University of Szeged

Department of Biotechnology

Single Cell analysis using regression

Thesis

Written by:

Attila, Beleon

Molecular Bionic Engineering
BSc student

Supervisors:

Gábor Rákhely, PhD Habil

Head of Department

Department of Biotechnology

University of Szeged

Péter Horváth, PhD

Director, Institute of Biochemistry

Group Leader BIOMAG

BRC HAS, Szeged

Szeged, 2019

Task

The task of the student is to design and implement trajectory visualiser tool based on existing active regression pipeline, and evaluate the performance on biological data.

More detailed explanation:

1. Give an overview of high-content screening and show the importance of algorithm development for large-scale biological data.

2. Give on overview of biological image processing methods. Compare classification and regression.

3. Evaluate the benefits of trajectory visualization based on regression.

4. Implement a software capable of visualize regression based trajectories of cell lines. The software should grant filtering options and provide a user-friendly environment to help biology experts to simultaneously evaluate large amount of data.

5. Run the software on a mitotic and a cellular differentiation biological data.

Abstract

The title of my topic is *Single Cell analysis using regression.* My task is to complement an already existing active regression software with trajectory visualization module and examine its capabilities in revealing hidden biological correlations. Test the developed methods on mitotic, and cellular differentiation data.

The first step was to be familiar with the Advanced Cell Classifier (ACC)[**?**] software, regression methods and learn to code in MATLAB what was also new for me. The programming language was given owing to the implementation of ACC. After the examination of regression results, I looked for a suitable tracking method to generate cell lines as the algorithm assign the coordinates of the regression result that belongs to the same cell on the original microscopy image.

I implemented a graphical user interface which calculates trajectory for each cell lines and visualize them on coordinate system. I also aimed at the optimization to speed up the run-time. Later I installed several filtering options and dynamic selection tools to be able to examine the data co-operated with biologist researchers to improve the efficiency of the module. During the development, I reviewed the literature of regression methods and machine learning.

Through a series of tests and examinations, the Trajectory module made it possible to observe the distribution and the dynamic of a newly discovered trans-differentiation of drosophila melanogaster blood cell.

Keywords: high-content screening, regression, trajectory, cell-tracking, single-cell, drosophila melanogaster, lamellocyte.

Contents

List of abbreviations

ACC	Advanced Cell Classifier	p.3
AL	Active learning	p.11
GUI	Graphical User Interface	p.16
HCS	High-content screening	p.10
ML	Machine Learning	p.11
SL	Supervised learning	p.11
TS	Training Set	p.11
USL	Unsupervised learning	p.11

Introduction

Microscopy methods develop rapidly. The improvement in resolution allows gathering more data from a single image. Large imaging scenarios, including high-content screening and tissue section imaging, became commonly used tools for discovering drugs, genes and understanding various tissue physiologies such as cancer heterogeneity. The performance of microscopes reached a level that allows creating millions of images in a short period. Manual image evaluation methods are no longer viable options. It is an urgent need for bioinformatics tools to discover, examine and deepen the understanding of the underlying biological processes.

Computational image processing in cell biology is a breakthrough field that widens the horizon in research, that makes able to get fast feedback from experiments. These techniques provide information about every single cell and the entire population at the same time. The possibility is given to apply multiple measurements within the same experiment or reuse samples.

In medical fields, modern automatic screening technology augmented with robust image processing forms a pipeline that offers possibilities for cancer detection in an earlier stage and chances to respond faster to increase the survivability of the patients.

Image analysis and machine learning are fields which intend to substitute knowledge of an expert and achieve the same analysis quality. Humans are losing focus as they are getting tired. The number of mistakes increasing by time. Machines perform the task much faster and in a more objective way.

Image analysis extracts parameters from raw images. The number of parameters can reach few hundreds, usually describe the morphology, texture and intensity properties of the cells. These properties of the cell are called features and organised in a vector. Most of the features are slightly representative for humans. To give an illustration, the value of the standard deviation of intensity, or the eccentricity of the cell can not interpret as an absolute value, but we can compare them to other cells and investigate the differences between them. Likewise, similarity and diversity help us to categorize and create classes of cells with similar properties and behaviours. It is an obvious decision to use machine learning to deal with this amount of data.

Machine learning is a field of computer science, which gives computers the ability to learn without being explicitly programmed. Two main categories can be distinguished. Unsupervised learning is generally used for structures revealing in data, based on local densities in the feature space. Supervised learning uses annotated samples to train a learner function. It attempts to find similarities in feature space in relations with the training strategy. The quality and abundance of features are crucial for the process to find similarities. Increasing the number of elements in the training set improves the performance of the prediction. The precise selection of examples for training is also essential to reduce time consumption and to avoid redundancy. Some procedures can help the user to avoid these problems. One of such we used is Active Learning. (will be described later)

Nature rarely instantaneous rather a continuous series of events. For the prediction, we used regression analysis techniques. These are statistical processes for estimating the relationships between dependent variables, suitable to examine continuous biological processes.

Visualization techniques perform major components in results evaluation. There are various approaches (graphs, heatmaps, histograms, animation, others) to present helpful information about the outcomes. The structure of the data and the nature of the needed information implies the technique to use to answer the raised questions. In this thesis, I attempt to present the pipeline mentioned above and the results.

Chapter 1

Pipeline

1.1 Sample preparation

As a rule, the first step is the planing of the experiment. We should raise questions about a given biological phenomenon, set goals what we would like to achieve. Following that, we should find out which cellular attributes or components can be a beneficial indicator for the examination. For example, if we are interested in the size of the cell or the nuclei, we should mark the membranes of the cell to make it visually distinguishable. To indicate mitotic phases, it is recommended to use combinations of markers, different colour of fluorescent markers to make visible tubulins, membranes and the genomic material.

./Figures/PipeLine/sample_prep.png

Figure 1.1: Simplified pipeline in system biology

Well-chosen indicators remarkably improve the performance of machine learning algorithms and the success rate of experiments. To choose the proper microscopy method is also an essential. The technology should be compatible with cell and markers.

1.2 High-content screening

High-Content Screening (HCS) is the product of the imaging technology revolution. The capability of screening cellular or even subcellular scale combined with high-throughput techniques dramatically speeds up the pipelines of cellular biology experiments.[?] HCS makes able to analyse a large number of samples in a relatively short period. This ability is crucial in live-experiments. Theoretically, if we would like to investigate continuous actions with many participants on an area that many times larger than our field-of-view, we have no clue what is happening on the whole field. If it takes lots of time to screen the entire area, the elapsed time will produce continues shift-error in our data. The shift-error can be such high it makes the evaluation impossible. HCS brakes this limit and gives the chance to prepare more advanced experiments that promising far-reaching results.

The improvement, compared to the previous generation of screening technology, the high-throughput screening, was an advance focusing technologies and advanced softwares. High-throughput microscopes had already installed with high-performance cameras and objectives. But scanning an area is not a simple task. Moving sideways on the sample will result the loss of focus. Laser-based auto-focus detection solved this problem.[?]

1.3 Image analysis

From the computer point of view, an image is represented as a matrix. Every pixel of an image represents a matrix element, and its value is the intensity of the pixel. We are interested in cellular properties on images. The first step is segmentation when a human or an algorithm are drawing contours around cells. Segmentation by hand is time-consuming, machines are preferable in this field, but there are some difficulties here. Usually, the cells are connected, as in a tissue sample, and it is not trivial how to design such algorithms. Convolutional Deep Neural Network is suitable for this task with high efficiency unless there is only 1 pixel is the boundary between two cells. This problem originates from the nature of the convolution operation, because convolution decreases the dimension of the image. If the boundary is only 1 pixel, we will lose this information by the convolution. Fortunately, we do not have to face this problem during our experiments, because cells were separated.

After the segmentation, we have the contours of cells, so we can create masks to cover areas that out of interest. The other algorithm start extraction the properties of the cells, name these as features. Features have based on morphology (area, shape, eccentricity, Euler number), pixel intensity (minimum-maximum, variance, standard deviation) plus texture (entropy, variance). These values have organised in vectors describing the cell. [**?**]

1.4 Machine learning

Machine learning (ML) is a discipline of Artificial Intelligence. ML uses self-optimizing functions. They are versatile because they do not need explicit instructions to solve problems. ML founded on computational-statistic, uses probabilities to predict the most probable outcome. Performance of ML measured by cross-validation[**?**] and calculation of errors.[**?**]

The machine identifies cells with the previously extracted feature vectors. This part is where we should raise questions about the biology behind. There are two different methods to chose. If we are lack of *a priori* information about the cells, the treatments, mutation or we use a brand new strain, the proper option to use is Unsupervised learning (USL). USL is suitable to find underlying structures in the data, without any user action, or previously given knowledge. USL is looking for high-density areas in feature space and categorize cells by similarity.

The other option is Supervised Learning (SL). In the case of SL, the expert has some expectation how the result should looks like, what aspect of cells are interested. SL needs a so-called Training Set (TS). TR is a subsample of the whole data, that represents a strategy given by the expert. The machine has trained with the TS, then predict the rest of the cells followed the strategy.

There are theories what is a good TS, and how to optimize the sampling for the TS. The performance of ML heavily dependent on the number of elements in the TS. Extending the TS is time-consuming, furthermore increasing the probability of redundancy. We used the Active Learning (AL) method for optimization. AL is asking "clever" question from the expert, by looking through the data and showing elements that are "hard" to predict.

The biologist expands the TS with these elements, then the AL continues its task. The result is a more precise, optimal TS with low redundancy.

ML is a widely used method in science. The reason is that ML is super fast and objective compared to a human. Modern implementations can perform the same accuracy as experts.

1.5 Classification and Regression

In the field of biology, ML algorithms are commonly used for performing classification. Practically, it classifies the instances into user-defined classes that represent phenotypes. The algorithm is iterating through the entire data and comparing them to the TS. Cells will be placed into that class where the distance in features space is minimum. One of the difficulties is to determine how many classes needed to answer the previously raised biological questions. Ordinarily, there are outlier cells that greatly different from the training set, or equally similar to two classes. These cells are near the theoretical boundary of a class. To solve the problem of the outliers we can create new classes, if we would like to increase the resolution, we should add new classes between two other classes, for those cells that are close to the boundary. Adding one extra class between two already existing class will increases the number of boundaries. This process going to get out of control until at some point we will have equal or more amount of classes than cells. To avoid this time-consuming practice, regression gives a handy solution.[**?**]

Differently to the classification, regression does not have classes but also has a continuous field. Regression is more than a continuous way of classification. If the number of class is equal to the number of cells, every class will contains only one cell, but distances in feature space between classes are still not visible this way, moreover, the process becomes pointless because it is not an automated classification anymore if we classify all the cells for the TS. Regression has the advantage to present the differences between cells in a continuous way. Theoretically, a two-dimensional field for regression can dramatically increase the performance and degrees of freedom, it is capable to learn the deeply complex network of paths until it can be represented as a planar graph. Regression also overcome the problem of objective prediction, because even an expert is not able to perceive slight differences between cells, or if they can it is impossible to tell the magnitude of difference between two cells in an objective way. **??**

Regression method also has difficulties. It is not obvious how to create consequent TS. AL methods can be applied to regression, it called active regression, that can increase the consistency of the TS.

1.6 Visualization

Visualization techniques play a fundamental role in evaluation of experiments. Three main domains of visualization are data, information and knowledge visualization, which indicate different levels of abstraction.

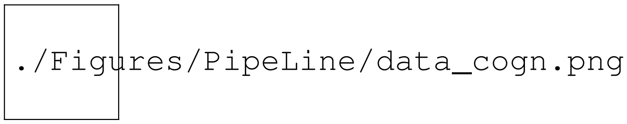

Figure 1.2: Russell Ackhoff's definition of data, information and knowledge in perceptual and cognitive space. [**?**]

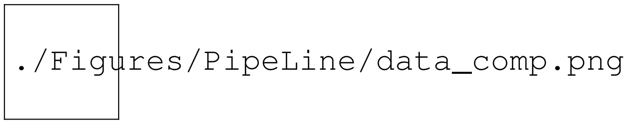

Figure 1.3: Min Chen's definitions of data, information and knowledge in computational space. [**?**]

The storage and representation of these levels can happen in two spaces. One is the computational space, other is perceptual and cognitive space (practically the human brain). Visualization is the most efficient and effective way to transfer data, information and knowledge from the computational space to the perceptual and cognitive space.

Some techniques were already implemented in ACC. Different heatmaps that represents the distribution of cells, and cluster-gram to show correlation between different treatments.

./Figures/heatmap.png

Figure 1.4: Heatmap of regression result, with resolution value 5 (felt), and 10 (right).

./Figures/distanceBasedCulstergram.png

Figure 1.5: Distance-based clustergram. Lines on the left side showing hierarchical connections between the different treatments.

Chapter 2

Trajectory module

Trajectory Plot is a multi functional visualization tool that contributes to our understanding of the continuous aspect of biology. Trajectory plot offers several features to investigate cell path and fate and compare the development of particular cells in time. This approach provides alternative measurements (abstract distances, timing, time-consuming of different stages, proliferation and differentiation speed). Decent frequency of time sampling is crucial. The shorter elapsed time between taken images results in smoother trajectories that could reveal remarkable details.

2.1 Input

In our lamellocyte experiment, a series of images were taken from all the chosen fields of the sample in every two hours. We ended up with 2115 images. After the segmentation part, we extracted the features. The name of the extracted features can be found in the supplementary.**??**

Biologist expert trained the Regression Plane with 147 images. The strategy was the following. After the evaluation of several machine learning algorithms, we found out that the best result was given by an algorithm called WEKA-RandomForest[**?**]. Using these parameters we predicted the regression coordinates of all cells.

2.1.1 Tracking

Before generating trajectories, we have to append the coordinates from different frames of the same cell. At first to solve this problem I used the "track.pro" function of John C. Crocker (1999)[**?**], later we modified this function to be more suitable for our experiments and integrated it into the ACC. This tracker function read one sequence of images of the same field with the original centre point coordinates of the cells. Compare the coordinates of the cell centres, in respect of the given threshold that determines the maximum allowed distance between the two centre points on two consecutive images. At this point, we should deal with some difficulties with tracking. The cells can drift away while the microscope moves its' sample table. The morphology of the cell alter by the time, and this also can cause the drift of the centre point. Moreover, segmentation failures can happen as well that brings out the disappearing of cells. Missing cells also can be caused by the boundaries of the field of view, cells can migrate out of the field. These problems become more complicated if cells can move fast or the field is crowded and there and cells close to each other. Proper determination of thresholds can be critical at this point. There are 3 different thresholds to set. The maximum distance of centre in pixels, the maximum number of frames that we allowed a cell to disappear, and the minimum length of a trajectory.

Two of these are already mentioned. The purpose of the minimum length of a trajectory is to avoid specks of dirt and trashes to count them as cells.

With this function and all these options, a tracking file is generated. Tracker function found 5378 unique cells.

Now all the requested information allowed to generate trajectories.

2.2 Implementation and appearance

The main part of the graphical user interface (GUI) is a coordinate system, called axes in MATLAB. Pushing the START button plots the trajectories onto this axes. I implemented several filtering and selection options, these are the followings:

The toolbar contains generally used functions. With the save button, user can save the

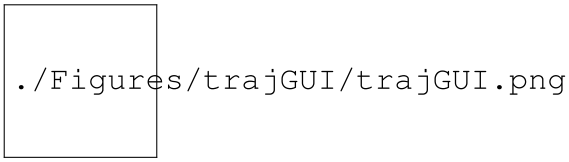

Figure 2.1: Trajectory module graphical user interface with test data

entire GUI with all the metadata and filtering options within, this makes easier and faster to reopen and continue a project instantly. There are possibilities to zoom, and there is a pan tool for grab and move actions of axes. With *data-tooltips* button is easy to check precise coordinate values of points. *Print* button sends a print command to the default printer if it is connected, and print the entire GUI. *SavePNG* button is for saving the main window as an image.

2.2.1 Visualization options

Visualization options section contains the *Start/Reset*, *All visible*, *Clear all* and *Animation* buttons. The *Start* button was already mentioned, but it also has a reset function that clears all the variables and filtering options and set back all the parameters to the default values.

Clear all button hides all the trajectories, *All visible* button does the opposite. Pushing the *Animation* button a dialogue window will appear, where the user gives parameters for the animation.

Figure 2.2: Input dialogue window for Animation

The given resolution parameter determines the number of dummy-points ordered to interpolate between two real points of the trajectory. Higher value makes smoother the animation. Given tail-length value determines the length of the tail in frames as a dimension. For example, if the given value is 2, then on the animated the tail length will be as long as 2 frames. Filename and path are also can selected. The result is a short video.

./Figures/trajGUI/anim4.png

Figure 2.3: Animation example

This animation is a powerful tool that makes visible the dynamic of movements, helps the user to see the differences in time and speed.

2.2.2 Filtering options

Filtering options contains three slides. The bottom slide called *Time-lapse* is similar to the animation. It allows to hide part of the trajectories, as decreases their length. The result shows all the trajectories, but only from the first frame to the given number of frame. The *Distance slider* filters by length, the *Frame slider* do the same by the number of the steps. There are *Negation* options on the upper right corner of the sliders. This feature doubles the combinations of the filtering rules to give more opportunity to the user. Only those trajectories stay visible that meet the requirements of filters, others are turn to invisible.

2.2.3 Selection options

This section allows to ignore or focus on a specific part of the axes. Users can draw polygons on the axes, then chose from three different options to filter by origin, destination, or transition like behaviours of the trajectories. *Negation* feature added here as well. There is a table that shows coordinates of drawn polygons, gives the opportunity filter in more precise way.

./Figures/trajGUI/freehand.png

Figure 2.4: Selecting areas by free hand polygon drawing tool

There is a status window to show whether the GUI is busy or not. On slower machines this becomes useful.

2.2.4 Grouping

There is a grouping function that I implemented. Users can define groups, add and remove trajectories to groups. At the top section, there are toggle buttons to make groups visible and highlight them with a chosen colour. This function gives us a helping hand to categorize trajectories, sort them in groups driven by similarities, or by different strategies.

./Figures/trajGUI/groups.png

Figure 2.5: Example of highlighted groups

Get ID button lists out all the visible trajectories with their image name, makes it easy to find out if they are treated differently.

2.2.5 Line Properties

I implemented a callback function to present specific pieces of information about a selected trajectory. Clicking on a trajectory opens a new window called *Line Properties*.

./Figures/trajGUI/lineprop2.png

Figure 2.6: Line properties

In this window, users can modify the visibility, the colour, the width of the selected trajectory and group membership. The table on the right shows the frames where the cell is visible (a.k.a. was found by the tracker). Picking a frame number from the list, then clicking the *Jump to Cell* button guides back the user to the main window of the ACC and shows the cell on the given frame.

There is a checkbox at the upper right, named *Thumbnails*. Turning this on makes the thumbnails of the cell visible on the axes, at their regression positions.

Show images button opens a new figure and shows all the images of the cell in a sequence, ordered left-to-right. This tool is also useful to the user to compare cells.

./Figures/trajGUI/thumbnails.png

Figure 2.7: Single trajectory with thumbnails shown

./Figures/trajGUI/lamelloShowIM.png

Figure 2.8: Result of Show Image function, the first frame is in the upper left corner, images are ordered left-to-right, top-to-bottom.

Animation button is appearing here too, functioning similar to the previously presented animation, but the program animates the regression path only the selected trajectory and simultaneously animates the evolution of the cell on the right side, by showing a sequence of images transforming one to the other, frame-by-frame.

./Figures/trajGUI/lamelloSingleAnim.png

Figure 2.9: Path animation of a single cell

Features button show the list of the features which was extracted by image processing software. The user can pick multiple features and generate a table of the values of chosen features in every frame. This is useful to compare specific attributes of cells.

All these functionalities provides better evaluation, in the spirit of deeper understanding of continuous biological processes. The project and the development are ongoing. The provided result in the lamellocyte experiment won a grant in promising to find molecular markers connected to plasmatocity trans-differentiation.

./Figures/trajGUI/featurePlot2.png

Figure 2.10: Left: list of features, right: selected feature values listed by frames

Chapter 3

Materials and methods

3.1 Lamellocyte project

Early third instar larvae (msn-Cherry; eater-GFP) were immune induced by wounding the cuticule with a 0,2 mm diameter Austerlitz Insect Pin®. Wounded larvae were kept on standard Drosophila food at 25 °C. Circulating blood cells were isolated 12 hours after wounding. Blood samples of 10 larvae were collected, pooled in 300 µl Schneider's medium (Lonza, Cat: 04-351 Q) supplemented with 10% FBS (Gibco, Cat: 10270), 0,01 mg/ml gentamicin (Sigma, Cat: G3632), 0,065 mg/ml penicillin (Sigma, Cat: P7794) and 0,1 mg/ml streptamicin (Sigma, Cat: S6501), and spread in a well of an 8 well µ-slide (Ibidi, Cat:80826). Samples were kept and microscopic analysis was carried out at 25 °C. [**?**]

PerkinElmer Operetta™ High Content Screening System was used for screening the sample. Screening time were 28 minutes, period of screening was 28 minutes, repeated every two hours, for 15 times.

During image processing we used three channels: greyscaled-brightfield image, eater-GFP fluorescent channel, msn-Cherry fluorescent channel. Images style transfer technology was applied to used to create synthetic images for augmentation. Mask R-CNN

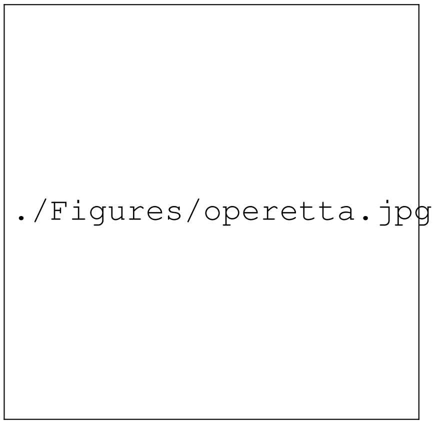

Figure 3.1: PerkinElmer Operetta

model was trained for image segmentation, to get multilevel annotated masks. During the feature extraction we used these masks in a Lipidanalyzer pipeline. These features were given to ACC for phenotypic classification and regression.

WEKA RandomForest [**?**] learner algorithm were used during phenotypic classification and regression.

3.2 Mitotic experiment

Mitotic data consists images and masks from experiment of Jan Ellenberg, Experimental and computational framework for a dynamic protein atlas of human cell division.[**?**]

The original dataset contains 3-dimensional, 31 layer images and masks for each cells. Number of cells is 505, number of frames 40. We used Maximum intensity projection method to create 2D images and masks.

For feature extraction we used CellProfiler^{TM}[**?**][**?**] pipeline. Extracted features can be found in Supplementary. **?? ?? ??**

Chapter 4

Experimental results

4.1 Lamellocyte project

The blood cells of Drosophila melanogaster - the hemocytes - play indispensable roles in the defense of the organism against microbes and parasitoid wasps. In the larva, three hemocyte types were described: the phagocytic plasmatocytes, the encapsulating lamellocytes and the melanizing crystal cells.

./Figures/Bobek.png

Figure 4.1: Network of differentiation transcriptomic factor in Drosophila Melanogaster [**?**]

Our lineage tracing studies revealed that plasmatocytes - which were previously believed to be terminally differentiating into lamellocytes upon immune induction. Following this observation, it was shown that lamellocytes are heterogeneous regarding their morphology and transgene expression patterns, which led to the definition of novel hemocyte types.

To better define hemocyte classes and to understand their developmental relation, we established a novel method which enables culturing hemocytes *ex vivo*. Hemocyte types are distinguished by the expression of specific *in vivo* transgenes, and differentiation routes are analyzed with machine learning.

With the trajectory module, we have showed that there is a continuum of lamellocytes in size, rather than strict phenotypes. **??**

Our approach serves as a basis for the identification of key factors instrumental in plasmatocyte-lamellocyte transition and leads to a better understanding of macrophage plasticity, in general.

4.2 Mitosis representation

We successfully demonstrated that our model can learn and predict a more complex, cyclic representation of a biological phenomenon, where ’x’ and ’y’ coordinates do not represent any specific features or dimension. Furthermore, we successfully performed valuable results, using such data of an experiment that not ideal to use for computational image processing. The fluorescent markers used in this experiment were not designed for classification or regression and images has low resolution. We used only one channel (nuclei channel) of the 2D images in our pipeline of feature extraction. We extracted only the morphological and texture features and no intensity features. With these parameters, it appeared that about 1/3 of the time, the prediction mix up cells in pro-meta phase and early-ana phase. We compared the performance of the machine and a biologist expert giving them the same information about these cells, and both performed evenly. It happened because of one frame before meta-phase and one frame after meta phase morphology of the nuclei - or at this point chromosomes - are look similar.

Chapter 5

Conclusion

Regression methods are developing quick, trying to overcome its own obstacles. In this thesis I attempt to briefly present use of image processing, machine learning and regression methods in single cell biology, and give an overview about the Trajectory module by showing its capability of evaluation on an ongoing biological experiment, and a given mitotic data-set.

Trajectory module has been implemented successfully, the lates version is available in the *BitBucket* repository of **BIOMAG**. The entire pipeline played important role to observe a newly discovered differentiation of immune cells in *drosophila melanogaster*. But there are still lot to do. Next phase we will repeat the experiment with higher resolution in frames, practically we will increase the frequency of screening to 1/hour or even more. Moreover, we will try to implement a new pipeline without markers, and examine the molecular composition of cells, to find signalization pathways.

Statement

Me, Beleon, Attila a Molecular Bionic Engineering BSc. student hereby state that I made my thesis at the Institute of Biotechnology, University of Szeged to acquire BSc diploma.

I declare that I did not defend my thesis at other degree program before, it is completely made by me, and I used only the cited sources.

I accept, that my thesis is placed in the library of Institute of Biotechnology, University of Szeged, in the section of locally available books.

Szeged, March 27, 2020 . . . .

signature

Acknowledgement

Hereby I wish to say thank you to **Peter Horvath** for allowing me to be part of BIOMAG, for guidance and a lot of helpful ideas. To **Rakhely Gabor** for the support. Special thanks to **Abel Szkalisity** for major help in MATLAB language and the extra hours of guidance in general programming and mathematics and for reviewing the manuscript and giving valuable comments. To **Viktor Honti** and **Gergely Istvan Varga** and **Ede Migh** for the exciting and fruitful cooperation. To **Tamas Balassa** and **Csaba Molnar** for the helping hand and the successful cooperation on mitotic projects. To **Filippo Piccinini** for help to write the documentation of the Trajectory GIU. To all other members of BIOMAG: **Krisztian Koos**, **Istvan Grexa**, **Akos Diosdi**, **Jozsef Molnar**, **Reka Hollandi**, **Gyukity-Sebestyen Edina**, **Maria Kovacs**, **Nikita Moshkov Csaba Voros**, **David Bauer**, **Dominik Hirling**, **Tasnadi Ervin**, **Tivadar Danka** and to the SingleCell guys: , **Andras Kriston**, , **Ferenc Kovacs** and , **Norber Bara** for sharing their experience on thesis writing, LaTeX usage and help in small everyday problems and sharing biological explanations. Finally to my **Family** for supporting me, and **Friends** for everyday help.

Chapter 6

Supplementary

./Figures/trajGUI/inputExample.png

Figure 6.1: Input data example of Trajectory module.

./Figures/trajGUI/lamelloTS.png

Figure 6.2: Training set of the lamellocyte project represented on the Regression Plane

./Figures/trajGUI/lamelloTraj.png

Figure 6.3: Trajectories, lamellocyte project

./Figures/trajGUI/Lamellocita_large.PNG

Figure 6.4: Cell that transferred to large size lamellocyte.

./Figures/trajGUI/Lamellocita_medium.PNG

Figure 6.5: Cell that transferred to medium size lamellocyte.

./Figures/trajGUI/Lamellocita_small.PNG

Figure 6.6: Cell that transferred to small size lamellocyte.

./Figures/trajGUI/noCrossing.png

Figure 6.7: Terminally differentiated lamellocytes has stable size.

./Figures/trajGUI/ind.png

Figure 6.8: Induced cells predicted on regression plane.

./Figures/trajGUI/Control.png

Figure 6.9: Induced cells predicted on regression plane.

./Figures/classVreg.png

Figure 6.10: Comparison of classification (left) and regression (right). Classification section *a,* 2 classes, *b,* 3 classes, *c,* 5 classes, red rectangles represents average cell size inside of each classes. Regression Plane *d,* y-axis represents a continuous distribution of cell size, x-axis represents green and red intensity distribution.

./Figures/featureList/lamelloF.png

Figure 6.11: Feature list used in Lamellocyte experiment.

./Figures/trajGUI/mitoTS.png

Figure 6.12: Training set of MitoCheck project.

./Figures/trajGUI/mitoTraj.png

Figure 6.13: Trajectories of MitoCheck project.

./Figures/trajGUI/anim1.png

Figure 6.14: Frame of animation of MitoCheck project.

./Figures/trajGUI/anim2.png

Figure 6.15: Frame of animation of MitoCheck project.

./Figures/trajGUI/anim3.png

Figure 6.16: Frame of animation of MitoCheck project.

./Figures/trajGUI/anim4.png

Figure 6.17: Frame of animation of MitoCheck project.

./Figures/trajGUI/anim5.png

Figure 6.18: Frame of animation of MitoCheck project.

./Figures/trajGUI/anim6.png

Figure 6.19: Frame of animation of MitoCheck project.

./Figures/featureList/mitoF1.png

Figure 6.20: Feature list used in MitoCheck experiment (part1).

./Figures/featureList/mitoF2.png

Figure 6.21: Feature list used in MitoCheck experiment (part2).

./Figures/featureList/mitoF3.png

Figure 6.22: Feature list used in MitoCheck experiment (part3).