interpreting these results due to significant standard deviations, limited sample sizes, and the genes used to locate the flagellum. Further research is necessary to fully comprehend the factors contributing to the observed differences in cell length and flagella measurements. For this, larger sample sizes should be used and segmentation masks should be filtered or improved to avoid spotty signals.

The second part of the third research objective involves fitting ellipses to kinetoplasts and nuclei. This analysis requires examining several plots showing the differences in the distribution of minor and major axis length, orientation, mean and maximum DNA content, and the area between kinetoplasts and nuclei. The first observation in figure 4.1 for 1K1N cells was that the major axis length of kinetoplasts seems to have two separate peaks at 7 and 10 pixels, hinting at a bimodal distribution. However, the peak at 7 pixels is much smaller than the peak at 10 pixels. The different sizes of the peaks could be explained by the fact that most of the cells are in the very early stages of the cell cycle. This theory is supported by looking at the number of samples from previous experiments, such as the cell and flagella lengths, where a significant decrease in the number of samples was observed from 1615 samples in 1K1N to 70 samples in 2K1N and only 19 samples in 2K2N. So, the cells are not only more often in 1K1N compared to 2K1N and 2K2N, but they are also more often in the very first cell cycle stage than in later stages. This bimodal distribution, with a larger left peak, could also be explained in the following way. There could be a minimum size to the kinetoplasts, which would explain the sudden and sharp spike. It could also be related to the cell division itself. After 2K2N, the cell divides, the organelles are smallest, and because cell populations undergo exponential growth, there are many more young cells than older ones. This would also explain the large class imbalances observed in previous experiments. However, errors in my methodology or the TrypTag GitHub repository that was used to classify the objects from the DNA channel as kinetoplasts can not be excluded.

Figure 4.2 was described after that. For this plot, two observations were expected. Firstly, the kinetoplast distribution should normalise again as two kinetoplasts are present that should have similar sizes. Secondly, the nucleus should start growing. The first expectation could not be confirmed. There was still a smaller peak at 10 pixels. However, it is no longer as clear as for 1K1N cells due to small sample sizes. However, it has to be mentioned that the kinetoplast distribution has a thicker right tail compared to 1K1N, which skews the mean higher, so the 2K1N kinetoplasts

are, on average, larger, but this is due to outliers. The second observation seems to be true. Not only does the mean major axis length increase between the 1K1N and 2K1N configurations. But there also seem to be two peaks at 26 and 36 pixels in the major axis length. This confirms that the major axis length can be used to identify the growth of the nucleus. The growth of the nuclei is also visible in the bottom right chart, showing the area. The mean area of the nuclei changed from 381.8 pixels in the 1K1N configuration to 490.90 pixels in the 2K1N configuration, clearly showing its growth.

The third figure, labelled A.19, was not thoroughly analysed as it does not contain any information relevant to the critical cell cycle stages that are being classified. During the course of this research, it was observed that the major axis lengths of kinetoplasts and nuclei followed a bimodal distribution. This indicates that the subcycle progress can be measured by using ellipses to measure the dimensions of these organelles. This discovery can be potentially useful to differentiate between closely related stages like stages 2 and 3, as well as stages 4 and 5.

To conclude, the third research objective has been achieved through the use of an improved MAT-based cell and flagella measurement, which enhances a previously used technique. Additionally, a new method has been developed to capture the morphology of the organelles using ellipses to gauge the subcycle progress. These advancements have made it possible to obtain more accurate measurements and better understand and track the morphology of the organelles. Additionally, I consider the first research objective as completed, as I used the findings from the literature and compared them to my new measurements.

In order to fulfil the fourth research objective, two minor questions need to be answered. Firstly, we need to revisit the scatterplot matrices. The findings revealed several interesting observations that require further explanation. The first observation is that different configurations seem to be localised but overlapping, indicating differences in morphology across various cell cycle stages and could hint at misclassification of KN configurations. Furthermore, it is unclear whether this result is truly due to different attributes in different stages or simply a consequence of the large class imbalance. The large class imbalance could cause the full range of 1K1N cells to be shown but limited ranges for 2K1N and 2K2N. Again, the class imbalance has two possible explanations. Either it could be caused by TrypTag's KN classification or is due to exponentially growing populations, as hypothesised earlier. The lack of distinct minimally overlapping clusters could be explained by the fact that biological processes happen continuously instead of instantaneously. The script also automatically calculates the Pearson correlation between all the unique pairs in the plot. The pairs with the highest Pearson correlations, and therefore with linear relationships, are the area of the kinetoplast and the major axis length of the kinetoplast with a coefficient of 0.92, the sum of the DNA in the kinetoplast and the sum of DNA in the nucleus with a coefficient of 0.89, the cell area and corrected cell length with a coefficient of 0.84 and lastly the area of the nucleus and the major axis length of the nucleus. However, these strong relationships reveal nothing new about the cell cycle stages.

Furthermore, there is also a lack of minimally overlapping clusters in the scatterplots of the different dimensionality reduction techniques, as seen in figures A.24, A.25 and A.26. This could mean that the cell cycle stages and KN configurations are not clearly differentiable by either pair of these features or by all features in combination. So the stages do not form clusters in this 11-dimensional feature space. However, this experiment also resulted in tables A.6 and A.7. Of particular interest is the first table. It shows the percentage contribution of each feature to the

first principal component. The first principal component is the basis vector in this 11-dimensional space that points in the direction with the most variance in the data. Looking at the percentage contributions of each feature of this first principal component tells us how much each feature contributes to the variance of that principal component. Overall, the first principal component explains 33.62% of the variance in the data. The most important contributors to that are the nucleus area, the cell area, the major axis lengths, the corrected cell length and the kinetoplast area. Those components contribute at least 10% to the variance explained by principal component one. The second principal component explains a further 18.44% of the variance. The three most important contributors to that are the sum of DNA content in kinetoplasts and nuclei and the corrected cell length. In total, those two components explain 52.06% of the variance in the data. With this approach, the best features can be extracted to build simpler classifiers. Noticeable is, that the normalised position of the kinetoplast and nucleus do not seem that important.

The last part of the fourth research objective was to find out whether ResNet-50, a deep convolutional neural network could be used together with PCA, a dimensionality reduction method, to find patterns or clusters similar to Eulenberg et al. 2017 [Eul+17]. As mentioned during the presentation of results, the created 2-dimensional plot of the principal components of the feature vector was not structured. Despite the improvements made, such as a uniform background, I believe there is one main reason for the neural network's shortcomings. The network was not trained on biological data or specifically on this dataset but rather on everyday objects such as cars, bicycles, trees, people, cats, and dogs. Nevertheless, it was expected that the network could pick up on more visual differences such as colour, size, texture, or shape differences. However, the only noticeable difference that was observed was that data points located in the top left corner of the plot contain images of small curled-up cells that appear to be brighter. In contrast, data points located on the opposite side, in the bottom right corner, contain darker and longer cells. Although my methodology had some shortcomings, there is room for improvement. One of the limitations of my approach is that the plots did not distinguish between different cell cycle stages, like in the scatter matrix plots. Perhaps, by making this change, one could observe some structure in the results. The results in Eulenberg et al. also show more continuous transitions in the principal components instead of clusters. Furthermore, ResNet-50 uses three channels, which should be used for the mNG and DNA channels instead of just repeating the phase contrast channel. However, I am sceptical about this approach because the neural network used in our analysis is unlikely to differentiate between the number of kinetoplasts or nuclei present within a single cell. To further enhance the accuracy of future results, I suggest fine-tuning a model on biological data, specifically on T. brucei data microscopy image data. Based on the experimentation with ResNet-50 feature vector visualisation, it can be concluded that it did not capture the cell's structure, size, colour, or shape as expected. Although there was a noticeable difference between the images when we compared the top left and bottom right corners, it wasn't significant enough to draw any concrete conclusions. Better results are expected with the improvements in colouring the different cell cycle stages, finetuning a network, and passing all three channels.

The fourth research objective is fulfilled. Unsupervised methods were applied to extract correlations and the most relevant features of principal components. However, no new and unexpected correlations were observed. The most important features of the principal components showcase how and which features could be selected to classify cell cycle stages. Unfortunately, the scatterplot matrix was not as useful as expected, as no clusters could be observed. Although features of a specific KN configuration seem to be localised in certain regions of the subplots, they were

overlapping. To see whether this overlap existed in more than two dimensions PCA, t-SNE and UMAP visualisation were created. However, they also did not reveal any clusters in the data useful for classification. This means that better features are needed to distinguish cell cycle stages easily. More importantly, the methodology should be improved first, as the features for kinetoplast and nuclei properties only looked at the kinetoplast or nucleus closest to the anterior end of the cell. Features like the distance between the nuclei and kinetoplasts should be considered if there are two of the same organelles like in 2K1N and 2K2N. Although the multivariate scatterplot analysis and the ResNet-50 feature vector visualisation did not achieve the goal of identifying clusters or novel correlations for cell cycle classification, valuable insights were gained that can inform future research.

It was not possible to fully achieve the fifth and sixth research objectives, which were to build and evaluate a classifier, due to limitations in the techniques used. One of the main obstacles to building a classifier was the lack of flagella information. Since the techniques used could not differentiate between one or multiple flagella within a cell, it was impossible to build a classifier, as the designed binary decision builds on top the KNF configurations. Moreover, it is uncertain whether the correlation found between subcycle progress and the major axis length of the kinetoplast and nucleus would be sufficient to produce clear classification results. However, the binary decision tree could be more refined. In the methodology section where the decision tree was designed, some assumptions were made about which features would be useful for telling the difficult stages 2 and 3, as well as 4 and 5 apart. To decide between stages two and three, it was proposed to use some combination of the major axis length of the kinetoplast and the length of the new flagellum. As a bimodal distribution in the major axis length of the kinetoplast was observed, with the two peaks being at around 7 pixels and 10 pixels, an initial decision boundary at 8.5 pixels could be used to differentiate stages two and three. The same can be applied to stages four and 5 for the nucleus. The proposed features were the relative position of the flagella and the size of the nucleus. Again, there is no information about the flagella. However, two peaks in the major axis lengths of the nuclei were observed at 26 and 36 pixels. With this, we can again linearly separate the two peaks and set a boundary for our decision to be 31 pixels.

However, through the previous experiments, the decision tree could be further refined. The final decision tree can be seen in figure 4.6. The two boundaries for deciding between stages two and three, and four and five, are 8.5 pixels in the major axis length of the kinetoplast and 31 pixels in the nucleus. In micrometres, these initial boundaries are at 0.87 µm and 3.1 µm. Therefore, the classifier would predict stage two if the cell has one kinetoplast, two flagella and a major axis length of the ellipse fitted to the kinetoplast below 0.87 µm and stage three above 0.87 µm. For a cell with two kinetoplasts and one nucleus, it would predict the fourth stage if the ellipse major axis length that is fitted to the nucleus is below 3.1 µm and stage five if it's above 3.1 µm. Although this binary decision would likely not perform exceptionally well with those linearly interpolated decision boundaries, it is a good starting point for future studies and the best I could do with the data available. Unfortunately, the evaluation of this classifier is left open.

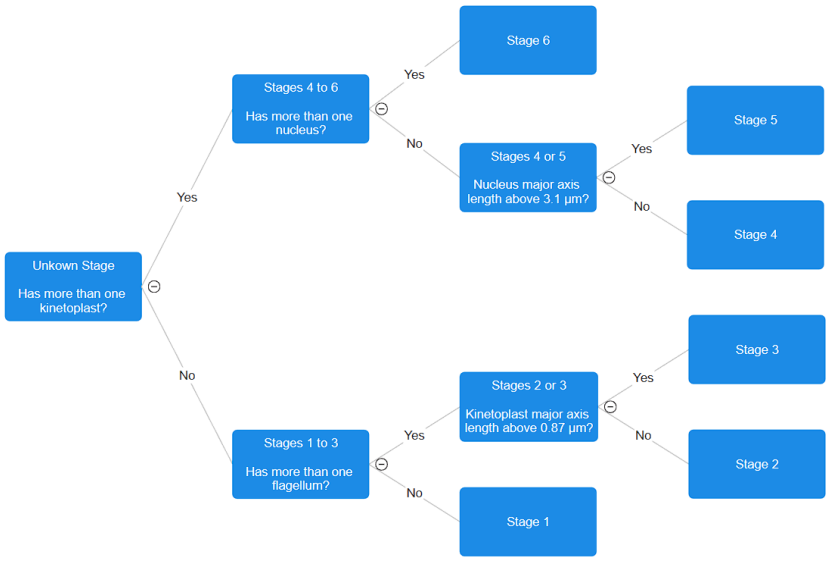

Figure 4.6 – The final binary decision tree for classifying procyclic Trypanosoma brucei with initial guesses for the decision boundaries between stages 2 and 3, as well as stages 4 and 5.

4.3 Implications and Discussion

The Implications and Discussion section explores the findings' broader impact and relevance. Here, I will identify how my contribution fits into the current understanding of the field and discuss the potential applications and consequences of the results in the context of the existing body of knowledge.

Through the improvements and findings in this thesis, future cell length measurements will have improved accuracy, by applying the corrective term to the MAT-based length measurement; the length of the flagellum can be automatically tracked, by using the same MAT-based approach; the subcycle progress of kinetoplasts and nuclei can be tracked more accurately, by using major axis length from fitted ellipses; procyclic Trypanosoma brucei could be classified efficiently, by using the binary decision tree constructed in this thesis; researchers gain a new resource to study the paraflagellar rod and the flagellum, by using the presented flagella dataset; Trypanosome researchers are exposed to new ideas from other fields like C. elegans and more recent machine learning focused approaches; researchers have access to new visualisations and tools, like the flagellum visualisations, multivariate scatterplots and other interactive charts; researchers gain valuable statistics about cell lengths and flagellum lengths for comparison and they have access to a fast and reliable alternative for segmenting the DNA channel. Ultimately, they increase accuracy in future studies and provide new features and a framework for classification, ideas for further research, and data and tools for more analyses. This thesis also validates existing approaches, like the MAT-based length measurements, as it was shown that it can be applied not only to cell lengths but also to flagella lengths and potentially even more.

However, this thesis provides many points for improvement. The segmentation masks of the paraflagellar rod could be improved by capturing finer details of the signals without creating patchy segmentation masks. Furthermore, the branched flagella skeletons need to be split in order to create accurate KNF configurations and more accurate flagellum length measurements and

enable the tracking of the flagellum throughout the cell cycle. For this, existing methods could be used, as pointed out in the literature review. Once KNF configurations are accurately captured, the final binary decision tree classifier could be evaluated. Additionally, more features could be tested like geometric, intensity and texture features, as well as better features for tracking the relative position of organelles. Moreover, even more modern neural networks could be tested and trained on Trypanosoma data to find visual features. This includes the use of the information of all three channels. The results could simply be replicated on a larger scale with more data to get more conclusive results of the statistics, as 2K1N and 2K2N samples are limited. This would also help find better decision boundaries for the classifier.

Now, on a more personal level, some of the limitations I faced. A major hurdle in writing the thesis was the missing flagella dataset, which I thought would be provided to me. This added significant overhead to the writing process on top of the already existing complexity of this interdisciplinary thesis topic. The writing itself was difficult at times due to English not being my native language. However, this fact was alleviated a bit by using Grammarly, which helps with spelling, grammar, and punctuation but also suggests how the clarity of sentences could be improved. The missing flagella dataset and the missing labelled data made the thesis harder and broader than it should have been. A significant portion of this thesis had to be dedicated to the construction, segmentation and evaluation of the flagella dataset, which was not planned before starting it. Ultimately, I think this led to the KNF configurations not working due to the segmentation approach, which was required for the classifier to work. Because of that, the classifier could not be evaluated. This is why I pivoted the focus of my research objectives towards the features themselves instead of the classifier being the focus. Overall, I think this was the correct decision, as otherwise, I would have no interesting results.

To summarise, it is possible to build a classifier for procyclic Trypanosoma brucei cells from microscopy images. However, further research is required to capture the flagellum accurately and to find better decision boundaries between the difficult stages. If those challenges can be solved, fully automated and efficient classification would be possible. Which would allow for more detailed and extensive studies without the need for human labelling. I hope that the methods and ideas presented shine new light on this research field and broaden the view of already existing solutions and novel approaches.

4.4 Conclusion and Future Work

This thesis advanced the understanding of cell cycle classification in procyclic Trypanosoma brucei based on images. This was accomplished by identifying gaps in the literature, cell cycle classification, and current implementations. The main gap is a lack of automated tools for detecting the KNF configurations, which is the first step to a more complete classification. For this, a custom flagella dataset was created by using the TrypTag GitHub repository. The diverse dataset includes 10 genes that are expressed in the paraflagellar rod and can be used to measure flagellum lengths. To include the flagellum as an additional source of information, self-developed segmentation techniques were employed using K-means clustering and simple matching of the flagellum and cell masks. Although the approach and the segmentation masks are not ideal for detecting the number of flagella within the cell, the flagella information provided a large amount of statistical data about its length during different KN configurations. Additionally, this thesis introduced the corrected MAT-based length measurement, which explains the discrepancy be-

tween automated and manual measurements found in a previous study. The study also introduces the use of ellipses to measure the dimensions of the kinetoplast and nucleus, particularly the major axis length of the fitted ellipses, to determine their subcycle progress. In addition, the study explores various techniques to identify correlations between different cell features, such as the cell area, corrected cell length, corrected flagella length, area of the kinetoplasts and nuclei, DNA content of the kinetoplasts and nuclei, the position of the kinetoplast and nucleus, and the major axis length of the ellipses fitted to the kinetoplasts and nuclei. Advanced techniques, such as principal component analysis on the feature vector of a pre-trained ResNet-50 convolutional neural network, were used to analyse the visual structure of the phase contrast images. The findings and improvements were then used to construct a three-stage binary decision tree model that can theoretically distinguish all six cell cycle stages, but still has to be evaluated. In addition to the implemented and tested techniques, this thesis provides a comprehensive literature review that includes innovative ideas from other research fields to further improve the segmentation, feature selection and classification of trypanosome cell cycle stages. With that, all of the research objectives and questions could be answered, except for the evaluation of the classifier. The main research question is whether microscopy images from TrypTag can be used to classify cell cycle stages in procyclic Trypanosoma brucei. The answer is that it is possible. But the classification approach detailed here, requires a more nuanced way to extract the KNF configurations, especially within earlier cell cycle stages.

Future work should focus on building upon the idea of using the flagella as an additional source of information. It is conceivable to develop masks for the flagella by utilising the paraflagellar rod component that can identify more than one flagellum growing within a single cell. Such masks must precisely capture the mNG signal without including the glow around these signals; otherwise, it would be impossible to distinguish between individual flagella. Implementing untangling algorithms is crucial for separating fused flagella masks within a cell, thereby enhancing the accuracy of the methodology introduced in this thesis. Once successful in classifying the complete KNF configuration, efforts should be directed towards implementing the three-stage binary decision tree outlined in this thesis. Utilising the introduced ellipse-fitting method, along with the newly acquired measurements of the flagellum, will be pivotal for distinguishing between stages 2 and 3 as well as stages 4 and 5. Further, creating an open-source dataset that labels all six stages would be an invaluable asset to the field, promoting a collaborative and accelerated advancement in understanding and classifying Trypanosoma brucei's cell cycle stages. While correlations or patterns in the multivariate scatterplots and the principal components of neural network feature vectors offer some insights, they may not be the most efficient and straightforward pathway for progress.

In conclusion, this thesis presents a comprehensive investigation of cell cycle classification in procyclic Trypanosoma brucei based on microscopy images. The findings and improvements have laid a foundation for further research and are poised to lead to better classification of the cell cycle stages. Allowing researchers to conduct more in-depth and accurate studies. Ultimately, this will help in understanding Trypanosomes and fundamental cell biology and alleviate the fundamental bottleneck of data analysis.

A Appendix

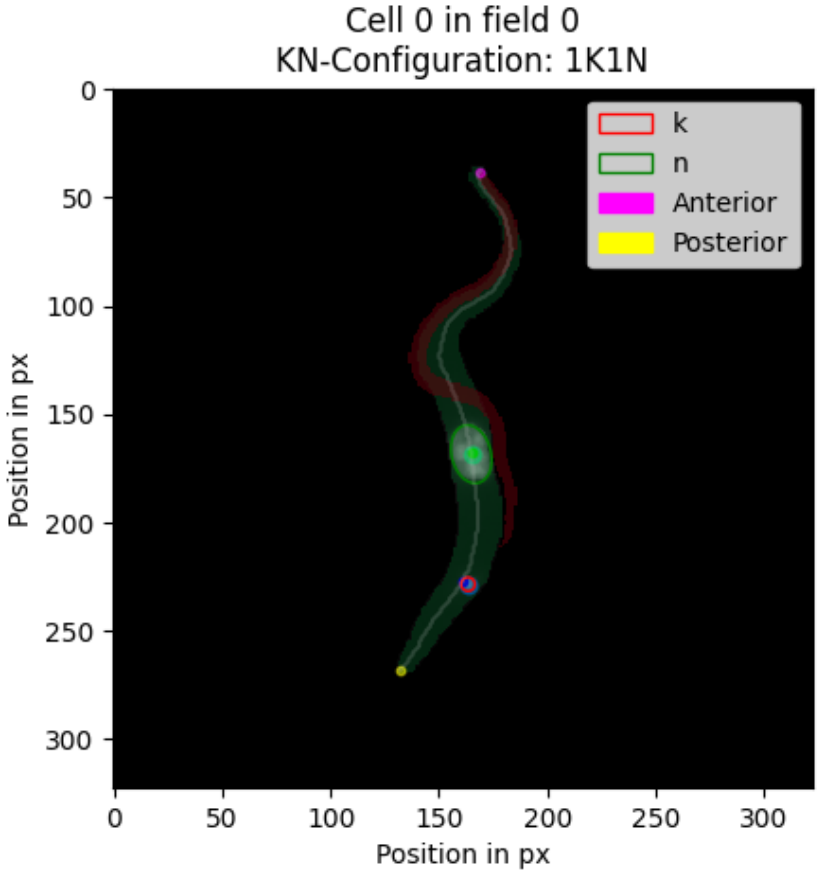

Figure A.1 – Generated Plot showing the structure of a 1K1N cell from gene Tb927.6.4140. It Highlights the paraflagellar-rod in red, the phase mask in green, the skeletonisation of the phase mask in gray, the anterior and posterior point of the cell as pink and yellow dots, the nucleus and kinetoplast in green and blue, and the nucleus and the kinetoplast are circled in green and red.

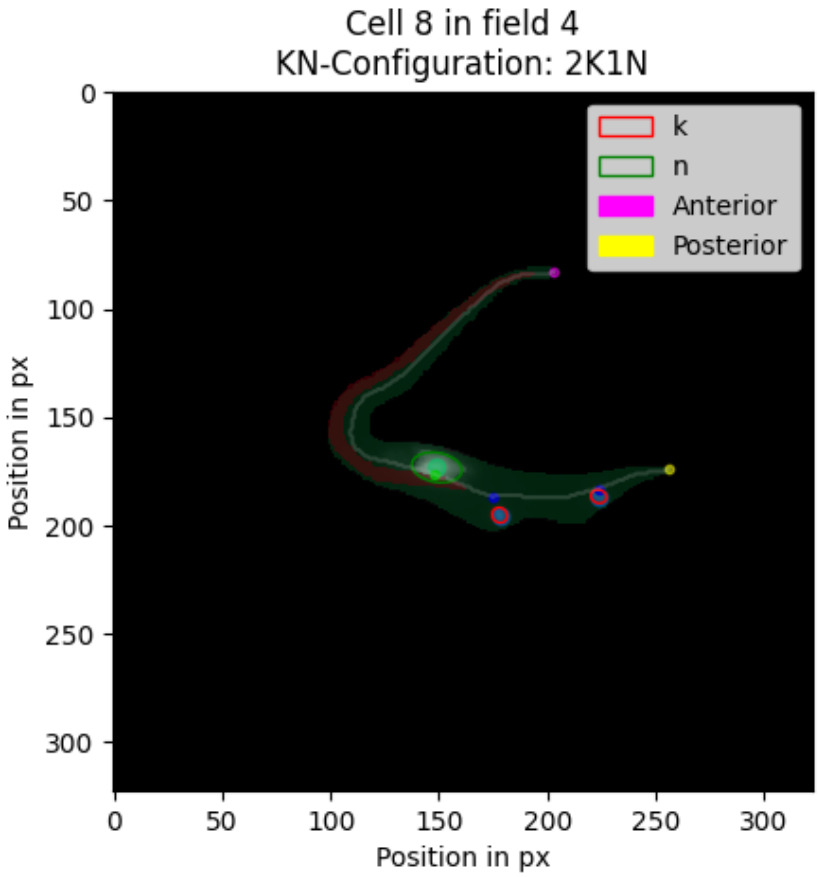

Figure A.2 – Generated Plot showing the structure of a 2K1N cell from gene Tb927.6.4140. It Highlights the paraflagellar-rod in red, the phase mask in green, the skeletonisation of the phase mask in gray, the anterior and posterior point of the cell as pink and yellow dots, the nucleus and kinetoplast in green and blue, and the nucleus and the kinetoplast are circled in green and red.

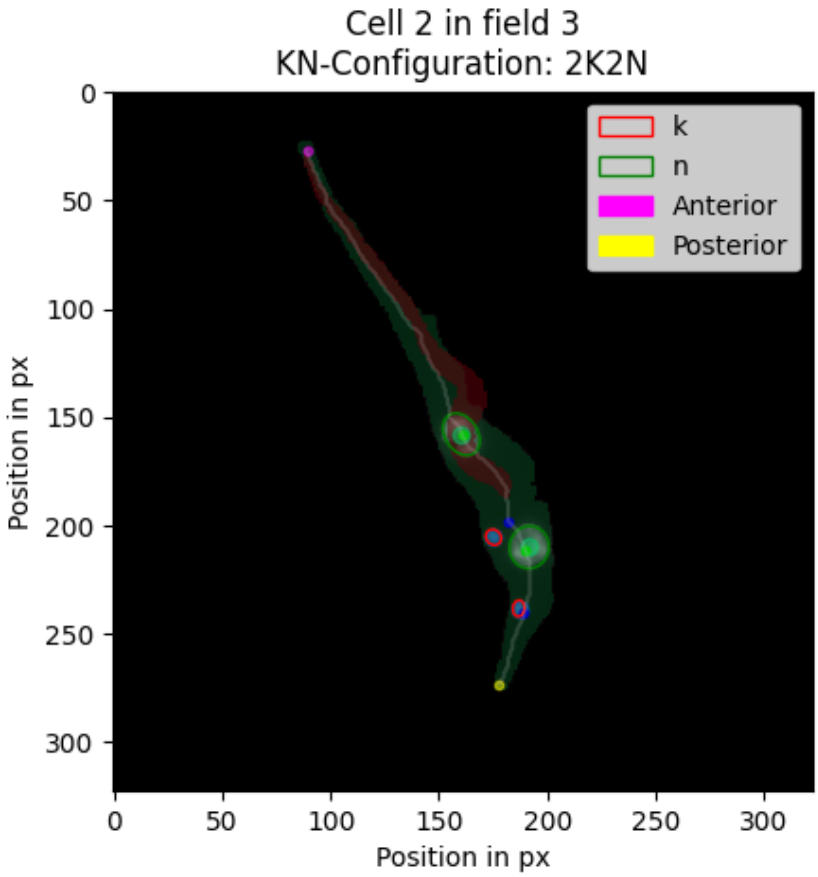

Figure A.3 – Generated Plot showing the structure of a 2K2N cell from gene Tb927.6.4140. It Highlights the paraflagellar-rod in red, the phase mask in green, the skeletonisation of the phase mask in gray, the anterior and posterior point of the cell as pink and yellow dots, the nucleus and kinetoplast in green and blue, and the nucleus and the kinetoplast are circled in green and red.

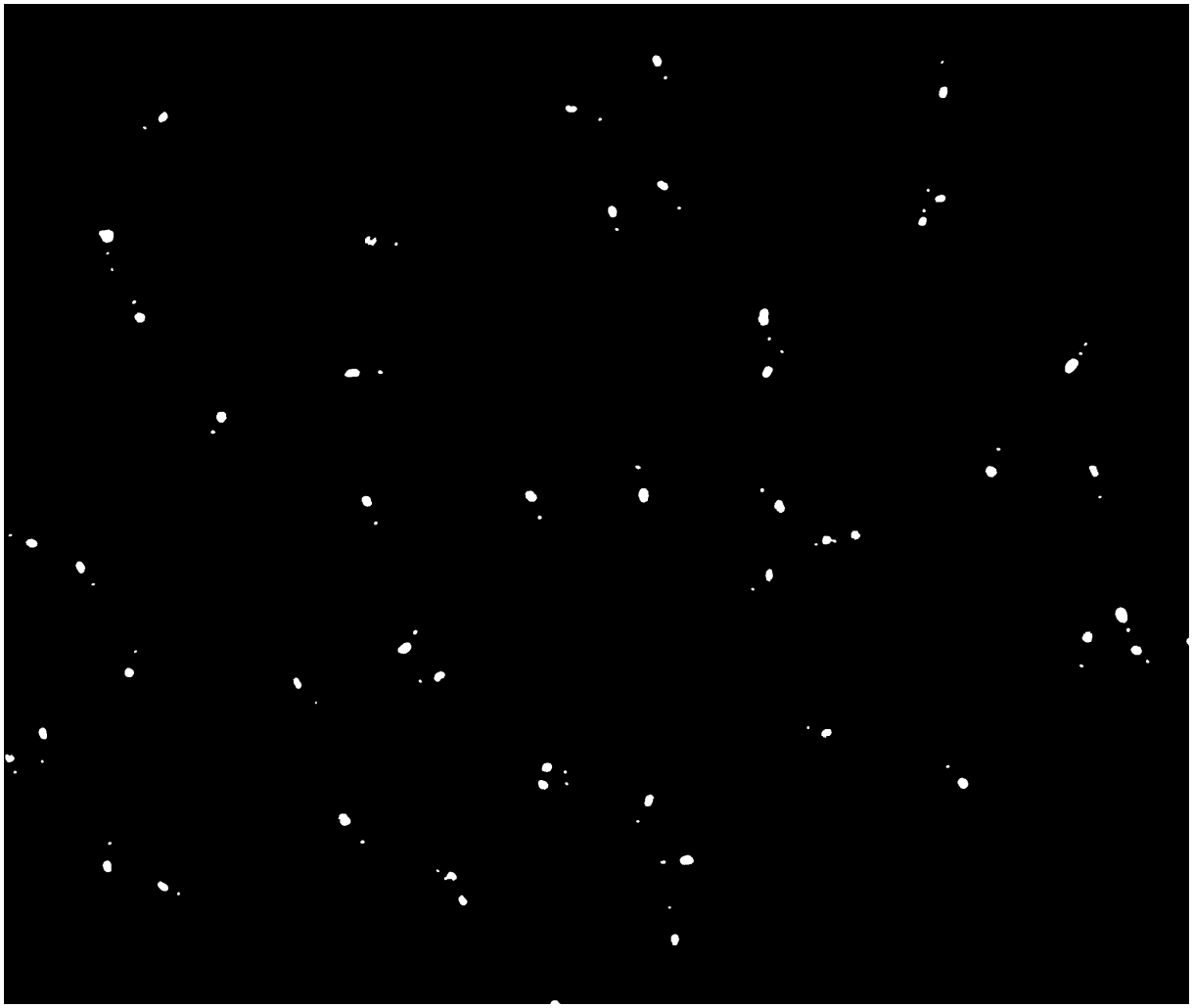

Figure A.4 – An unfiltered segmented field image of the DNA channel using K-Means clustering with K=2 showing the masks for the nuclei and kinetoplasts

Figure A.5 – An unfiltered segmented field image of the MNG channel using K-Means clustering with K=2 showing the masks for the flagella

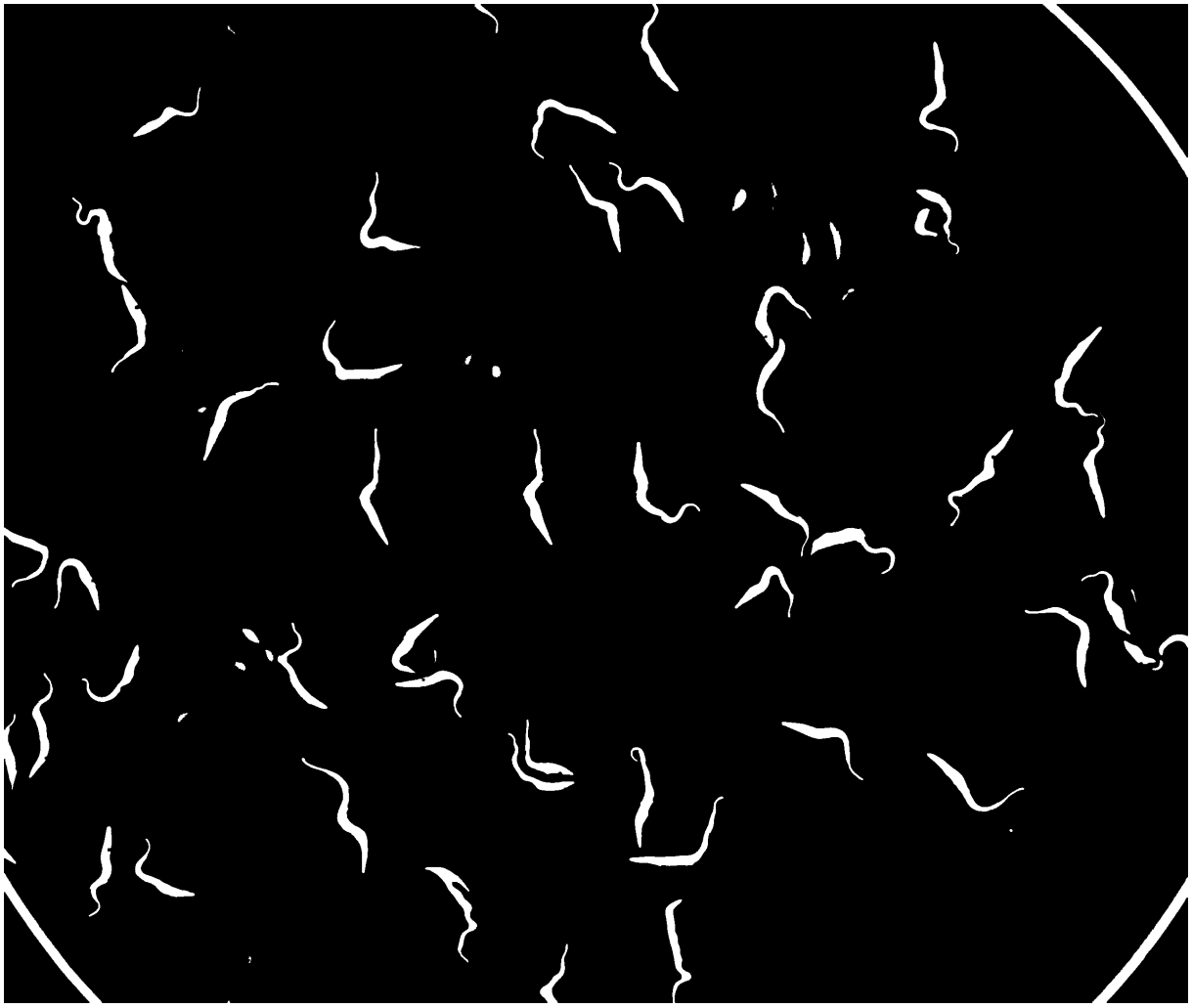

Figure A.6 – An unfiltered segmented field image of the phase contrast channel using K-Means clustering with K=4 showing mostly the cell masks but also some noise and the border of the cell dish

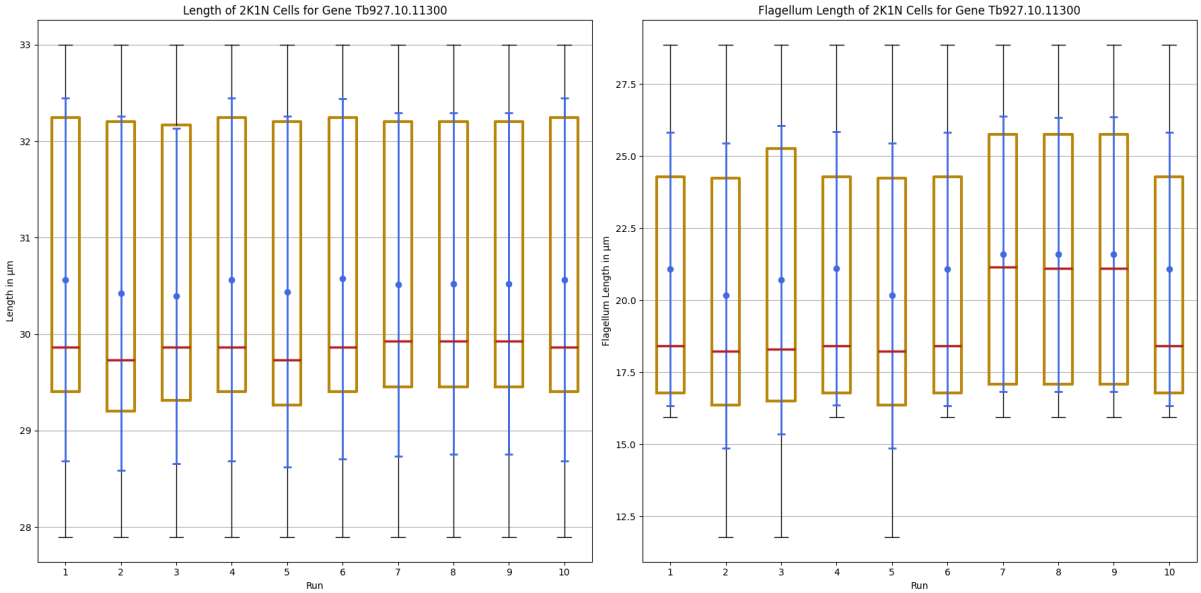

Figure A.7 – Boxplot comparing the results of the cell and flagellum length measurements across ten runs for 2K1N cells of the gene Tb927.10.11300. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

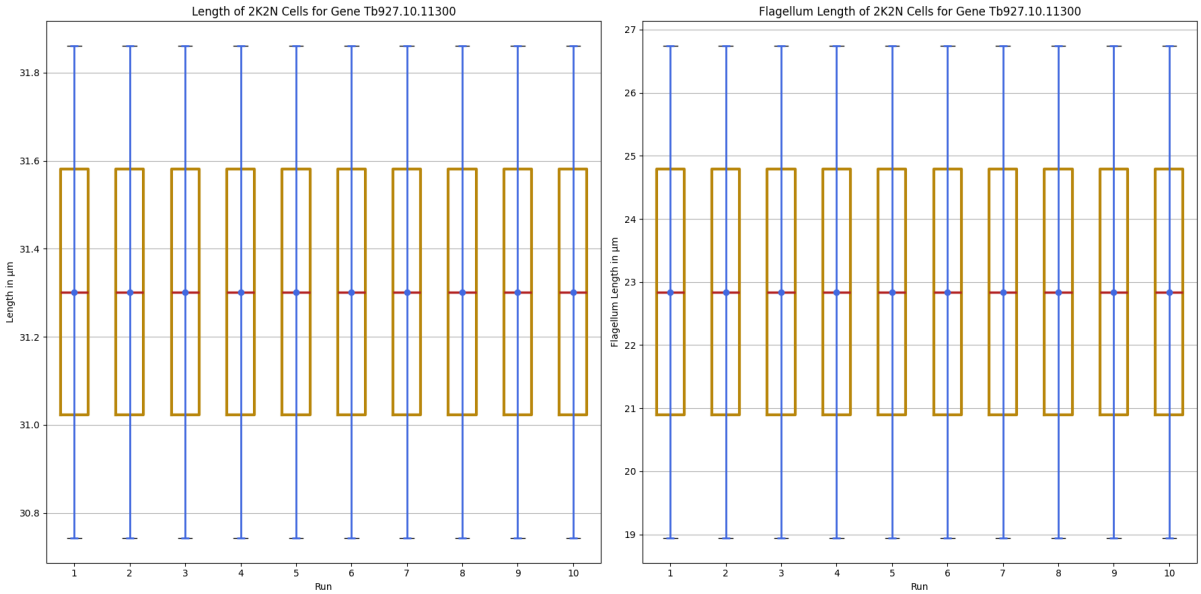

Figure A.8 – Boxplot comparing the results of the cell and flagellum length measurements across ten runs for 2K2N cells of the gene Tb927.10.11300. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

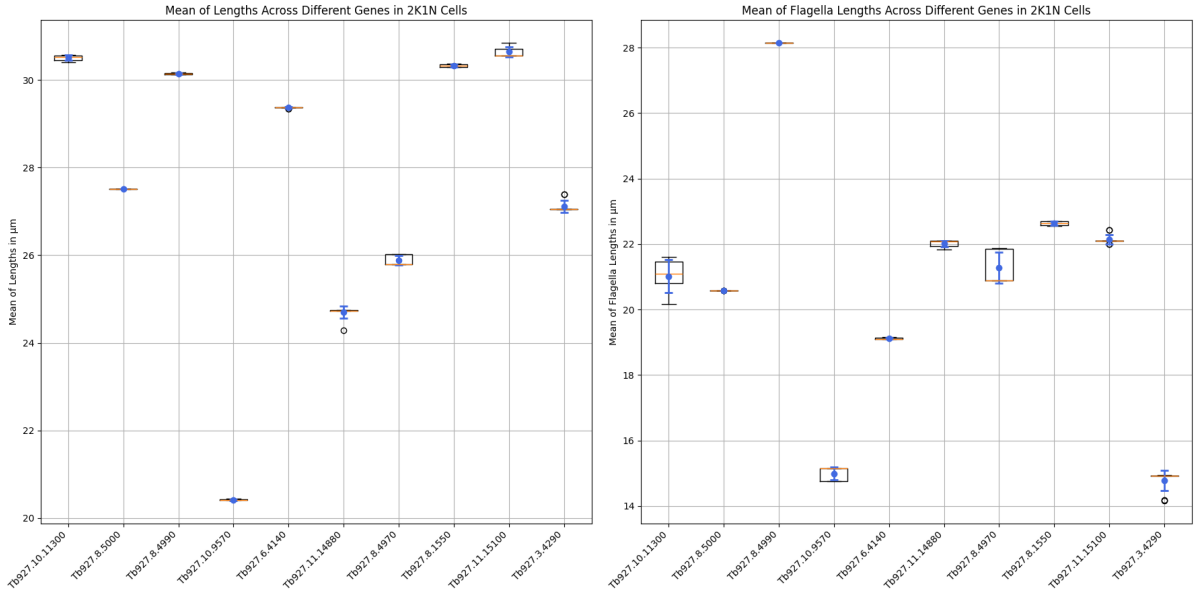

Figure A.9 – Boxplot comparing the mean of the cell and flagellum length measurements across ten runs for 2K1N cells for all genes. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

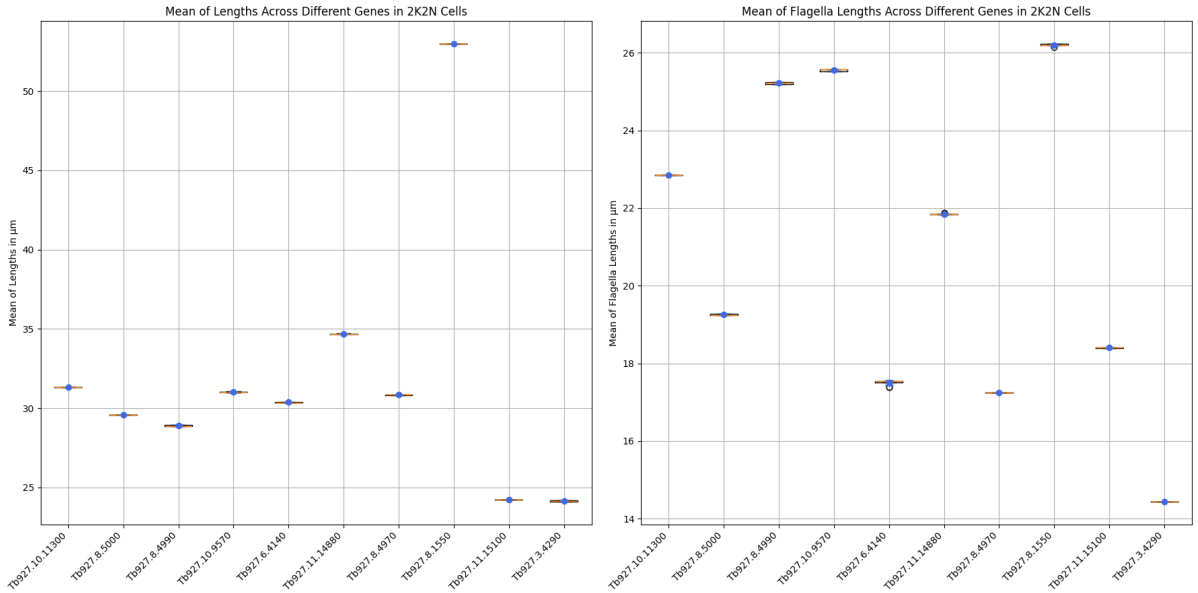

Figure A.10 – Boxplot comparing the mean of the cell and flagellum length measurements across ten runs for 2K2N cells for all genes. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

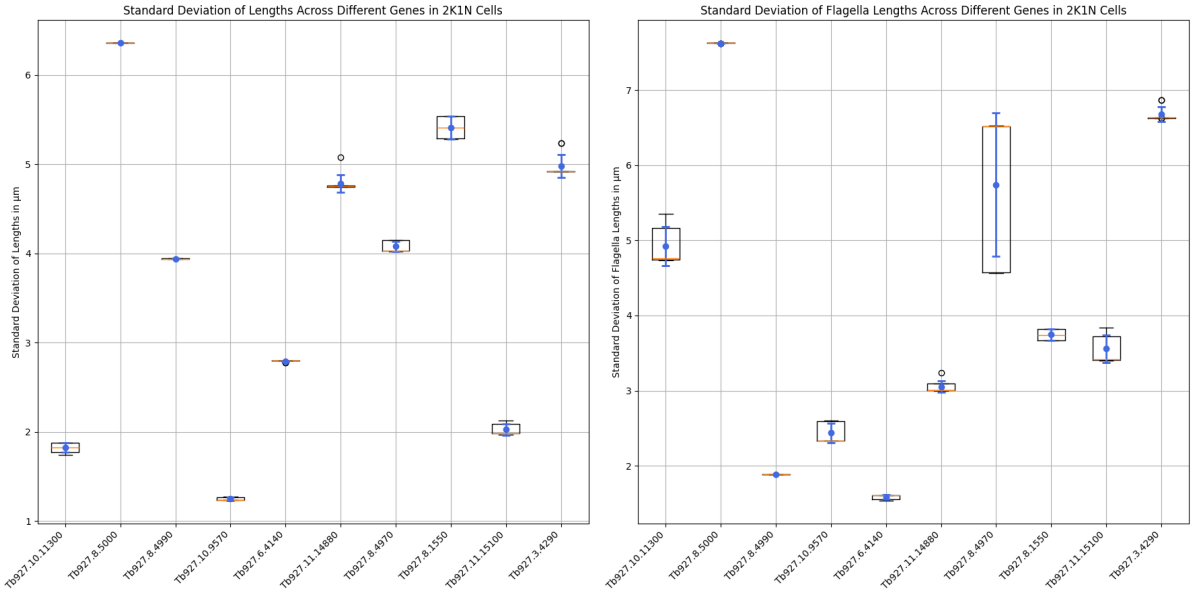

Figure A.11 – Boxplot comparing the standard deviation of the cell and flagellum length measurements across ten runs for 2K1N cells for all genes. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

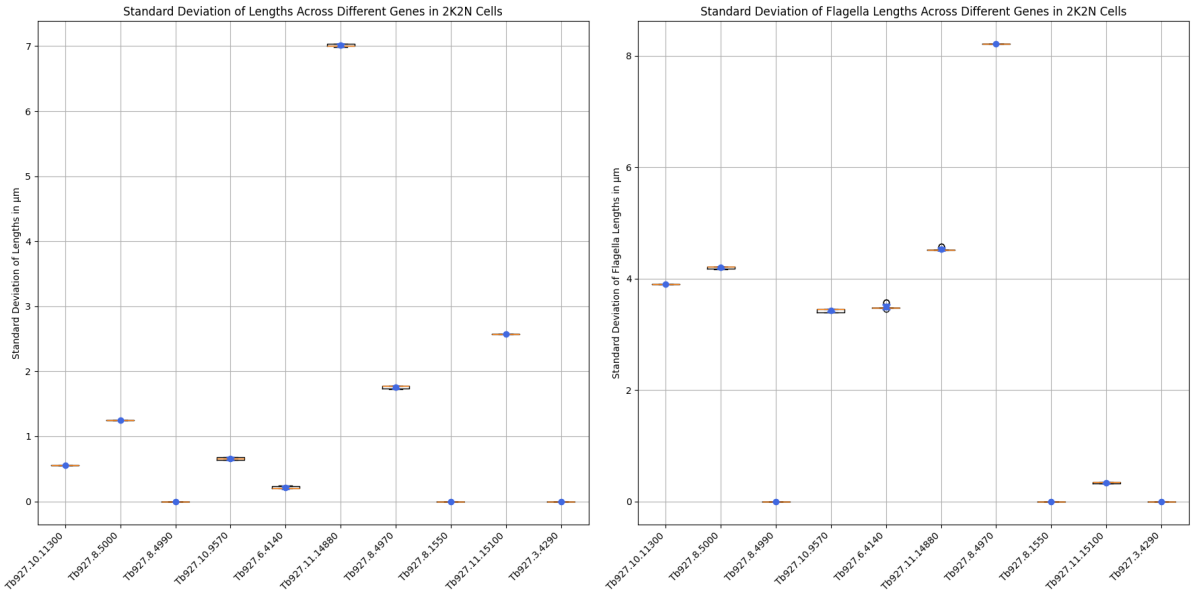

Figure A.12 – Boxplot comparing the standard deviation of the cell and flagellum length measurements across ten runs for 2K2N cells for all genes. The mean and standard deviation are highlighted as blue error bars, and the red line is the median.

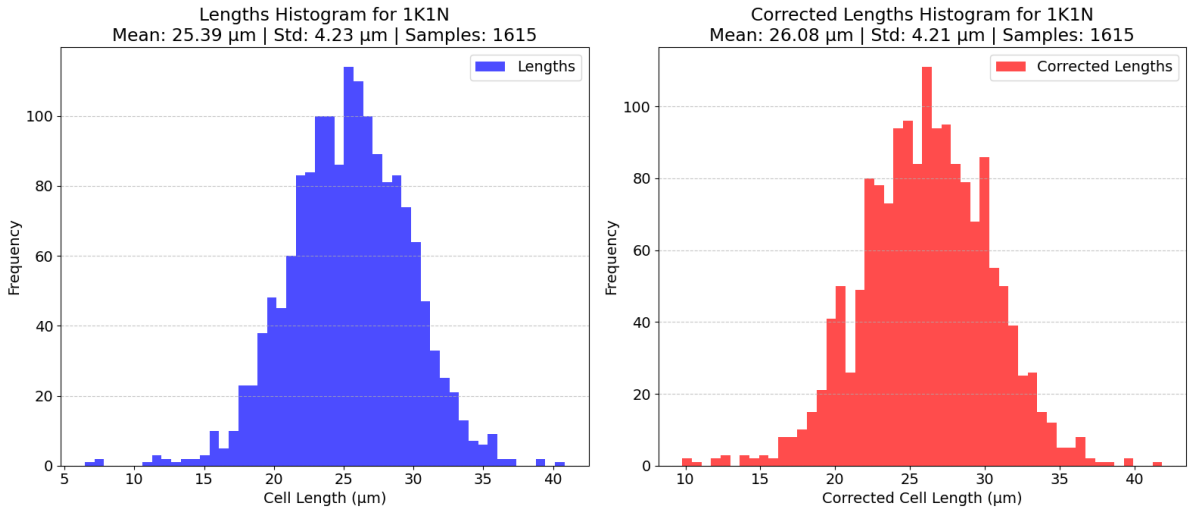

Figure A.13 – Two histograms for the length of 1K1N cells. On the left in blue the unadjusted length as calculated by TrypTag. On the right in red the adjusted length of the cell.

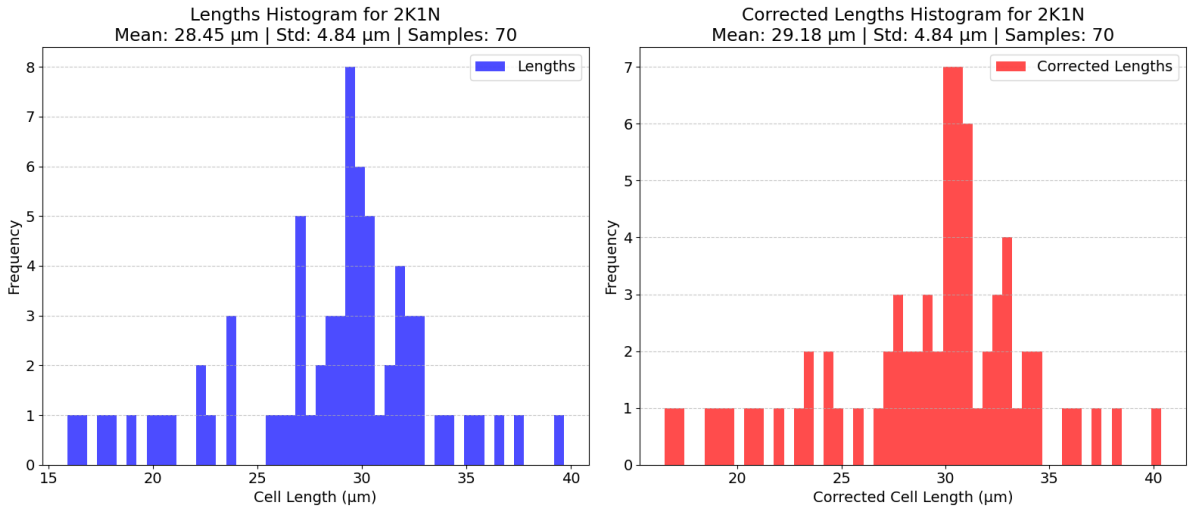

Figure A.14 – Two histograms for the length of 2K1N cells. On the left in blue the unadjusted length as calculated by TrypTag. On the right in red the adjusted length of the cell.

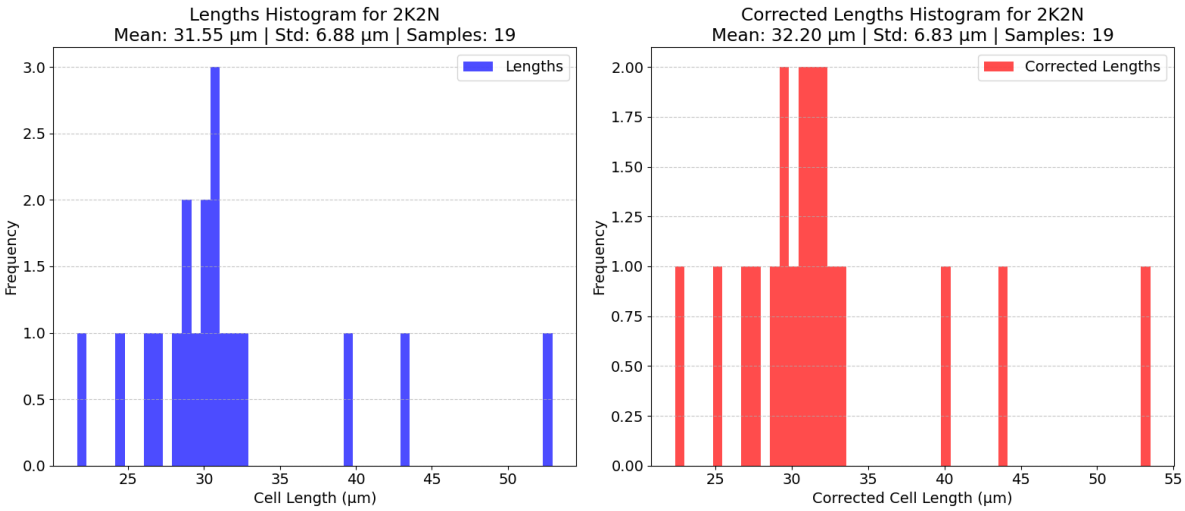

Figure A.15 – Two histograms for the length of 2K2N cells. On the left in blue the unadjusted length as calculated by TrypTag. On the right in red the adjusted length of the cell.

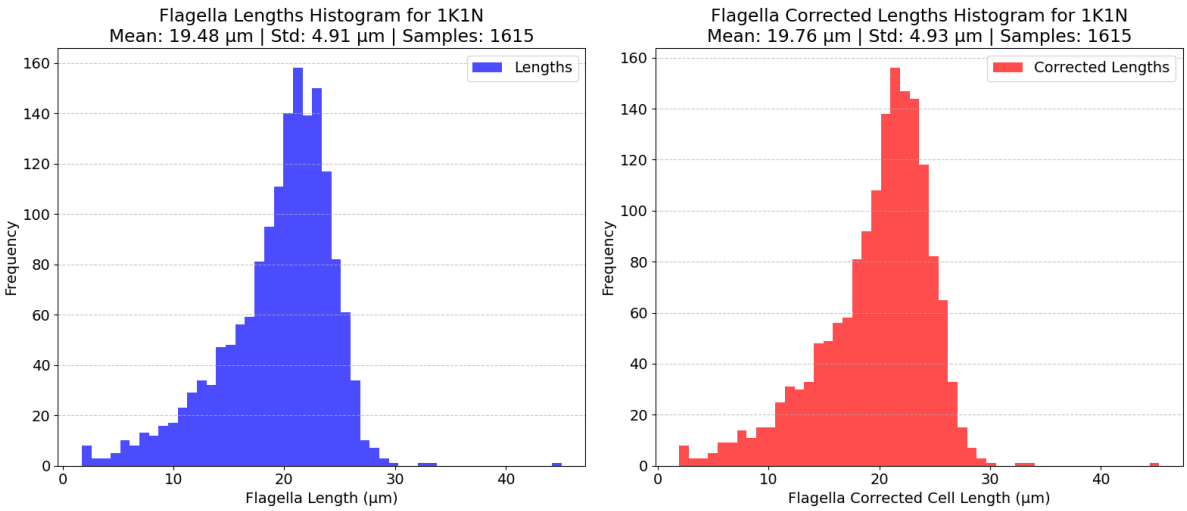

Figure A.16 – Two histograms for the lengths of the paraflagellar rod of 1K1N cells. On the left in blue the unadjusted length using the TrypTag implementation. On the right in red the adjusted length of the paraflagellar rod.

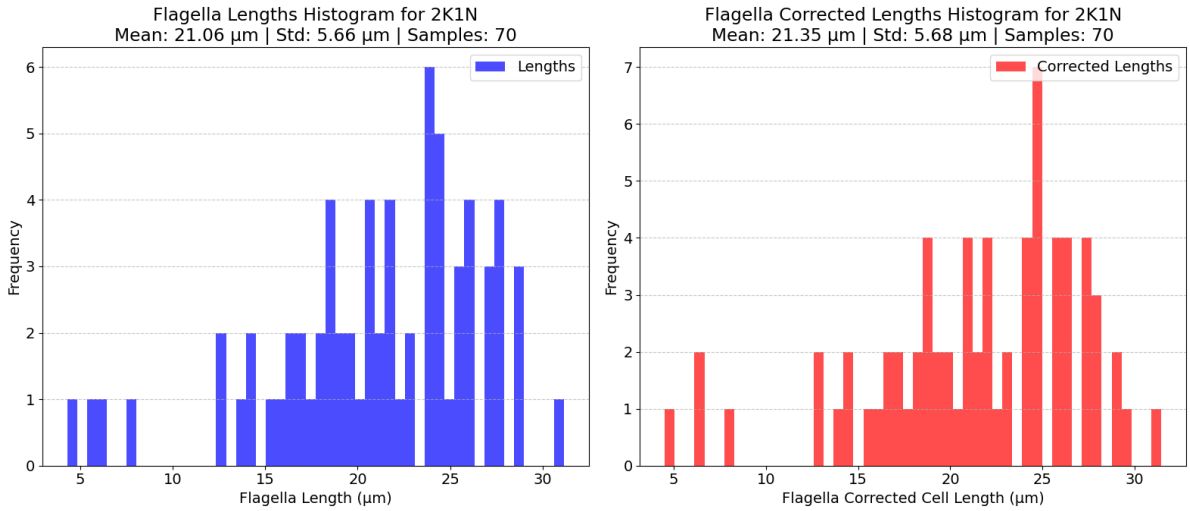

Figure A.17 – Two histograms for the lengths of the paraflagellar rod of 2K1N cells. On the left in blue the unadjusted length using the TrypTag implementation. On the right in red the adjusted length of the paraflagellar rod.

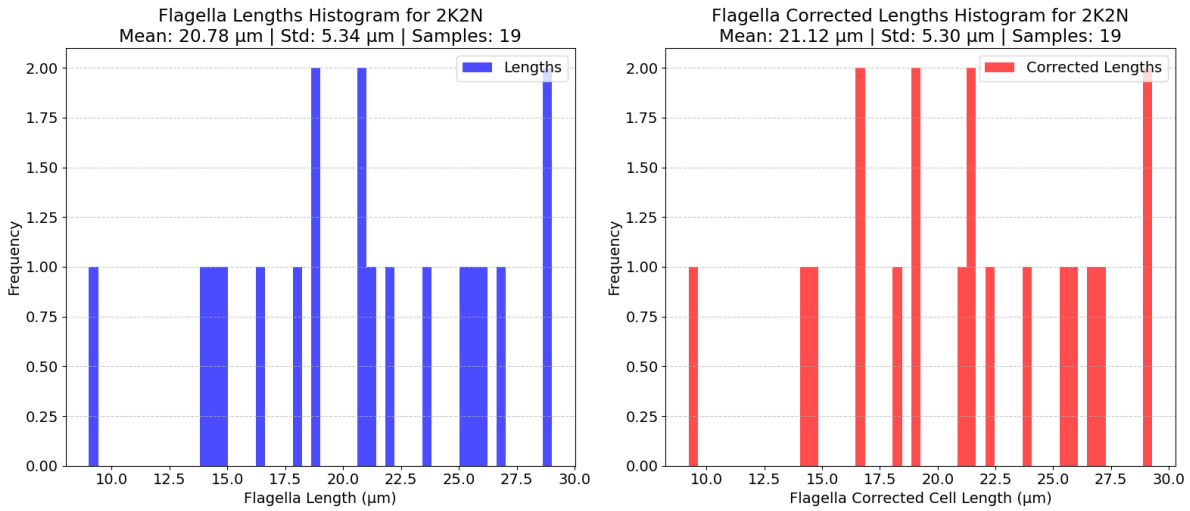

Figure A.18 – Two histograms for the lengths of the paraflagellar rod of 2K2N cells. On the left in blue the unadjusted length using the TrypTag implementation. On the right in red the adjusted length of the paraflagellar rod.

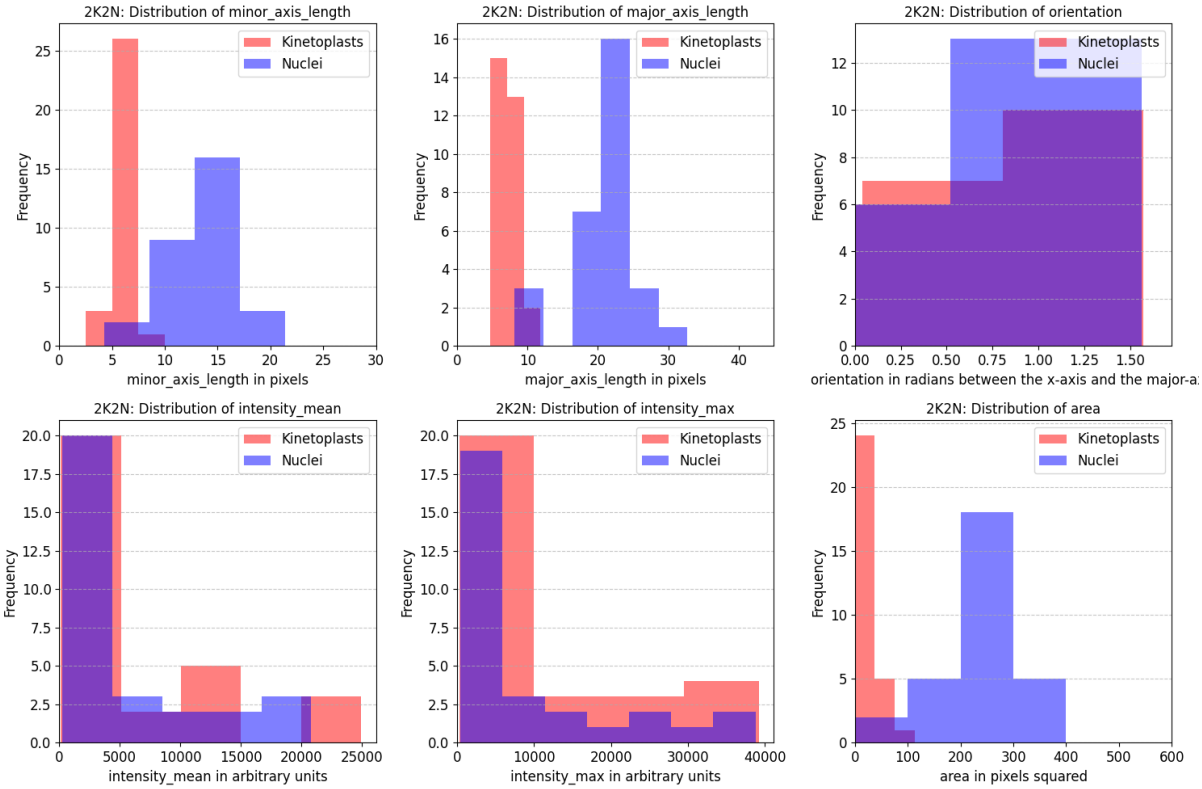

Figure A.19 – A plot with six subplots comparing different properties of 2K2N cells

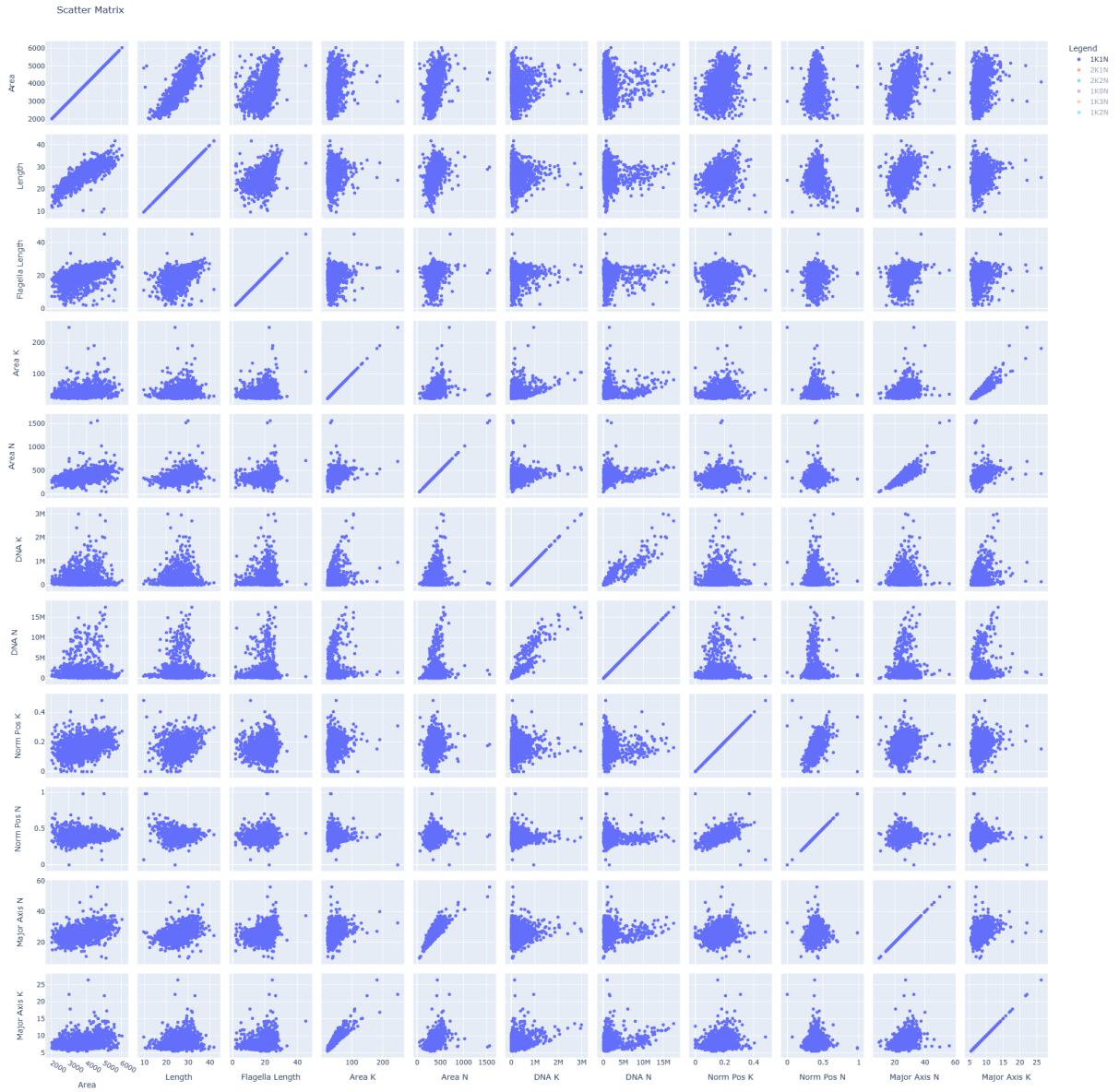

Figure A.20 – Multivariate Scatter Matrix for comparing different properties of 1K1N cells.

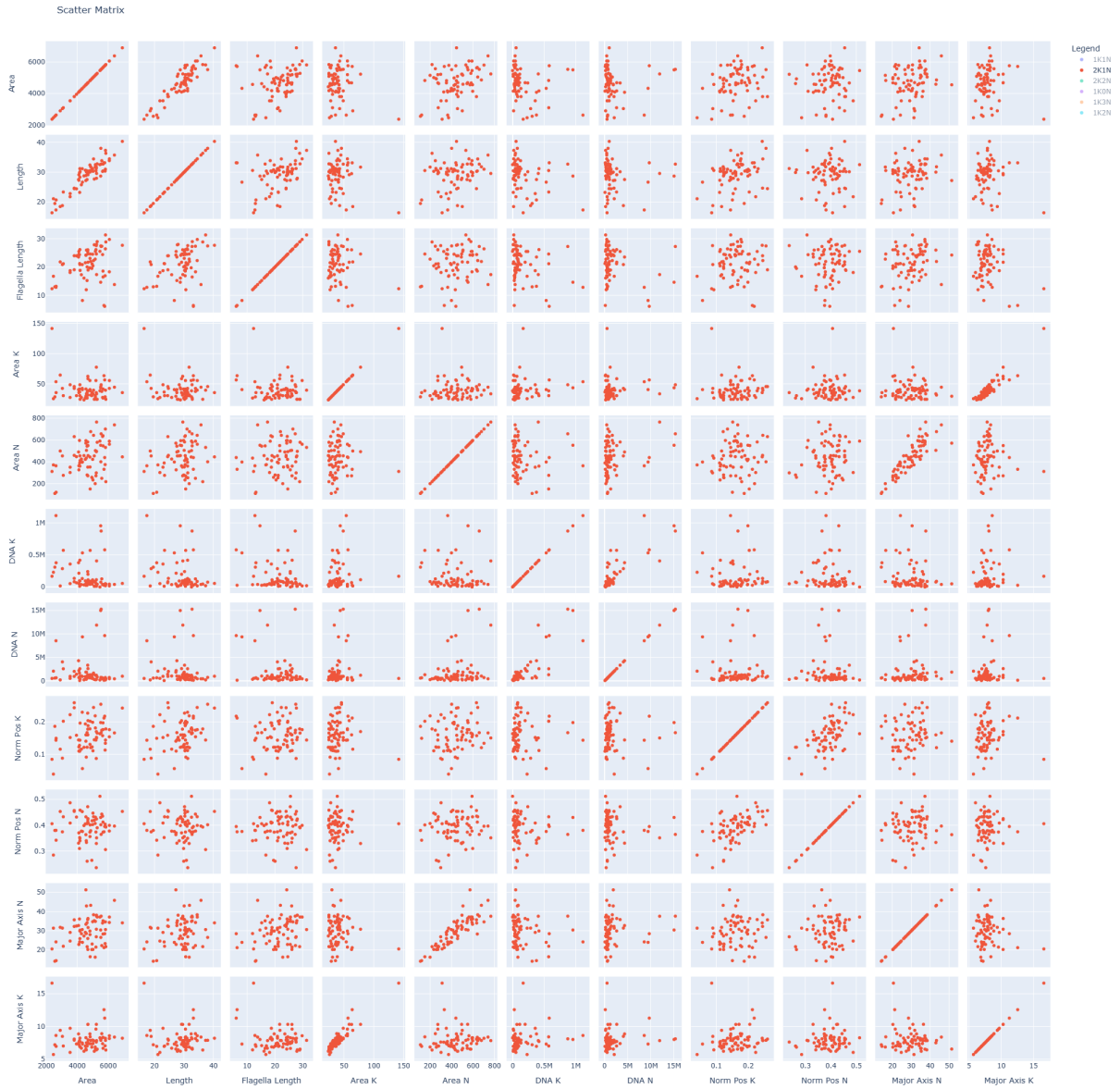

Figure A.21 – Multivariate Scatter Matrix for comparing different properties of 2K1N cells.

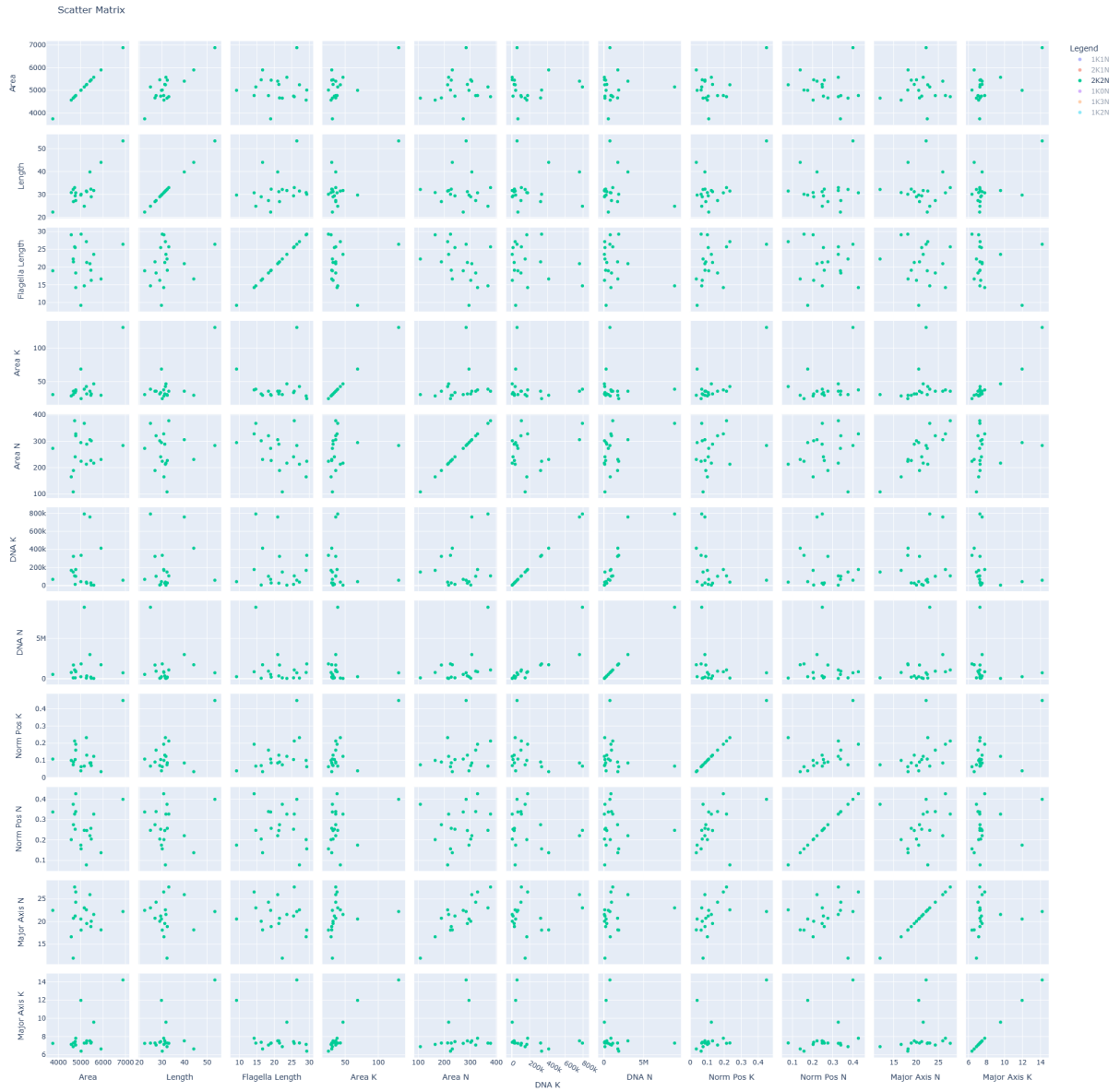

Figure A.22 – Multivariate Scatter Matrix for comparing different properties of 2K2N cells.

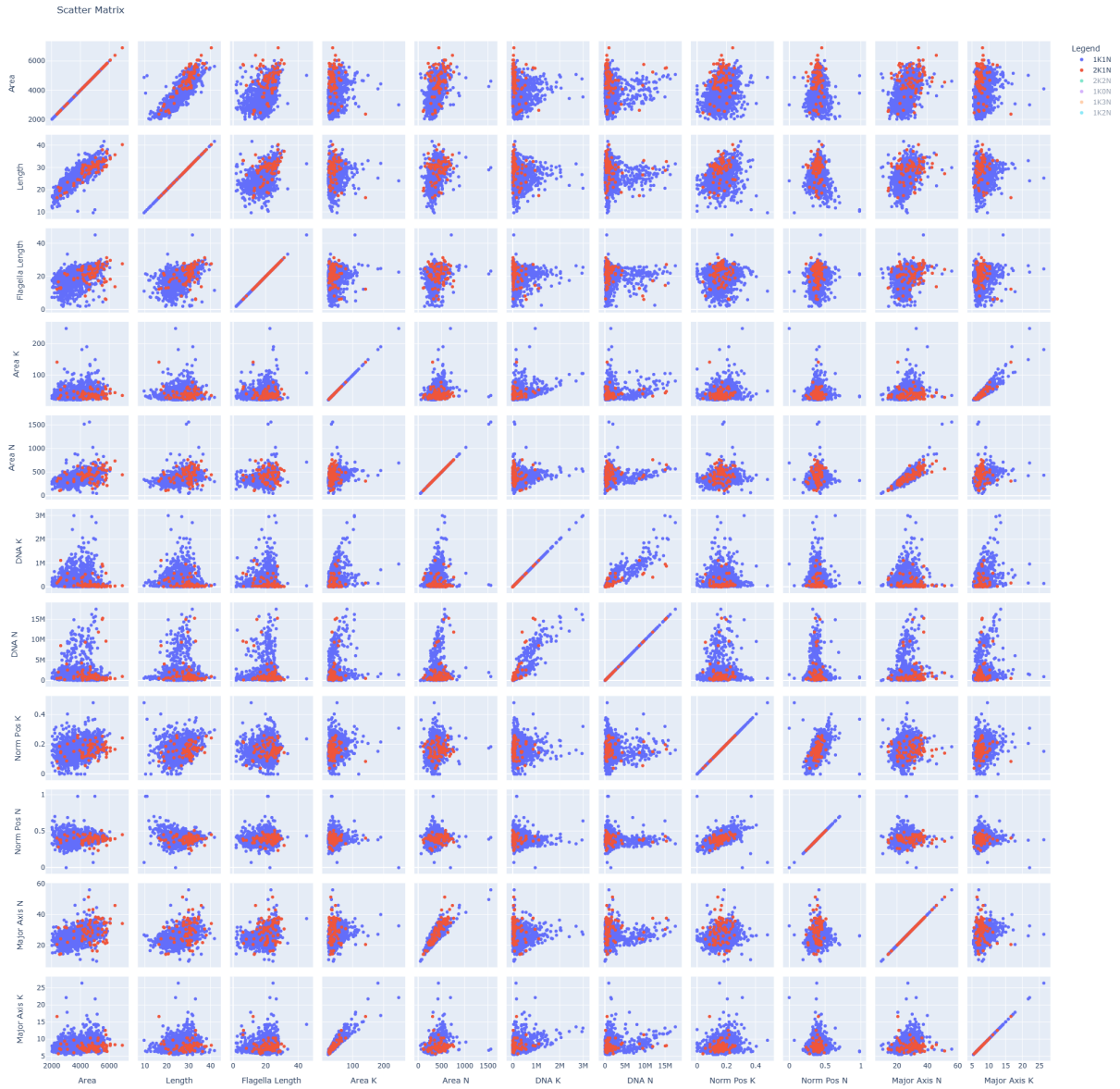

Figure A.23 – Multivariate Scatter Matrix for comparing different properties of 1K1N and 2K1N cells.

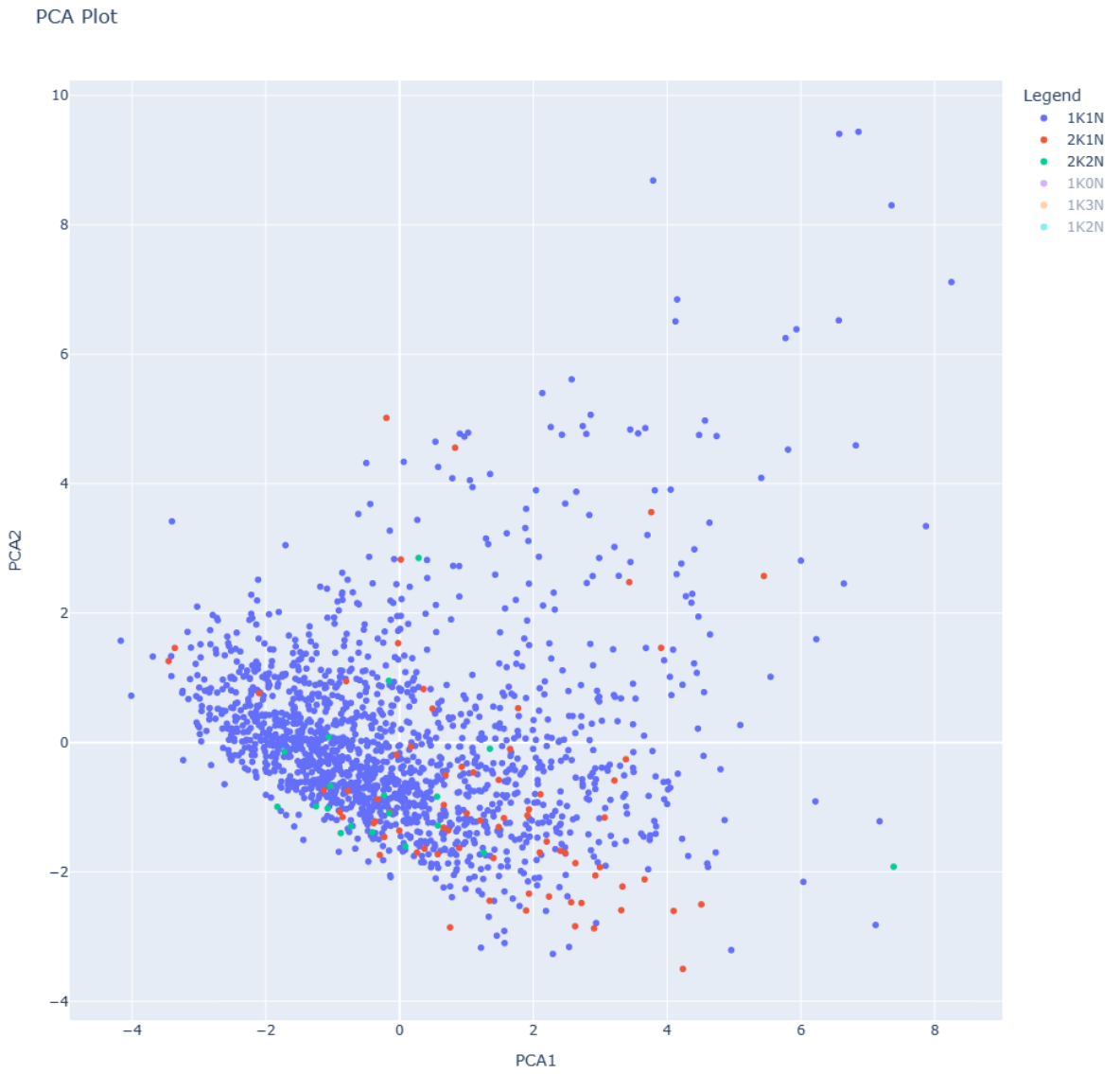

Figure A.24 – Scatterplot showing the two principal components of the standardised features from the scatter matrices for 1K1N, 2K1N and 2K2N cells.

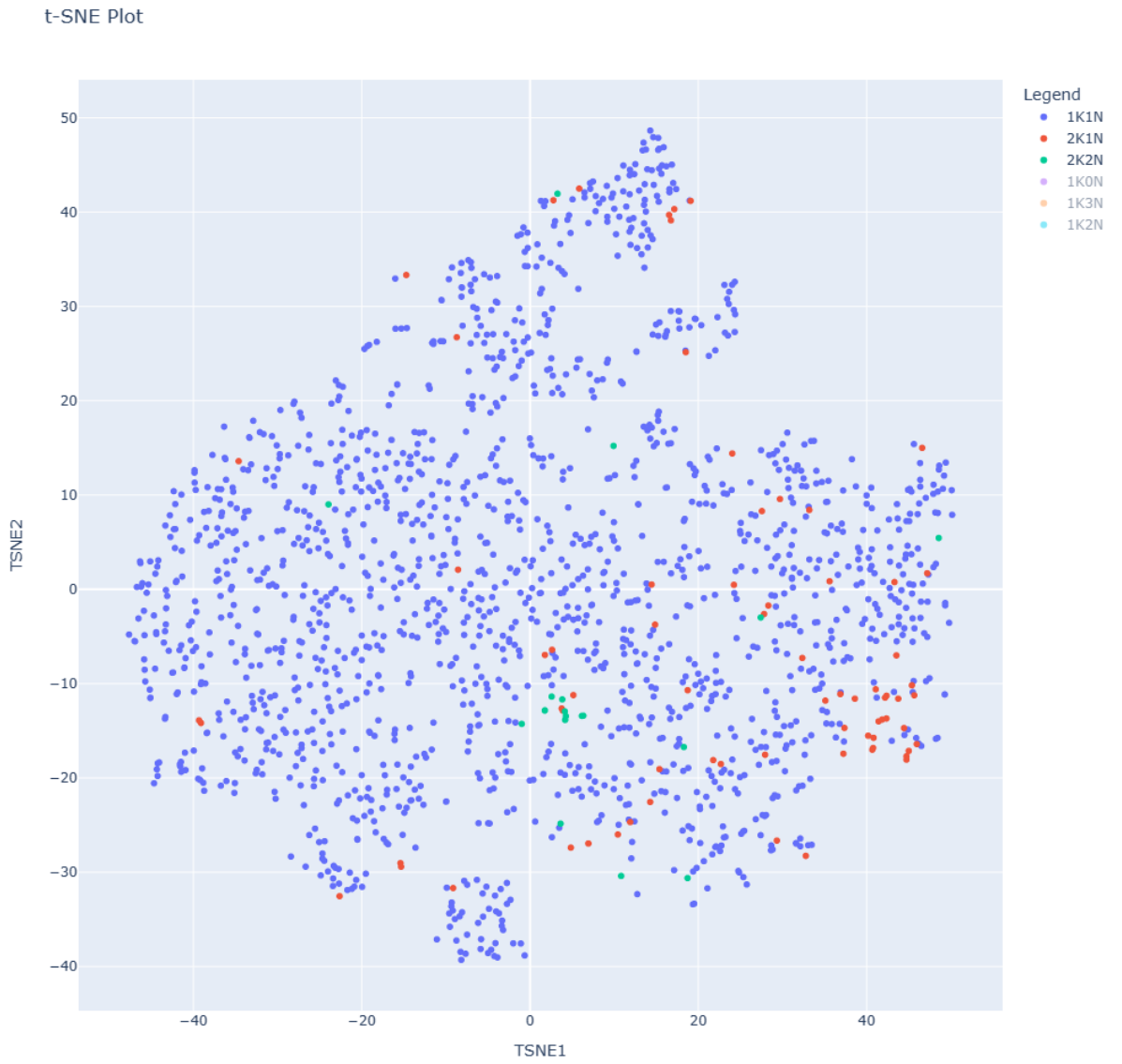

Figure A.25 – Scatterplot showing t-SNE embeddings of standardised features from the scatter matrices for 1K1N, 2K1N, and 2K2N cells.

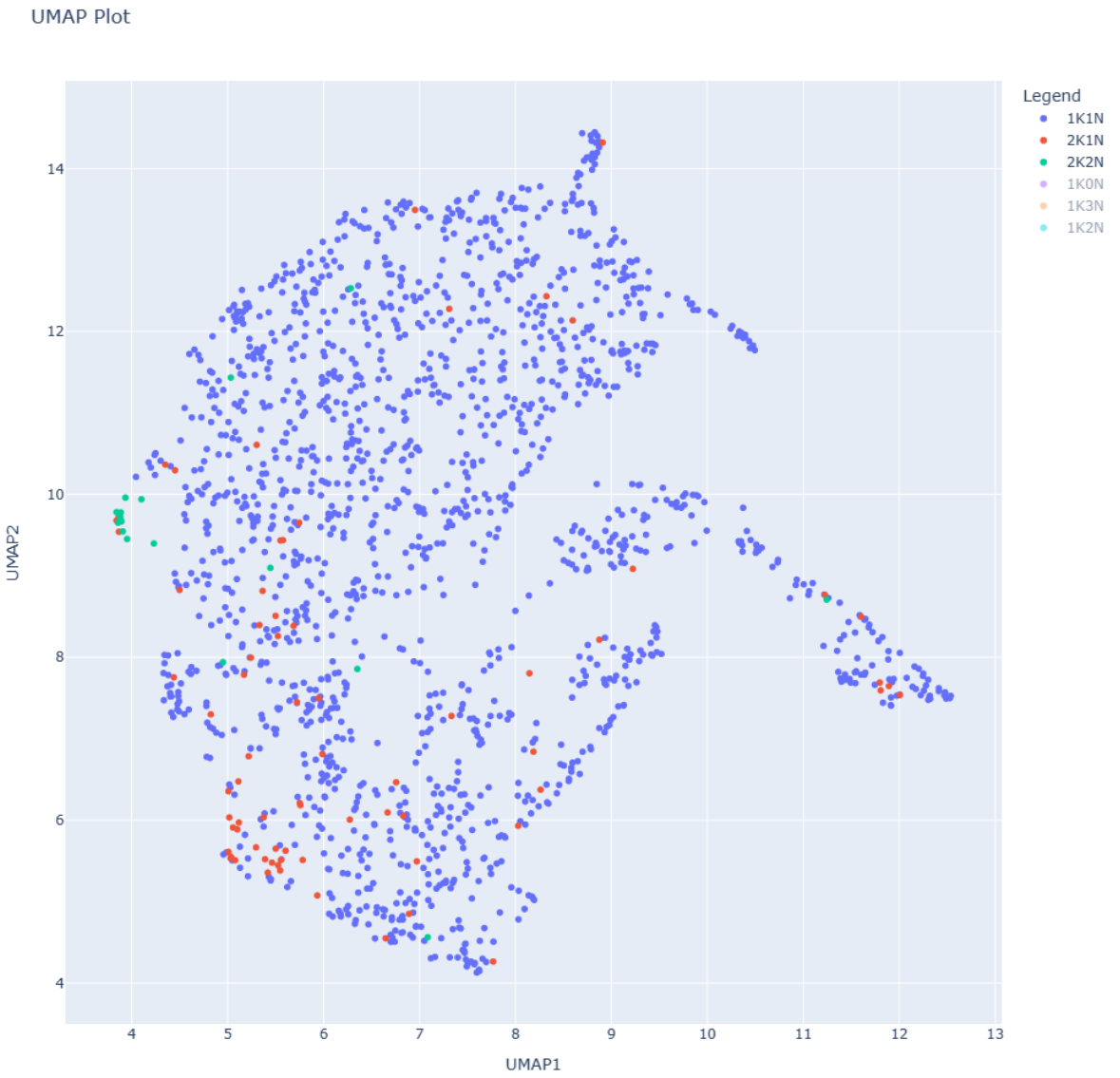

Figure A.26 – Scatterplot showing UMAP embeddings of standardised features from the scatter matrices for 1K1N, 2K1N, and 2K2N cells.

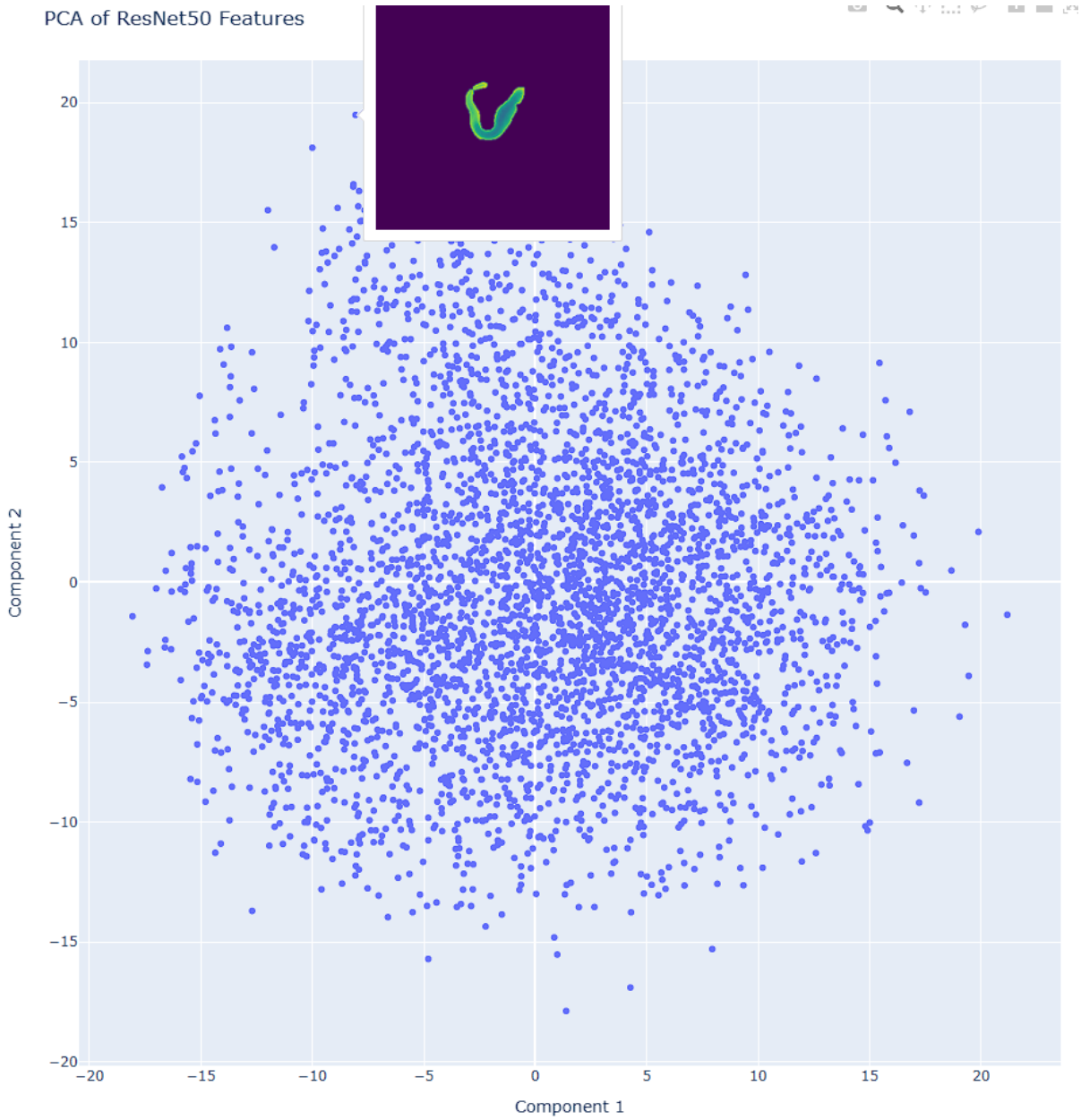

Figure A.27 – Visualisation of the two most important components of the PCA of the ResNet-50 feature Vector. Highlighted in this image is the cell with the largest component two of the PCA.

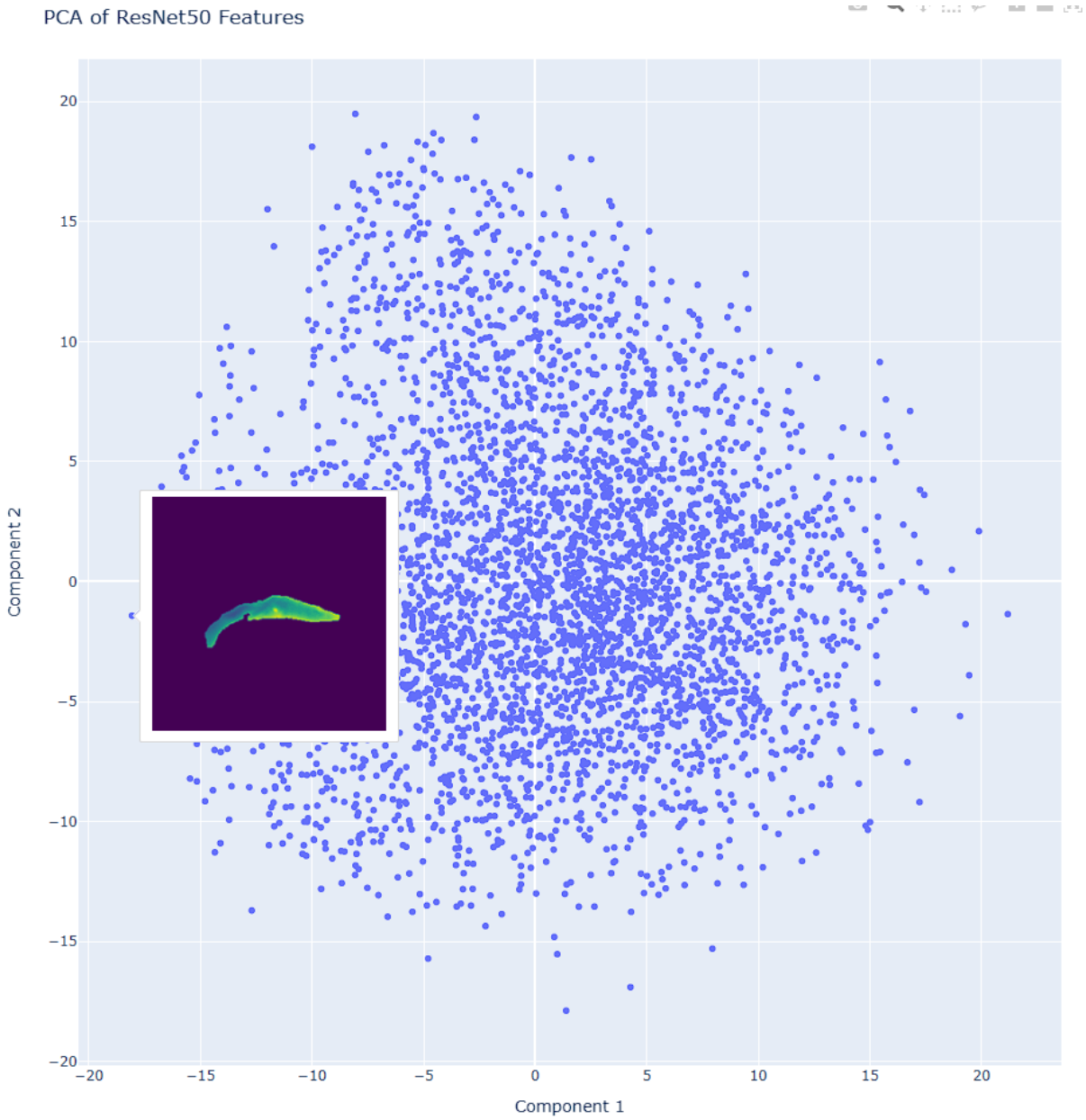

Figure A.28 – Visualisation of the two most important components of the PCA of the ResNet-50 feature Vector. Highlighted in this image is the cell with the smallest component one of the PCA.

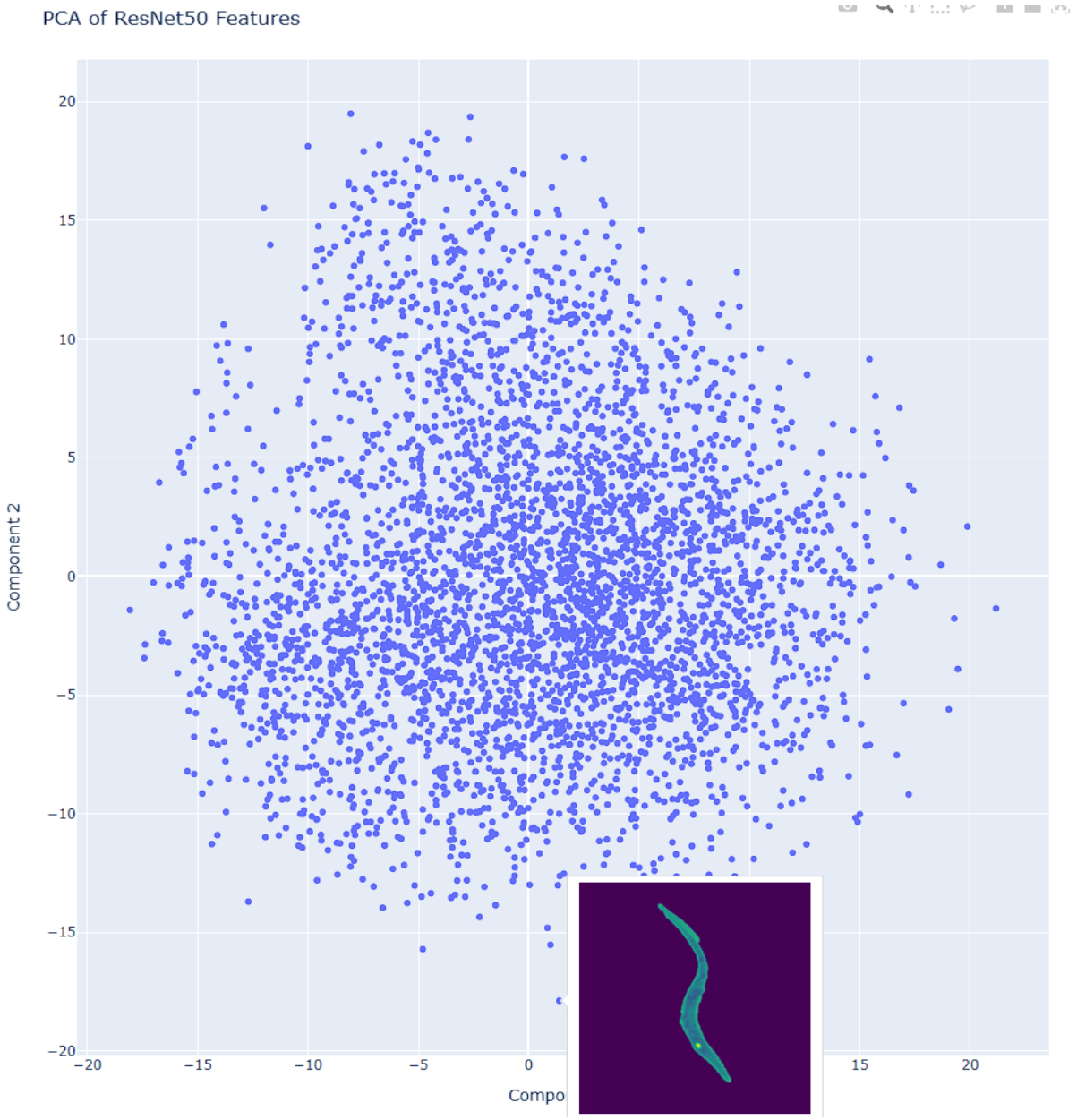

Figure A.29 – Visualisation of the two most important components of the PCA of the ResNet-50 feature Vector. Highlighted in this image is the cell with the smallest component two of the PCA.

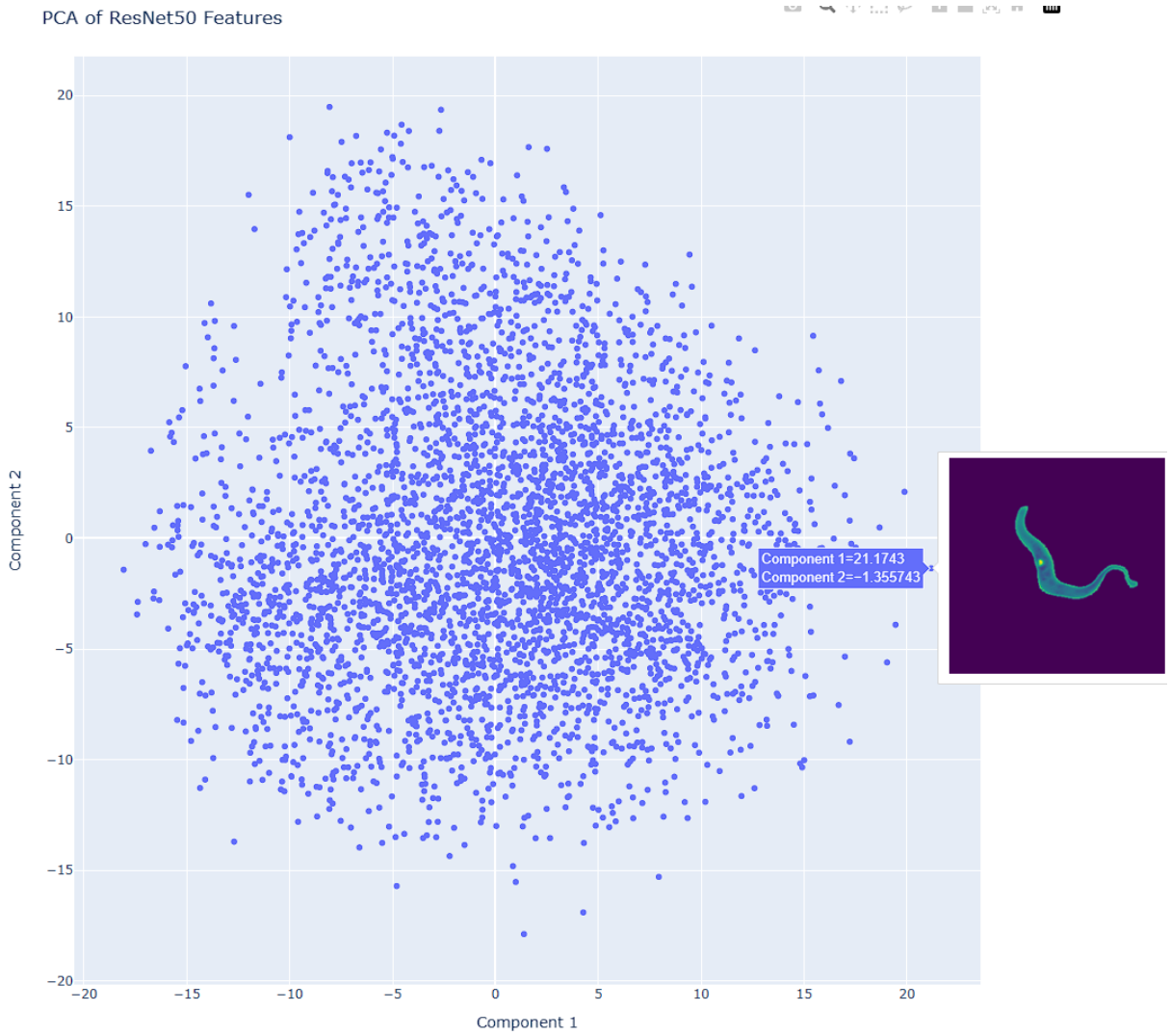

Figure A.30 – Visualisation of the two most important components of the PCA of the ResNet-50 feature Vector. Highlighted in this image is the cell with the largest component one of the PCA.

Figure A.31 – Visualisation of the two most important components of the PCA of the ResNet-50 feature Vector. Highlighted in this image is a cell with both components close to zero.

Table A.1 – The table of the selected genes for the flagella dataset with the columns GeneID, Type, Form and URL.

Gene ID	Type	Form	URL
Tb927.10.11300	n	procyclic	`http://preview.tryptag.org/?query=Tb927.10.11300`
Tb927.8.5000	n	procyclic	`http://preview.tryptag.org/?query=Tb927.8.5000`
Tb927.8.4990	n	procyclic	`http://preview.tryptag.org/?query=Tb927.8.4990`
Tb927.10.9570	c	procyclic	`http://preview.tryptag.org/?query=Tb927.10.9570`
Tb927.6.4140	c	procyclic	`http://preview.tryptag.org/?query=Tb927.6.4140`
Tb927.11.14880	c	procyclic	`http://preview.tryptag.org/?query=Tb927.11.14880`
Tb927.8.4970	c	procyclic	`http://preview.tryptag.org/?query=Tb927.8.4970`
Tb927.8.1550	c	procyclic	`http://preview.tryptag.org/?query=Tb927.8.1550`
Tb927.11.15100	c	procyclic	`http://preview.tryptag.org/?query=Tb927.11.15100`
Tb927.3.4290	n	procyclic	`http://preview.tryptag.org/?query=Tb927.3.4290`

Table A.2 – Statistics for Kinetoplasts with different KN configurations

Configuration	Property	Number of Samples	Mean	Standard Deviation
KN	minor_axis_length	2977	6.117863	1.128607
KN	major_axis_length	2977	8.170174	2.109085
KN	orientation	2977	0.035280	0.985719
KN	intensity_mean	2977	6869.816854	6683.709997
KN	intensity_max	2977	11287.726570	10299.182123
KN	area	2977	42.668122	19.840344
KKN	minor_axis_length	438	6.376605	1.570224
KKN	major_axis_length	438	8.356584	2.776793
KKN	orientation	438	0.106304	0.946963
KKN	intensity_mean	438	5737.336808	6260.974746
KKN	intensity_max	438	9661.328767	10040.225582
KKN	area	438	46.721461	30.940066
KKNN	minor_axis_length	30	5.733498	0.972028
KKNN	major_axis_length	30	7.422211	1.020546
KKNN	orientation	30	0.220625	0.941206
KKNN	intensity_mean	30	5769.574530	6864.795986
KKNN	intensity_max	30	10173.700000	11762.875054
KKNN	area	30	35.633333	11.979102

Table A.3 – Statistics for Nuclei with different KN configurations

Configuration	Property	Number of Samples	Mean	Standard Deviation
KN	minor_axis_length	2977	17.719891	2.850791
KN	major_axis_length	2977	26.119837	5.001020
KN	orientation	2977	0.017756	0.900847
KN	intensity_mean	2977	4819.845473	5596.776821
KN	intensity_max	2977	8105.871347	8939.308932
KN	area	2977	381.799798	123.466726
KKN	minor_axis_length	219	18.483859	3.676592
KKN	major_axis_length	219	32.358157	8.826989
KKN	orientation	219	0.051534	0.904021
KKN	intensity_mean	219	5866.511463	6730.025697
KKN	intensity_max	219	10196.378995	10866.630535
KKN	area	219	490.899543	175.359797
KKNN	minor_axis_length	30	13.369848	2.971094
KKNN	major_axis_length	30	20.905398	4.513790
KKNN	orientation	30	0.006075	1.105021
KKNN	intensity_mean	30	5174.097608	6345.971312
KKNN	intensity_max	30	8824.333333	11198.964962
KKNN	area	30	233.800000	72.691311

Table A.4 – Statistics for Kinetoplasts for all configurations

Property	Number of Samples	Mean	Standard Deviation
minor_axis_length	4521	6.266201	1.489727
major_axis_length	4521	8.330654	2.501235
orientation	4521	0.055207	0.980217
intensity_mean	4521	6515.722264	6551.818192
intensity_max	4521	10750.479761	10189.148691
area	4521	45.487945	28.869420

Table A.5 – Statistics for Nuclei for all configurations

Property	Number of Samples	Mean	Standard Deviation
minor_axis_length	4271	17.711442	3.274604
major_axis_length	4271	26.400503	6.603265
orientation	4271	0.022260	0.909284
intensity_mean	4271	4899.279762	5751.140751
intensity_max	4271	8312.945446	9251.621770
area	4271	389.575978	179.416108

Table A.6 – PCA1 Feature Contributions

Feature	Contribution to PCA1%
Area of the Nucleus	15.500934
Cell Area	14.846747
Major Axis Length of the Nucleus	13.774215
Major Axis Length of the Kinetoplast	11.858795
Corrected Cell Length	11.184351
Area of the Kinetoplast	10.266544
Corrected Flagellum Length	6.467653
Normalised Kinetoplast Position	5.930460
Summed DNA content inside the Nucleus	3.864630
Summed DNA content inside the Kinetoplast	3.646111
Normalised Nucleus Position	2.659561

Table A.7 – PCA2 Feature Contributions

Feature	Contribution to PCA2%
Summed DNA content inside the Kinetoplast	29.692792
Summed DNA content inside the Nucleus	26.472459
Corrected Cell Length	10.598723
Area of the Kinetoplast	8.357192
Cell Area	7.955162
Major Axis Length of the Kinetoplast	7.235855
Corrected Flagellum Length	5.566028
Major Axis Length of the Nucleus	1.537946
Normalised Kinetoplast Position	1.341390
Normalised Nucleus Position	0.893476
Area of the Nucleus	0.348976

List of Figures

List of Tables

Bibliography

[Ach+23] OpenAI Josh Achiam et al. “GPT-4 Technical Report”. In: 2023. URL: https://arxiv.org/abs/2303.08774 (cit. on p. 1).

[Ben+17] Corinna Benz, Frank Dondelinger, Paul G. McKean, and Michael D. Urbaniak. “Cell cycle synchronisation of Trypanosoma brucei by centrifugal counter-flow elutriation reveals the timing of nuclear and kinetoplast DNA replication”. en. In: *Scientific Reports* 7.1 (Dec. 2017), p. 17599. DOI: 10.1038/s41598-017-17779-z. URL: https://www.nature.com/articles/s41598-017-17779-z (visited on 08/29/2023) (cit. on p. 11).

[Bil+23] Karen Billington, Clare Halliday, Ross Madden, Philip Dyer, Amy Rachel Barker, Flávia Fernandes Moreira-Leite, Mark Carrington, Sue Vaughan, Christiane Hertz-Fowler, Samuel Dean, Jack Daniel Sunter, Richard John Wheeler, and Keith Gull. “Genome-wide subcellular protein map for the flagellate parasite Trypanosoma brucei”. en. In: *Nature Microbiology* 8.3 (Feb. 2023), pp. 533–547. DOI: 10.1038/s41564-022-01295-6. URL: https://www.nature.com/articles/s41564-022-01295-6 (visited on 08/29/2023) (cit. on pp. 16, 20, 22, 24).

[Che+20] Long Chen, Martin Strauch, Matthias Daub, Xiaochen Jiang, Marcus Jansen, Hans-Georg Luigs, Susanne Schultz-Kuhlmann, Stefan Krussel, and Dorit Merhof. “A CNN Framework Based on Line Annotations for Detecting Nematodes in Microscopic Images”. In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. Iowa City, IA, USA: IEEE, Apr. 2020, pp. 508–512. DOI: 10.1109/ISBI45749.2020.9098465. URL: https://ieeexplore.ieee.org/document/9098465/ (visited on 08/29/2023) (cit. on p. 12).

[Cro+18] Thomas W.M. Crozier, Michele Tinti, Richard J. Wheeler, Tony Ly, Michael A.J. Ferguson, and Angus I. Lamond. “Proteomic Analysis of the Cell Cycle of Procylic Form Trypanosoma brucei”. en. In: *Molecular & Cellular Proteomics* 17.6 (June 2018), pp. 1184–1195. DOI: 10.1074/mcp.RA118.000650. URL: https://linkinghub.elsevier.com/retrieve/pii/S1535947620322404 (visited on 08/29/2023) (cit. on p. 11).

[DB23] Maurice Deserno and Katarzyna Bozek. “WormSwin: Instance segmentation of C. elegans using vision transformer”. en. In: *Scientific Reports* 13.1 (July 2023), p. 11021. DOI: 10.1038/s41598-023-38213-7. URL: https://www.nature.com/articles/s41598-023-38213-7 (visited on 08/29/2023) (cit. on pp. 3, 12).

[Dos+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV] (cit. on p. 1).

[DSW17] Samuel Dean, Jack D. Sunter, and Richard J. Wheeler. "TrypTag.org: A Trypanosome Genome-wide Protein Localisation Resource". en. In: *Trends in Parasitology* 33.2 (Feb. 2017), pp. 80–82. DOI: 10.1016/j.pt.2016.10.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S1471492216301921 (visited on 08/29/2023) (cit. on pp. 2, 20).

[Eul+17] Philipp Eulenberg, Niklas Köhler, Thomas Blasi, Andrew Filby, Anne E. Carpenter, Paul Rees, Fabian J. Theis, and F. Alexander Wolf. "Reconstructing cell cycle and disease progression using deep learning". en. In: *Nature Communications* 8.1 (Sept. 2017), p. 463. DOI: 10.1038/s41467-017-00623-3. URL: https://www.nature.com/articles/s41467-017-00623-3 (visited on 08/29/2023) (cit. on pp. 12 sq., 44 sq., 62).

[Hal+19] Clare Halliday, Karen Billington, Ziyin Wang, Ross Madden, Samuel Dean, Jack Daniel Sunter, and Richard John Wheeler. "Cellular landmarks of Trypanosoma brucei and Leishmania mexicana". en. In: *Molecular and Biochemical Parasitology* 230 (June 2019), pp. 24–36. DOI: 10.1016/j.molbiopara.2018.12.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0166685118302251 (visited on 08/29/2023) (cit. on pp. 20, 22).

[Heb+21] Laetitia Hebert, Tosif Ahamed, Antonio C. Costa, Liam O'Shaughnessy, and Greg J. Stephens. "WormPose: Image synthesis and convolutional networks for pose estimation in C. elegans". en. In: *PLOS Computational Biology* 17.4 (Apr. 2021). Ed. by Dina Schneidman-Duhovny, e1008914. DOI: 10.1371/journal.pcbi.1008914. URL: https://dx.plos.org/10.1371/journal.pcbi.1008914 (visited on 08/29/2023) (cit. on pp. 3, 12).

[HW66] Cecil A. Hoare and Franklin G. Wallace. "Developmental Stages of Trypanosomatid Flagellates: a New Terminology". en. In: *Nature* 212.5068 (Dec. 1966), pp. 1385–1386. DOI: 10.1038/2121385a0. URL: https://www.nature.com/articles/2121385a0 (visited on 08/29/2023) (cit. on p. 6).

[Jum+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A.A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596 (7873 Aug. 2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2 (cit. on p. 1).

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (cit. on p. 1).

[LY09] Dongju Liu and Jian Yu. "Otsu method and K-means". In: vol. 1. 2009, pp. 344–349. DOI: 10.1109/HIS.2009.74 (cit. on p. 25).

[Min+11] Neha Minocha, Devanand Kumar, Kalpana Rajanala, and Swati Saha. "Kinetoplast Morphology and Segregation Pattern as a Marker for Cell Cycle Progression in Leishmania donovani1: KINETOPLAST AS CELL CYCLE STAGE MARKER". en. In: *Journal of Eukaryotic Microbiology* 58.3 (May 2011), pp. 249–253. DOI: 10.1111/j.1550-7408.2011.00539.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1550-7408.2011.00539.x (visited on 08/29/2023) (cit. on p. 1).

[Rob+] Derrick R. Robinson, Trevor Sherwin, Aspasia Ploubidou, Edward H. Byard, and Keith Gull. *Microtubule Polarity and Dynamics in the Control of Organelle Positioning, Segregation, and Cytokinesis in the Trypanosome Cell Cycle*. URL: http://rupress.org/jcb/article-pdf/128/6/1163/1478521/1163.pdf (cit. on pp. 10, 60).

[Rou87] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". en. In: *Journal of Computational and Applied Mathematics* 20 (Nov. 1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7. URL: https://linkinghub.elsevier.com/retrieve/pii/0377042787901257 (visited on 08/29/2023) (cit. on p. 26).

[SG13] Christoph Sommer and Daniel W. Gerlich. "Machine learning in cell biology – teaching computers to recognize phenotypes". en. In: *Journal of Cell Science* (Jan. 2013), jcs.123604. DOI: 10.1242/jcs.123604. URL: https://journals.biologists.com/jcs/article/doi/10.1242/jcs.123604/263567/Machine-learning-in-cell-biology-teaching (visited on 08/29/2023) (cit. on pp. 11, 26).

[Sha+08] Reuben Sharma, Lori Peacock, Eva Gluenz, Keith Gull, Wendy Gibson, and Mark Carrington. "Asymmetric Cell Division as a Route to Reduction in Cell Length and Change in Cell Morphology in Trypanosomes". en. In: *Protist* 159.1 (Jan. 2008), pp. 137–151. DOI: 10.1016/j.protis.2007.07.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S1434461007000508 (visited on 08/28/2023) (cit. on p. 1).

[SHC08] T. Nicolai Siegel, Doeke R. Hekstra, and George A.M. Cross. "Analysis of the Trypanosoma brucei cell cycle by quantitative DAPI imaging". en. In: *Molecular and Biochemical Parasitology* 160.2 (Aug. 2008), pp. 171–174. DOI: 10.1016/j.molbiopara.2008.04.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S016668510800090X (visited on 08/29/2023) (cit. on pp. 10 sq.).

[Tho53] Robert L. Thorndike. "Who belongs in the family?" en. In: *Psychometrika* 18.4 (Dec. 1953), pp. 267–276. DOI: 10.1007/BF02289263. URL: http://link.springer.com/10.1007/BF02289263 (visited on 08/29/2023) (cit. on p. 26).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All You Need". In: 2017. URL: https://arxiv.org/pdf/1706.03762.pdf (cit. on p. 1).

[WG90] R. Woodward and K. Gull. "Timing of nuclear and kinetoplast DNA replication and early morphological events in the cell cycle of *Trypanosoma brucei*". en. In: *Journal of Cell Science* 95.1 (Jan. 1990), pp. 49–57. DOI: 10.1242/jcs.95.1.49. URL: https://journals.biologists.com/jcs/article/95/1/49/60576/Timing-of-nuclear-and-kinetoplast-DNA-replication (visited on 08/29/2023) (cit. on p. 7).

[WGG11] Richard J. Wheeler, Eva Gluenz, and Keith Gull. “The cell cycle of Leishmania: morphogenetic events and their implications for parasite biology: The cell cycle of Leishmania”. en. In: *Molecular Microbiology* 79.3 (Feb. 2011), pp. 647–662. DOI: 10.1111/j.1365-2958.2010.07479.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2958.2010.07479.x (visited on 08/29/2023) (cit. on pp. 10, 22).

[WGG12] Richard J Wheeler, Keith Gull, and Eva Gluenz. “Detailed interrogation of trypanosome cell biology via differential organelle staining and automated image analysis”. en. In: *BMC Biology* 10.1 (Dec. 2012), p. 1. DOI: 10.1186/1741-7007-10-1. URL: https://bmcbiol.biomedcentral.com/articles/10.1186/1741-7007-10-1 (visited on 08/29/2023) (cit. on pp. 2, 9 sq., 16 sq., 39).

[WGS19] Richard J. Wheeler, Keith Gull, and Jack D. Sunter. “Coordination of the Cell Cycle in Trypanosomes”. en. In: *Annual Review of Microbiology* 73.1 (Sept. 2019), pp. 133–154. DOI: 10.1146/annurev-micro-020518-115617. URL: https://www.annualreviews.org/doi/10.1146/annurev-micro-020518-115617 (visited on 08/29/2023) (cit. on pp. 6–8).

[Whe+13] Richard J. Wheeler, Nicole Scheumann, Bill Wickstead, Keith Gull, and Sue Vaughan. “Cytokinesis in Trypanosoma brucei differs between bloodstream and tsetse trypomastigote forms: implications for microtubule-based morphogenesis and mutant analysis”. en. In: *Molecular Microbiology* 90.6 (Dec. 2013), pp. 1339–1355. DOI: 10.1111/mmi.12436. URL: https://onlinelibrary.wiley.com/doi/10.1111/mmi.12436 (visited on 08/29/2023) (cit. on p. 1).

[Zha+17] Mengdi Zhao, Jie An, Haiwen Li, Jiazhi Zhang, Shang-Tong Li, Xue-Mei Li, Meng-Qiu Dong, Heng Mao, and Louis Tao. “Segmentation and classification of two-channel C. elegans nucleus-labeled fluorescence images”. en. In: *BMC Bioinformatics* 18.1 (Dec. 2017), p. 412. DOI: 10.1186/s12859-017-1817-3. URL: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1817-3 (visited on 08/29/2023) (cit. on pp. 12, 25, 41, 43).