University of Bordeaux

College of Science and Technology

Master in Bioinformatics - Computational Biology

School year 2019-2020

Design of a Research and Development Project
Load specifications

Alexandre Cornier, Martin Drance, Ali Cuhadar, Abdelghani Neuhaus

Team leader : Andreas Frick
Project supervisor : Marie Beurton-Aimar

Table of contents

Resume

Autism spectrum disorders (ASD), is a neurodevelopmental disorder that primarily affects socio-emotional communication and therefore social reciprocity combined with the presence of restricted and stereotypical interests in young people and adults with autism. Some children will show autistic symptoms very early in their development or global developmental delay, which will alert parents. Other children have no language or motor delay, it is only in school that the symptoms may become manifest.

Today, re-education is possible depending on the level of severity but there is no targeted therapeutic approach. The object of our project is to improve the functioning of a tool intended to improve the treatment of autism therapeutically.

Partie 1

Subject presentation

In this section, we will present the elements used for the project

1.1 State of art

Atypical processing of sensory information (for example, tactile, visual and auditory information) is now considered as a key phenotype of ASD and can be strongly determinant for other basic autistic phenotypes. Information from the different senses is processed in the neocortical circuits and can be measured by electrophysiological or imaging approaches. Measurements of sensory information processing and perception could also provide objective biomarkers essential to complete the evaluation of social, communicational and behavioral alterations and to quantify the therapeutic results.

Studies are currently being carried out at the neurological center of the University of Bordeaux, Andreas Frick's team is particularly interested in the different areas of the brain in mutant mouse models of autism spectrum disorders. For this, the center is equipped with a very efficient electron microscope to obtain high-resolution images (NDPI format) section by section of the brain. Once the images are recovered, they must be compared to a reference atlas, in particular Allen brain atlas. This is the Allen Institute project that seeks to combine genomics with neuroanatomy by creating gene expression maps for the mouse brain.

To compare the control and mutant brains, Andreas Frick and his team developed a computer program designed to help align images of brain slices to the Allen Brain Atlas, and count cells in different regions. Currently, the software is used to view sections of the brain of mice. It is possible to superimpose these sections on a reference atlas to identify the areas of interest in the images. The images used are in TIFF or NDPI formats. After the selection of points or areas of interest, we obtain a spreadsheet with information on clicks. Currently, some features have been coded and others have not been finished. This research protocol is at the beginning and it is very specific, there is, therefore, no real equivalence, the final objective of the software being to be able to open, superimpose images (in color or not) of different formats with the atlas as easily as possible and to save the data of the analysis carried out.

1.2 Main goal

Our project is a continuation of the work carried out by Rémi Proville. They asked us to optimize the workspace, that is to say to the sorts of features and remove the one that seems superfluous (in particular to make the main window larger, to solve the problems related to the contrast and the brightness). We will, therefore, carry out several readings of the code in order to locate the elements to be modified.

Then once our software is "clean", improves some features and makes the interface more accessible. Currently, some selection sliders are too sensitive or reset : our goal is to improve them. As it is a project in development we must also make the program more modular and therefore adapt to the next steps. this requires documentation of the code to be further developed and the implementation of writing rules. If the most important elements have been corrected or added, we will try to allow the use of different types of images with the software.

1.3 Tools used

a) Python

Python is a powerful and easy to learn programming language. It has high-level data structures and allows a simple but effective approach to object-oriented programming. Because its syntax is elegant, its typing is dynamic and it is interpreted, Python is an ideal language for scripting and rapid development of applications in many fields and on most platforms.

Python is an interpreted language : like PHP, a Python program does not require a compilation step in machine language to function. The code is interpreted at run-time. Python nevertheless has an interesting characteristic : as in Java, Python code is compiled in byte-code (intermediate format) before launching, which optimizes its performance. This programming language has strong dynamic typing : this means that typing, although not checked during "compilation", Python performs consistency checks on the types used, and allows you to explicitly transform a variable from one type to the other. As part of our project, python is an ideal programming language because it has a very large community and therefore a large variety of libraries available.

Figure 1 : Python logo

b) Python libraries

The Atlaser software code is programmed using a number of add-ons :

- **OpenSlide** is a C library that provides a simple interface for reading whole-slide images, also known as virtual slides, which are high-resolution images used in digital pathology
- **NumPy** library allows you to perform numerical calculations with Python. It introduces easier management of arrays of numbers

— **Python Imaging Library**, or **PIL**, is one of the basic libraries for manipulating images in Python

— **Pathlib** is a module that offers classes representing the file system with the appropriate semantics for different operating systems. The path classes are divided into pure paths, which provide purely computation without I / O, and concrete paths, which inherit from pure paths and also provide I / O operations.

— **Scikit-image** is a collection of algorithms for image processing

— **PyQt5** is a library that lets you use the Qt GUI framework from Python. Qt itself is written in C++. By using it from Python, you can build applications much more quickly while not sacrificing much of the speed of C++

— The **python-javabridge** package makes it easy to start a Java virtual machine (JVM) from Python and interact with it. Python code can interact with the JVM using a low-level API or a more convenient high-level API

— **Pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language

— **Python-Bioformats** is a Python wrapper for Bio-Formats, a standalone Java library for reading and writing life sciences image file formats. Bio-Formats is capable of parsing both pixels and metadata for a large number of formats, as well as writing to several formats. Python-bioformats uses the python-javabridge to start a Java virtual machine from Python and interact with it.

— **PyQtGraph** is a pure-python graphics and GUI library built on PyQt4 / PySide and NumPy. It is intended for use in mathematics / scientific / engineering applications

— The **Logging** package is a means of tracking events that happen when some software runs. The software's developer adds logging calls to their code to indicate that certain events have occurred. An event is described by a descriptive message which can optionally contain variable data (i.e. data that is potentially different for each occurrence of the event). Events also have an importance which the developer ascribes to the event ; the importance can also be called the level or severity.

c) Current code

The current program consists of 4 files in py format, 3 files in npy format (number table corresponding to all the information on the atlas), 2 files in nrrd format and one file in JSON format (corresponding to the tree structure of the brain areas). As for Python files, there are 3711 lines of code. We believe that we must intervene on a maximum of 20 to 40 % of the code for certain files.

d) Images acquisition

The neuroscience center has in these infrastructures a scanner of Nanozoomer blades HAMAMATSU. This new technology makes it possible to digitally reproduce optical content of a histological glass slide. Reading is done on a computer screen via proprietary viewing software (NDP View) which reproduces the functions of a microscope.

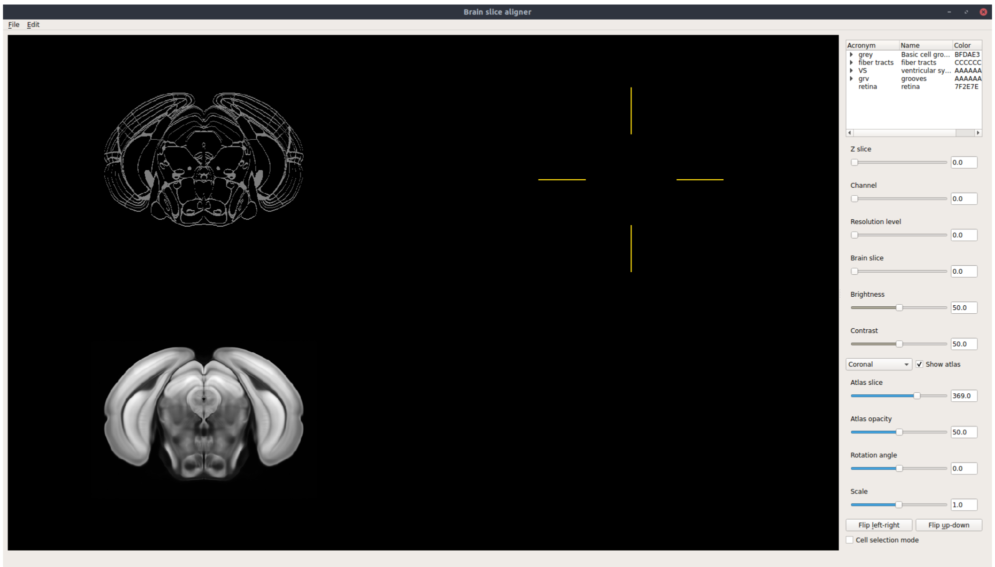
We end up with files in NDPI format which cannot always be directly opened with current image analysis software such as ImageJ. NDPITools is a set of software that can convert NDPI files into standard TIFF files, with the option of cutting into smaller pieces (in JPEG or TIFF format) which will more easily hold in RAM. They are accompanied by a set of plugins for ImageJ, which allow you to use the software directly from ImageJ, graphically.

HAMAMATSU
PHOTON IS OUR BUSINESS

Figure 2 : Hamamatsu logo

e) Current Atlaser software

We are not starting our project from scratch since we have a large working base which is the current software. The goal will be to modify it and make it more accessible while respecting the client's requests, presented in the following part.



Acronym	Name	Color
grey	Basic cell gro...	BFDAE3
fiber tracts	fiber tracts	CCCCCC
VS	ventricular sy...	AAAAAA
grv	grooves	AAAAAA
retina	retina	7F2E7E

Z slice 0.0
Channel 0.0
Resolution level 0.0
Brain slice 0.0
Brightness 50.0
Contrast 50.0
Coronal ☑ Show atlas
Atlas slice 369.0
Atlas opacity 50.0
Rotation angle 0.0
Scale 1.0

□ Cell selection mode

Figure 3 : Atlaser software

Partie 2

Needs for the project

In this section, we will present the needs discussed during the meetings with our client and his team and what we agreed on

2.1 Functional needs

During the development of the software, several functionalities were implemented and commented out for certain reasons; the others are functional but pose a problem for the user. The current software allows us to view slices of mice brains and navigate between them. Images handled are in TIFF or NDPI format. Currently, the user can select areas of interest by clicking on them, overlay it on a mouse brain atlas. During selecting points, when the image is superimposed on the atlas, the region where it is located is saved for each point. These data can be exported in the form of a spreadsheet.

Following our various discussions with Mr.Frick and his team, we identified various elements to modify or implement.

First of all, concerning **the graphic interface**, it is necessary to have greater visibility on the left part (image and atlas). Therefore, we will remove the 3D atlas present in the window at the bottom left to give more space to the treated elements, that is to say, the brain slices and their analyzes with the atlas. The right part which corresponds to a zoom area (for a precise selection of points) will be placed in a corner to leave more room for the image and the atlas. Two additional elements discussed are the zoom and the selection : the zoom is not synchronized on the two windows, the cross which allows the selection is too large and the pointing area not precise enough. So, we will synchronize the two windows for zooms, the cursor will be modified for a better selection.

Still, concerning the selection of points, one of the problems encountered when saving data is that if the user clicked in the same place, each click is recorded (we can, consequently, have the same point many times). Also, the user cannot delete a click made : he must cancel all clicks until the one he want to delete. To overcome these two problems, we will modify the code so that the same pixel can be selected several times but saved only once; and we are going to implement a tool allowing to delete any point, regardless of its position in the selection history. Finally, some buttons or scrolls do not work or are not used : we will delete them.

Then, regarding **images and the atlas** (left window). Currently, the atlas and images are fully synchronized in terms of movement, zoom and *"click to determine the brain area"*. The problem is that the user can currently only modify the image via rotations and zooms, which causes problems when overlapping the two elements.

The third main point concerns **the processing of images before and during the opening**. First of all, with regard to TIFF images, we will improve the sharpness once opened. Indeed, the images lose resolution when opening. We will, consequently, modify the existing code in order to set the resolution to a suitable value.

Now for the NDPI images. These are obtained by acquisition under a microscope. It is presented in the form of a single image with all the slices. They are in two colors (red and blue). The main problem is that when they are opened, they are processed and separated automatically, and then they are converted to TIFF. This separation is poorly done because there are missing bits of images. We will, therefore, modify the code to correct this problem, but also to allow the user to cut the image himself.

Finally, the buttons managing the contrast and brightness of open images have defects : resetting the modification (of contrast or brightness) when doing certain actions, etc... These problems will be corrected.

2.2 Non-functional needs

We think it would be interesting to be able to separate the two colors of the image (having a TIFF image with the red marker and a TIFF image with the blue marker) in order to be able to analyze the two.

Next, we would like to create a help tab where the user can easily access the various commands or shortcuts if he needs to.

Also, we find it advisable to add more details when identifying the selected area of the brain. Currently, only the smallest sub-area is displayed ; we would like to modify this so that the user can have at least three levels of hierarchies.

Partie 3

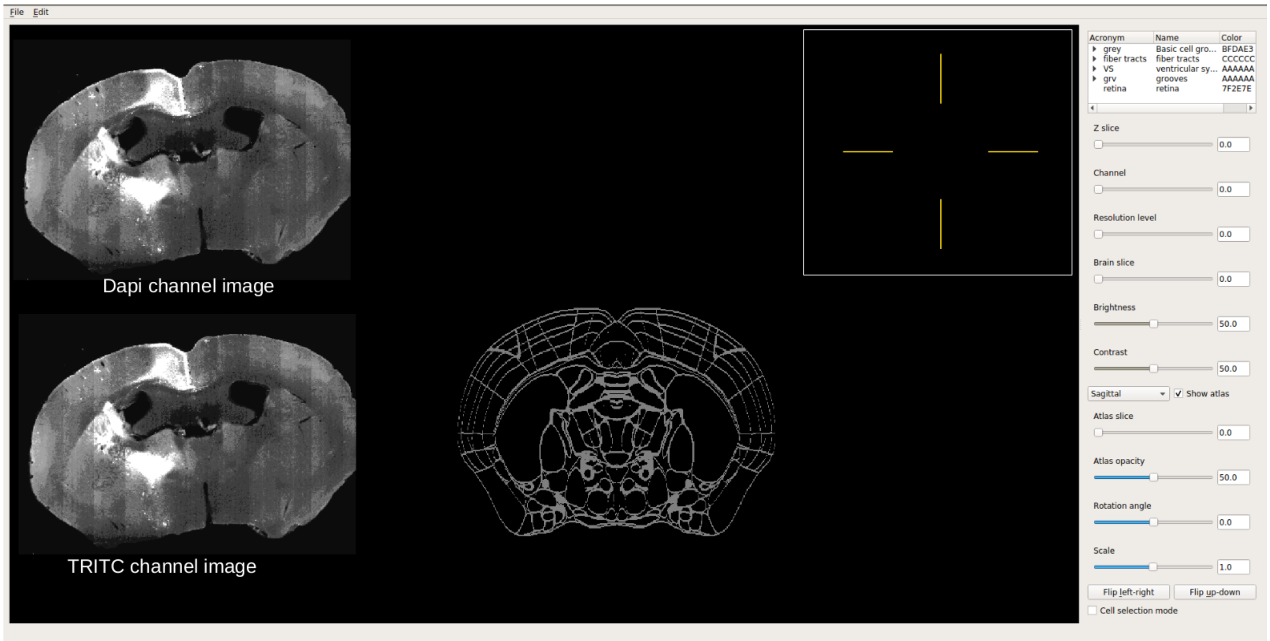
Tool concept

3.1 Graphic interface

Regarding the working window, we are going to make the window larger so that the user can open several images to see the different markings. We will also make the selection window smaller (with the cross) for :

- Free up space
- Make a more precise selection

Then, at the level of the window on the right side, we will keep only the smallest sub-area (but we will have more information on selection and saving). The model below gives an overview of the new organization of the interface :



Acronym	Name	Color
grey	Basic cell gro...	BFDAE3
fiber tracts	fiber tracts	CCCCCC
VS	ventricular sy...	AAAAAA
grv	grooves	AAAAAA
retina	retina	7F2E7E

Z slice 0.0

Channel 0.0

Resolution level 0.0

Brain slice 0.0

Brightness 50.0

Contrast 50.0

Sagittal ☑ Show atlas

Atlas slice 0.0

Atlas opacity 50.0

Rotation angle 0.0

Scale 1.0

□ Cell selection mode

Figure 4 : Model of the software, and above all of two new working spaces (visualization and selection)

Also, open images will have a label (to differentiate the two channels).

3.2 Buttons and widgets

It is important to be able to handle the atlas and the image separately. To overcome this problem, we will create two separate boxes that will allow modifying :

- in the image : zoom, rotation, contrast, brightness, and movement,
- in the atlas : zoom, rotation, movement, opacity, display it or not.

3.3 Software tools

It will be possible to open NDPI images and select the zones that we want to export to TIFF. It will be possible to open several images at the same time (red and blue)

Then, with regard to TIFF images, we will improve the sharpness once opened. Currently, the images lose resolution when opening. The reset issue concerning the brightness will be corrected.

Regarding the selection of points, we will change the fact that there is a multi-selection on the same point (saving data from the same point). Also, when saving, we will revise the code so that more information about the selected area appears

3.4 Quantification of work

The simplest elements to deal with at first glance are the improvement of the main window, the correction of the different buttons (zoom, brightness, contrast, displacement, etc ...), the addition of an accessible help in the software (via a help tab), the synchronization of the zoom for the two windows. We think that the biggest part of the work will consist of the correction concerning the opening of the images and the possibility of having images separated into two channels.

Web and bibliographic references

- **Hamamatsu website** : https ://www.hamamatsu.com/eu/en/product/index.html
- **NumPy documentation** : https ://numpy.org/devdocs/
- **OpenSlide documentation about Hamamatsu pictures** : https ://openslide.org/formats/hamamatsu/
- **PIL documentation** : https ://he-arc.github.io/livre-python/pillow/index.html
- **PyQt5 documentation** : https ://www.riverbankcomputing.com/static/Docs/PyQt5/