Project 12 : Artificial Intelligence to quantify neurodegeneration

Project owners:
BROCARD Jacques (PLATIM)
MOLLEREAU Bertrand (LBMC)

Tutor:
MEYER Alexandre (LIRIS)

Students:
ALIZADEH-NOIRET Grégoire
DESAINTJEAN William
LAHAIE Thomas
LEON FOUN LIN Ravy

Master 2 BioInformatics - 2022/2023

Table of Contents

1 Project Presentation

1.1 Stakeholders

1.1.1 The LBMC

The Laboratory of Biology and Modelling of the Cell (LBMC) is a laboratory located at the *École Normale Supérieure* (ENS) of Lyon. Research teams from the LBMC study various aspects of the cell, from proliferation to differentiation and apoptosis. We work with the team whose work on "Regulated cell death and genetics of neurodegeneration". The project is conducted by the group's leader Bertrand Mollereau. We also collaborate with Marion Celle, the research engineer. The team uses a fluorescence-based method in their experiments, which is why we also work with the PLATIM.

1.1.2 The PLATIM

The Imaging and Microscopy Core Facility or *Plateau d'Imagerie / Microscopie* (PLATIM) is also located at the ENS. They provide a wide range of optical microscopy as well as analysis solutions. Jacques Brocard is part of this laboratory and works in collaboration with the Mollereau team. Jacques Brocard is the author of various programs which pre-process and configure the data that we will use for our model.

1.2 Context

Lipid dysregulation research is a new approach to understand mechanisms behind neurodegenerative diseases such as Parkinson's and Alzheimer's diseases which affect the brain, or the Age-Related Macular Degeneration, which affects the retina.

To study the link between neurodegeneration and lipid metabolism, a team of researchers from Mollereau's group uses *Drosophilia melanogaster* (fruit fly) as a model for this work.

The experimental strategy with *Drosophilia* is to knock out genes of interest (e.g. fatty acid transport protein 1 or FATP1 [8]) in order to measure the phenotypical impact of these genes on fly eye tissues. Precisely, researchers focus the measurements on ommatidia [Figure 1], the functional units of arthropods' eyes.

Drosophilia species have an average of 800 ommatidia per eye. Each ommatidium is composed of a cluster of eight photoreceptor cells, enumerated from R1 to R8 [3]:

- R1 to R6 : photoreceptor cells for scotopic vision. They specifically express rhodopsin 1 (RH1).
- R7 and R8 : other photoreceptor cells that express an other type of rhodopsin.

If the inactivation of a gene of interest leads to neurodegeneration, photoreceptor cells will die and ommatidia development will be affected.

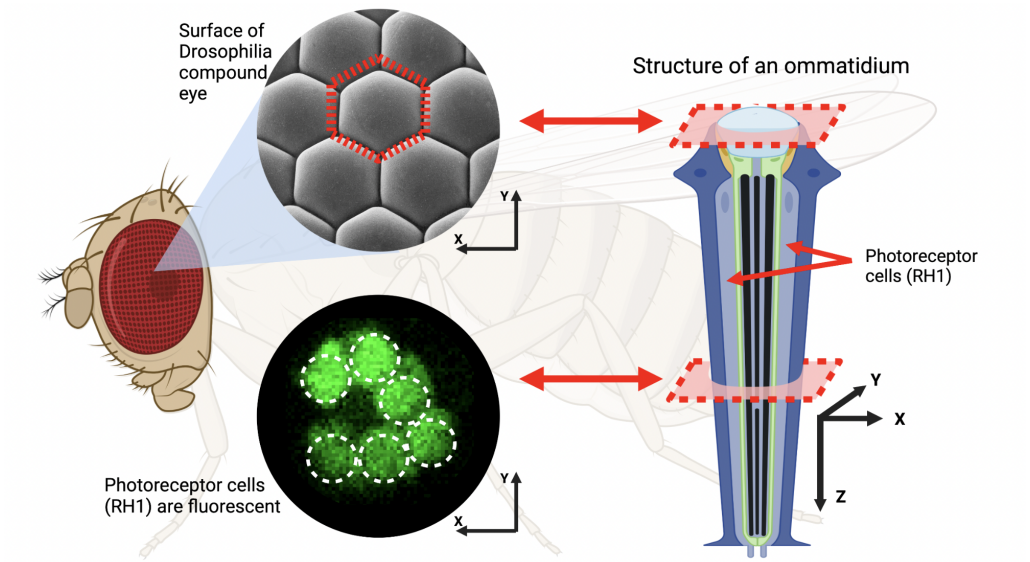

Figure 1: *Drosophilia* eye structure and image localization. To the left, the microscopic views corresponding to anatomical sections of ommatida on the right side.

To check the survival of these cells, researchers performed a confocal microscopy imaging 1.4 [1]. The flies are put to sleep with CO_2, and placed on their side in agar-agar, so that the eyes are oriented towards the top.

This is made possible thanks to the gene-fusion with Green Fluorescent Protein (GFP) which is used as a reporter of RH1 expression. In fact, GFP is placed under the control of the RH1 promoter; thus, if RH1 is expressed, the photoreceptor cell appears fluorescent, which indicates that it is alive.

The phenotypical characterization is based on the number of photoreceptor cells per ommatidium. For wild type, we can expect the 6 receptors (R1 to R6) that express RH1 to be fluorescent and visible with the microscopy. This quantitative approach allows the use of statistical tools and provides robust results. However, the cell counting is currently hand-made, laborious and experimenter-dependent.

In order to simplify the photoreceptor-counting procedure, Jacques Brocard is looking forward to developing an automatic algorithm derived from deep learning networks. This is the main goal of the present project.

1.3 Objectives

The aim of this project is to use a deep learning approach to count the number of photoreceptors on pictures of *drosophila* ommatidia. We will use images realized by the LBMC and annotated by an expert from the team in order to train our models.

Once the training on images from wild type and different genetic *Drosophila* mutants is complete, we will use this model on a test image set.

At the end, the model will be generalized using another dataset of the same type, or presenting mild degeneration.

1.4 Resources

For the project, several images of *Drosophila* eyes have been realized by Bertrand Mollereau's team by using confocal microscopy imaging. 310 vignettes of ommatidia have been extracted from those

images.

For one *Drosophila* eye, several pictures are taken at different depths (Z-axis) to see every receptor of the retina. As shown below, these photoreceptors are organized in a dome-like shape. These images need to be pre-processed to be used in a program:

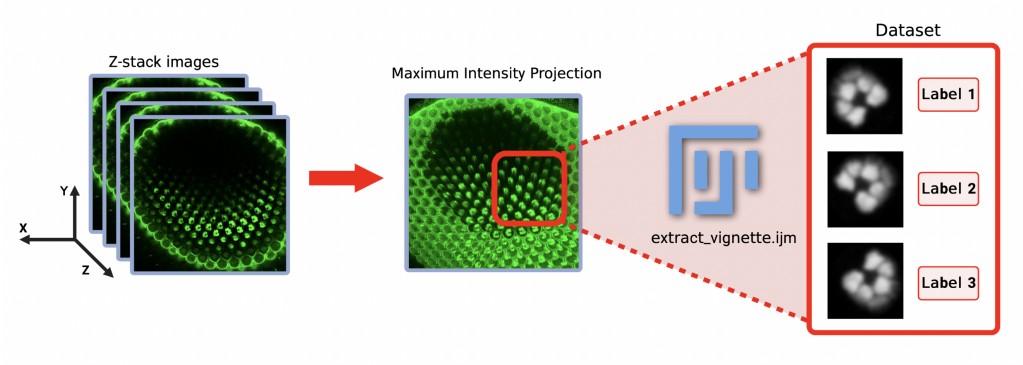

Figure 2: Images made by the experimental team. The first images represent the stack of images taken at different depths (the Z-axis). The second image is a reconstruction of the original image using the stack of images. The last three pictures represent the vignettes and their labels, made with ImageJ from the reconstructed images.

After the confocal microscopy, Jacques Brocard used Fiji ImageJ [7], an open source image processing package, to superimpose all the pictures on the Z-axis to see all the ommatidia of the eyes in one image. Then the software asks the user to place each ommatidium in a circle of adapted size, and produces vignettes of 60 $\times$ 60 pixels (in our case, 310 images were generated from several pictures). These will be analyzed using deep learning networks.

Jacques Brocard used Fiji ImageJ again to compute the number of receptors, represented by several white pixels in a circular form on the images by thresholding [10]. This method is not precise enough but is faster than manual annotation. For that, he counts the number of stains, by decreasing the gradient of color on the image's pixels, from white to black. It produces a first labelling to use in machine learning approaches.

The expert will also manually label each picture, in order to validate the program. Although this method is more precise, it is user-sensitive and takes more time.

For now, we have 310 vignettes of ommatidia extracted from several total eye images of wild type *Drosophila*. Jacques Brocard will provide lots of new images, including mutants, for the second project period.

We will be able to use Jacques Brocard's python [9] notebooks in a "Google Colab" environment, that were written during a course organized by the CNRS and titled "Image analysis training with deep learning", in 2021. These notebooks describe how to use classical networks (LeNet-5, VGG16, ResNet50 [2], UNET, ...) for image analysis using the TensorFlow/Keras framework.

For instance, in the notebook about LeNet-5, pictures of hand-written digits from the MNIST database are classified (0-9). Another notebook uses VGG16 to classify skin pictures as "naevus" or "melanoma" (skin cancer).

After exploring these approaches, we will try other neural network architectures as there is a large number of python libraries to use machine learning or deep learning (e.g scikit-learn [5]).

1.5 Acceptability Criteria

As a criterion, Jacques Brocard suggested to keep the error margin below 5% for the test set. It is also known in Artificial Intelligence (AI) that a good model has a certain balance between bias and

likelihood. To evaluate this, we will focus our work on having a model with the best loss function, and a balance between specificity and sensitivity. This will be done by computing the confusion matrix and the Receiver Operating Characteristic (ROC) curve.

2 System Features

2.1 Functional Requirements

Requirements	Must	May
Count the number of photoreceptors per ommatidium	✓	
Detect ommatidia in an image and crop them		✓
Remove noise from an ommatidia image		✓

2.2 Non-Functional Requirements

Requirements	Must	May
The network's predictions must have a correct accuracy $\approx 95\%$	✓	
Investigated methods need to be documented	✓	
The user interface may be easy to use		✓

3 Constraints

3.1 Deadlines

03/10 - 14/10: First period of project

14/10: Deadline for the bill of specifications

31/10 - 13/11: Second period of project

28/11 - 09/12: Third period of project

15/12: Return of the final product and project defense

3.2 Technical Constraints

Presently, we do not have enough data (310 vignettes only) and they have incorrect labels since no expert in Bertrand Mollereau's team has taken time to annotate them properly. Therefore, Jacques Brocard used an automatic but inaccurate method base on an ImageJ script to assign an approximate number of photoreceptors to each vignette. We may still train a model with these poorly labeled vignettes but it will give better results once the data have been reviewed by experts. This should be done before the second phase of the project.

Meanwhile, the small amount of data can be overcome with data augmentation: rotating images or using their mirror representations are methods which can create new data with the same quality.

We are not limited by the power of our machines: the Google Colab notebook allows us to launch our different trials of machine learning on google's servers, even with GPU, for free. This will be sufficient to use the different methods, even the most resource-consuming.

4 Project phases

4.1 Planning

03/10 - 14/10 : First meeting with the researchers and the tutors, start of the research and exploration on the different models, and writing of the bill of specifications.

30/09 - 11/11 : Creations of the models: each member of the group will focus his work on the implementation of one method. We will test several possibilities, which we will adapt to our problematic:

- A basic neural network originally used to identify written numbers: it's the Google Colab notebook given by Jacques Brocard. We can try adapting it to count receptors.
- VGG19 [6] or ResNet50[2], two neural networks that already exist in Keras.
- Use a network that already exists and works well on our data. For example: Github, advised by Alexandre Meyer. It will be interesting to test it.
- A non-supervised method: if the experts can not label the images in the future, or if the training requires too many images to be labeled by a human, it can be an interesting solution to try. [4]

To make these networks work with our data, we will probably have to solve a number of difficulties, such as not having enough labeled data. We will then consider data augmentation, by rotating or mirroring the vignettes we already have.

28/11 - 09/12 : Evaluation and comparison of the models. Redaction of the final work and preparation for the project defense

4.2 Quality assurance plan

To assess the quality of a given method, we can check and compare several metrics:

- Its sensitivity (detected positives / all real positives) and specificity (detected negatives / all real negatives),
- Draw the ROC curve, which displays the relationship between the true positive rate and the false positive rate.
- The error rate of the final counting on the test set of images.

4.3 Final Product

The final product will be a python Notebook. The input will be a set of 60×60-pixel ommatidia images realized in confocal microscopy imaging and the model will return the number of photoreceptors with a probability of confidence. The code will be available on a GitLab or GitHub repository with the documentation. We will add a written report to the repository to explain our choices to implement the different methods, and to compare the results with different indicators.

Bibliography

[1] Alexis Gambis et al. 'Two-color in vivo imaging of photoreceptor apoptosis and development in Drosophila'. In: *Developmental biology* 351 (Mar. 2011), pp. 128–34. DOI: 10.1016/j.ydbio.2010.12.040.

[2] Brett Koonce. 'ResNet 50'. In: *Convolutional neural networks with swift for tensorflow*. Springer, 2021, pp. 63–72.

[3] Bertrand Mollereau and Pedro M Domingos. 'Photoreceptor differentiation in Drosophila: from immature neurons to functional photoreceptors'. In: *Developmental dynamics: an official publication of the American Association of Anatomists* 232.3 (2005), pp. 585–592.

[4] *MS Windows NT Kernel Description*. https://medium.com/@joel_34096/k-means-clustering-for-image-classification-a648f28bdc47#:~:text=K%2DMeans%20Clustering%20can%20be,Here's%20how.&text=K%2Dmeans%20clustering%20is%20an,cluster%20with%20the%20nearest%20centroid. Accessed: 2010-09-30.

[5] Fabian Pedregosa et al. 'Scikit-learn: Machine learning in Python'. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[6] Venkatesan Rajinikanth et al. 'A customized VGG19 network with concatenation of deep and handcrafted features for brain tumor detection'. In: *Applied Sciences* 10.10 (2020), p. 3429.

[7] Johannes Schindelin et al. 'Fiji: an open-source platform for biological-image analysis'. In: *Nature methods* 9.7 (2012), pp. 676–682.

[8] Daan M. Van Den Brink et al. 'Physiological and pathological roles of FATP-mediated lipid droplets in Drosophila and mice retina'. In: *PLOS Genetics* 14.9 (Sept. 2018), pp. 1–25. DOI: 10.1371/journal.pgen.1007627. URL: https://doi.org/10.1371/journal.pgen.1007627.

[9] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[10] Wikipedia contributors. *Thresholding (image processing) — Wikipedia, The Free Encyclopedia*. [Online; accessed 13-October-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Thresholding_(image_processing)&oldid=1100706643.