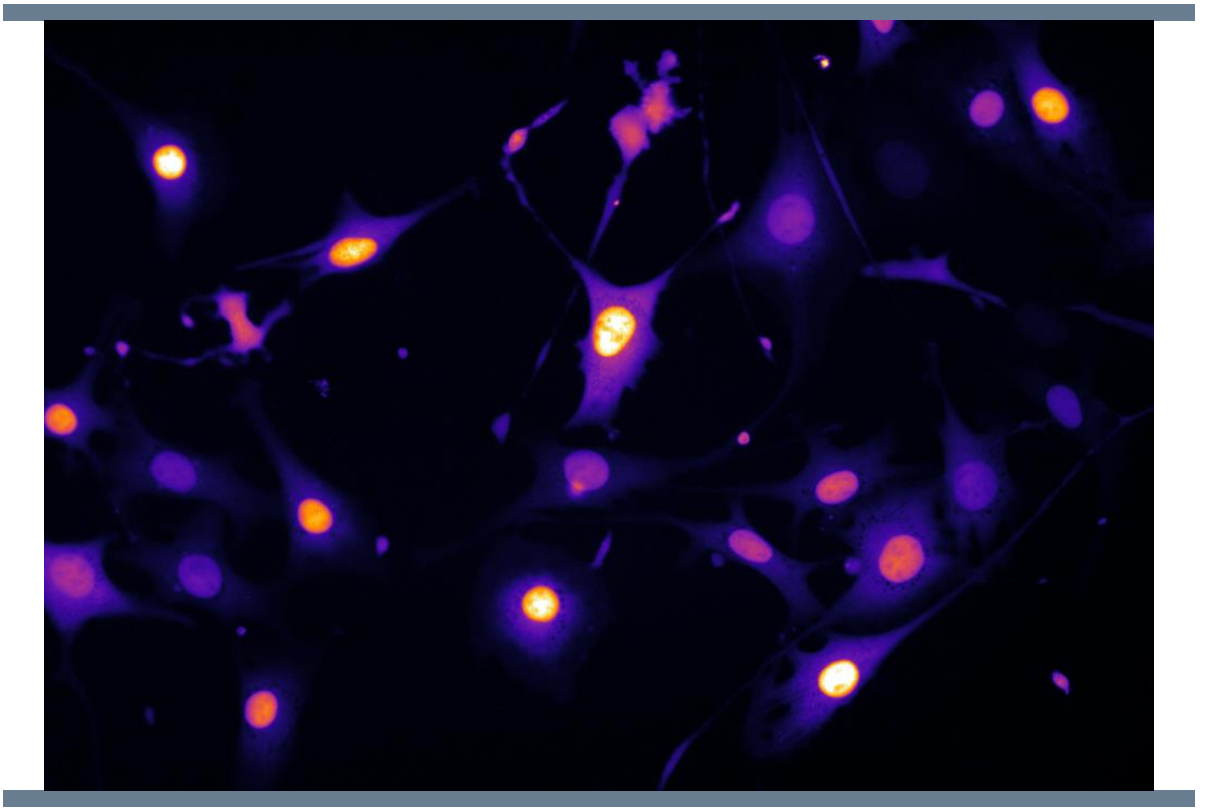


Cell Classifier for Biological Research

Separating the bad from the good

Bachelor Thesis

Degree program:	Bachelor of Computer Science
Author:	Steven Cardini
Thesis advisor:	Prof. Dr. Bernhard Anrig
Expert:	Dr. René Bach
Project partner:	Dr. Maciej Dobrzyński, Cellular Dynamics Lab of Prof. Dr. Olivier Pertz, University of Berne
Date:	18.01.2018

Abstract

Time-lapse microscopy extracts multiple measurements of single cells as a function of time (time series). However, the cell segmentation procedure in time-lapse microscopy is error-prone due to various reasons, such as false image segmentation or cells undergoing division or apoptosis. Thus, statistical methods are needed to detect and remove falsely segmented cells from data sets to avoid biased results. So far, no well-established methods that reliably identify these falsely segmented outlier time series have been proposed. In this work, we aimed to close this gap by evaluating three approaches to identifying outlier time series: (1) principal component analysis (PCA), (2) a sliding window procedure, and (3) a machine learning approach with decision tree classifiers. We found that the PCA approach is not reliable, since it produces too many false positives (39%). The other two approaches are better suited for the problem: They exhibited accuracy rates and true positive rates of around 90%, and false positive rates of 10-20%, which are satisfactory values. The results presented in this work expand our knowledge regarding reliable methods to identify outlier time series and will contribute to large-scale unsupervised time series processing from single-cell microscopy experiments.

Contents

1 Introduction

The study of cellular heterogeneity is a rapidly developing field in the biological sciences. When cells within a population of genetically identical cells are studied individually, there is usually some degree of variability among them [1]. For example, identical cells can respond differently to external stimuli such as activators or inhibitors of cellular signaling pathways. One central unanswered question is how much of this heterogeneity results from the random nature of biochemical reactions at the level of a single cell, and how much can be explained by external factors such as a cell's microenvironment.

The Cellular Dynamics Lab at the University of Bern is studying cellular heterogeneity in the context of the MAPK (mitogen activated protein kinase) pathway inside human cells. Figure 1 illustrates this pathway. It can be activated by growth factors such as epidermal growth factor (EGF) [2], fibroblast growth factor (FGF) [3] or nerve growth factor (NGF) [4] that bind to receptors on the surface of the cell (for example RTK, see Figure 1). The activation of the pathway leads to a cascade of protein phosphorylation, including Ras, Raf, MEK and ERK (extracellular signal-regulated kinase), which changes the involved proteins into an active state. As a result, the MAPK pathway regulates transcription factors in the nucleus that are associated with cell fate decisions and thereby plays a crucial role in cell proliferation, differentiation or death. A substantial proportion of human cancer types are associated with mutations in proteins of the MAPK pathway [5].

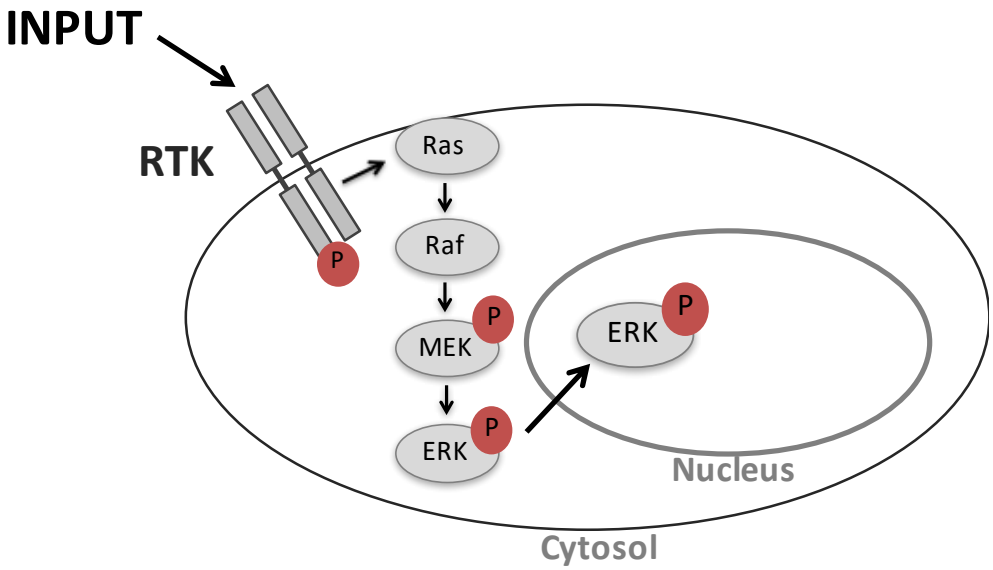


Figure 1: MAPK pathway. Ligands such as growth factors binding to specific cell receptors (here represented as RTK: receptor tyrosine kinase) activate the pathway. Only a subset of the involved proteins is depicted. The protein ERK translocates from the cytosol to the nucleus because of MAPK pathway activation, where it regulates the activity of transcription factors that are associated with cell fate decisions. The illustration is provided by C. Dessauges, Unibe.

When the MAPK pathway is active, ERK gets phosphorylated (=activated) in the course of the signaling cascade in the cytosol and translocates to the nucleus [6]. Variability in ERK activity levels between cells of a population affects the fate of single cells. For example, in many cancer types, the uncontrolled cell proliferation results from mutations that lead to high activity levels of the MAPK pathway. Drug interventions aim to inhibit this activity, however responses to such treatments vary significantly and will thus not always affect targeted cells.

It is not only the static activity level of ERK that determines the cell fate but rather the entire ERK dynamics that matters. PC-12 cells are a classic illustration of this phenomenon. Transient activation of ERK (induced by EGF) leads to cell proliferation, while sustained ERK activity (due to NGF) induces differentiation into a neuron-like cell phenotype. Here, different growth factors induce various ERK dynamics leading to distinct cell fates. However, a recent study from Pertz Lab [7] demonstrated that fates can be manipulated independently of the growth factor. A proliferation-inducing growth factor EGF could also lead to cell differentiation (a different cell fate) if applied in a pulsed manner with a specific frequency. Notably, cell-to-cell variability in ERK dynamic responses could be minimized by pulsed stimulations leading to more coherent dynamic responses and a larger fraction of differentiated cells. Under physiological conditions, cell-to-cell variability is arguably beneficial to maintain balance between proliferation and differentiation; if all cells were to differentiate, there would be no cells left to maintain the population. On the other hand, understanding mechanisms that can reduce variability is crucial for designing efficient treatments targeted at pushing most cells towards a certain fate, e.g. apoptosis in cancer or differentiation in nerve reconstruction. The Pertz Lab is currently investigating these processes at the level of single cells.

In the past, the study of single cell behavior was practically impossible. Instead, techniques from molecular biology like Western blotting were used to study the responses of entire cell populations [8]. As these techniques simply average the response of all cells, important information such as cell-to-cell variability is neglected.

Microscopy has evolved significantly in the past decades. Simple light microscopes evolved into modern electron microscopes and fluorescence microscopes. Time-lapse fluorescence microscopy as well as advances in the field of computer vision have laid the foundation to investigate cellular heterogeneity.

Today it is possible to capture image series of cells under a fluorescence microscope and record their responses to external stimuli over time. To evaluate these experiments, individual cells need to be identified, or segmented, from the image series. There are some software tools that are capable of automatized cell segmentation (for example, see [9]). However, these segmentation techniques are not 100% accurate for all cells. Before data analysis, biologists thus need to invest a significant amount of time excluding falsely segmented single cells from their data sets to avoid biased results.

Considering these limitations, the aim of this project is to evaluate statistical approaches and machine learning algorithms based on their efficiency to identify falsely segmented cells.

2 Problem statement

A subset of the time series that are generated in the experiments cannot be used for data analysis, because they contain false measurements and are thus biased. Reasons for such false measurements include:

- The nucleus of the cell was not properly segmented from the images due to poor signal-to-noise ratio or due to sudden cell movement.
- The cell was undergoing cell division during the experiment.
- The cell was dying (apoptosis) during the experiment.

False segmentation can occur when two neighboring cells are close, such that the tracking algorithm considers two separate objects to be the same in some of the images. Dim objects represent a further problem for accurate segmentation: their outline cannot be identified reliably.

During cell division, single cells divide and produce two separate cells. Tracked objects decrease their size significantly in this case, which causes the time series to change abruptly. On the other hand, before cells die, their volumes compact into a sphere, which has an impact on the time series as well.

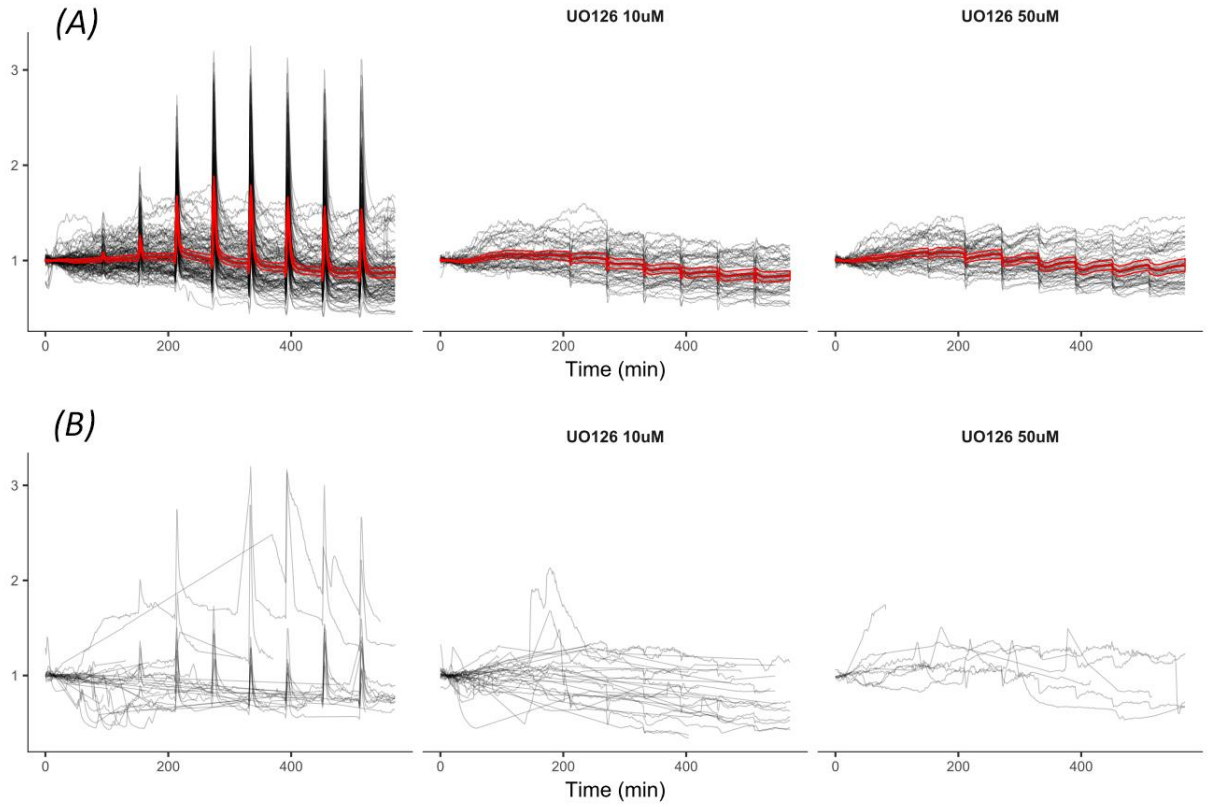


Figure 2: Time series of the 1 / "mean intensity of nuclear ERK-KTR" per nucleus in response to stimulation of the opto-FGF receptor with light at 60' intervals. Time series are normalized individually with respect to their mean behavior between time 0' and 30'. The upper part (A) shows individual cell responses (black lines) and mean cell responses (red lines) over time for good cells in the absence (left) and in the presence of two different concentrations of ERK inhibitor UO126 (middle and right panels). Red lines indicate the population mean and 95% confidence intervals. The lower part (B) shows individual cell responses over time for bad cells for the same conditions. Cell classification (good vs. bad) was done manually by Dr. M. Dobrzynski.

Cells or time series characterized by false measurements are deemed "bad" in this project, while the remaining cells are considered "good". For example, Figure 2 shows time series

of the mean ERK-KTR in single nuclei. Figure 2A contains plots of good time series, while Figure 2B shows examples of bad time series. A subset of the time series of bad cells do not follow the mean trajectory of the population (red line in Figure 2A) and contain abrupt value changes, see Figure 2B.

Such bad time series can be viewed as outliers with respect to the "shape" of their trajectories. Various methods exist to identify outliers in a set of values (for example, see [10]). Time series, however, increase the complexity of this problem by adding a time dimension to the data and thus greatly enlarge the dimensionality of the data. Approaches to identify outlier time series are not well established.

This work focuses on identifying such outlier time series. We investigated three different approaches to identify time series outliers:

- Reduction of time series with principal component analysis and classification according to a few principal components.
- Aggregation of time series to a few measures of its characteristics (features) and classification according to rules learned by machine learning.
- Analysis of time series from measurement point to measurement point to detect outlier values ("sliding window" procedure) without aggregation of the time series.

We compared these approaches with the help of specific performance metrics (see *4.4 Data analysis and evaluation*). For subsequent data analysis, it is important to detect and exclude as many bad time series as possible to reduce bias.

3 Project definition and management

3.1 Previous work

In previous work [11] we made use of various machine learning models to exclude bad time series from the input data sets. We trained and tested these models on aggregated time series. While the accuracy of these models is reasonable (85-90%), the number of false positives can be far too high (>70%).

Furthermore, we found the annotation of the training and test data sets to be problematic. We investigated three approaches to annotate the data sets. Annotation based on linear regression of fluorescence intensity needs a threshold value that differs across experiments. Another method to automatically annotate the data sets is based on identifying outlier values from aggregated time series. However, we concluded that the best method is to annotate the data sets manually.

3.2 Scope of work

The scope was defined in August 2017:

The purpose of this project is to examine statistical approaches and machine learning algorithms to eliminate outlier time series. The various approaches shall be evaluated using manually annotated data sets. The deliverable of the project will be a web application for biologists that facilitates the elimination of poorly segmented cells using one of the previously evaluated approaches. It will be developed using the R Shiny framework and / or the Python programming language. The input of the application will be time series data sets that contain information about cell shape features as well as fluorescence intensities. The user should be able to adjust input parameters that are relevant for the classification and get some adequate feedback to perform quality control after classification.

3.3 Organization

3.3.1 Place of work

The author worked in the Cellular Dynamics Lab of Prof. Dr. O. Pertz approximately once per week. The remaining work on the project was done at BFH or at the author's home.

3.3.2 Work planning

The work schedule was documented in a project journal and made available to the project advisors on the GitLab platform of BFH. Tasks were planned with a Trello board [12]. The tasks in the board were attributed to three states: either "To do", "In progress" or "Done".

3.3.3 Source code

Data analysis was carried out with the R programming language [13] and the open-source integrated development environment RStudio [14]. Source code versions were tracked with Git and made available to the project advisors on the GitLab platform of BFH.

Separate R scripts were written for each of the three approaches. A fourth file (*functions.R*) contains functions that are common to all approaches, such as data import. This file also contains global variables that are shared among all R scripts, such as the names of the

selected variables. Furthermore, it contains the class *dataset_parameters*, which handles data set file paths for data imports.

To ensure that R scripts and functions work as intended, they were tested with small data sets that contained artificial values.

The source code will be made available to the public on GitHub. The author will explain and hand over the code to Dr. M. Dobrzynski by the end of the project.

3.4 Communication

The progress in the project was communicated to the project advisors according to Table 1:

Table 1: Progress Communication

Person	**Mode of communication**
Prof. Dr. B. Anrig	Project meeting at BFH every two weeks Biweekly progress report (email)
Dr. R. Bach	Introductory meeting at BFH Biweekly progress report (email)
Dr. M. Dobrzynski	Discussion in the lab once a week

3.5 Project schedule

3.5.1 Planned schedule

The initial schedule was planned in September 2017 and is listed in Table 2.

Table 2: Planned project schedule

Weeks	**Task**
38 – 41	Feature engineering and statistical approaches to identify outliers: - Check for correlations and reduce the number of variables in the input data sets - Use principal component analysis (PCA) to reduce dimensionality in the input data sets - Evaluate some statistical approaches for outlier detection
42 – 45	Machine learning algorithms: - Revise the manual classification of input data sets to create suitable training and testing data sets - Determine which machine learning models are best suited for the problem - Optimize and compare models
46 – 49	Implementation of a web app: - Implement a web application based on the R Shiny framework - Incorporate the best machine learning algorithm into the web application
50 – 03	Documentation: - Write the final report - Prepare the poster for the final day in Bern and the content for the BFH book - Finalize other work

3.5.2 Alterations to the planned schedule

Weeks 38 to 45 went according to the planned project schedule. Then, the following decisions lead to changes to the original schedule:

- Week 46: The results were not satisfactory enough. Weeks 46 and 47 were used to improve cell classification procedures. The implementation of the web application should be started only in week 48.

- Week 48: New results from the previous two weeks seemed promising, especially new insights from principal component analysis. Discussion with M. Dobrzynski lead to the prioritization of further analysis over the implementation of the web application. Furthermore, writing of the project report was initialized.

- Week 51: Because the insights from the results are more important than the web application, the project focus is set entirely on the project report. The web application will not be part of the project deliverables.

4 Experiment Pipeline

This section describes the pipeline from cell culture to the data sets that were used in this project. All steps before the data analysis were carried out by the experimental team of Pertz lab and are not results from this project.

4.1 Biological Experiments and Microscopy

The biological experiments underlying this work were done using NIH3T3 fibroblast cells [15]. These are mobile human cells that originate from the connective tissue. The cells were genetically engineered to have two biosensors: one marker of the nucleus (NucMem) and one marker of ERK activation (ERK-KTR). ERK-KTR is a fluorescent biosensor that translocates from the nucleus to the cytosol in response to phosphorylation of ERK [16]. Additionally, the cells express a modified version of FGF receptor that is activated by laser light pulses (opto-FGF receptor) [17]. The FGF receptor is an important activator of the MAP kinase pathway [5].

During the experiment, a fluorescence microscope was used to capture image series of two color channels over a period of time: the nuclear channel (NucMem) as well as the channel of ERK-KTR. Images were recorded once per minute. Hence, the position of ERK within the cells was monitored over time.

Some of the cells were treated with UO126, an inhibitor of ERK [18] as a control. Figure 3 illustrates the time series of ERK activation (fluorescence intensity emitted by ERK-KTR within the nuclei of the cells). After stimulation with laser light pulses (dashed vertical lines), the MAP kinase pathway is activated and ERK gets activated (upper plot, red line). If an inhibitor is present, activation of ERK is strongly reduced as expected (lower plot).

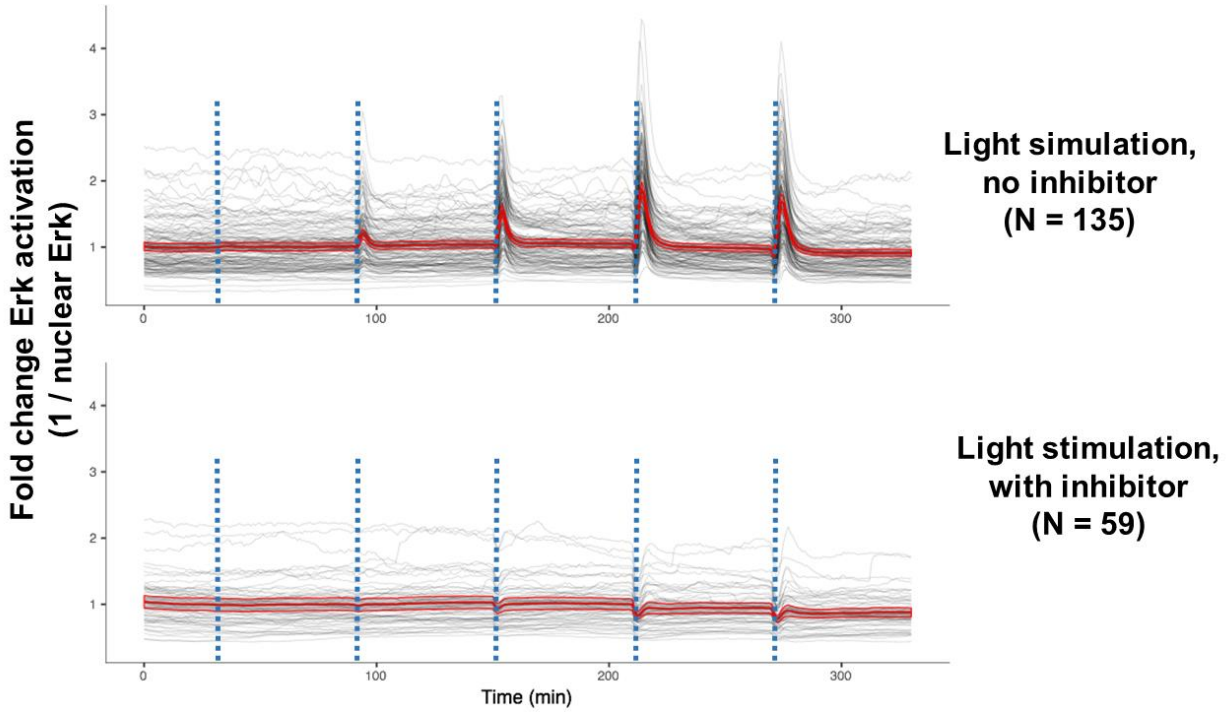


Figure 3: Nuclear ERK activity as function of time in response to light stimulation (indicated by dashed vertical lines) in NIH3T3 fibroblast cells with opto-FGFR receptor. Light pulses of 100ms were applied at 60min intervals with intensities of 2, 10, 20, 50, and 100% of the maximum light intensity. The red lines correspond to the mean of all cell responses. The figure is provided by Dr. M. Dobrzynski, Unibe.

Hence, the result of the first step are image series with two channels: one channel for the nucleus (NucMem) and one for ERK (ERK-KTR). To evaluate the experiments, the image series need to be converted to data that can be analyzed in subsequent steps. To do this, the cells are segmented from the images and various measurements (features) are extracted.

4.2 Cell segmentation and feature extraction

The open source image analysis software CellProfiler [19] was used to segment the cells and nuclei and to extract features. CellProfiler can extract various sorts of shapes, sizes, intensities and texture from objects in the time-lapse microscopy images. The objects can be whole cells or the cell nuclei for example. An illustration of segmented cells is shown in Figure 4.

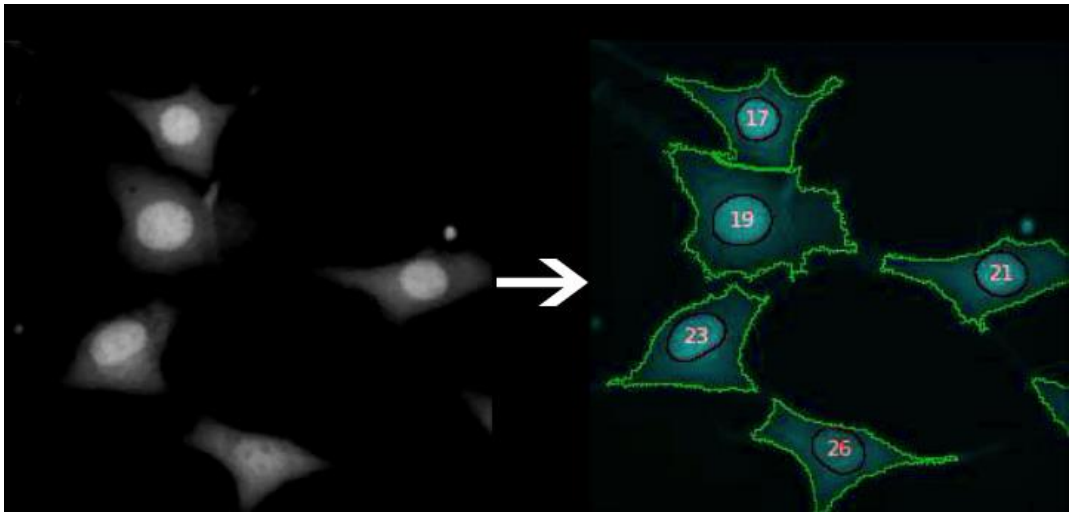


Figure 4: Image segmentation. Shown on the left is the image of fluorescent signal from the ERK-KTR biosensor. The right image shows the result of the image segmentation. The CellProfiler software identifies cell nuclei based on another biosensor (NucMem) expressed only in the nucleus. The cytosol is identified based on the signal from the ERK-KTR biosensor. The images are provided by Dr. M. Dobrzynski, Unibe.

With the chosen experimental setup, cell outlines are well visible when the biosensor ERK-KTR is in the cytosol. This corresponds to the situation when ERK gets activated. After the activation subsides, the biosensor translocates back to the nucleus. The fluorescence signal in the cytosol then decreases drastically, and cell outlines are hardy visible. CellProfiler did a poor job in identifying the cell outlines in these situations, which resulted in faulty segmentation of the cells in many cases. The nucleus on the other hand is well visible all the time thanks to the nuclear biosensor NucMem. For this reason, we decided to ignore features of whole cells and instead focus exclusively on features of the nuclei.

Since the features are measured over time, each feature is recorded as a time series of measurements. Hence, a time series is the measurement of one feature for one cell at subsequent time points.

The output of this step is a data set of time series of various features (variables) for each cell. The data set exists in the form of a CSV file and is the input to our experiments. Table 3 gives a brief overview of the file formats used in the preparation of the data sets.

Table 3: File formats used during the data preparation pipeline

	Microscopy	**Cell segmentation and feature extraction**
Output	Image series for each recorded color channel (pixels)	Data set with time series of specific variables (features) from the nuclei (see next section or *9.1 Variables in the data sets* in the appendix for more information about the data sets)
Format	nd2 (propietary format for Nikon microscopes)	csv

4.3 Data sets

The data sets that were used in this project were made available by Dr. M. Dobrzynski. They contain time series with 43 variables in total. These variables are listed in the appendix, see *9.1 Variables in the data* sets. They can be attributed to the following categories:

- 16 nucleus shape variables (variable IDs 3-18)
- 16 fluorescence intensity variables (variable IDs 19-34)
- 11 metadata variables (variable IDs 1, 2 and 35-43)

We used two different data sets in this project. Both were generated on different days but with identical experimental setups. The datasets are listed in Table 4.

Table 4: The datasets used in this project

Notation in this work	**Identification**
Dataset 2017_08	20170812_dose_response_same_FOVs
Dataset 2017_07	20170713_dose_response_same_FOVs

To evaluate the classifiers that were created during this project, Dr. M. Dobrzynski annotated the cells in these datasets manually. This was done in two steps: First, the images of the cells were revised by eye. Cells with segmentation errors in the nuclei as well as cells undergoing cell division and dying cells were annotated as bad cells. Second, the time series of nuclear ERK concentration were plotted. Time series that contained obvious outliers were annotated as bad cells as well.

Annotations were added as Boolean values to the data sets, see Table 5.

Table 5: Cell class overview

Cell class	**Notation in the data sets**	**Description**
Bad	FALSE	Cells that contain bad time series.
Good	TRUE	Refers to cells that do not contain bad time series.

Unless otherwise stated, all results in this project were obtained from dataset 2017_08.

4.4 Data analysis and evaluation

Variables in the data sets have different value ranges and units, see *9.1.3 Variable types and ranges* in the appendix. Thus, we rescaled them to have a mean value of 0 and a standard deviation of 1 to make the results comparable across variables.

We analyzed the data sets with RStudio [14]. We used the following R packages:

- *dplyr* [20] and *data.table* [21] for efficient data manipulation
- *ggplot2* [22] for easy and interactive plots
- *traj* [23] to extract various measurements from time series

- *rpart* [24] to generate decision tree classifiers
- *ROCR* [25] to create ROC curves

To evaluate classifiers, we present the results in the form of confusion matrices, see Table 6. Confusion matrices facilitate the calculation of various performance metrics [26], see Equation 1.

Table 6: Depiction of a confusion matrix used in this work

		actual class	
		bad	*good*
predicted class	*bad*	True negative (TN)	False negative (FN)
	good	False positive (FP)	True positive (TP)

Equation 1: Performance metrics used in this work

$$True\ positive\ rate\ (TPR) = \frac{TP}{TP + FN}$$

$$False\ positive\ rate\ (FPR) = \frac{FP}{TN + FP}$$

$$Accuracy = \frac{(TP + TN)}{TP + FP + TN + FN}$$

To avoid as much bias as possible from the data sets, our goal is to minimize the number of false positives (see *2 Problem statement*). Hence, a low false positive rate is important for this work. The number of measured cells (N = 238) is large enough that when some of them are removed as false negatives (n = 46), there is still a significant number left for the analysis (n = 192).

5 Results

5.1 Variable selection

As a preprocessing step, analyses were carried out to reduce the number of variables in the input data sets. All initially collected variables are listed and explained in *9.1 Variables in the data* sets in the appendix.

5.1.1 Nuclear Shape Variables

We expected a subset of the shape variables to be redundant due to the roundish shape of the nuclei (different variables describe the same features of the nucleus). To investigate this hypothesis, we calculated the correlation coefficients of each pair of nuclear shape variables. The complete table with all parameters is available in the appendix, *9.2 Correlation coefficients of all nuclear shape variables*. Table 7 depicts a subset of variables (only those that contained a high correlation coefficient).

Table 7: Bivariate Pearson correlation coefficients of nuclear shape variables. Values greater than 0.85 are displayed in bold.

Variable names	Area	Major axis length	Max Feret diameter	Max radius	Mean radius	Median radius	Min Feret diameter	Minor axis length	Perimeter
Area									
Major axis length	**0.87**								
Max Feret diameter	**0.89**	**0.99**							
Max radius	**0.91**	0.62	0.65						
Mean radius	**0.95**	0.73	0.75	**0.98**					
Median radius	**0.95**	0.75	0.78	**0.95**	**0.98**				
Min Feret diameter	**0.93**	0.65	0.69	**0.98**	**0.97**	**0.95**			
Minor axis length	**0.92**	0.64	0.67	**0.99**	**0.98**	**0.95**	**1.00**		
Perimeter	**0.97**	**0.92**	**0.94**	0.83	**0.88**	**0.89**	**0.87**	**0.86**	

As can be seen in Table 7, the *area* variable correlates highly with all other listed variables (smallest $r = .87$). Despite of their high correlation, the variables *max Feret diameter, min Feret diameter* and *perimeter* were kept in the data set in addition to the *area* variable. The Pertz lab found those variables to be good measures in the data analysis (personal communication). The remaining variables from Table 7 were not used in this work.

Nuclear shape variables that are not listed in Table 7 did not show high correlation to other variables in the data set (see full variable list in the appendix, *9.1 Variables in the data sets*). Still, we discarded two of these remaining variables. The variable *Euler number* describes topological properties of a structure. In the case of 2D objects, its value represents the number of "holes" the object contains [27]. Since the nuclear marker was homogeneously distributed across the nucleus in the experiments associated with this project, the segmented nuclei never contained "holes". Therefore, the Euler number in the data set was constantly 1 and we excluded it from the dataset.

The values of the variable *orientation* fall in the range between -90 and +90 degrees. Since this variable shows large variations in value, we excluded this variable from the data sets

as well. Hence, we selected nine nuclear shape variables to be kept in the dataset, see Table 8.

Table 8: Selected Nuclear Shape Variables

Nuclear shape variables
Area
Compactness
Eccentricity
Extent
Form factor
Max Feret diameter
Min Feret diameter
Perimeter
Solidity

5.1.2 Fluorescence Intensity Variables

The data set contains each fluorescence variable twice: the raw readout of the sensor as well as the readout values with illumination correction. We considered only variables with illumination correction and discarded the raw readouts.

Also, the data set contains fluorescence readouts within the cytosol as well as within the nucleus. Since the segmentation of the cytosol was not optimal, we did not consider fluorescence variables concerning the cytosol.

Hence, we included four fluorescence intensity variables in the data sets, see Table 9.

Table 9: Selected Fluorescence Intensity Variables

Fluorescence intensity variables
Integrated intensity of ERK-KTR biosensor
Integrated intensity of the nuclear marker
Mean intensity of ERK-KTR biosensor
Mean intensity of the nuclear marker

These fluorescence intensity variables do not have high correlations among each other and do not significantly correlate with the selected nuclear shape variables. The correlation coefficients of all selected variables are available in the appendix, *9.3 Correlation coefficients of all selected variables*.

5.1.3 Metadata

Metadata is added to the data sets by the CellProfiler software as additional information. However, not all metadata is required in this work.

Variables *Image_Metadata_Site* and *objNuc_TrackObjects_Label* together uniquely identify a cell. Hence, we used these two variables to generate unique cell identifiers to facilitate the subsequent data analysis.

We used variable *RealTime* (time of experiment in minutes) to sort the input data sets according to the time points for each cell. Variables *objNuc_Location_Center_X* and *objNuc_Location_Center_Y* were used to identify the cell in the microscopy images. Other metadata from the dataset was not used in this project.

5.2 Cell Classification

We used the selected variables from section 5.1 to predict whether time series are good or bad. We implemented and evaluated three different approaches.

In a first attempt, we used principal component analysis (PCA) to reduce the dimensionality in the data set (section 5.2.1) and then used the principal components with the best ability to separate the two cell classes to determine outliers and predict cell classes.

In another approach, we chose an iteration over the time series using a sliding window procedure (section 5.2.2). In each iteration, we analyzed the values within the window to check whether there is an outlier value. We then used the sum of these outlier values for cell class prediction.

Finally, we tested a supervised machine learning approach with decision tree classifiers (section 5.2.3). We calculated several measures from the time series that we then used as input features to our classifiers.

5.2.1 Classification with Principal Component Analysis (PCA)

PCA is a common method in data science to reduce the dimensionality of data sets, where a set of points is transformed into its principal components [28]. PCA can furthermore facilitate the detection of outliers in data sets (for example, see [29]). Because of these properties, we used PCA as a first step in identifying outlier time series.

We first transformed the time series of each cell into their principal components using PCA (see Figure 5A). We only used the principal components that cumulatively accounted for 95% of the variance in the data (measured by the eigenvalues of the covariance matrix) for further analyses. The variance contribution for the first 20 principal components are listed in *9.4.1 Cumulative contribution to variance*.

In a second step, we analyzed the degree to which these remaining principal components can distinguish between the two cell classes. The Bhattacharyya distance is a statistical measure of the similarity of two probability distributions [30]. We used it to measure the separation between the distributions of good and bad time series for each principal component. Then, we visualized the Bhattacharyya distances as bar plots (see Figure 5B). The larger the distance in these plots, the better a principal component distinguishes between the two cell classes. All Bhattacharyya distance values and visualization of the class separations are listed in *9.4.2 Class separability (Bhattacharyya distance)*.

Finally, we selected the principal component with the largest Bhattacharyya distance value and used the distribution of time series in function of that principal component to predict the cell classes (Figure 5C). The Mahalanobis distance is a measure of the distance between a single point and a whole distribution [31]. We calculated the Mahalanobis distance between each time series and the distribution of time series and used a threshold value for the Mahalanobis distance to determine if a time series is an outlier (larger than the chosen threshold value) or not.

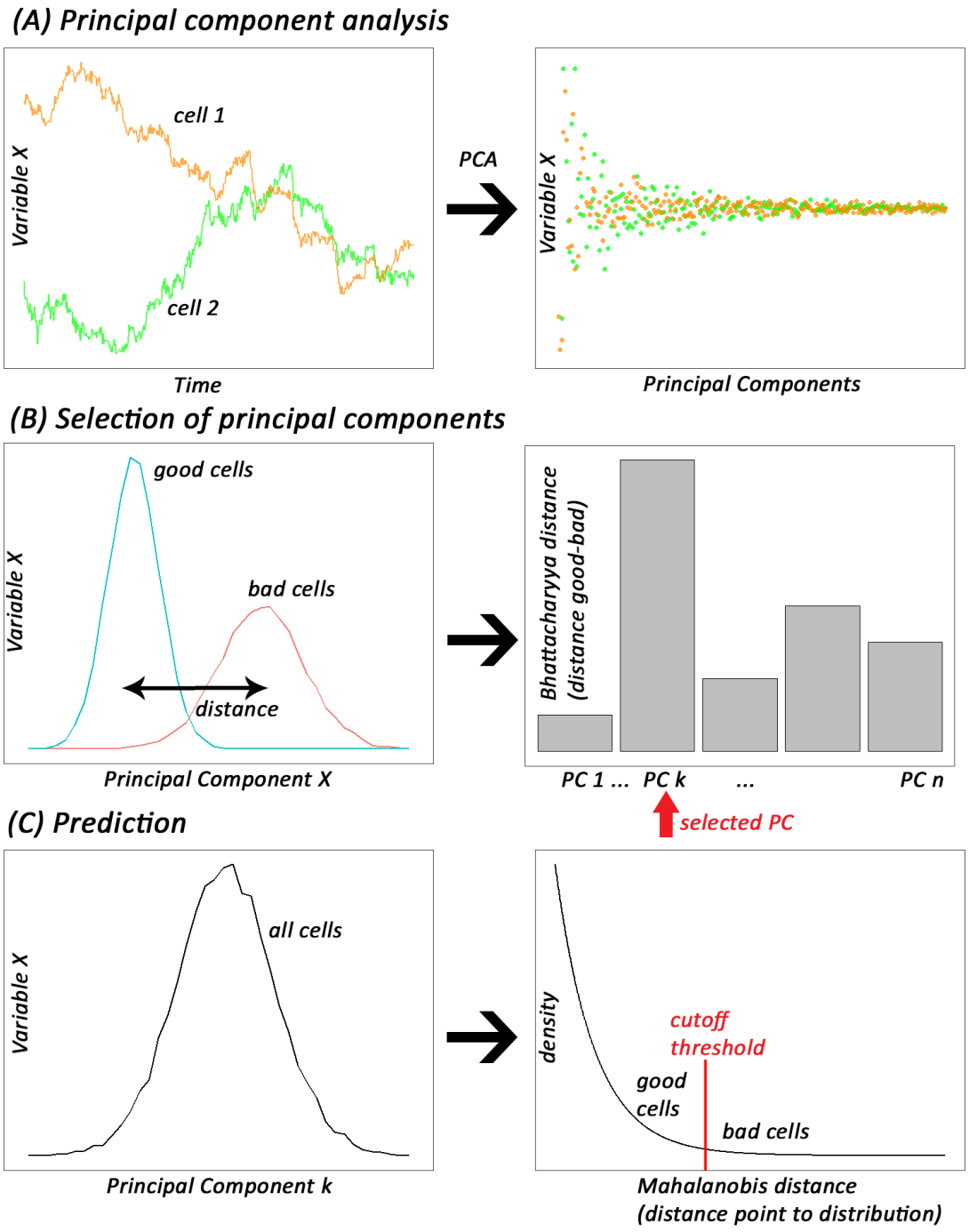


Figure 5: Schematics of outlier selection procedure based on PCA. (A) Time series were first converted to principal components with PCA. (B) The first few principal components were tested for their cell class separability power. The Bhattacharyya distance was used to measure the separability. (C) Principal components with the best class separability power were chosen for classification. The Mahalanobis distance was calculated for all points (cells) and a cutoff threshold was determined to label good and bad cells. The procedure was carried out for each variable separately (Variable X is a placeholder for any variable in the figure).

The Bhattacharyya distances for all variables are visualized in Figure 20 in the appendix. The first and second principal components of all variables demonstrate poor class separation. The third and fourth principal components however are more promising. The best results are presented in Figure 6.

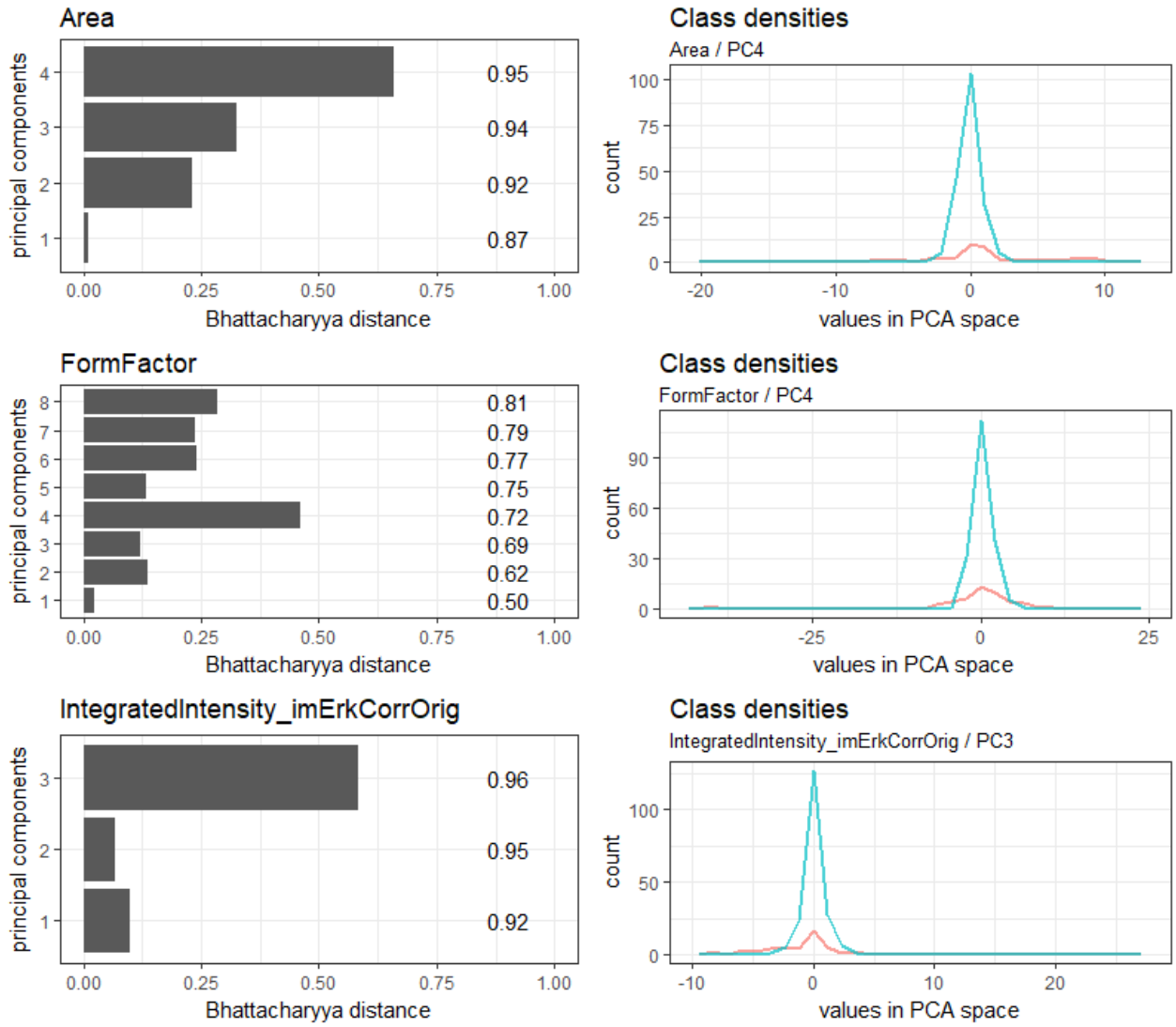


Figure 6: Bar plots of the Bhattacharyya distances for the different variables are shown on the left side. The number on the right side of the bar plots indicate the cumulative contribution to variance of the principal components. The class densities in PCA space are shown on the right.

The shape variables *area, perimeter, max Feret diameter* and *form factor* all had a relatively high Bhattacharyya distance associated with their fourth principal component. Since *area*, *perimeter* and *max Feret diameter* are strongly correlated (see section 5.1.1), only *area*, which had the best success rate among these three variables, was used further along with *form factor*. With respect to the intensity variables, only the third principal component of the integrated intensity of ERK showed a meaningful class separation.

Hence, the fourth principal components of variables area and form factor as well as the third principal component of integrated intensity of biomarker ERK-KTR were used for further analysis. The Bhattacharyya distances as well as the class densities for these variables are shown in Figure 6. The class densities represent the distribution of time series in PCA space and visualize how well the selected principal components separate the two cell classes. Two-dimensional class separation between all pairs of these variables is shown in Figure 7.

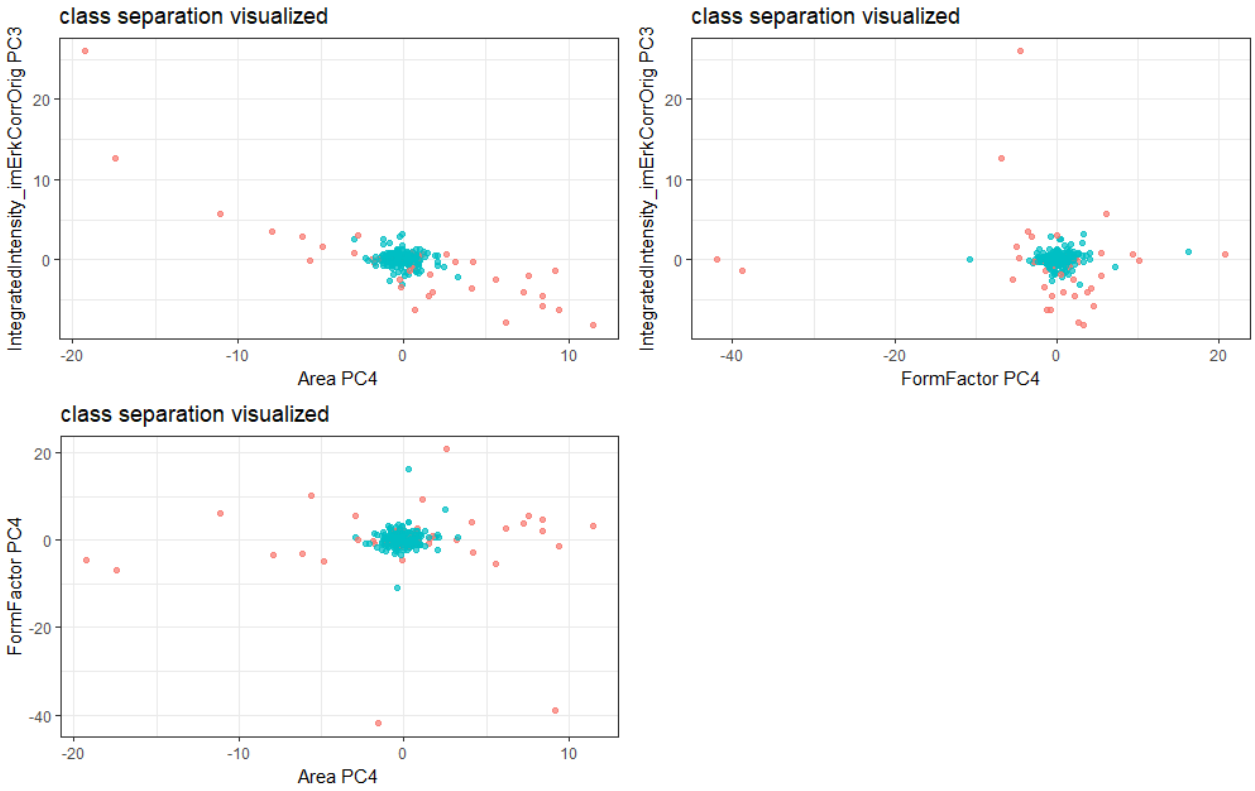


Figure 7: 2D class separation for all combinations of selected variables. Good time series are represented as green points, bad time series are red.

The scatter plots in Figure 7 indicate that the selected principal components are suitable for separating the two cell classes. We calculated the Mahalanobis distances of time series for each of the three variables. Plots of these distances are shown in Figure 8. A subset of bad cells contains outlier time series with respect to the distributions of these variables. A large proportion of bad time series, however, have small Mahalanobis distances and are therefore similar to the good time series.

We treated time series with a Mahalanobis distance larger than the value resulting from the equation described in Equation 2 as outliers and labeled the corresponding cells as bad cells.

Equation 2: Determination of outlier time series based on their Mahalanobis distance (MD) and a tolerance factor (TF)

$$\boldsymbol{outliers}: MD \geq mean(MD) + TF * IQR(MD)$$

Hence there were three separate cutoff values for cell classification (one for each selected variable principal component). We used each cutoff value for cell classification individually as well as in combination with the other cutoff values.

The results for tolerance factors 0.25, 0.5 and 1.0 are listed in *9.4.3 Classification results*. They are robust and change only slightly with the different factors. For further evaluations, we set the tolerance factor to 0.5. These results are listed in Table 10.

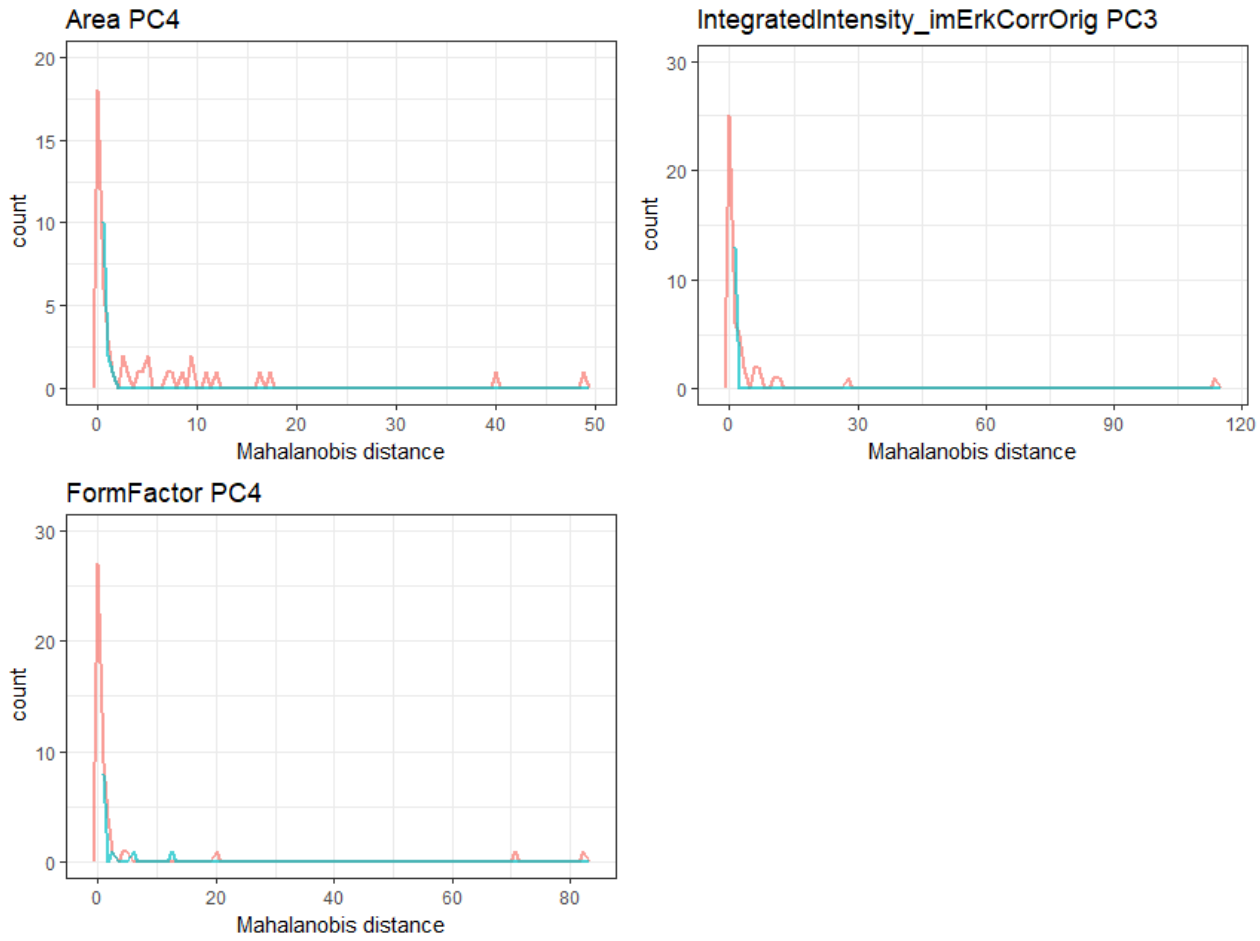


Figure 8: Mahalanobis distances of the selected principal components. The y axes of the plots are cropped to better visualize small values (the green lines have far higher maximum values). Green lines represent good time series, while red lines represent bad time series.

Table 10: Results for dataset 2017_08

	TPR	**FPR**	**accuracy**
area	99.0%	56.5%	88.2%
form factor	98.4%	76.1%	84.0%
intensity	96.9%	60.9%	85.7%
area + form factor	97.4%	50.0%	88.2%
area + intensity	96.4%	45.7%	88.2%
form factor + intensity	95.3%	43.5%	87.8%
area + form factor + intensity	94.8%	39.1%	88.2%

The true positive rate is reasonable for all variable combinations. Most of the good cells are correctly identified by these classifiers. On the other hand, the true negative rate of this approach is not satisfying. Many bad cells are falsely labeled as good cells.

The classification with all three variables combined resulted in the best true negative rate. The confusion matrix for this approach can be found in Table 11. The results are plotted in Figure 9.

Table 11: Confusion Matrix for classification with all cutoff values combined with data set 2017_08

		actual	
		bad	*good*
predicted	*bad*	28	10
	good	18	182

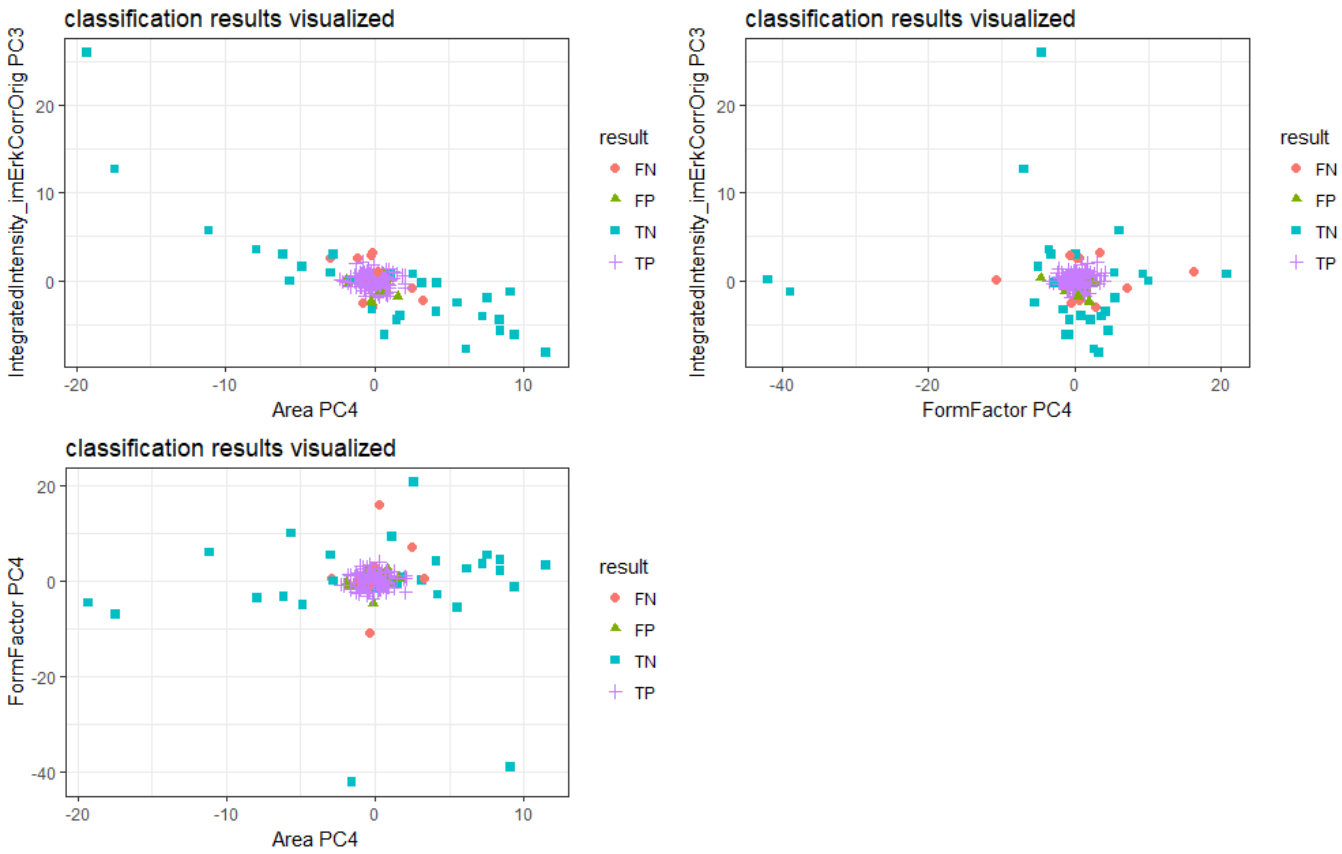


Figure 9: Visualization of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) for cell classification on dataset 2017_08 with all cutoff values combined

We repeated this classification procedure with data set 2017_07 to test if the approach is transferrable to data from other experiments. The results are listed in Table 12.

Table 12: Results for dataset 2017_07

	TPR	**FPR**	**accuracy**
area	98.5%	46.8%	87.5%
form factor	89.2%	54.8%	78.5%
intensity	96.9%	66.1%	81.6%
area + form factor	88.1%	27.4%	84.4%
area + intensity	95.4%	33.9%	88.3%
form factor + intensity	87.1%	37.1%	81.3%
area + form factor + intensity	86.1%	22.6%	84.0%

Again, the combination of all three variables lead to the best true negative rate. Interestingly, the results for this data set are even better than the ones in the other data set (see Table 10 and Table 12). The confusion matrix for data set 2017_07 and all three variables combined is presented in Table 13, the results are plotted in Figure 10.

Table 13: Confusion Matrix for classification with all cutoff values combined with data set 2017_07

		actual	
		bad	*good*
predicted	*bad*	48	27
	good	14	167

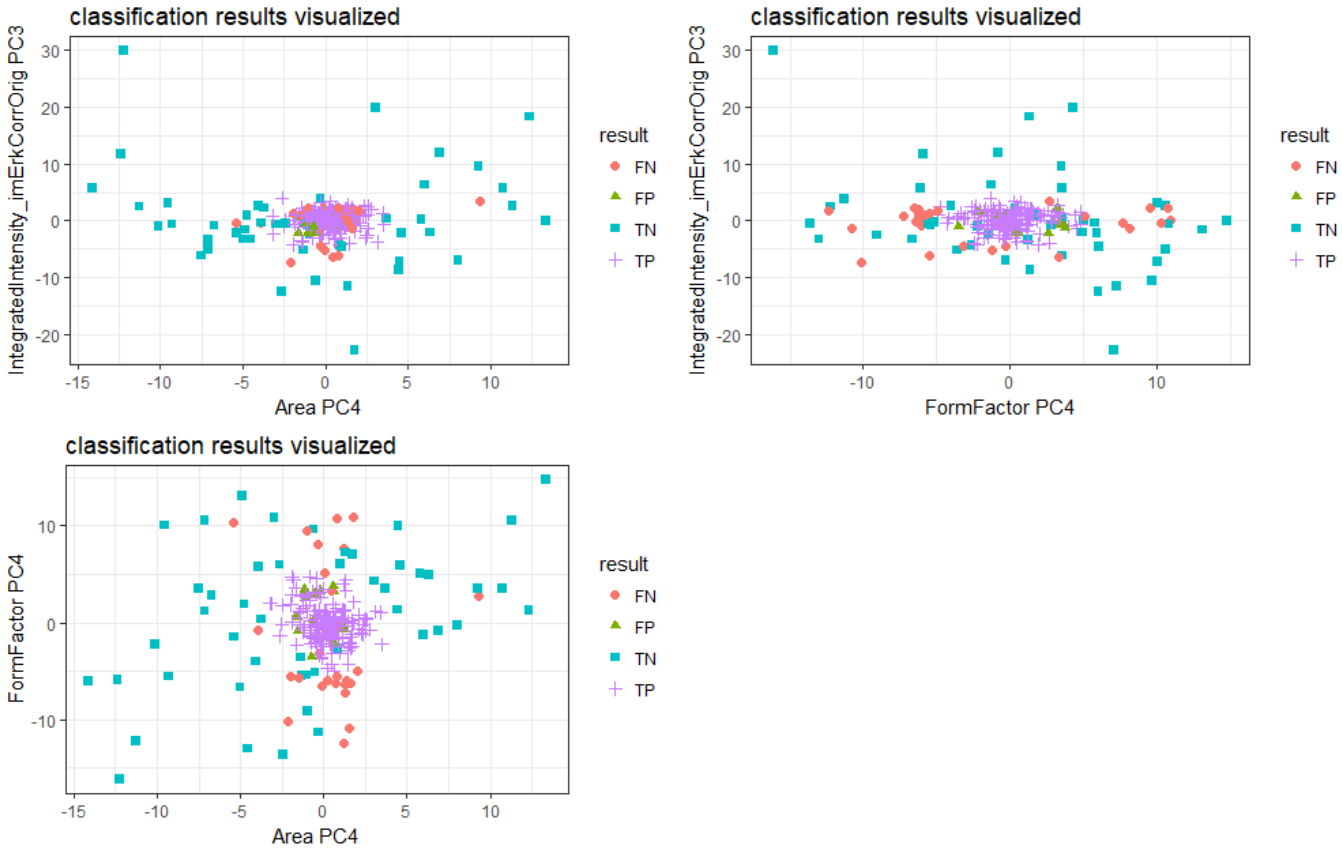


Figure 10: Visualization of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) for classification on dataset 2017_07 with all cutoff values combined

5.2.2 Classification with a Sliding Window Approach

In another approach, we used a modified sliding window procedure to detect outlier time series [32]. In this method, a "window" of a constant number of time points *W* (width, or window size) is moved from left to right along the time series (see Figure 11).

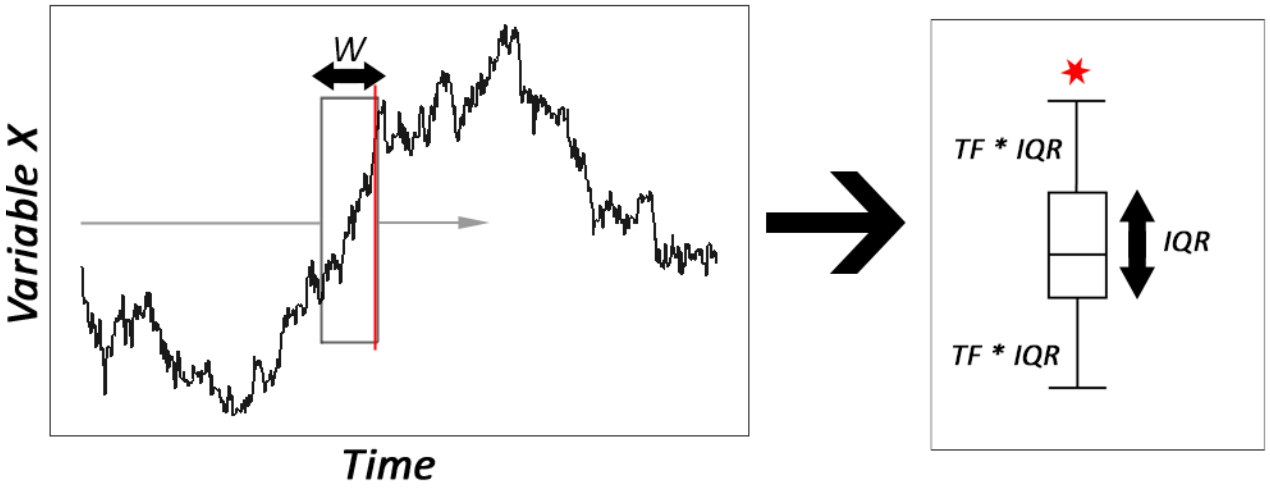


Figure 11: Outlier detection with the sliding window procedure. A window of constant width W is moved along the time series. Each iteration checks whether the last value within the window is an outlier. This check is represented on the right side. The value is an outlier if it is larger than the third quartile plus the inter-quartile range (IQR) multiplied with some tolerance factor (*TF*) or if it is lower than the first quartile minus the IQR multiplied with the same *TF*. Finally, if the sum of outlier values is larger than a threshold value, the corresponding cell is labelled as a bad cell.

In each iteration, we compare the last value within the window against the distribution of all values contained in the window. If the last value is significantly higher or lower than the distribution, we count it as an outlier value. The lower and upper threshold values are calculated with a tolerance factor (*TF*) according to Equation 3:

Equation 3: Threshold values for outlier identification. *TF* = tolerance factor, *IQR* = inter-quartile range.

$$upper\ threshold = 3rd\ quartile(window\ values) + TF * IQR(window\ values)$$
$$lower\ threshold = 1st\ quartile(window\ values) - TF * IQR(window\ values)$$

If *T* corresponds to the total number of time points in a time series, the number of positions a window moves along a time series (or number of iterations) can be calculated according to Equation 4:

Equation 4: Number of iterations for a window over a time series. *T* = total number of time points, *W* = window size.

$$number\ of\ iterations = T - W + 1$$

We repeat this procedure for all cells and time series (i.e. all selected variables). Finally, if the sum of outlier values over all variables for a cell is greater than a pre-defined threshold value, we label the cell as a bad cell. We calculate this threshold value according to Equation 5:

Equation 5: Threshold value for bad cells

$$sum(outlier\ values) > 0.1\% * number\ of\ variables * number\ of\ iterations$$

We evaluated the procedure with all selected variables from *5.1 Variable selection* and with various combinations of the parameters windows size *W* and tolerance factor *TF*. We found that some variables, especially *min Feret diameter*, have many outlier values

irrespective of the cell being good or bad. Following this analysis, we decided to use only the variables listed in Table 14 in the subsequent sliding window approach.

Table 14: Variables used in the approach

Variable name
FormFactor
Perimeter
Solidity
Integrated intensity of the nuclear marker

The confusion matrices for data sets 2017_08 and 2017_07 with $W = 40$ and $TF = 5$ are shown in Table 15 and Table 16 respectively. Table 17 summarizes the results.

Table 15: Confusion Matrix for data set 2017_08

		actual	
		bad	*good*
predicted	*bad*	35	16
	good	11	176

Table 16: Confusion Matrix for data set 2017_07

		actual	
		bad	*good*
predicted	*bad*	55	19
	good	7	175

Table 17: Results of the sliding window approach

	TPR	**FPR**	**accuracy**
data set 2017_08	91.7%	23.9%	88.7%
data set 2017_07	90.2%	11.3%	89.8%

The results are comparable to those of the PCA-based approach (section 5.2.1). The false positive rate, however, is significantly better with the sliding window approach.

To visualize the distribution of good and bad time series with this approach, the sum of outlier values per time series are plotted as stacked histograms in Figure 12. The plots indicate that bad time series show a clear tendency to have higher sums of outlier values.

Figure 13 shows the ROC curve for this procedure with parameters $W = 40$ and $TF = 5$ for data set 2017_08. The curve is generated by varying the threshold value that is used to identify bad time series (see Equation 5). The ROC curve shows that the true positive rate grows fast when decreasing the maximum possible threshold value and the false positive rate starts to grow quickly only for small threshold values, which is a satisfactory result.

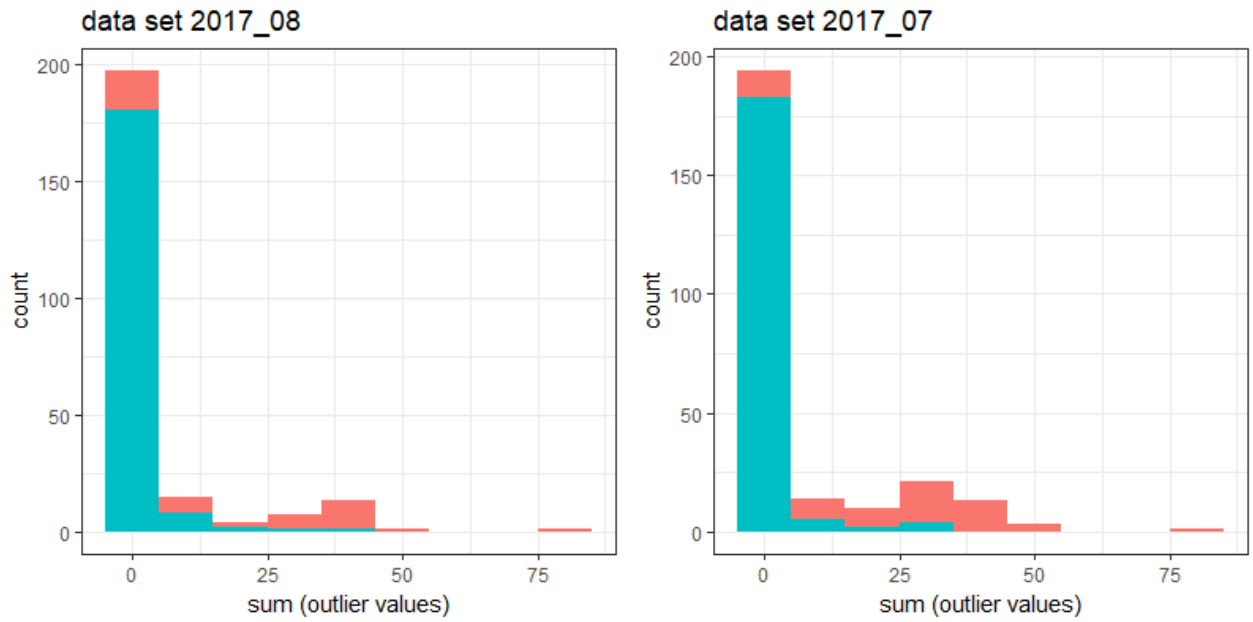


Figure 12: Stacked histograms containing the distributions of the sum of outlier values for the two cell classes (“good” in green, “bad” in red). The plots show that bad time series tend to have larger sums of outlier values.

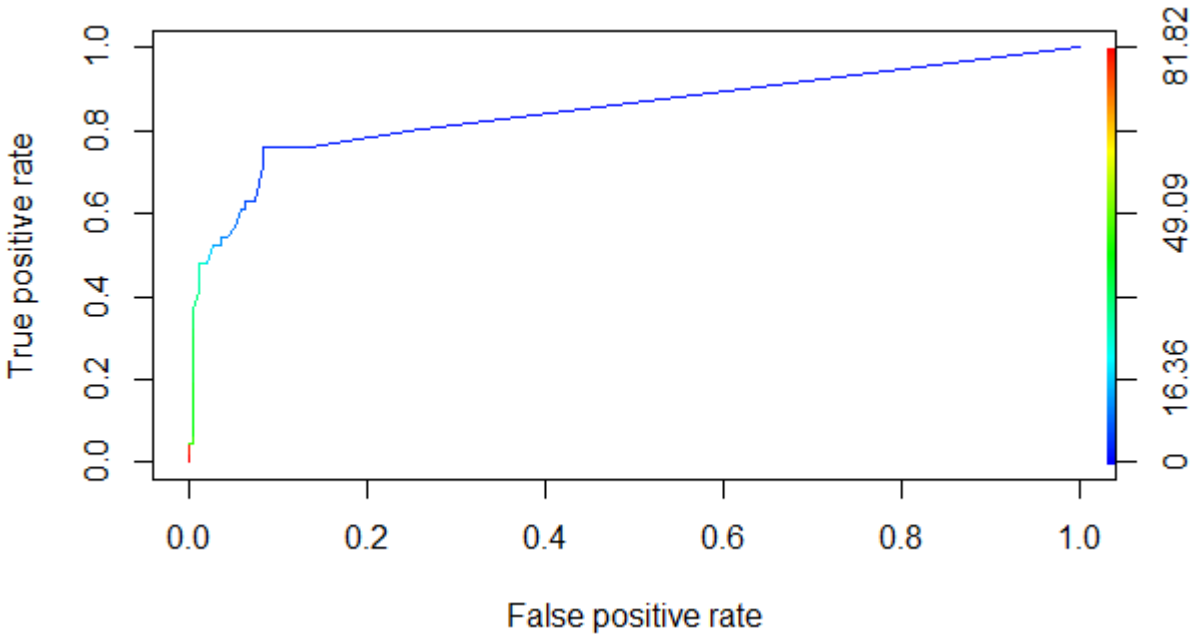


Figure 13: ROC curve for sliding window procedure. The false positive rate is plotted against the true positive rate. The color of the curve corresponds to the threshold value used to

5.2.3 Classification with Supervised Machine Learning

As an alternative to directly detecting outlier time series, we used a supervised machine learning approach to categorize the time series. In supervised learning, a machine attempts to infer classification rules from a set of features (variables) of multiple annotated instances (records). Various algorithms exist for this purpose. Because rules can be intuitively represented by trees, we chose a supervised learning approach with decision tree classifiers in this work. We used the R package *rpart* [24] to create the decision trees.

The R package *traj* [33] provides a method that calculates various measures of time series / trajectories, see Table 18. We calculated these measures for each time series (Figure 14A). With 13 selected variables, we extracted 312 features per cell. These features then served as input to the machine learning model (Figure 14B). In a final step, we evaluated the model by testing its accuracy on the training data set for control as well as on a separate test data set (Figure 14C).

Table 18: Features extracted from the time series

ID	Description
m1	Range (minimum value subtracted from the maximum value)
m2	Mean-over-time
m3	Standard deviation (SD)
m4	Coefficient of variation (CV)
m5	Change (first value subtracted from the last value)
m6	Mean change per unit time
m7	Change relative to the first score
m8	Change relative to the mean over time
m9	Slope of the linear model
m10	R^2: Proportion of variance explained by the linear model
m11	Maximum of the first differences
m12	SD of the first differences
m13	SD of the first differences per time unit
m14	Mean of the absolute first differences
m15	Maximum of the absolute first differences
m16	Ratio of the maximum absolute difference to the mean-over-time
m17	Ratio of the maximum absolute first difference to the slope
m18	Ratio of the SD of the first differences to the slope
m19	Mean of the second differences
m20	Mean of the absolute second differences
m21	Maximum of the absolute second differences
m22	Ratio of the maximum absolute second difference to the mean-over-time
m23	Ratio of the maximum absolute second difference to mean absolute first difference
m24	Ratio of the mean absolute second difference to the mean absolute first difference

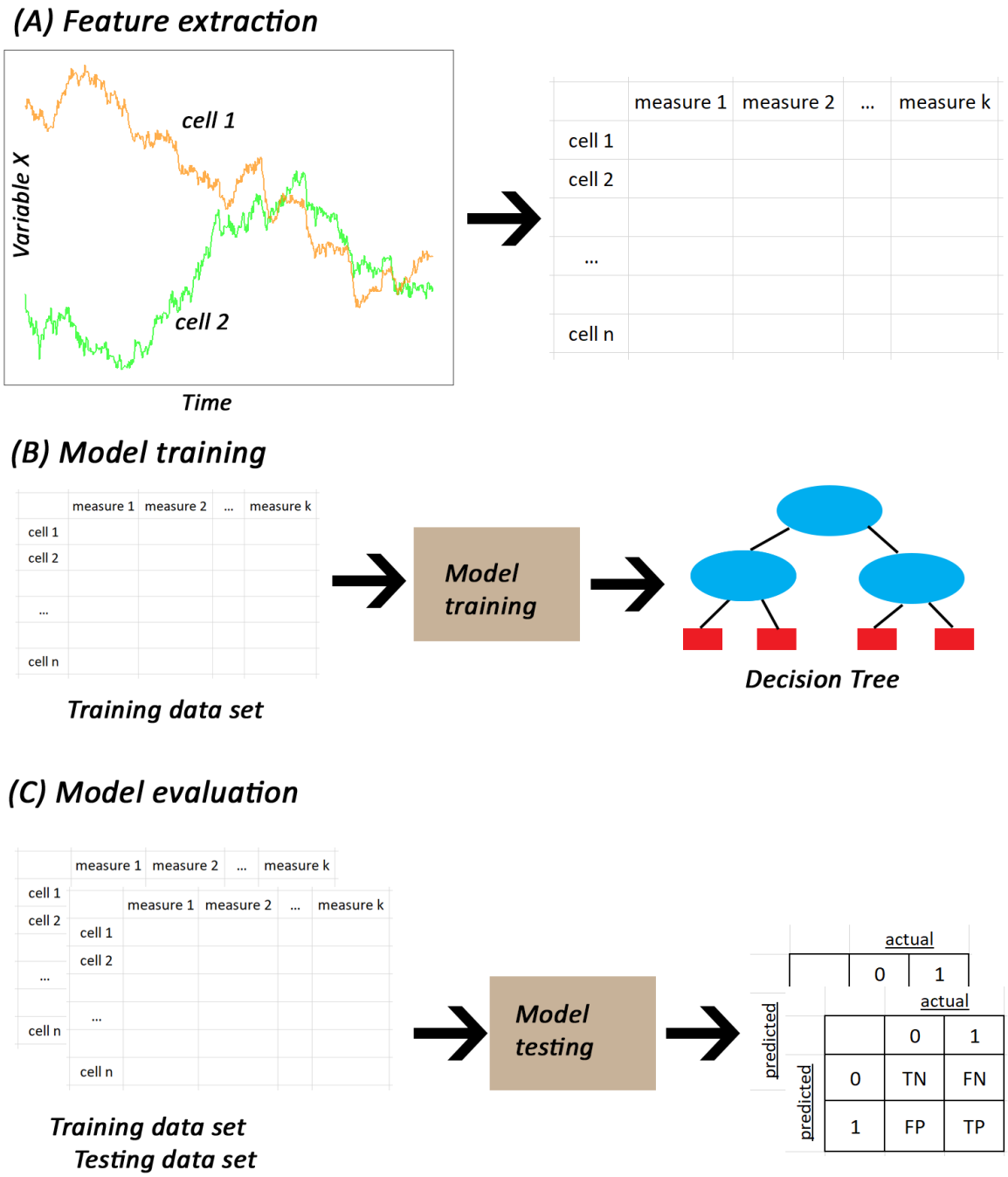


Figure 14: Overview of the supervised learning method. (A) Various features were extracted from each time series. (B) A decision tree model was then trained based on the extracted features and the manual annotation from a training data set. (C) The model was then tested with the training data set (as control) and a separate testing data set.

The resulting decision tree splits on three features (see Figure 15) and has an overall accuracy of 93% for the training data set. When applied to the independent test data set, the accuracy drops to 84%. To reduce the effect of overfitting, we optimized the pruning of the tree. The *rpart* package uses a complexity parameter (cp) to control at which point the growth of the tree stops during the tree construction. The stop happens as soon as a

further split does not improve the model by at least the value provided by cp and the tree is pruned at that node [34]. The effects of increasing the value for cp are shown in Figure 16.

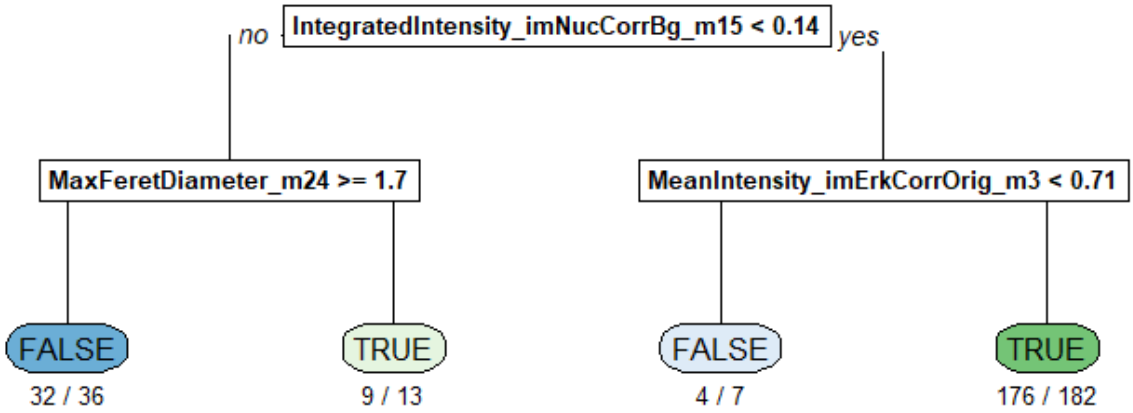


Figure 15: Decision tree with default value for complexity parameter (cp)

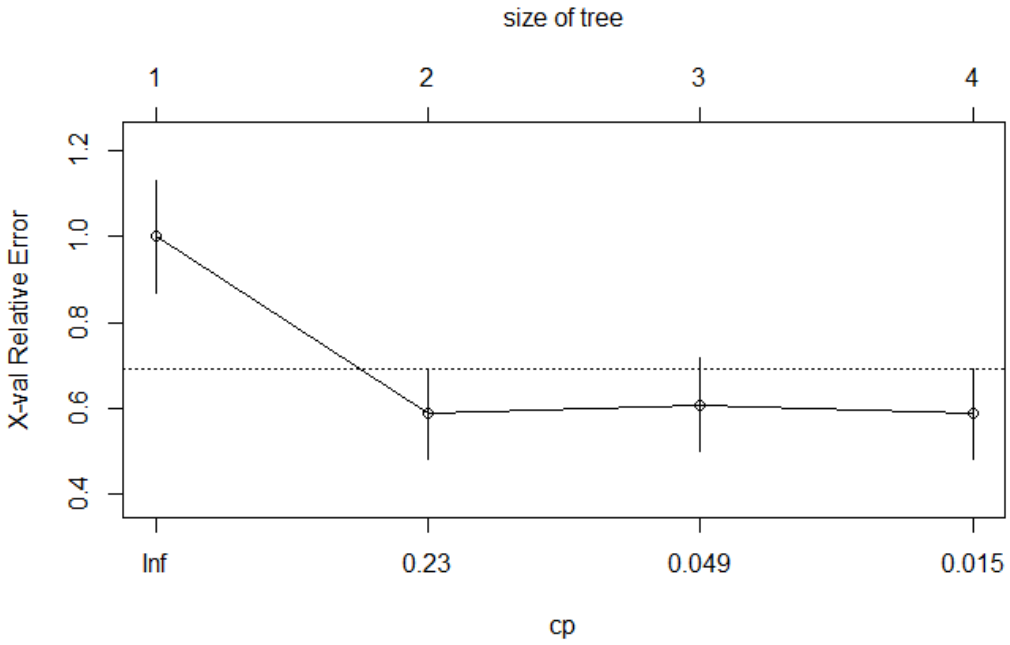


Figure 16: Overview of how the reduction of the complexity parameter (cp) influences the relative error in the model and the size of the tree.

The plot in Figure 16 shows comparably low error for a pruned two-level tree and the three-level tree shown in Figure 15. Hence, we retained the two-level tree. The resulting tree is shown in Figure 17.

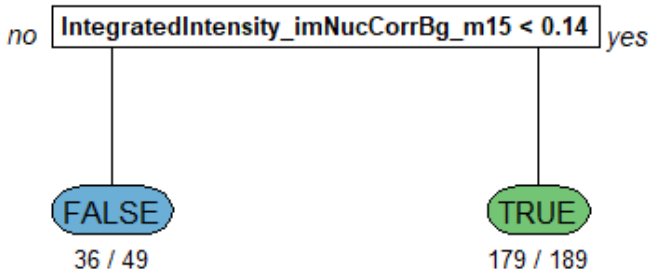


Figure 17: The pruned decision tree

Table 19 and Table 20 contain the confusion matrices for the training and testing data sets, Table 21 summarizes the results. With both data sets, we reached an accuracy and a true positive rate of more than 90%. Analogously to the sliding window approach, we found the false positive rate to be significantly better with the test data set (11.3%) than with the data set that was used for training (21.7%), see Table 21.

Table 19: Confusion Matrix for training data set

		actual	
		bad	*good*
predicted	*bad*	36	13
	good	10	179

Table 20: Confusion Matrix for testing data set

		actual	
		bad	*good*
predicted	*bad*	55	17
	good	7	177

Table 21: Results

	TPR	**FPR**	**accuracy**
data set 2017_08 (training)	93.2%	21.7%	90.3%
data set 2017_07 (testing)	91.2%	11.3%	90.6%

Figure 18 shows the ROC curve for the pruned tree with the training data set. The curve is generated by varying the threshold value of the feature *IntegratedIntensity_imNucCorrBg_m15* in the tree and plotting the false positive rate against the true positive rate. The ROC curve shows that the true positive rate grows fast when decreasing the threshold value and the false positive rate starts to grow quickly only for small threshold values, which is a satisfactory result.

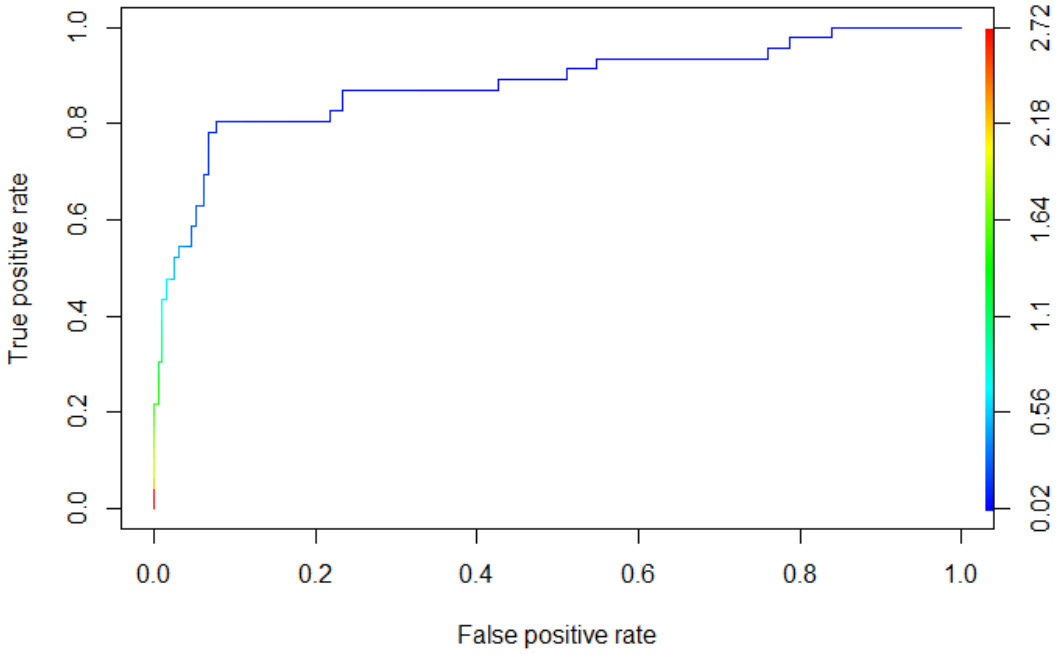


Figure 18: ROC curve for the pruned tree. The false positive rate is plotted against the true positive rate. The color of the curve corresponds to the threshold value of IntegratedIntensity_imNucCorrBg_m15.

Figure 19 shows probability density plots of IntegratedIntensity_imNucCorrBg_m15 per class for both data sets. As expected from the rule of the decision tree, large values are more likely to be associated with bad cells.

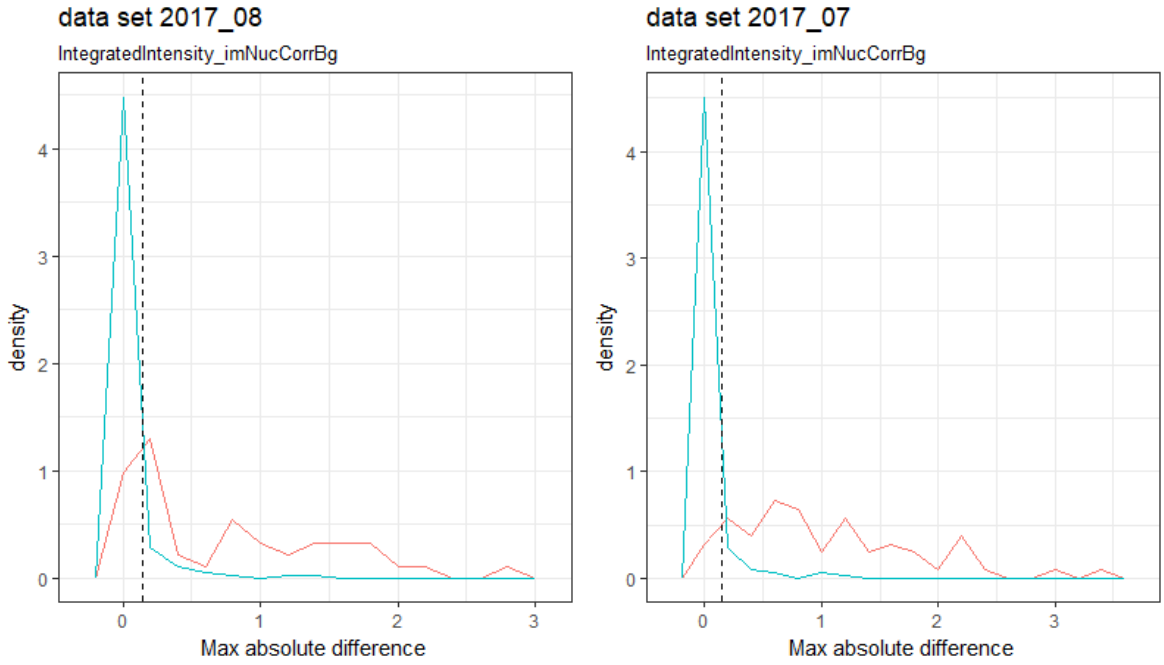


Figure 19: Probability density plots of IntegratedIntensity_imNucCorrBg_m15 for the two cell classes (“good” in green, “bad” in red). Data set 2017_08 was the training dataset, 2017_07 was the test data set. The dashed vertical line is the threshold found by the decision tree. The plots show that bad time series tend to have larger values, which confirms the rule found by the decision tree.

6 Discussion

The aim of this work was to develop and evaluate methods to identify outliers in time series of cells. To our knowledge, there are no established methods for this problem at the moment. Hence, we developed and evaluated three approaches.

A single time series is a point in a highly dimensional space where each dimension corresponds to a time point. Thus, in the first approach, we used principal component analysis (PCA) to reduce the high dimensionality of the time series. Then, in a manually annotated data set, we selected principal components that could best separate outlier time series from the rest. This allowed us to establish a strongly reduced space based on these principal components. This procedure enabled us to transform the problem of finding outlier time series to a less complex problem of finding outlier values.

This approach produced a high true positive rate, meaning that most of the correctly segmented cells were recognized as such. A positive point of this procedure is that the algorithm associated with it has a low execution time. The results however were not completely satisfactory. There was a relatively large number of falsely segmented cells with time series that do not qualify as outliers according to this method. This finding is represented by a high false positive rate (39%).

The PCA based procedure has some limitations: First, one needs to select principal components along which two classes of annotated time series are separated the most. This decision is based on the Bhattacharyya distance between the distributions of two classes. Even though the Bhattacharyya distance is designed to measure the amount of overlap between two populations, the distance may be high despite overlap of points from two distributions. There might exist methods that are more sensitive to such a situation. The second limitation is the introduction of a cutoff threshold for the outlier points in the reduced component space. We calculate a Mahalanobis distance between a point and the rest of the population and choose a threshold beyond which points are deemed outliers. Thus, in the PCA based approach, a user needs to choose two thresholds, and the choice will depend on a particular data set.

In a second approach, we introduced a window and considered whole time series by sliding the window from point to point. In each iteration, we determined whether the last value within the window is an outlier with respect to all values in the window. If the sum of outlier values is larger than a threshold value, the time series is considered an outlier and is discarded.

The clear advantage here is that a time series are considered in its entirety and the method is sensitive to abrupt changes in the time series, which is exactly the type of behavior we would like to identify and discard. The results are satisfactory: The true positive rate is comparable to the PCA-based approach, and the false positive rate is significantly lower (maximum value of 24%). There are, however, some limitations. Again, manual intervention is required. A single time series may have multiple outlier points. Thus, one needs to determine a threshold for the number of outlier points within the time series beyond which a time series is deemed an outlier. Furthermore, this method has a significantly longer execution time than the PCA-based approach (minutes instead of seconds).

In a third approach, we wanted to avoid manual interventions and used supervised learning to get computerized estimations of falsely segmented time series. We created decision trees because trees intuitively represent the learnt rules of a machine. This, in turn, facilitates the implementation of the algorithm in a productive environment. The high dimensionality in the data sets is a real challenge for this method as well. We reduced the dimensionality by calculating 24 aggregated measures of the time series and used those as input features for the supervised learning model.

A considerable limitation in this approach is the aggregation of whole time series to few measures. Since information contained in the trajectories is lost, caution is required to aggregate data in such a way that the outlier dynamics (i.e. abrupt changes) is captured by those measures. Nevertheless, this approach yielded the best results: The true positive rate is comparable to the other two methods and the false positive rate was slightly lower than the false positive rate of the sliding window approach (a maximum FPR of 22% instead of 24%). Notably, these results were achieved with a decision tree splitting on a single feature: the maximum value of the absolute first differences of the integrated intensity of the nuclear marker. It will be interesting to extend our sliding window approach with this feature and evaluate if the results get better.

A common limitation of all three classification approaches is their requirement of predefined threshold values. More research is needed to determine whether the findings from this work are generalizable and can be reproduced with data sets from further experiments.

An interesting insight from this work is that a number of time series that were manually annotated as bad were not identified as outliers by any of the methods. We could not decrease the false positive rate to a value below 10%. This indicates weak points of all the approaches. The dimensionality reduction into a couple of principal components in the first method may be too crude, the sliding window approach or the aggregation in the decision tree approach may be insensitive to the type of sudden changes existing in those time series. Further research is required to fine tune the methods.

A further major insight from this work indicates which variables are relevant to identify falsely segmented cells. We suspected certain variables such as the area of the nucleus or the integrated intensity of the nuclear marker to be important for this task, which we could confirm. However, we found that other variables could be helpful as well, such as the form factor of the nucleus or aggregated values of variables.

The results presented in this work will contribute to large-scale unsupervised time series processing from single-cell microscopy experiments. In a next step, we will integrate our findings into an application that will be used to inspect and clean data sets from our experiments. We will publish the application (open source), such that researchers all over the world can make use of our approaches to identifying outlier time series.

List of figures

7 List of tables

8 Bibliography

[1] S. J. Altschuler and L. F. Wu, "Cellular Heterogeneity: Do Differences Make a Difference?," *Cell,* pp. 559-563, 14 May 2010.

[2] B. Burgering, A. de Vries-Smits, R. Medema, P. van Weeren, L. Tertoolen and J. Bos, "Epidermal Growth Factor Induces Phosphorylation of Extracellular Signal-Regulated Kinase 2 via Multiple Pathways," *Molecular and Cellular Biology,* pp. 7248-7256, December 1993.

[3] C. Creuzet, J. Loeb and G. Barbin, "Fibroblast Growth Factors Stimulate Protein Tyrosine Phosphorylation and Mitogen-Activated Protein Kinase Activity in Primary Cultures of Hippocampal Neurons," *Journal of Neurochemistry,* p. 1541–1547, April 1995.

[4] L. Pang, C. F. Zheng, K. L. Guan and A. R. Saltiel, "Nerve growth factor stimulates a novel protein kinase in PC-12 cells that phosphorylates and activates mitogen-activated protein kinase kinase (MEK)," *Biochemical Journal,* pp. 513-519, 15 April 1995.

[5] B. Alberts, D. Bray, K. Hopkin, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, Lehrbuch der Molekularen Zellbiologie, Weinheim: Wiley, 2005.

[6] T. G. Boulton, G. D. Yancopoulos, J. S. Gregory, C. Slaughter, C. Moomaw, J. Hsu and M. H. Cobb, "An insulin-stimulated protein kinase similar to yeast kinases involved in cell cycle control," *Science,* pp. 64-67, 6 July 1990.

[7] H. Ryu, M. Chung, M. Dobrzyński, D. Fey, Y. Blum, S. S. Lee, M. Peter, B. N. Kholodenko, N. L. Jeon and O. Pertz, "Frequency modulation of ERK activation dynamics rewires cell fate," *Molecular Systems Biology,* 27 November 2015.

[8] S. Cooper and C. Bakal, "Accelerating Live Single-Cell Signalling Studies," *Trends in Biotechnology,* pp. 422-433, May 2017.

[9] OMICtools, "IMAGE SEGMENTATION SOFTWARE TOOLS," [Online]. Available: https://omictools.com/image-segmentation-category. [Accessed 13 January 2018].

[10] S. Santoyo, "A Brief Overview of Outlier Detection Techniques," 11 September 2017. [Online]. Available: https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561. [Accessed 7 January 2018].

[11] S. Cardini, "Usable Cell Selector (Project report)," Bern, Switzerland, 2017.

[12] T. Inc., "Trello," [Online]. Available: https://trello.com/. [Accessed 28 12 2017].

[13] T. R. Project, "R: What is R?," [Online]. Available: https://www.r-project.org/about.html. [Accessed 12 January 2018].

[14] RStudio, "RStudio official website," [Online]. Available: https://www.rstudio.com/. [Accessed 15 06 2017].

[15] G. J. Todaro and H. Green, "Quantitative Studies of the Growth of Mouse Embryo Cells in Culture and Their Development Into Established Lines," *The Journal of Cell Biology,* pp. 299-313, May 1963.

[16] S. Regot, J. J. Hughey, B. T. Bajar, S. Carrasco and M. W. Covert, "High-sensitivity measurements of multiple kinase activities in live single cells," *Cell,* p. 1724–1734, 19 June 2014.

[17] N. Kim, J. M. Kim, M. Lee, C. Y. Kim, K.-Y. Chang and W. D. Heo, "Spatiotemporal control of fibroblast growth factor receptor signals by blue light," *Chemistry & Biology,* pp. 903-912, 17 July 2014.

[18] M. F. Favata, K. Y. Horiuchi, E. J. Manos, A. J. Daulerio, D. A. Stradley, W. S. Feeser, D. E. Van Dyk, W. J. Pitts, R. A. Earl, F. Hobbs, R. A. Copeland, R. L. Magolda, P. A. Scherle and J. M. Trzaskos, "Identification of a novel inhibitor of mitogen-activated protein kinase kinase," *Journal of Biological Chemistry,* pp. 18623-18632, 17 July 1998.

[19] J. T. F. A. B. M. L. D. M. K. L. V. R. C. H. G. E. K. C. A. Kamentsky L, "Improved structure, function, and compatibility for CellProfiler: modular high-throughput image analysis software," *Bioinformatics,* 2011.

[20] H. Wickham, R. Francois, L. Henry and K. Müller, "dplyr: A Grammar of Data Manipulation," [Online]. Available: https://cran.r-project.org/web/packages/dplyr/index.html. [Accessed 28 12 2017].

[21] M. Dowle, A. Srinivasan, J. Gorecki, T. Short, S. Lianoglou and E. Antonyan, "data.table: Extension of 'data.frame'," [Online]. Available: https://cran.r-project.org/web/packages/data.table/index.html. [Accessed 28 12 2017].

[22] H. Wickham and W. Chang, "ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics," [Online]. Available: https://cran.r-project.org/web/packages/ggplot2/index.html. [Accessed 28 12 2017].

[23] M.-P. Sylvestre and D. Vatnik, "traj: Trajectory Analysis," [Online]. Available: https://cran.r-project.org/web/packages/traj/index.html. [Accessed 28 12 2017].

[24] T. Therneau, B. Atkinson and B. Ripley, "rpart: Recursive Partitioning and Regression Trees," [Online]. Available: https://cran.r-project.org/web/packages/rpart/index.html. [Accessed 28 12 2017].

[25] T. Sing, O. Sander, N. Beerenwinkel and T. Lengauer, "ROCR: Visualizing the Performance of Scoring Classifiers," [Online]. Available: https://cran.r-project.org/web/packages/ROCR/index.html. [Accessed 28 12 2017].

[26] Wikipedia, "Sensitivity and specificity definitions," [Online]. Available: https://en.wikipedia.org/wiki/Sensitivity_and_specificity#Definitions. [Accessed 28 12 2017].

[27] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Upper Saddle River, New Jersey: Pearson Prentice Hall, 2008.

[28] M. Sugiyama, Introduction to Statistical Machine Learning, Waltham, MA, USA: Elsevier Inc., 2016.

[29] B. N. Saha, N. Ray and H. Zhang, "Snake Validation: A PCA-Based Outlier Detection Method," *IEEE Signal Processing Letters,* pp. 549 - 552, 2 May 2009.

[30] A. K. Bhattacharya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society,* p. 99–109, 1943.

[31] P. C. Mahalanobis, "On the generalised distance in statistics," *Proceedings of the National Institute of Sciences of India,* pp. 49-55, 15 April 1936.

[32] K. Bulteel, E. Ceulemans, R. J. Thompson, C. E. Waugh, I. H. Gotlib, F. Tuerlinckx and P. Kuppens, "DeCon: a tool to detect emotional concordance in multivariate time series data of emotional responding," *Biological Psychology,* pp. 29-42, April 2014.

[33] M.-P. Sylvestre, J. McCusker, M. Cole, A. Regeasse , E. Belzile and M. Abrahamowicz, "Classification of patterns of delirium severity scores over time in an elderly population," *International psychogeriatrics,* pp. 667-680, 2006.

[34] S. f. S. ETHZ, "rpart.control documentation," [Online]. Available: https://stat.ethz.ch/R-manual/R-devel/library/rpart/html/rpart.control.html. [Accessed 22 12 2017].

[35] A. E. Carpenter and T. R. Jones, "CellProfiler Manual," 2014.

9 Appendix

All data in the appendix is derived from data set 2017-08, unless otherwise stated.

9.1 Variables in the data sets

9.1.1 Variable overview

ID	Variable name	Abbreviation
1	Image_Metadata_Site	
2	objNuc_TrackObjects_Label	
3	objNuc_AreaShape_Area	Area
4	objNuc_AreaShape_Compactness	Compactness
5	objNuc_AreaShape_Eccentricity	Eccentricity
6	objNuc_AreaShape_EulerNumber	Euler number
7	objNuc_AreaShape_Extent	Extent
8	objNuc_AreaShape_FormFactor	Form factor
9	objNuc_AreaShape_MajorAxisLength	Major axis length
10	objNuc_AreaShape_MaxFeretDiameter	Max Feret diameter
11	objNuc_AreaShape_MaximumRadius	Max Radius
12	objNuc_AreaShape_MeanRadius	Mean radius
13	objNuc_AreaShape_MedianRadius	Median radius
14	objNuc_AreaShape_MinFeretDiameter	Min Feret diameter
15	objNuc_AreaShape_MinorAxisLength	Minor axis length
16	objNuc_AreaShape_Orientation	Orientation
17	objNuc_AreaShape_Perimeter	Perimeter
18	objNuc_AreaShape_Solidity	Solidity
19	objNuc_Intensity_IntegratedIntensity_imErk	
20	objNuc_Intensity_IntegratedIntensity_imErkCorrOrig	integrated intensity of ERK-KTR biosensor
21	objNuc_Intensity_IntegratedIntensity_imNuc	
22	objNuc_Intensity_IntegratedIntensity_imNucCorrBg	integrated intensity of nuclear biosensor
23	objNuc_Intensity_MeanIntensity_imErk	
24	objNuc_Intensity_MeanIntensity_imErkCorrOrig	mean intensity of ERK-KTR biosensor
25	objNuc_Intensity_MeanIntensity_imNuc	
26	objNuc_Intensity_MeanIntensity_imNucCorrBg	mean intensity of nuclear biosensor
27	objCyto_Intensity_IntegratedIntensity_imErk	
28	objCyto_Intensity_IntegratedIntensity_imErkCorrOrig	
29	objCyto_Intensity_IntegratedIntensity_imNuc	
30	objCyto_Intensity_IntegratedIntensity_imNucCorrBg	
31	objCyto_Intensity_MeanIntensity_imErk	
32	objCyto_Intensity_MeanIntensity_imErkCorrOrig	
33	objCyto_Intensity_MeanIntensity_imNuc	
34	objCyto_Intensity_MeanIntensity_imNucCorrBg	
35	objNuc_Location_Center_X	
36	objNuc_Location_Center_Y	
37	objNuc_ObjectNumber	
38	RealTime	
39	Metadata_Well	
40	Stimulation_duration	
41	Stimulation_wl	
42	Stimulation_intensity	

43	Stimulation_treatment	

9.1.2 Variable description

The descriptions were partly retrieved from [35].

ID	Description
1	The site of the cells. Per experiment, multiple sites are recorded.
2	The temporary object id used by CellProfiler prior to tracking the objects.
3	The actual number of pixels in the nucleus.
4	The variance of the radial distance of the object's pixels from the centroid divided by the area.
5	The eccentricity of the ellipse that has the same second-moments as the region. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. The value is between 0 and 1. (0 and 1 are degenerate cases; an ellipse whose eccentricity is 0 is a circle, while an ellipse whose eccentricity is 1 is a line segment.)
6	The number of objects in the region minus the number of holes in those objects, assuming 8-connectivity.
7	The proportion of the pixels in the bounding box that are also in the region. Computed as the Area divided by the area of the bounding box.
8	Calculated as $4*\pi*Area/Perimeter^2$. Equals 1 for a perfectly circular object.
9	The length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region.
10	The Feret diameter is the distance between two parallel lines tangent on either side of the object (imagine taking a caliper and measuring the object at various angles). The maximum Feret diameter is the largest possible diameter, rotating the calipers along all possible angles.
11	The maximum distance of any pixel in the object to the closest pixel outside of the object. For skinny objects, this is 1/2 of the maximum width of the object.
12	The mean distance of any pixel in the object to the closest pixel outside of the object.
13	The median distance of any pixel in the object to the closest pixel outside of the object.
14	The minimum Feret diameter is the smallest possible diameter, rotating calipers along all possible angles.
15	The length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region.
16	The angle (in degrees ranging from -90 to 90 degrees) between the x-axis and the major axis of the ellipse that has the same second-moments as the region.
17	The total number of pixels around the boundary of each region in the image.
18	The proportion of the pixels in the convex hull that are also in the object, i.e. ObjectArea / ConvexHullArea. Equals 1 for a solid object (i.e., one with no holes or has a concave boundary), or <1 for an object with holes or possessing a convex/irregular boundary.
19	The sum of the pixel intensities within the nucleus (raw readout).
20	The sum of the pixel intensities within the nucleus (with illumination correction).
21	The sum of the pixel intensities within the nucleus (raw readout).
22	The sum of the pixel intensities within the nucleus (with illumination correction).
23	The average pixel intensity within the nucleus (raw readout).
24	The average pixel intensity within the nucleus (with illumination correction).
25	The average pixel intensity within the nucleus (raw readout).
26	The average pixel intensity within the nucleus (with illumination correction).
27	The sum of the pixel intensities within the cytosol (raw readout).
28	The sum of the pixel intensities within the cytosol (with illumination correction).
29	The sum of the pixel intensities within the cytosol (raw readout).
30	The sum of the pixel intensities within the cytosol (with illumination correction).
31	The average pixel intensity within the cytosol (raw readout).

32	The average pixel intensity within the cytosol (with illumination correction).
33	The average pixel intensity within the cytosol (raw readout).
34	The average pixel intensity within the cytosol (with illumination correction).
35	The pixel X coordinates of the center of mass of the cells.
36	The pixel Y coordinates of the center of mass of the cells.
37	The numeric label assigned to each identified object according to the arrangement order by the CellProfiler software.
38	The time since the start of the experiment, measured in minutes.
39	The well number of the plate used in the experiment.
40	Text describing the stimulation (laser pulses) duration.
41	Text describing the wavelength of the stimulation (laser pulses).
42	Text describing the intensity of stimulation.
43	Text describing the treatment of the cells (inhibitor concentrations if present).

9.1.3 Variable types and ranges

ID	R data type	Unit	Minimum	1st quartile	Median	Mean	3rd quartile	Maximum
1	Integer	dimensionless	0	2	4	3.614	5	7
2	Integer	dimensionless	1	10	19	19.77	28	46
3	Numeric	Pixels	177	387	483	515.5	598	1828
4	Numeric	dimensionless	0.999	1.015	1.038	1.061	1.083	2.546
5	Numeric	dimensionless	0.01488	0.51737	0.63601	0.61646	0.73347	0.97731
6	Integer	dimensionless	1	1	1	1	1	1
7	Numeric	dimensionless	0.3824	0.7389	0.7654	0.7573	0.7855	0.9024
8	Numeric	dimensionless	0.2352	0.8508	0.8837	0.8703	0.9059	0.9719
9	Numeric	Pixels	15.87	26.02	28.63	29.2	31.71	84.78
10	Numeric	Pixels	15.52	25.5	28.02	28.63	31.06	90.82
11	Numeric	Pixels	5	9.22	10.77	10.9	12.21	19.1
12	Numeric	Pixels	2.014	3.807	4.251	4.314	4.742	7.19
13	Numeric	Pixels	2	3.606	4	3.858	4.123	6.403
14	Numeric	Pixels	10	18	21	21.36	24.04	38.18
15	Numeric	pixels	11.02	18.87	21.64	22.15	24.95	39.11
16	Numeric	degrees	-90	-45.015	2.165	1.313	47.079	90
17	Numeric	Pixels	48.87	75.01	83.25	85.17	92.66	205.32
18	Numeric	dimensionless	0.6456	0.9474	0.9527	0.9507	0.9573	1.0026
19	Numeric	integrated intensity units	6.383	24.411	30.219	36.352	40.498	253.7
20	Numeric	integrated intensity units	4.038	20.125	25.497	31.23	34.922	244.326
21	Numeric	integrated intensity units	11.48	30.87	40.03	45.24	54.89	177.98
22	Numeric	integrated intensity units	2.916	12.71	19.025	23.154	29.096	119.404
23	Numeric	mean intensity units	0.02024	0.05337	0.06641	0.06982	0.08107	0.22104
24	Numeric	mean intensity units	0.01155	0.04388	0.05627	0.05985	0.07061	0.21134
25	Numeric	mean intensity units	0.04394	0.07132	0.08343	0.0872	0.09955	0.20587

26	Numeric	mean intensity units	0.01169	0.02880	0.03880	0.04401	0.05532	0.16257
27	Numeric	integrated intensity units	0.0232	16.0192	29.3743	36.5888	45.7853	440.1916
28	Numeric	integrated intensity units	0.0155	10.7535	20.0215	26.1747	32.255	388.2141
29	Numeric	integrated intensity units	0.04167	26.02673	45.29696	51.76968	67.67177	259.24964
30	Numeric	integrated intensity units	0.00908	3.58706	6.05107	7.02527	8.92333	46.28113
31	Numeric	mean intensity units	0.02236	0.029	0.03146	0.03291	0.03516	0.011308
32	Numeric	mean intensity units	0.01501	0.01926	0.02131	0.02294	0.02484	0.10203
33	Numeric	mean intensity units	0.03473	0.04471	0.05044	0.0501	0.05519	0.06837
34	Numeric	mean intensity units	0.004546	0.006217	0.006766	0.006939	0.007387	0.020446
35	Numeric	Pixels	17.29	268.56	481.55	484.83	675.41	1007.25
36	Numeric	Pixels	11.65	265.9	477.65	488.06	702.96	992.35
37	Numeric	dimensionless	1	11	20	20.85	30	50
38	Integer	minutes	0	142	285	284.5	427	569
39	Integer	dimensionless	1	3	5	4.614	6	8
40	Character	values: "100ms"						
41	Character	values: "470/10 + ND5"						
42	Character	values: "2-5-10-20-40-60-80-100-100% light"						
43	Character	values: "UO126 50uM" / "UO126 10uM" / ""						

9.2 Correlation coefficients of all nuclear shape variables

ID / ID	3	4	5	7	8	9	10	11	12	13	14	15	16	17	18
3	1.00	-0.20	-0.26	0.04	0.04	0.87	0.89	0.91	0.95	0.95	0.93	0.92	0.01	0.97	0.23
4	-0.20	1.00	0.76	-0.60	-0.81	0.26	0.21	-0.53	-0.43	-0.38	-0.48	-0.50	-0.02	-0.04	-0.36
5	-0.26	0.76	1.00	-0.45	-0.60	0.20	0.15	-0.58	-0.45	-0.39	-0.56	-0.58	-0.01	-0.15	-0.18
7	0.04	-0.60	-0.45	1.00	0.68	-0.23	-0.22	0.26	0.22	0.19	0.19	0.22	0.04	-0.10	0.45
8	0.04	-0.81	-0.60	0.68	1.00	-0.31	-0.29	0.36	0.29	0.25	0.25	0.29	0.03	-0.17	0.67
9	0.87	0.26	0.20	-0.23	-0.31	1.00	0.99	0.62	0.73	0.75	0.65	0.64	0.00	0.92	0.11
10	0.89	0.21	0.15	-0.22	-0.29	0.99	1.00	0.65	0.75	0.78	0.69	0.67	0.00	0.94	0.12
11	0.91	-0.53	-0.58	0.26	0.36	0.62	0.65	1.00	0.98	0.95	0.98	0.99	0.01	0.83	0.35
12	0.95	-0.43	-0.45	0.22	0.29	0.73	0.75	0.98	1.00	0.98	0.97	0.98	0.02	0.88	0.37
13	0.95	-0.38	-0.39	0.19	0.25	0.75	0.78	0.95	0.98	1.00	0.95	0.95	0.01	0.89	0.36
14	0.93	-0.48	-0.56	0.19	0.25	0.65	0.69	0.98	0.97	0.95	1.00	1.00	0.02	0.87	0.22
15	0.92	-0.50	-0.58	0.22	0.29	0.64	0.67	0.99	0.98	0.95	1.00	1.00	0.02	0.86	0.25
16	0.01	-0.02	-0.01	0.04	0.03	0.00	0.00	0.01	0.02	0.01	0.02	0.02	1.00	0.00	0.01
17	0.97	-0.04	-0.15	-0.10	-0.17	0.92	0.94	0.83	0.88	0.89	0.87	0.86	0.00	1.00	0.07
18	0.23	-0.36	-0.18	0.45	0.67	0.11	0.12	0.35	0.37	0.36	0.22	0.25	0.01	0.07	1.00

9.3 Correlation coefficients of all selected variables

ID / ID	3	4	5	7	8	10	14	17	18	20	22	24	26
3	1.00	-0.20	-0.26	0.04	0.04	0.89	0.93	0.97	0.23	0.67	0.63	0.08	0.11
4	-0.20	1.00	0.76	-0.60	-0.81	0.21	-0.48	-0.04	-0.36	-0.12	-0.18	-0.04	-0.16
5	-0.26	0.76	1.00	-0.45	-0.60	0.15	-0.56	-0.15	-0.18	-0.15	-0.18	-0.03	-0.12
7	0.04	-0.60	-0.45	1.00	0.68	-0.22	0.19	-0.10	0.45	0.03	0.08	0.03	0.14
8	0.04	-0.81	-0.60	0.68	1.00	-0.29	0.25	-0.17	0.67	0.04	0.14	0.05	0.21
10	0.89	0.21	0.15	-0.22	-0.29	1.00	0.69	0.94	0.12	0.57	0.53	0.03	0.04
14	0.93	-0.48	-0.56	0.19	0.25	0.69	1.00	0.87	0.22	0.59	0.58	0.06	0.13
17	0.97	-0.04	-0.15	-0.10	-0.17	0.94	0.87	1.00	0.07	0.62	0.58	0.04	0.07
18	0.23	-0.36	-0.18	0.45	0.67	0.12	0.22	0.07	1.00	0.13	0.20	0.00	0.14
20	0.67	-0.12	-0.15	0.03	0.04	0.57	0.59	0.62	0.13	1.00	0.64	0.74	0.31
22	0.63	-0.18	-0.18	0.08	0.14	0.53	0.58	0.58	0.20	0.64	1.00	0.34	0.79
24	0.08	-0.04	-0.03	0.03	0.05	0.03	0.06	0.04	0.00	0.74	0.34	1.00	0.36
26	0.11	-0.16	-0.12	0.14	0.21	0.04	0.13	0.07	0.14	0.31	0.79	0.36	1.00

9.4 Principal Component Analysis

9.4.1 Cumulative contribution to variance

PC / ID	3	4	5	7	8	10	14	17	18	20	22	24	26
PC1	0.868	0.623	0.693	0.419	0.505	0.770	0.862	0.836	0.401	0.920	0.949	0.888	0.939
PC2	0.917	0.739	0.805	0.527	0.620	0.835	0.919	0.885	0.549	0.947	0.983	0.940	0.979
PC3	0.937	0.799	0.859	0.600	0.687	0.866	0.942	0.907	0.643	0.958	0.991	0.957	0.986
PC4	0.951	0.829	0.886	0.638	0.725	0.892	0.955	0.924	0.698	0.966	0.993	0.971	0.989
PC5	0.960	0.856	0.901	0.670	0.751	0.911	0.964	0.936	0.732	0.973	0.995	0.978	0.991
PC6	0.969	0.879	0.914	0.698	0.773	0.929	0.970	0.948	0.745	0.978	0.996	0.982	0.993
PC7	0.976	0.897	0.925	0.722	0.791	0.943	0.974	0.957	0.759	0.982	0.997	0.985	0.995
PC8	0.980	0.910	0.934	0.738	0.805	0.952	0.978	0.963	0.770	0.985	0.997	0.987	0.996
PC9	0.983	0.922	0.941	0.752	0.816	0.959	0.981	0.969	0.778	0.988	0.998	0.989	0.996
PC10	0.985	0.930	0.947	0.763	0.826	0.964	0.983	0.973	0.785	0.990	0.998	0.990	0.997
PC11	0.987	0.937	0.951	0.772	0.832	0.968	0.984	0.976	0.791	0.992	0.999	0.992	0.997
PC12	0.989	0.943	0.955	0.780	0.839	0.971	0.986	0.978	0.796	0.993	0.999	0.993	0.997
PC13	0.990	0.949	0.958	0.786	0.844	0.974	0.987	0.980	0.802	0.994	0.999	0.993	0.998
PC14	0.991	0.954	0.961	0.792	0.849	0.977	0.988	0.981	0.806	0.995	0.999	0.994	0.998
PC15	0.992	0.958	0.963	0.797	0.853	0.980	0.989	0.983	0.811	0.996	0.999	0.995	0.998
PC16	0.993	0.962	0.965	0.802	0.857	0.982	0.990	0.984	0.815	0.996	0.999	0.995	0.998
PC17	0.994	0.965	0.967	0.807	0.861	0.983	0.991	0.986	0.819	0.997	0.999	0.996	0.998
PC18	0.994	0.968	0.969	0.812	0.864	0.985	0.991	0.987	0.823	0.997	0.999	0.996	0.999
PC19	0.995	0.970	0.971	0.816	0.868	0.986	0.991	0.988	0.826	0.997	1.000	0.997	0.999
PC20	0.996	0.972	0.972	0.820	0.871	0.988	0.992	0.989	0.830	0.998	1.000	0.997	0.999

9.4.2 Class separability (Bhattacharyya distance)

PC / ID	3	4	5	7	8	10	14	17	18	20	22	24	26
PC1	0.010	0.002	0.007	0.007	0.023	0.004	0.006	0.004	0.064	0.100	0.018	0.033	0.004
PC2	0.231	0.059	0.014	0.053	0.136	0.217	0.129	0.307	0.268	0.068	0.071	0.035	0.002
PC3	0.326	0.072	0.008	0.054	0.121	0.259	0.181	0.260	0.264	0.583	0.226	0.095	0.039
PC4	0.659	0.101	0.074	0.028	0.461	0.587	0.260	0.636	0.318	0.076	0.510	0.006	0.109
PC5	0.801	0.192	0.051	0.055	0.133	0.543	0.200	0.498	0.086	0.739	0.559	0.083	0.252
PC6	0.372	0.309	0.049	0.051	0.240	0.484	0.381	0.568	0.001	0.129	0.235	0.207	0.178
PC7	0.507	0.266	0.103	0.115	0.237	0.305	0.312	0.672	0.143	0.400	0.792	0.284	0.532
PC8	0.409	0.331	0.104	0.066	0.283	0.490	0.293	0.710	0.039	0.317	0.763	0.055	0.265
PC9	0.471	0.305	0.023	0.079	0.190	0.352	0.190	0.494	0.215	0.515	0.549	0.058	0.440
PC10	0.578	0.136	0.076	0.131	0.227	0.468	0.109	0.344	0.116	0.451	0.706	0.014	0.014
PC11	0.477	0.402	0.017	0.056	0.123	0.564	0.105	0.421	0.052	0.564	0.848	0.175	0.262
PC12	0.173	0.395	0.097	0.025	0.097	0.289	0.109	0.698	0.065	0.279	0.664	0.119	0.198
PC13	0.238	0.289	0.136	0.130	0.173	0.680	0.099	0.406	0.441	0.415	0.615	0.115	0.252
PC14	0.474	0.120	0.124	0.109	0.293	0.183	0.222	0.142	0.129	0.158	0.329	0.061	0.283
PC15	0.729	0.221	0.121	0.100	0.117	0.538	0.241	0.695	0.146	0.292	0.215	0.112	0.244
PC16	0.700	0.190	0.038	0.098	0.176	0.406	0.165	0.437	0.160	0.161	0.293	0.174	0.233
PC17	0.249	0.275	0.071	0.142	0.207	0.436	0.148	0.413	0.004	0.201	0.673	0.084	0.251
PC18	0.634	0.171	0.147	0.090	0.125	0.431	0.179	0.328	0.044	0.262	0.832	0.107	0.296
PC19	0.158	0.156	0.122	0.039	0.019	0.482	0.077	0.316	0.190	0.478	0.177	0.144	0.343
PC20	0.764	0.148	0.180	0.061	0.154	0.437	0.079	0.216	0.157	0.080	0.799	0.083	0.184

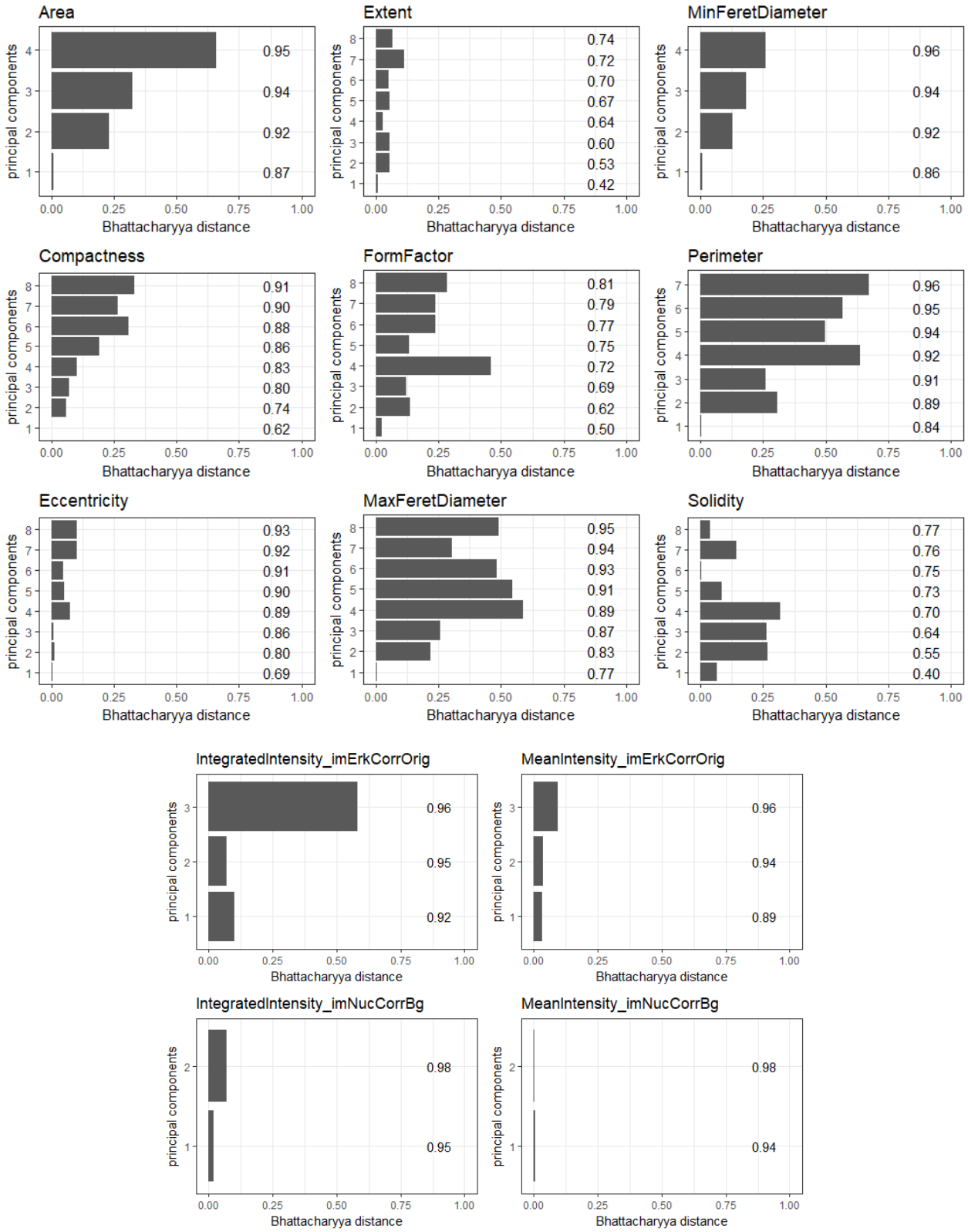
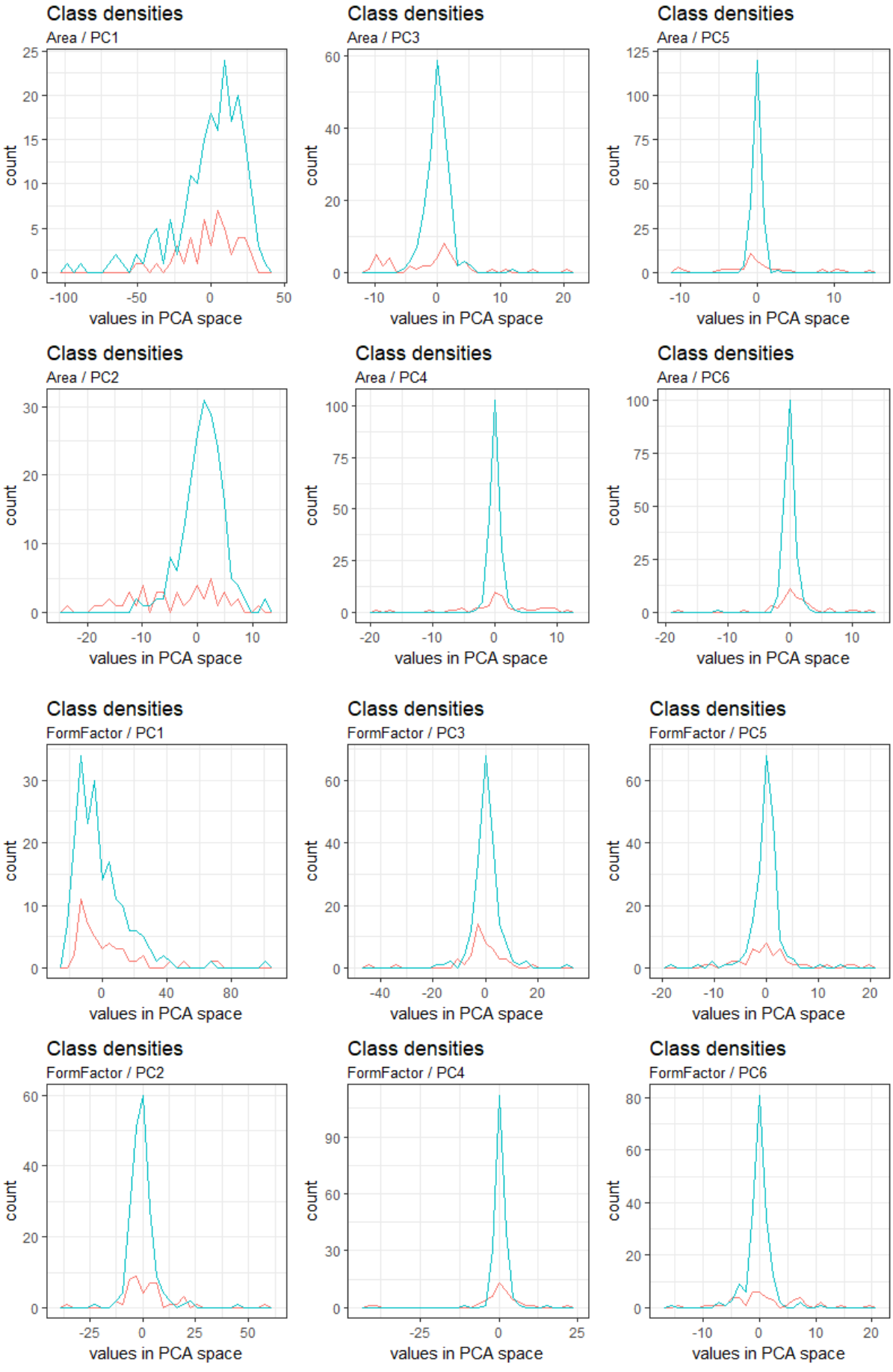
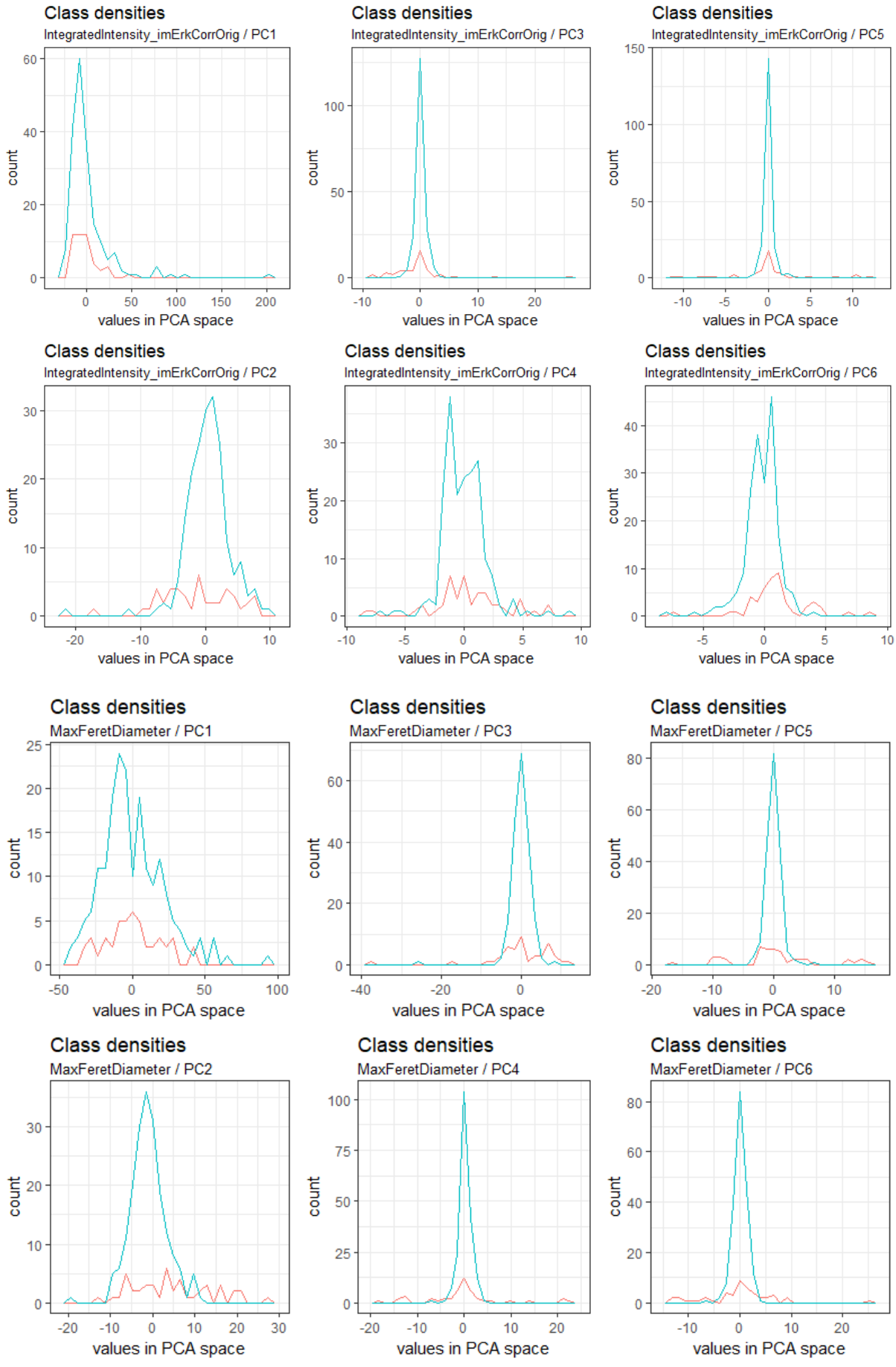
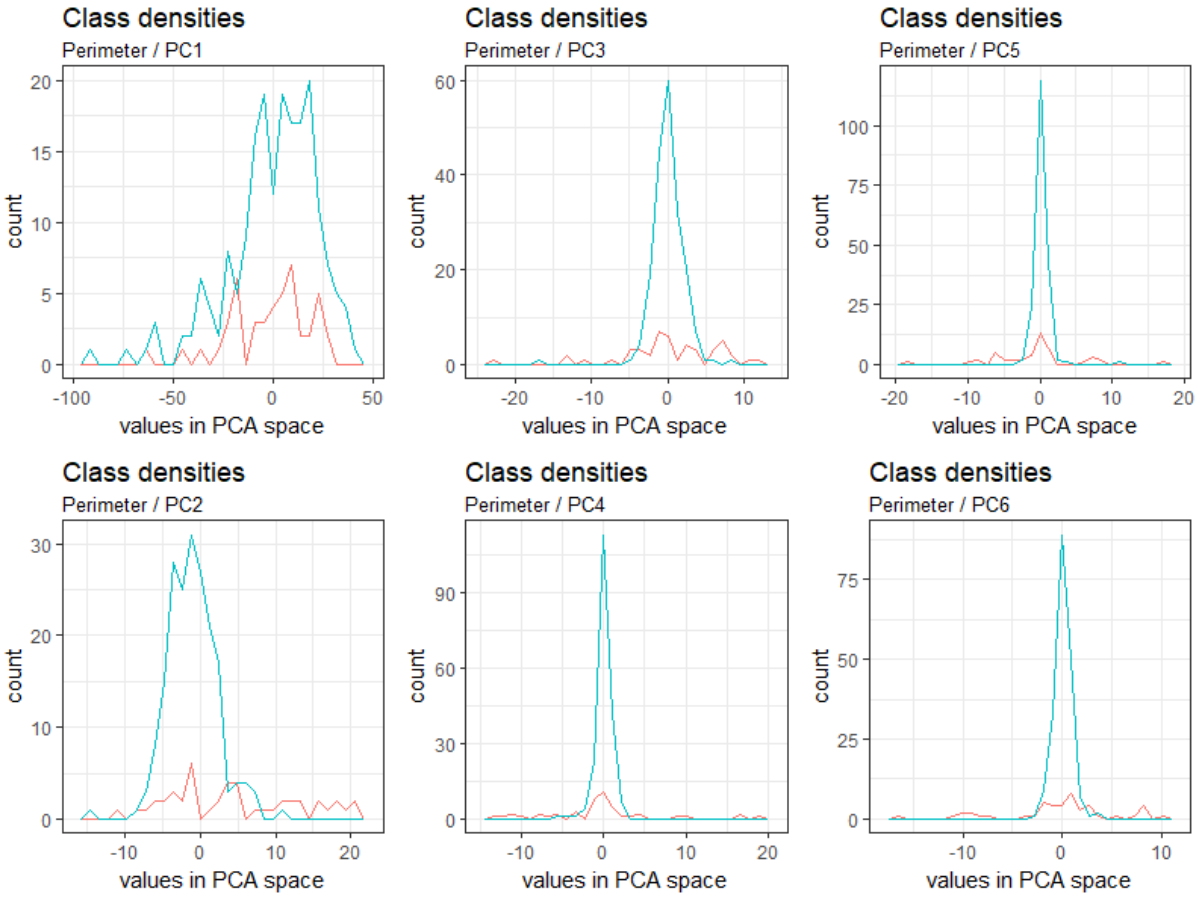


Figure 20: Plots showing Bhattacharyya distance between the two cell classes per feature and principal component. The numbers on the right side of the plots indicate the cumulative contribution to variance by the principal components.

Class separation visualized:


Class densities
Area / PC1
count
values in PCA space
Class densities
Area / PC3
count
values in PCA space
Class densities
Area / PC5
count
values in PCA space
Class densities
Area / PC2
count
values in PCA space
Class densities
Area / PC4
count
values in PCA space
Class densities
Area / PC6
count
values in PCA space
Class densities
FormFactor / PC1
count
values in PCA space
Class densities
FormFactor / PC3
count
values in PCA space
Class densities
FormFactor / PC5
count
values in PCA space
Class densities
FormFactor / PC2
count
values in PCA space
Class densities
FormFactor / PC4
count
values in PCA space
Class densities
FormFactor / PC6
count
values in PCA space


Class densities
IntegratedIntensity_imErkCorrOrig / PC1
Class densities
IntegratedIntensity_imErkCorrOrig / PC3
Class densities
IntegratedIntensity_imErkCorrOrig / PC5
Class densities
IntegratedIntensity_imErkCorrOrig / PC2
Class densities
IntegratedIntensity_imErkCorrOrig / PC4
Class densities
IntegratedIntensity_imErkCorrOrig / PC6
Class densities
MaxFeretDiameter / PC1
Class densities
MaxFeretDiameter / PC3
Class densities
MaxFeretDiameter / PC5
Class densities
MaxFeretDiameter / PC2
Class densities
MaxFeretDiameter / PC4
Class densities
MaxFeretDiameter / PC6
count
values in PCA space



9.4.3 Classification results

Dataset 2017-08

Table 22: Confusion Matrix, all for threshold value of 0.25

Area PC 4			**Area PC 4 + Form Factor PC 4**		
	bad	good		bad	good
bad	20	2	bad	23	5
good	26	190	good	23	187
Form Factor PC 4			**Area PC 4 + Integrated Erk PC 3**		
	bad	good		bad	good
bad	11	3	bad	25	7
good	35	189	good	21	185
Integrated Erk PC 3			**Form Factor PC 4+ Integrated Erk PC 3**		
	bad	good		bad	good
bad	18	6	bad	26	9
good	28	186	good	20	183
			All three variables		
				bad	good
			bad	28	10
			good	18	182

Table 23: Confusion Matrix, all for threshold value of 0.5

Area PC 4	**Area PC 4 + Form Factor PC 4**

	bad	good
bad	20	2
good	26	190

	bad	good
bad	23	5
good	23	187

Form Factor PC 4		
	bad	good
bad	11	3
good	35	189

Area PC 4 + Integrated Erk PC 3		
	bad	good
bad	25	7
good	21	185

Integrated Erk PC 3		
	bad	good
bad	18	6
good	28	186

Form Factor PC 4+ Integrated Erk PC 3		
	bad	good
bad	26	9
good	20	183

All three variables		
	bad	good
bad	28	10
good	18	182

Table 24: Confusion Matrix, all for threshold value of 1.0

Area PC 4		
	bad	good
bad	20	2
good	26	190

Area PC 4 + Form Factor PC 4		
	bad	good
bad	23	5
good	23	187

Form Factor PC 4		
	bad	good
bad	10	3
good	36	189

Area PC 4 + Integrated Erk PC 3		
	bad	good
bad	25	7
good	21	185

Integrated Erk PC 3		
	bad	good
bad	17	5
good	29	187

Form Factor PC 4+ Integrated Erk PC 3		
	bad	good
bad	25	8
good	21	184

All three variables		
	bad	good
bad	28	10
good	18	182

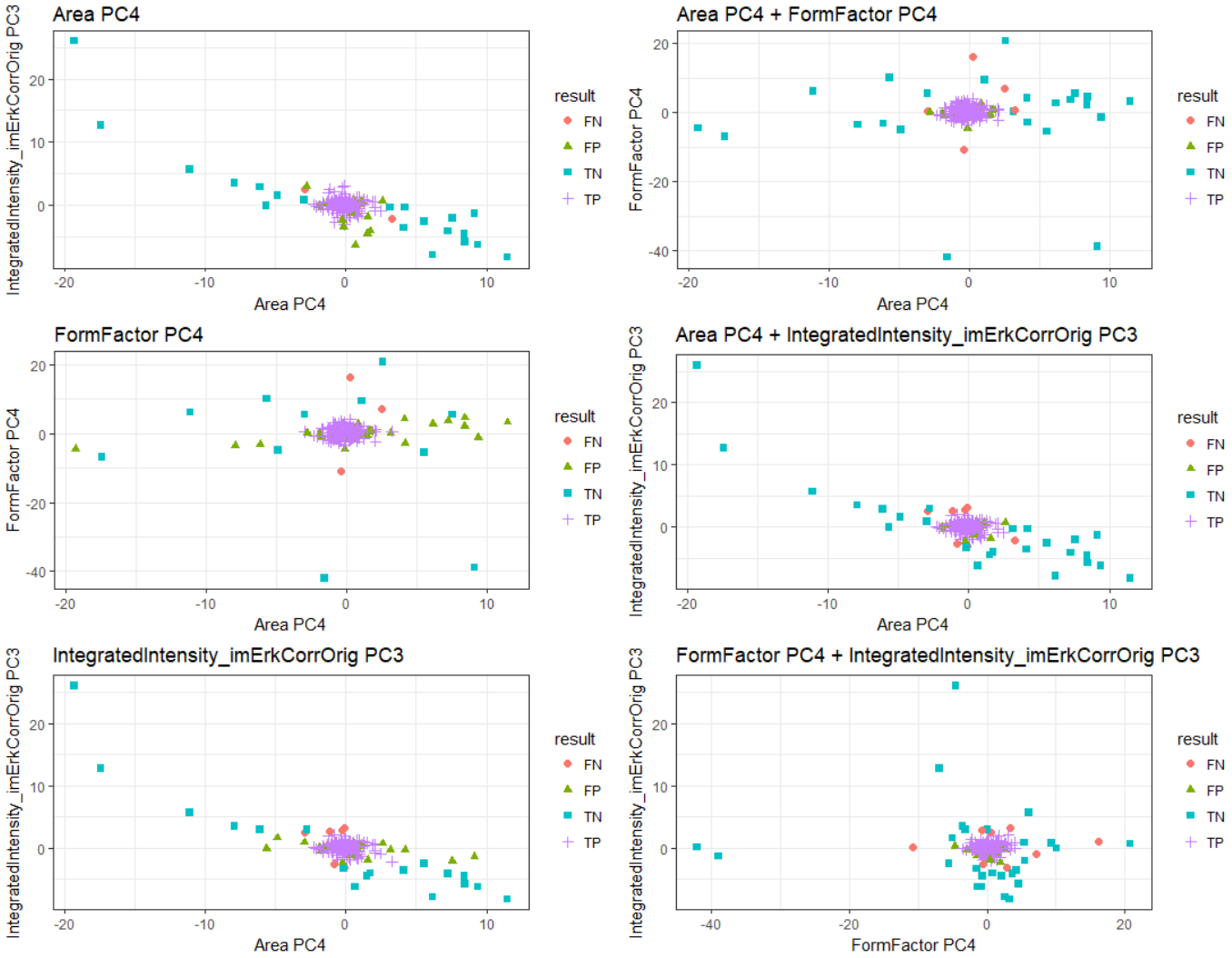


Figure 21: Plots for T = 0.5

Dataset 2017-07

Table 25: Confusion Matrix, all for threshold value of 0.25

Area PC 4		
	bad	good
bad	33	5
good	29	189

Area PC 4 + Form Factor PC 4		
	bad	good
bad	45	29
good	17	165

Form Factor PC 4		
	bad	good
bad	29	25
good	33	169

Area PC 4 + Integrated Erk PC 3		
	bad	good
bad	41	12
good	21	182

Integrated Erk PC 3		
	bad	good
bad	21	7
good	41	187

Form Factor PC 4+ Integrated Erk PC 3		
	bad	good
bad	40	30
good	22	164

All three variables		
	bad	good
bad	48	34
good	14	160

Table 26: Confusion Matrix, all for threshold value of 0.5

Area PC 4		
	bad	good
bad	33	3
good	29	191

Area PC 4 + Form Factor PC 4		
	bad	good
bad	45	23
good	17	171

Form Factor PC 4		
	bad	good
bad	28	21
good	34	173

Area PC 4 + Integrated Erk PC 3		
	bad	good
bad	41	9
good	21	185

Integrated Erk PC 3		
	bad	good
bad	21	6
good	41	188

Form Factor PC 4+ Integrated Erk PC 3		
	bad	good
bad	39	25
good	23	169

All three variables		
	bad	good
bad	48	27
good	14	167

Table 27: Confusion Matrix, all for threshold value of 1.0

Area PC 4		
	bad	good
bad	31	3
good	31	191

Area PC 4 + Form Factor PC 4		
	bad	good
bad	40	19
good	22	175

Form Factor PC 4		
	bad	good
bad	23	17
good	39	177

Area PC 4 + Integrated Erk PC 3		
	bad	good
bad	37	7
good	25	187

Integrated Erk PC 3		
	bad	good
bad	19	4
good	43	190

Form Factor PC 4+ Integrated Erk PC 3		
	bad	good
bad	33	20
good	29	174

All three variables		
	bad	good
bad	42	22
good	20	172

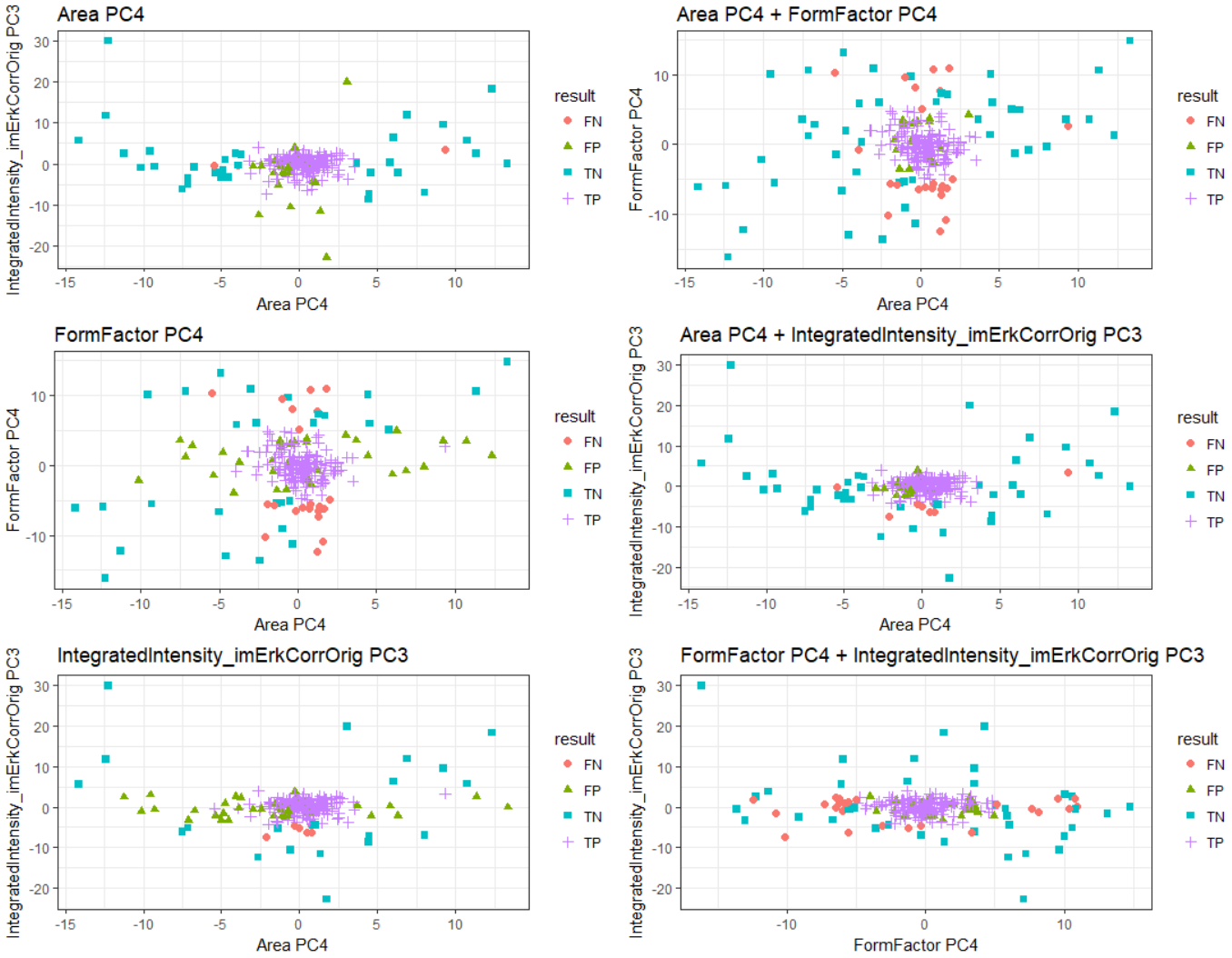


Figure 22: Plot for T = 0.5

11 Declaration of Authorship

I hereby certify that I composed this work completely unaided, and without the use of any other sources or resources other than those specified in the bibliography. All text sections not of my authorship are cited as quotations, and accompanied by an exact reference to their origin.

Place, date:

Signature: