Automatic Identification of Parallel Growth Among Neuronal Dendritic Branches

Inbar DAHARI 205561582
Shahar MOYAL 307934679

Ariel UNIVERSITY

Internal advisor: Prof. Boaz BEN MOSHE
External advisor: Prof. Danny BARANES
and Dr. Refael MINNES

September 2019

Contents

1 Abstract

The structure of neuronal dendritic trees plays a role in the activity of neuronal networks. The ramification of dendrites in neuronal cell in culture involves parallel and unparallel growth among neighboring dendritic branches.

Our hypothesis is that parallel vs non-parallel growth ratio is non-random.

During this work we will investigate the dendrite's architecture to show that the cells communicate, and the dendrite growth is not random. In order to achieve this goal, we grew nerve cells on a two-dimensional substrate. We used dendrite specific fluorescent dye to mark the dendrites and imaged the dendrite network using a confocal microscope.

The existing analysis technique, needed during image processing, involves semi-automatic approaches and some manual interventions, which makes the work strenuous and time consuming.

In this project we introduce a new technique, that allows direct measurement of dendrite morphology from fluorescence images of dendritic networks. We use segmentation to distinguish dendritic branches from the image background and accumulate a database on the spatial direction of each branch. The database is then used to calculate the prevalence of pairs, triplets, foursomes etc. of parallel dendritic branches.

The results will be compared to a simulation, based on simple combinatorics, of random growth, with realistic biological conditions to substantiate that there is indeed communication between the cells.

2 Main Goals Of Our Project

To test that we developed automatic quantification of the ratio in the neuronal networks and compared its value to that generated by combinatorial random growth simulation of dendritic branches.

The goals of our project were:

- Automatic identification
- Efficiency
- Performing individual analyzes
- Accumulating biological data from other cell types

Current morphological analyses typically involve a significant component of computer-assisted manual labor, which is very time-consuming and is susceptible to operator bias. The existing semi-automatic approaches largely reduce user efforts. For example 'ImageJ', an open source image processing package with a vast collection of community-developed biological image analysis tools, that's for detecting the lines, for each image and image, you should mark by yourself the dendrites over the image and then the system calculate the data. However, some manual interventions, such as setting a global threshold for segmentation, are still needed during image processing.

Instead of that we create automatic identification. Working time in our system is greatly reduced especially when a large number of images are needed to process. Moreover, we perform individual analyzes according to the researcher's requirements, furthermore, for the long term it can be used to other cell types, not just neural cells.

Our approach calls a coordinated effort to help meet challenges and create novel opportunities for academic sectors to explore aspects of biological and chemical diversity that cannot be accessed through natural mechanisms.

3 Algorithms

Our project use different types of algorithms since from one hand we need to detect lines from fluorescence images of dendritic networks and on the other hand we need to make multiple analyzes on the received information.

3.1 Segmentation

The most important and basic level of our system is the image segmentation. This is a process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and change the representation of an image into something that is more meaningful and easier to analyze. In order to reach maximum identification we use two main functions :

HoughLinesP()- Probabilistic Hough Line Transform
The Hough Line Transform is a transform used to detect straight lines in an input image with M pixels defined as edge points and a parameter space which is divided into NrxN accumulators. We chose to use this function since it more efficient implementation and it gives as output the extremes of the detected lines (X0,Y0,X1,Y1). To apply the Transform, first an edge detection pre-processing is desirable.

The Canny edge detection algorithm
Canny Edge Detection is used to detect the edges in an image. It accepts a gray scale image as input turn it to binary image, and uses a multistage algorithm. This method accepts the following parameters:

- Image after blurring, using GaussianBlur(src, (5, 5), 0). Blurring convolving the image with a low-pass filter kernel. It is useful for removing noise, it removes high frequency content (e.g: noise, edges) from the image resulting in edges.
- Edges- a Mat object representing the destination (edges) for this operation.
- Threshold1− a variable of the type double representing the first threshold for the hysteresis procedure. This parameter is changeable, and we found that the ideal range is between 110-200. Also, as the chosen number is low, the lines are more detectable.
- Threshold2 − a variable of the type double representing the second threshold for the hysteresis procedure. We recommend not to apply this parameter.

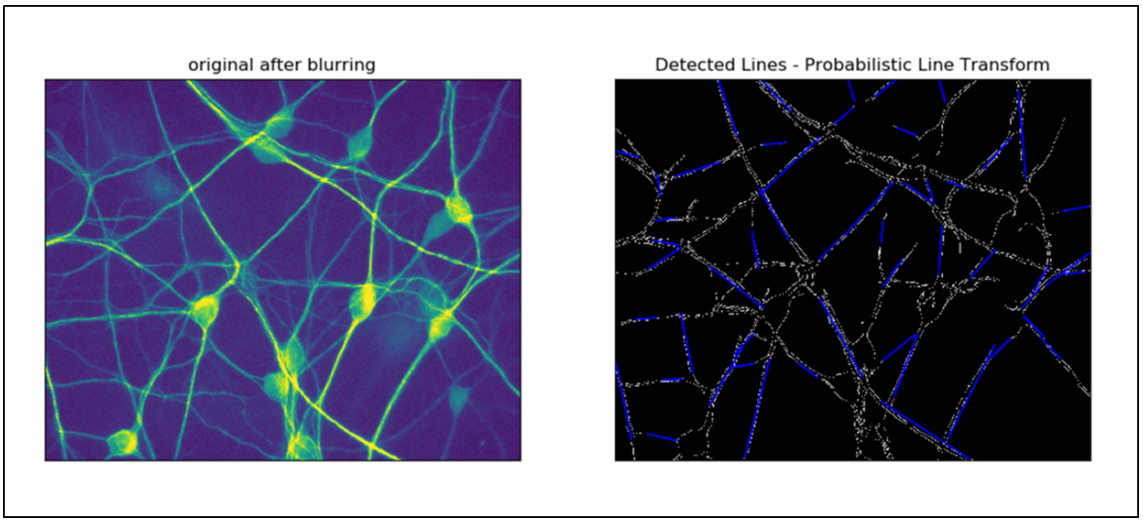

Figure 1: Image after blurring and segmentation.

3.2 Merging line segments

The two functions we described above creates an image with multiple lines and duplicates. Namely, it is often required to link or to merge line segments belonging to the same line. In view of this problem, we have used a line merge and deduplication function for the linking or merging of edge line segments belonging to a single line is proposed; the segments can be non-overlapping or overlapping, either fully or partially. The criteria for merging based on proximity of segments and on directional difference are included in the process and therefore you must update the parameters of: minimum distance to merge and minimum angle to merge.

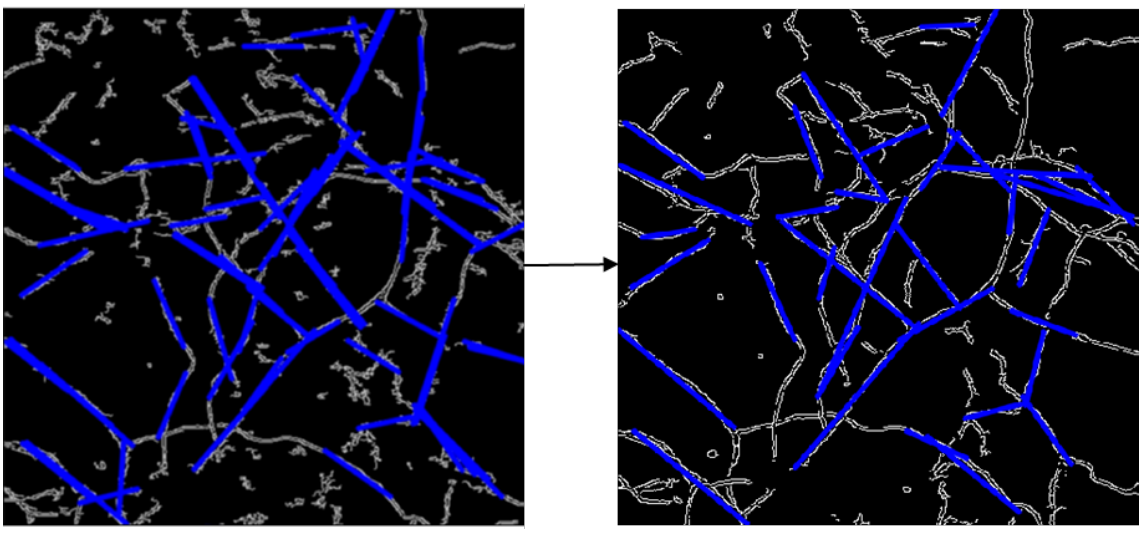

Figure 2: Image before and after merging.

3.3 Dendritic information

The Information obtained for each line marked including coordinate, angle and length:

Coordinate
Once the segmentation has been performed, we obtain the coordinates of each marked line. Each line is represented by four numbers((x0,y0)(x1,y1)), which are the two endpoints of the detected line segment. These coordinates are the first data information and with them we can get the length and angle in space.

Angle
math.atan2((y0 – y1), (x0 – x1)) return atan(y / x), in radians. The result is between -pi and pi. The vector in the plane from the origin to point (x, y) makes this angle with the positive X axis. The default value of angle will be in degrees(in a 180 degree range).

Length
Calculate distance between two points by math.sqrt((x1 - x0) ** 2 + (y1 - y0) ** 2) and its Return the length of each line.

3.4 Analysis

Perform various analyzes on the data obtained:

Classification
Classification of parallel groups and NOT parallel using map() function. Since the parallelism of the lines is calculated by comparing the angles, in this study we have defined that parallelism can be with deviation of +angle(=180/(number of lines)).We calculated for each line separately, we took its angle and made a range of its by adding the +angle(=180/(number of lines)) and checked which neighbors have an angle that is within that range.

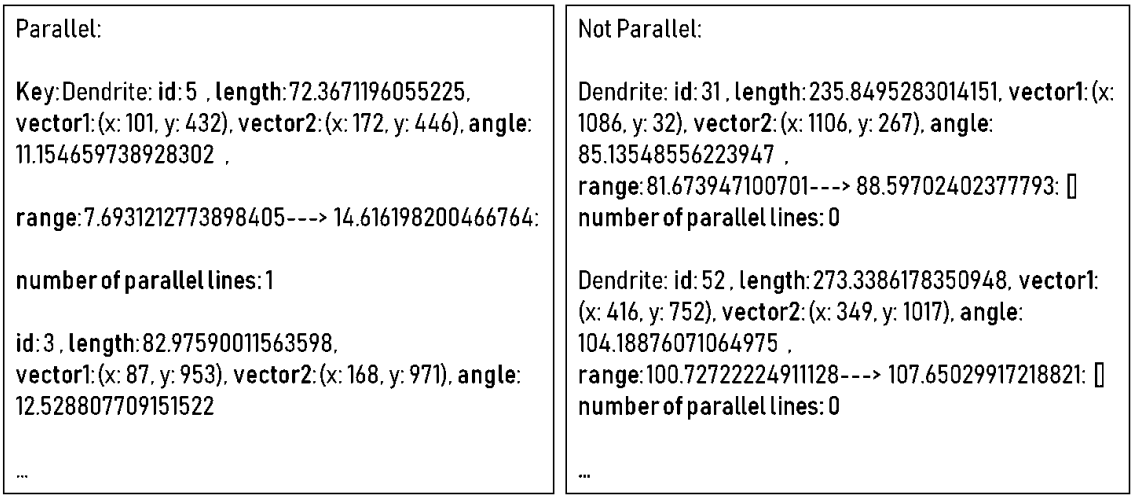

Figure 3: Classification information.

Angular results
To show the angular distribution and clustering of close-angled dendrites we used Bar graph and angular scatter plot. In each graph the columns on the X axis divide by ratio of 180/(number of lines). In the bar graph, we can see several lines exist in each range of angles, and in the scatter plot we can see the id of the dendrite and his angle.

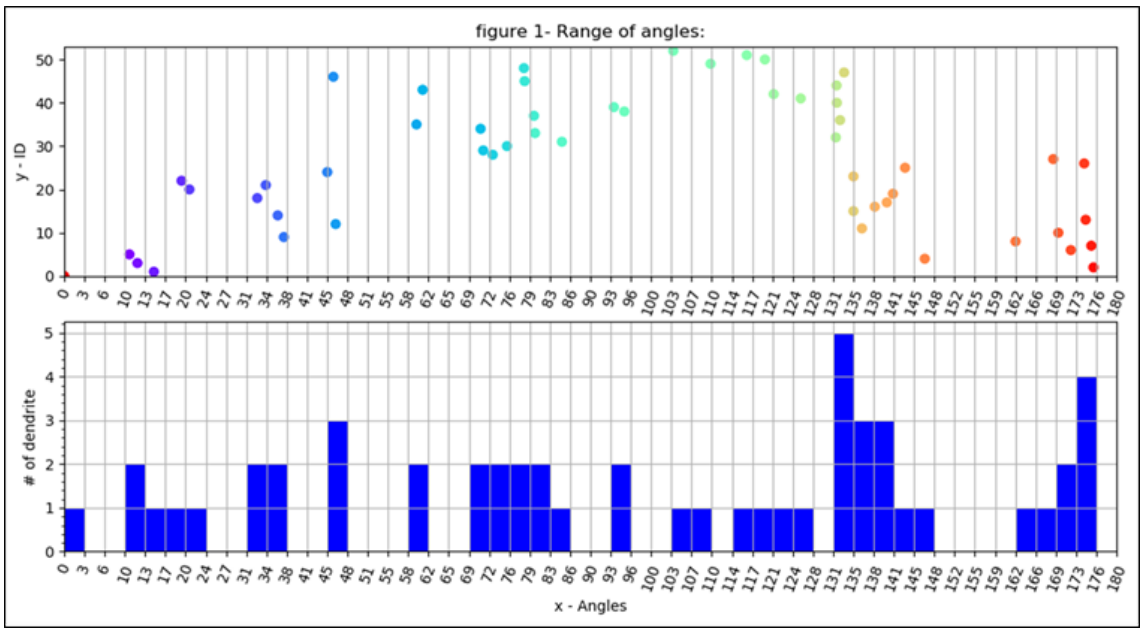

Figure 4: Angular scatter plot and bar graph.

The shortest distance between parallels groups
Given two lines, in 2 dimensions, defined by numpy.array pairs (a0,a1,b0,b1) return the closest points on each segment and their distance using normalization. Because we have defined that parallelism can be with deviation of + - angle (=180/(number of lines)) the distance between two lines is not fixed; that's why we chose this function, who find the shortest distance between two line segments. A line segment is defined by two endpoints. So for example one of the line segments (AB) would be defined by the two points A (x1,y1) and B (x2,y2) and the other (CD) would be defined by the two points C (x1,y1) and D (x2,y2).

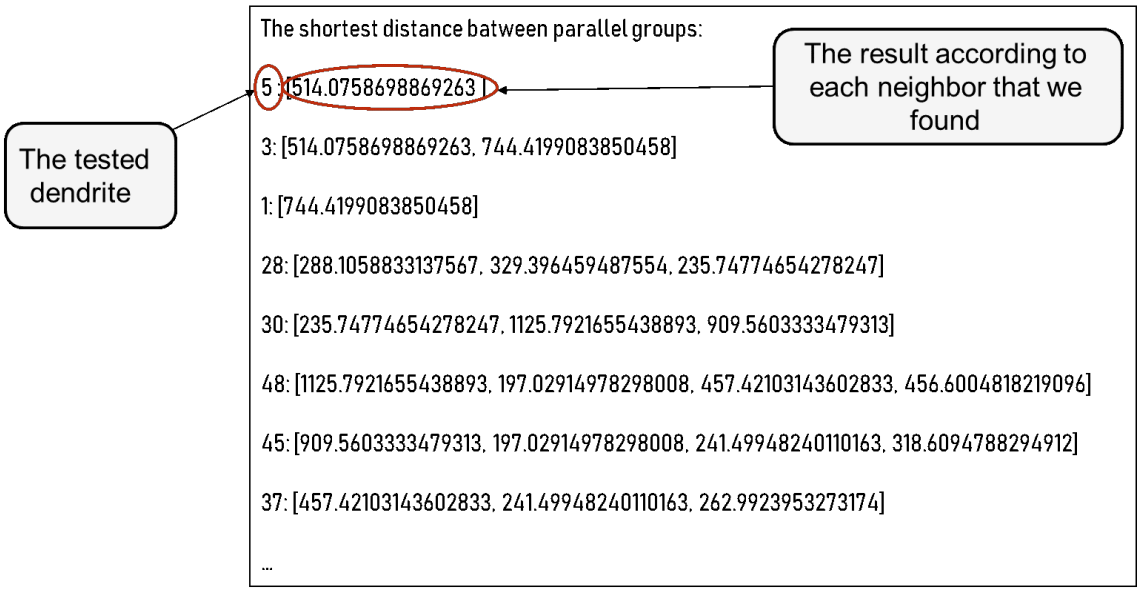

Figure 5: Shortest distance between every parallel group.

Length average of parallel group and not parallel group
Displays the lengths of the parallel and non-parallel group by Bar graph and the average of each group using average function. We assume that NOT parallel group should be longer than parallel group.

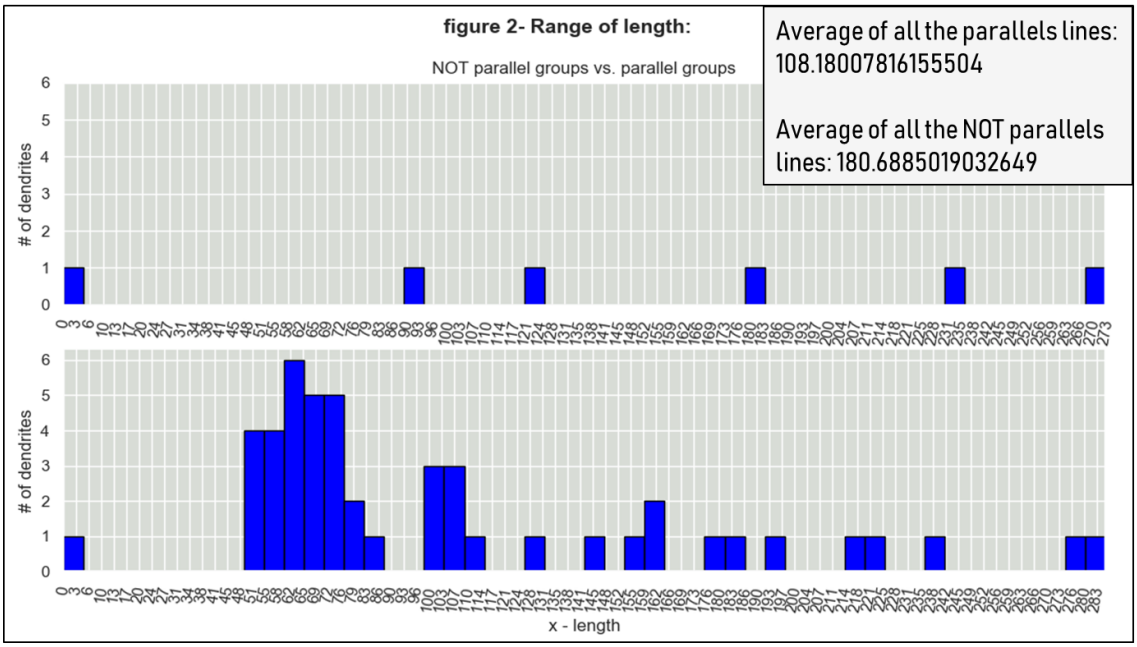

Figure 6: Length graph in parallel and not parallel group and their average.

4 System Details

Our system runs on Windows 10 using Python with the following main packages:

- OpenCV library- designed to solve computer vision problems
- NumPy- fundamental package for scientific computing library.
- Matplotlib- a Python 2D plotting library
- PyQt5- In order to manage the GUI.

The software operates in two main steps:

4.1 Updating settings and parameters part

To use the interface, you must first select an image from the computer files and then update the settings for line recognition (for straight line marking: lower and upper threshold For Line Merge: Minimum Angle to Merge, and Minimum Distance to Merge, Angle Units). Because each image individual and different, we recommend "playing" with the settings in order to try and achieve maximum segmentation.

4.2 Outcome part

After all the parameters updates, in this part, the former step that the system does, by pressing the "ok" button, is to create several figures files of the segmentation, the indication of each dendrite that identified and the four analyzes that performed.

4.3 GUI

User interface which the user insert the image and receive a comparison, of the original image and the segmentation image, dendrite data and corresponding line tax.

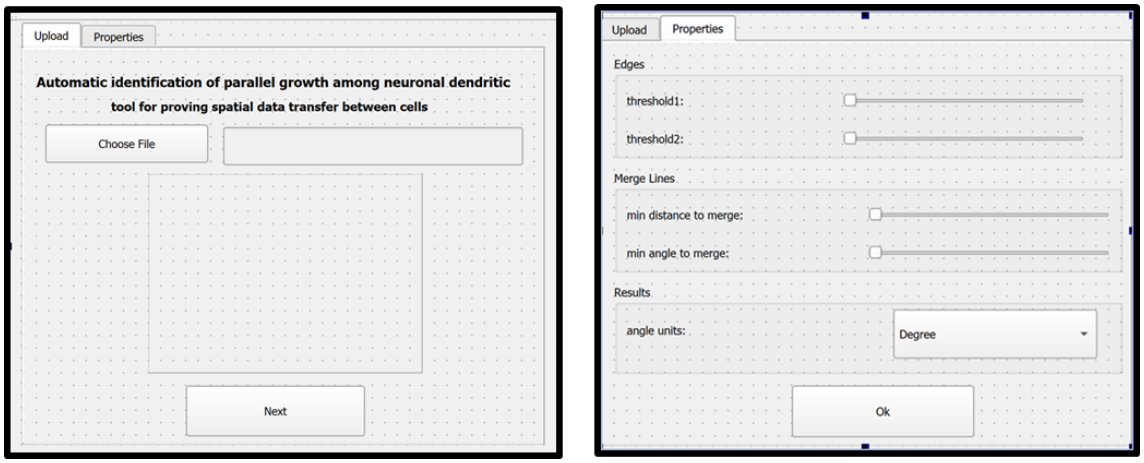

Figure 7: Updating settings and parameters page.

5 Results

The result show that occurrence of parallel growth of real dendritic branches is dramatically more frequent than the value found in the combinatorial random simulation, indicating that such type of growth is not random.

6 Conclusions

According to our experiments we came in with a several conclusions:

- First, We would like to stress out that the analysis of the results is inclusive and not individual.
- Second, from the images test that we saw there are difference in marking the lines between zoom in images and zoom out images. To get a good image with high-grade lines, the image better be enlarged.
- Third, marking the lines is not 100 present strict. The dendrites in the resulting images are sometimes marked shorter than they are. Since we want to show as much as possible that there is a significant difference in relation to parallel and non-parallel lines, the divergence of the marking is correct and negligible.
- Fourth, one of our most important conclusions, from the analysis of the obtained information and the discovery of parallel dendritic branch teams, we can clearly see that there are groups of angles with a significant approximation. In a situation where there was no connection between the dendrites, we should have received very few groups of parallel dendrites. This reinforces our hypothesis, that parallel vs non-parallel growth ratio is non-random.
- Another conclusion to justice our claim, later in the study we can grow dendrites, in the second time, in the lab, however this time we will change one of the parameters.Then we can insert the image into the system, get data and see if there is a change in the parallel growth and actually in the inter-cellular communication.

7 Future Work

The developed system can be used to identify branch-branch junctions, as they influence neuronal activity, as well as for accumulating biological data from other cell types, and perhaps even from tissues.

In addition we would like to note that the project is a part of a manuscript in preparation executive by Prof. Danny BARANES head of Molecular Biology Department and Dr. Refael MINNES from the Physics Department.