Skeletonization and fractal analysis of microglial cells in neonatal brain

Cheng-Hao Lin

A thesis

Submitted in partial fulfillment of the

Requirements for the degree of

Master of Science

University of Washington

2021

Committee:

Elizabeth Nance

David Beck

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Skeletonization and fractal analysis of microglial cells in neonatal brain

Cheng-Hao Lin

Chair of the Supervisory Committee:

Elizabeth Nance

Department of Chemical Engineering

Microglial cells can assume a range of morphologies that exist on a continuum, spanning from ramified, to bushy, to an amoeboid phenotype. Studies show that the morphology of microglial cells can be correlated with severity of injury and response to treatment in traumatic brain injury and neurodegenerative diseases such as Alzheimer's disease, underpinning the importance of understanding microglial morphology. In recent years, fractal dimension has become a proven measure of microglial morphology. However, research that analyzes fractal dimension often requires multiple software platforms to perform the analysis, making the analysis inefficient and computationally expensive. In this work, we introduce a compiled, automated Python package to analyze microglial morphology called Skeletonized Cell Analysis of Regional Features, or SCARF. By applying proper thresholding techniques, images are segmented, binarized, and skeletonized to simplify the complicated cell structure of microglia. Individual objects are then accessed via a number of morphological features and compared between different treatment groups or brain regions. From this Python package, we can also measure the fractal dimension of extracted skeletonized objects to quantify the morphological complexity of the microglial cells in response to injury and treatment. This study serves as an inspiration for future automated work in image analysis, as an easily obtainable package for relevant researchers, with gentle learning curve and can be easily customized to accommodate different field of study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

First, I would like to thank my advisor, Professor Elizabeth Nance, for all the advice and support over past two years. Your expertise and insight have always guided me when I'm facing questions in research. Outside of research, you are always generous to provide support to help me with my career plan and development. I am grateful to have you as my advisor in UW.

I would also like to thank Hawley for all the guidance in research. You have always been a positive impact, and there is always something to learn from you. I really appreciate all the help I received from you.

I must also thank my parents for all the financial and emotional support throughout my life. I really appreciate that I don't have to worry for the financial situation and can pursue high level education.

Many thanks to my roommate Peter for all the emotional support. You always cheer me up when I'm not feeling well. Thanks to my friend Joe, Jack and Henry for constantly checking in with me during the pandemic.

Thank you all for everyone that has helped me in any form. My deepest appreciation to have you all supported me in the past two years.

CHAPTER 1: Introduction

1.1 Brain cells morphology and brain disease

Brain cells play important roles in normal brain functions. Vital aspects in our daily life, such as perception, sensation, emotion and speech, depend on a range of cell types in the brain. These functions are related to the morphological structures of brain cells. In addition, in diseased brains, the morphological heterogeneity of brain cells are found to be altered compare to healthy brains.[1−3] For example, in the brain affected by Alzheimer's disease, different levels of astrocyte heterogeneity are observed during different phases of the disease.[1,2] Thus, changes in morphological heterogeneity of brain cells can be an indicator of disease onset and disease progression.

1.2 Behavior of microglial cells and their morphology

Microglial cells are the resident macrophages of the central nervous system (CNS), accounting for 5-20% of total glial cell population within the CNS parenchyma.[4,5] Microglial cells are conventionally known for their roles in immune defense. In recent research, their ability of actively interacting with neurons and maintaining the functions of synapses are showcased.[6] While resting in their steady state, microglial cells are highly ramified with branched processes.[7] These processes constantly survey the nearby environment and actively contact synapses to modify and eliminate the synaptic structure.[8] Microglial cells are activated when homeostasis is disturbed due to insult or manipulation to the brain. In this upregulated state, microglial cells appear to be bushy, or in some cases amoeboid. The morphological differences between resting and different activation states can be drastic and distinct. Post injection of an inflammatory stimulus, microglial cell activation is detected by analyzing the morphological phenotype across microglial cells population using fractal dimension analysis.[9] Microglial cells also show regional difference in morphological phenotype and response to inflammation .[10] Therefore, microglial cell morphology can be used as a measure of changes in their surrounding environment.

1.3 Relation between microglial cells morphology and CNS disorders

The relationship between microglial cells and neurodegenerative diseases that are closely related to aging, such as Alzheimer's disease and Parkinson's disease, has been discussed in recent years. Research shows that the morphological features of microglial cells gradually change as the brain ages.[11] Additionally, early and highly localized activations of microglial cells are recorded after traumatic brain injury (TBI) .[12] These results, as well as many others, suggest that measuring morphology to analyze microglial activation state can be one indicator of pathological onset and severity in both chronic disease and acute injuries.

1.4 Quantification of microglial cells features with automated work pipeline

Conventional analysis of cell morphology requires extensive human intervention.[13] When dealing with large sets of images, it is time-consuming for researchers to identify the cells of interest and make comparisons to existing data one image at a time. Studies of microglial cell morphology using software have emerged over the past decade.[14–16] However, these studies often use different software in each step of analysis, reducing the reproducibility and efficiency of the workflow. To automate the image processing and increase efficiency, I developed the Skeletonized Cell Analysis of Regional Features (SCARF) package. The idea of SCARF is adapted from diff_register, a Python package developed by former Nance lab member Dr. Chad Curtis. SCARF is designed to extract microglial cell features from the skeletonized images. The SCARF package is constructed with Python, one of the most popular programming language and data analysis tools. SCARF features a simple yet customizable work pipeline that requires only minimum knowledge in programming for the user. SCARF is fully written with Python and open-source libraries. With SCARF, the microglial morphological features as well as fractal dimensions can be analyzed in an automated way. Figure 1-1 shows the proposed workflow for SCARF.

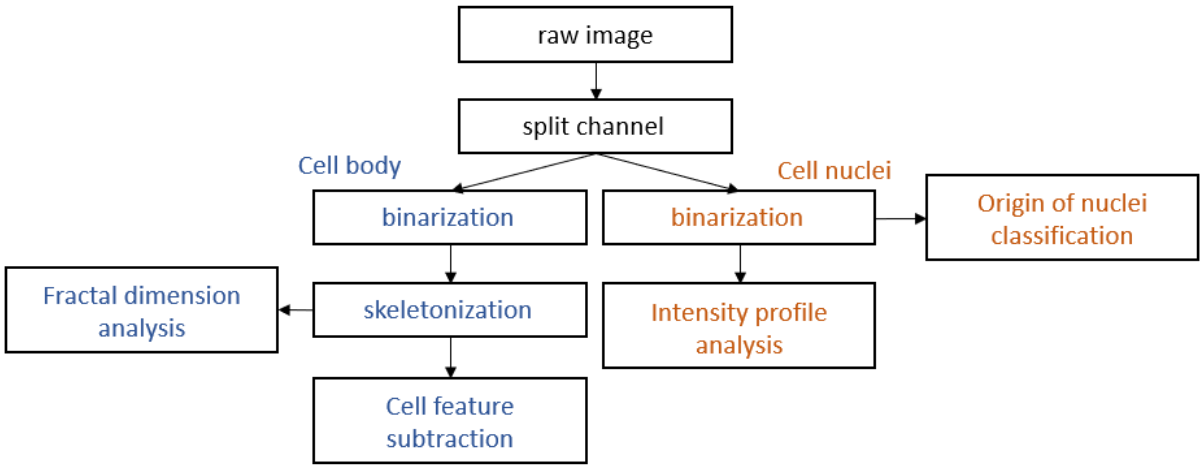


Figure 1-1 The schematic diagram of proposed workflow of SCARF.

CHAPTER 2: Methods

The SCARF package analyzes fluorescence microscopy images. Images are provided as .tiff files by former Nance lab member Kate Hildahl for package development and result evaluation.

2.1 Computational setup

The SCARF package is finalized and compiled in Python 3.8. This package uses built-in library in Python, primarily NumPy [17] and Scikit-Image [18], as well as some external library and package such as Skan.[19] To run the package, all necessary libraries must be installed to the local Python file location. All the version numbers of necessary libraries are recorded in the GitHub repository (https://github.com/Nance-Lab/scarf). A detailed setup instruction of the package can be found in the GitHub repository.

2.2 Image treatment before processing

The SCARF package is designed to process and analyze fluorescence microscopy images with two channels, Iba1 stained for microglial cell body and DAPI stained for cell nuclei. A set of image treatment is applied to simplify downstream analysis, as well as prevent possible error in rare occasion. First, channel splitting separates Iba1 and DAPI channel for individual access. For each channel, a distinct variable name is assigned for downstream usage. After that, image pixels are normalized to a fixed range from 0 to 255. For compatibility with NumPy in downstream analysis, the data type of pixels array is also reassigned as 8-bit unsigned integer (uint8).

2.3 DAPI stained channel for cell nuclei

In order to capture the objects, the DAPI stained channel of the image undergoes a series of thresholding, segmentation, and object extraction. Individual objects on the image can then be further accessed.

2.3.1 Image binarization

First, we apply Otsu thresholding function with Scikit-Image on the image array of normalized DAPI stained channel. Otsu is one of the most robust thresholding methods, and it is popular for precise thresholding when the objects are not extremely complicated. Cell nuclei tend to be amoeboid, so it is simple enough for Otsu to achieve a good threshold. The output of Otsu function is then used to map out the object in binary form, consisted with only true or false in the image array. Then, the potential small holes generated by the thresholding is filled with binary_fill_holes function in Scikit-Image. Finally, objects that is smaller than 50 square pixels are removed from the binary image array. The removal of small objects prevents the background noises from being considered as actual objects.

2.3.2 Object-based analysis for cell nuclei

Individual objects from the binarized image can be accessed using regionprops function from Scikit-Image. The regionprops function returns a list of region properties for each object respectively, containing multiple attributes such as coordinate and area of the object. By marking the coordinate of individual object on Iba1 channel skeletonized image, objects that locate in the microglial cell body area are considered microglial cell nuclei. This can separate microglial cell nuclei from other nuclei in the image.

2.4 Iba1 stained channel for microglial cells

The objects in Iba 1 channel of the image are segmented into binarized form with various approach of thresholding. Different pipeline of binarization is applied depending on the intensity ratio between object and background, orientation of objects and background noises. After binarization, objects are skeletonized and extracted for downstream analysis.

2.4.1 Image binarization for image without quality loss

For optimized binarization result, a unique set of image array processing is developed to address the complexity of microglial cell body. First, the object filter array and the background filter array are established in order to calculate the binary mask array. The object filter array is calculated from local area, whereas the background filter array is calculated from a wide area. For each pixel, the value of object filter is the median intensity within area of structuring element star(2) with respect to the pixel coordinate. The value of background filter is determined with similar approach, but using star(15) instead. The binary mask array is calculated with the two filters:

$$binary\ mask\ array = \begin{cases} true, if\ (object\ filter - 0.5 \times background\ filter) > 60 \\ false, if\ (object\ filter - 0.5 \times background\ filter) \leq 60 \end{cases}$$

The closing function in Scikit-image is then applied to eliminate small dark spot in object and connect small bright cracks. Finally, objects with area smaller than 350 square pixels are removed from the binary image.

2.4.2 Image binarization for image with severe background noises

In the case when background noises are unavoidable in the image, object can be better filtered with certain adjustments in the binarizing process. To capture the edge of the objects in a noisy background, using star(4) instead of star(15) as the structuring element when calculating background filter. The binary mask array is then calculated as:

$$binary\ mask\ array = \begin{cases} true, if\ (object\ filter - 0.65 \times background\ filter) > 60 \\ false, if\ (object\ filter - 0.65 \times background\ filter) \leq 60 \end{cases}$$

Changing the multiplier from 0.5 to 0.65 can reduce the chance of falsely include background noise as part of the object.

2.4.3 Image binarization for image including object with discontinued intensity

Erosion method can help fixing objects with discontinued intensity. First, calculate object filter array in area of structuring element star(3). Based on the object filter, the edge filter array is obtained by applying Prewitt edge detection function in Scikit-Image. The opening function in Scikit-Image is then applied on the edge filter array to remove small bright spot and link the small dark cracks together. The background filter array is established by taking the mean value of the edge filter array. The edge filter array and the background filter array are used to calculate the binary mask array:

$$binary\ mask\ array = \begin{cases} true, if\ edge\ filter > background\ filter \\ false, if\ edge\ filter \leq background\ filter \end{cases}$$

Using the returned binary mask array, the binary image array is constructed with the reconstruction function running erosion method. Finally, Small holes and objects smaller than 350 square pixels are removed to clean up the binary image.

2.4.4 Image skeletonization

The final binary image is skeletonized using skeletonize function in Scikit-Image. In some cases, the skeletonized objects appear to be discontinued at the thinner part of the objects. In order to address the discontinuity in the objects, the dilation function is then applied to connect small fraction at the end of the object. By Using the dilation function, better object connectivity can be achieved. The separated skeletonized objects are also used in the fractal analysis process to obtain the fractal dimension.

2.4.5 Object extraction and removal of unwanted objects

I next ran regionprops function on the skeletonized image to obtain the list of regional property. In the case when the original image has overlapping objects or objects that are too hard to recognize, the removal of problematic object is possible. After manually assigning the coordinate of problematic object, the removeObject function in SCARF package can remove the identified problematic object from the regional property list. This excludes the object from downstream branch feature analysis.

2.5 Branch feature subtraction and analysis

The Skeleton function in Skan is used to obtain relevant information in the skeletonized image. The summarize function can return a pandas [20] DataFrame for easier access of the result data. The output DataFrame contains data such as branch length distribution that can be used to further analyze the branch feature difference between different treatment groups and regions.

2.6 Fractal dimension calculation with FracLac

The fractal dimension of extracted objects is calculated with FracLac,[21] an extension plugin in ImageJ.[22] Individual objects go through the box counting fractal and lacunarity analysis method in FracLac. This method calculates the Minkowski dimension by setting up grid and counting boxes that covers the object. For this calculation, the grid number is set to 1.

CHAPTER 3: Image process pipeline development and cell feature subtraction

In some cases, microscopy images contain a defect that can interfere with the image processing. To avoid such interference, I developed two pipelines that can accommodate severe background noises and discontinuity in pixel intensity which are added to SCARF. The downstream analysis results using the improved pipeline are shown later in the chapter.

3.1 Thresholding and binarizing for different image conditions

Since the image is binarized at the beginning of the process, different binarization function and parameters can have a huge impact to downstream process and final analysis. I developed three binarization pipelines to cover the majority of the image conditions. Figure 3-1 shows the flow chart of three pipelines. When the result is still poor with these adapted pipelines, manual adjustment of parameter is also possible to deal with troublesome images that contain both background noises and discontinued objects.

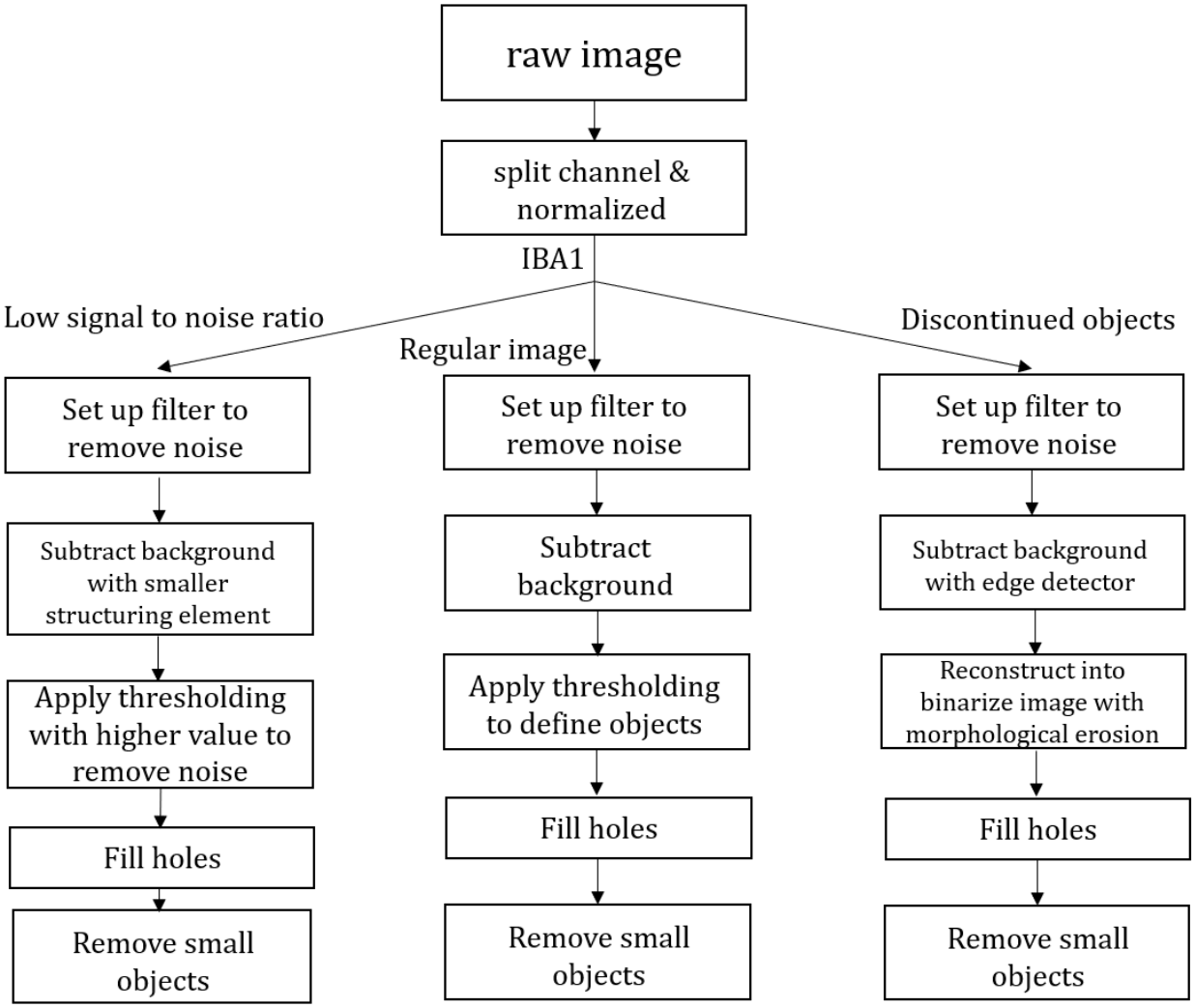


Figure 3-1 The flow chart diagram of three different binarization pipelines in Iba1 channel. To obtain better binarized images, use different treatment process according to the image condition.

3.1.1 Functions and parameter choice and adjustment for image with low signal to noise ratio

To avoid the background noises being considered as part of the objects, I adjusted the parameter used in regular binarization process. Constructing the background filter array with a smaller structuring element result in a sharper recognition along the edge of the object and reduces the possibility of including the noise spots into the object. As a trade-off, however, the outline of the object is not captured accurately if the intensity of cell body is not uniform. This pipeline is recommended when the background noise is causing objects to overlap. Different size of structuring elements and the binarizing result are compared in Figure 3-2. A list of functions and the corresponding adjustable parameters are shown in Table 3-1.

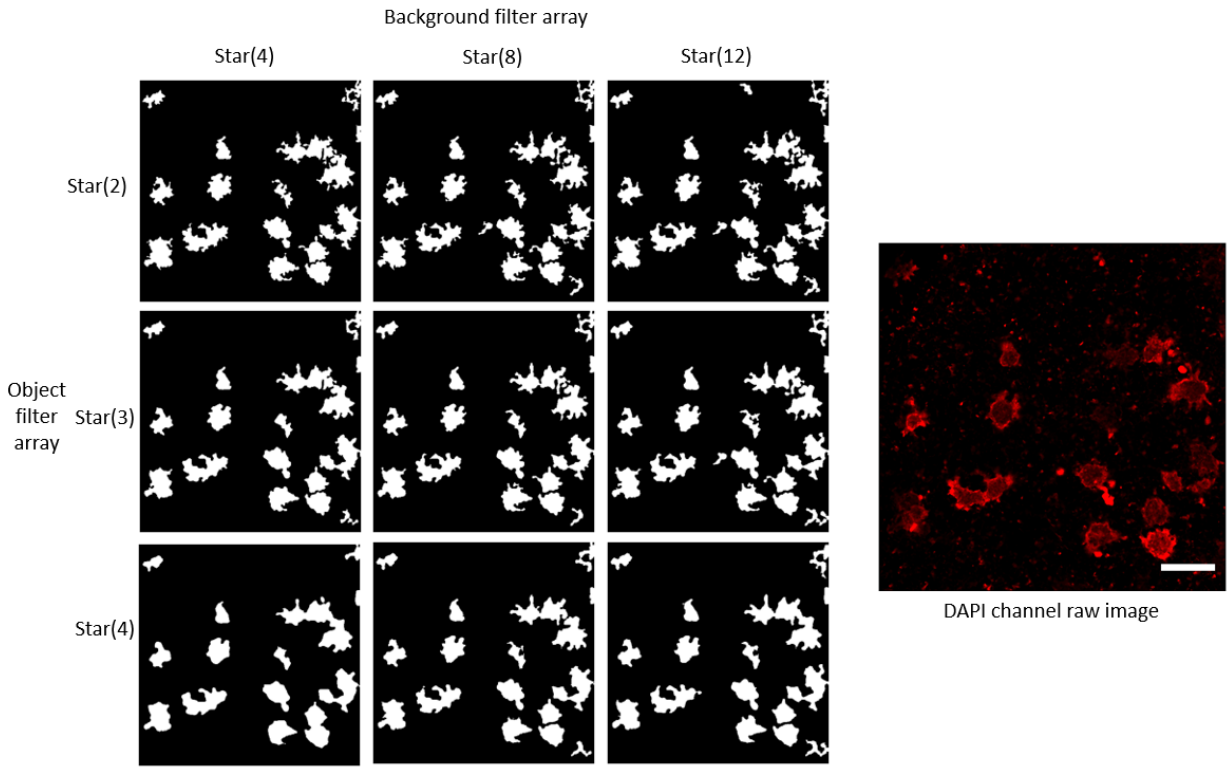


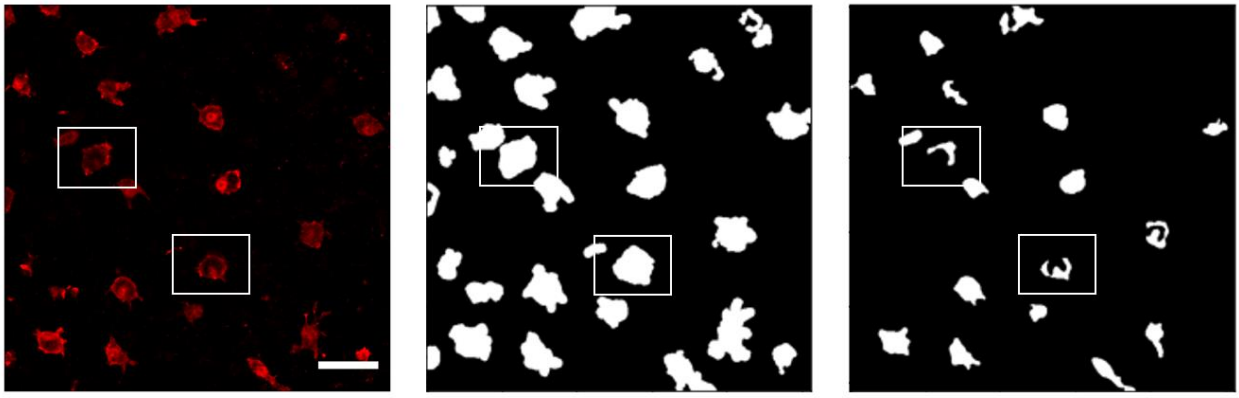
Figure 3-2 The effect of different structuring element sizes on the binarized result. For object filter array, using smaller structural element provides sharper edge recognition.

Table 3-1 Functions and parameter adjustment for regular image and image with low signal to noise ratio.

Function	Parameter for regular image	Parameter for image with low S/N ratio	note
filt_image = median()	selem = star(2)	selem = star(2)	Set up object filter
backg = median()	selem = star(12)	selem = star(4)	Set up background filter
filt_min = filt_image – n * background	n = 0.5	n = 0.6	Calculate binary filter
biim = filt_image > m	m = 60	m = 60	
full_im = closing()	selem = star(1)	selem = star(1)	Generate binary image with closing function

3.1.2 Functions and parameter choice and adjustment for image with discontinued intensity

To fill up the discontinuity intensity in objects, the reconstruction function is introduced to the pipeline. By setting up the binary mask array with edge detector, the binary image before reconstruction is slightly dilated comparing to regular process. The binary image is then reconstructed by erosion to fill the dark part inside the object. Figure 3-3 compares the reconstruction method and the regular binarizing process. A list of function and the adjustable parameter for this pipeline is shown in Table 3-2.



DAPI channel raw image | Binarized image using erosion method | Binarized image using regular method

Figure 3-3 The comparison between regular and erosion method when the intensity is discontinued. Compared to the regular method, the binarized image using erosion method captures the edge of the discontinued objects accurately.

Table 3-2 Functions used in erosion method.

Function	**Parameter**	**note**
filt_image = rank.mean()	selem = star(3)	Set up object filter
edge = prewitt()	N/A	Set up edge filter with Prewitt method
openim = ()	selem = star(2)	Generate temporary binary image with opening function
filledim = reconstruction()	Method = 'erosion'	Generate final binary image with erosion method
Filled_im = binary_fill_holes()	N/A	

3.2 Branch feature subtraction

The branch features of skeletonized objects are calculated and summarized with Skan. Images for this project were taken using a confocal scanning laser microscope organotypic brain slices obtained from the ferret brain. Ferrets were postnatal (P) day 21-23. Briefly, brain slices were exposed to oxygen-glucose deprivation (OGD) for different exposure times, as we've previously described.[23] Non-OGD exposed slices serves as normal controls (NC). At specified experimental endpoints, slices were fixed in formalin and stained with DAPI and anti-Iba1 for microglia. Imaging settings were optimized to capture microglial and nuclear morphology. Images were taken by former Nance Lab member Kate Hildahl. Images were exported as .tiff files and shared via the Nance Lab shared Google Drive.

3.3 Results

The distributions of skeletonized microglial cell branch length in different treatment groups and regions are calculated and compared. Figure 3-4 shows the swarm plot and box plot of cell branch length between 3 treatment groups, and the histogram is also shown as Figure 3-5.

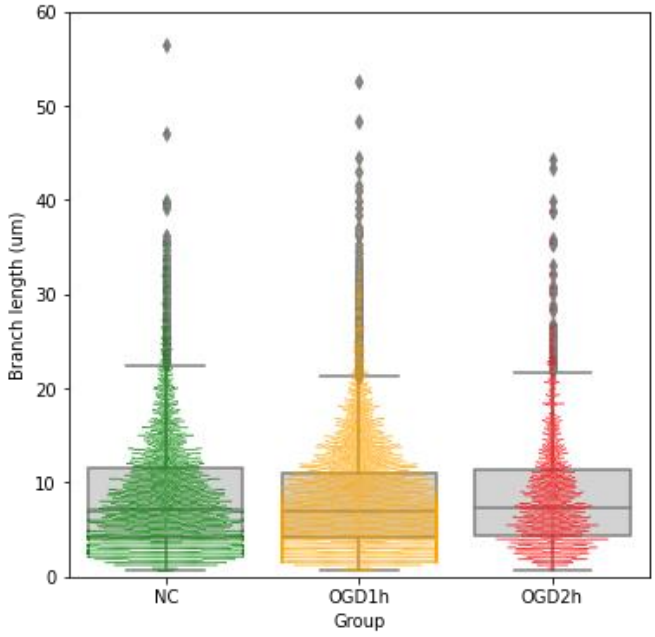


Figure 3-4 Distribution of skeletonized microglial cell branch length in organotypic ferret brain slices exposed to 1hr OGD or 2hr OGD, and compared to NC. The three horizontal lines in the middle box of the box plot represent 25th, 50th and 75th percentile.

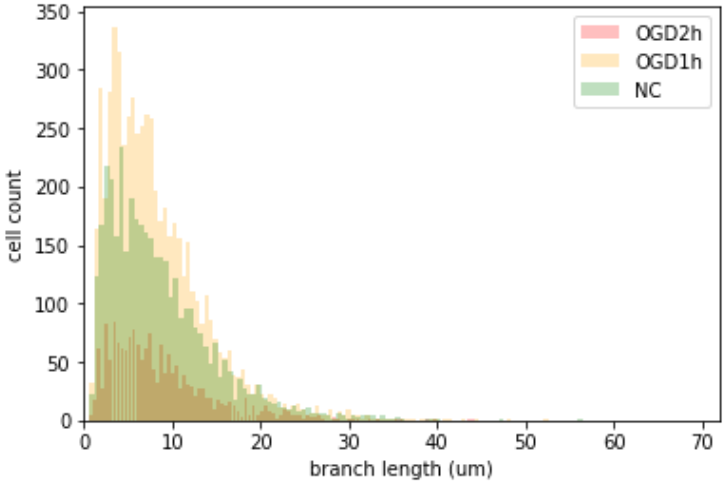


Figure 3-5 Histogram of skeletonized microglial cell branch length of 1hr OGD, 2hr OGD and NC.

The regional difference in microglial branch features is also compared in Figure 3-6.

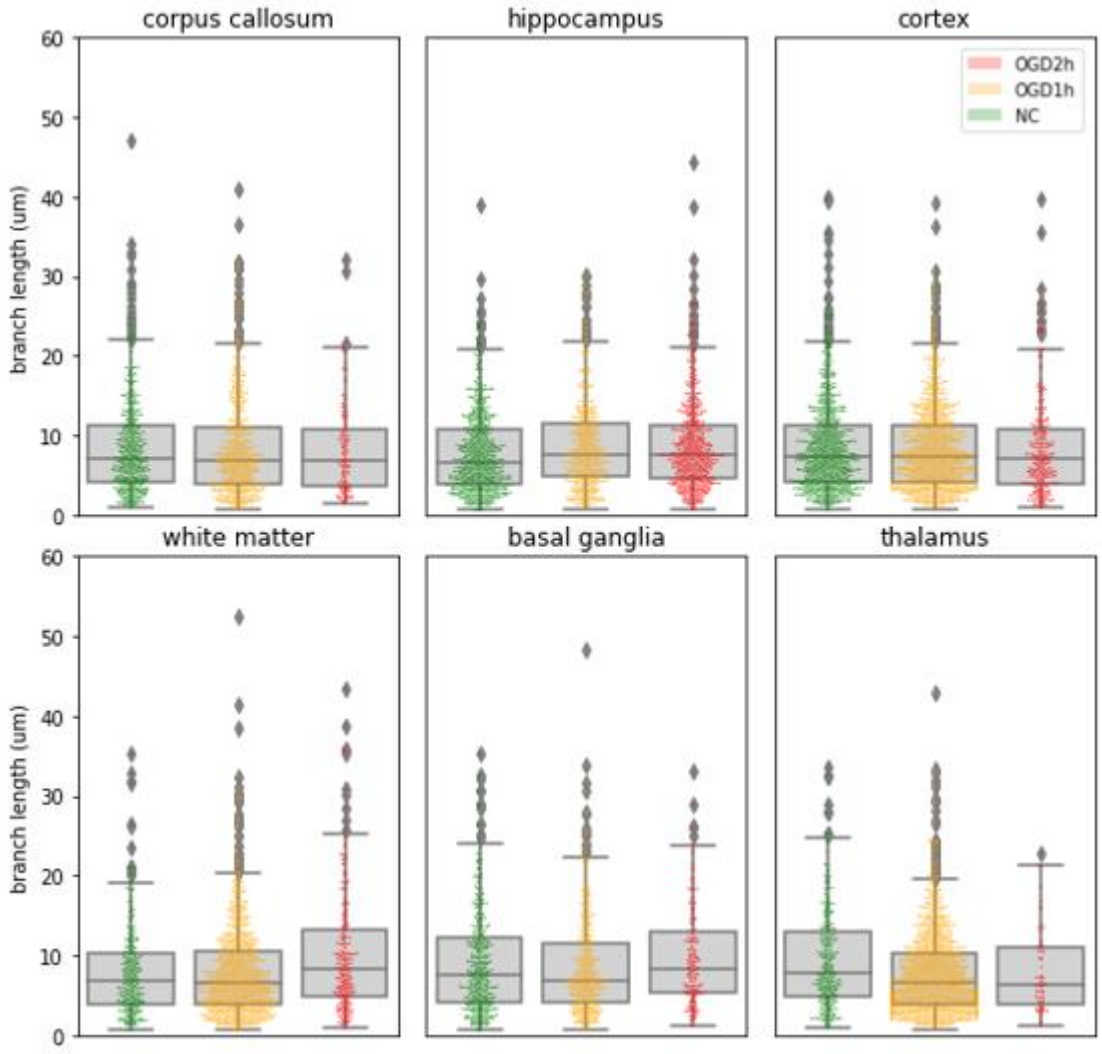


Figure 3-6 Distribution of skeletonized microglial cell branch length in different brain regions for 1 h OGD, 2 h OGD, and NC conditions

3.4 Discussion

The skeletonized microglial cell branch length appears to be similarly distributed in different treatment groups. The number of branches in 2hr OGD treatment group is significantly lower than the other two treatment groups. This is possibly due to the higher cell death rate results from longer exposure time of OGD, as well as the remaining living microglial cells taking a less ramified activation state. Due to the nature of skeletonization process, the more amoeboid the cell is, the harder for the skeletonization function to pick up the distinct branch feature from the process. This explains the

distribution curve of 2hr OGD being slightly flatter than the other two, as the skeletonization process may not pick up the feature correctly.

The distribution of branch length in different region also appears to be highly similar. The branch count is lower in regions such as basal ganglia, indicating that microglial cells in these regions are less ramified and possibly having lower morphological heterogeneity.

The distribution results being similar show that for such kind of analysis, it is better to only pick several images that are in good condition than using sets of images. That way, the slight distinction in branch features can be captured by skeletonization process and present in the result easier.

3.5 Conclusion

The number of branches on skeletonized cells in 2hr OGD group is lower than the other groups, although the distribution of branch lengths is comparable across all groups. The differences between regions are relatively minimal, but still provides some indication in microglial cells behavior. The cell count to branch count ratio can possibly be an indicator for the overall microglial cell activation state in different conditions. The cell count to branch count ratio in different regions are compared in Figure S3-1.

The study demonstrates the ability to correctly capture cell features in skeletonized form and analyze branch feature and distribution with SCARF package.

CHAPTER 4: Fractal analysis of skeletonized microglial cells

Fractal dimensions analysis has been proven to be a good indicator to describe the morphological complexity in cell morphology studies.[9,16] To calculate the fractal dimensions for an individual cell, the cell body must be extracted from the raw image. With the skeletonization process in SCARF, individual skeletonized cells can be extracted from the image and be analyzed. In this chapter, the fractal dimensions of the OGD image set provided in Chapter 3 are analyzed using the SCARF skeletonization process and FracLac fractal dimensions calculation.

4.1 Fractal dimensions and microglial cell phenotype

Quantifying the morphological features of microglial cells can be challenging. During different activation state, microglial cells show a wide range of phenotypes that is hard to classify by human eye and manually. In recent years, fractal dimension has proven to be a solid indicator of morphological complexity in microglial cells.[14,16] Fractal dimension is a measurement of self-similarity in different scale of the image. When N is the number of new parts and S is the scale, the fractal dimension D_F of an image is calculated as:

$$N = s^{D_F}, D_F = \frac{ln\ N}{ln\ S}$$

The visualization of box-counting method to calculate fractal dimensions is shown in Figure 4-1.[24]

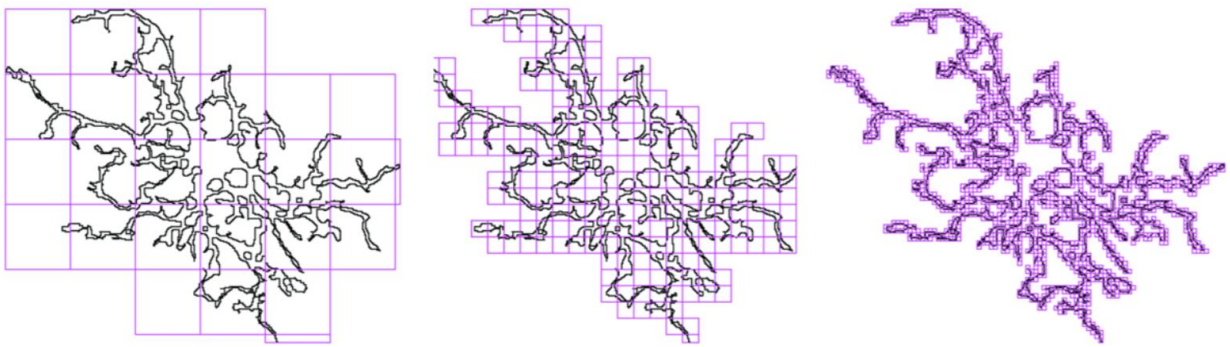


Figure 4-1 Visualization of the box-counting method to calculate fractal dimensions. Shape detail is captured by the box in different scales. As the scale of the object increase, more boxes are needed to capture the shape detail.

4.2 Fractal dimensions of skeletonized cells

The fractal dimension of the skeletonized objects can be calculated with the box counting method in FracLac. The same set of ferret brain slice microscopy images in Chapter 3 is used to analyze the fractal dimension difference between different treatments and brain regions.

4.3 Results

The fractal dimensions of individual skeletonized objects are calculated with FracLac and saved to Python as Pandas DataFrame. Objects that are below 300 square pixels are not analyzed, since these objects are often too small for the skeletonization to capture the branches distinctly. Fractal dimensions that are below 5% percentile or above 95% percentile are excluded to prevent bias from extreme cases of skeletonization.

4.3.1 Fractal dimension distribution between different treatment groups

The distributions of fractal dimension for skeletonized microglial cells for 1hr OGD, 2hr OGD and normal control are compared below as swarm and box plot in Figure 4-2. The histograms for different treatment group are also compared in Figure 4-3.

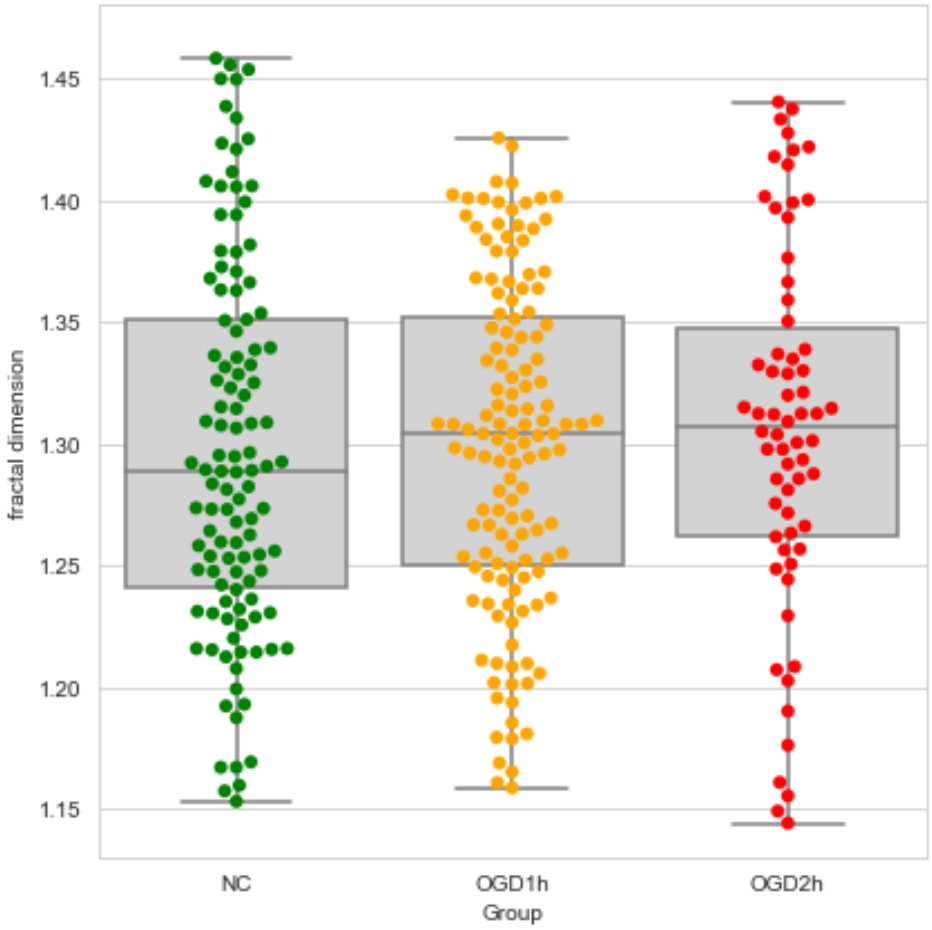


Figure 4-2 Distribution of fractal dimension of 1hr OGD, 2hr OGD and NC. The three horizontal lines in the middle box of box plot represent 25th, 50th and 75th percentile.

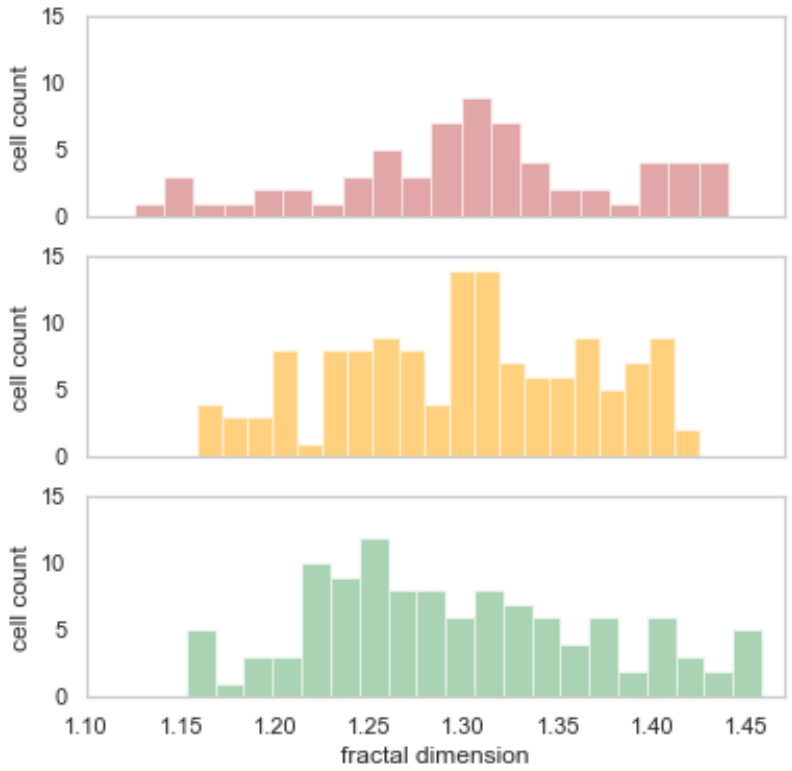


Figure 4-3 Histograms of fractal dimension of 1hr OGD, 2hr OGD and NC.

4.3.2 Fractal dimension distribution between different regions

The fractal dimension distributions of different brain region are compared in Figure 4-4.

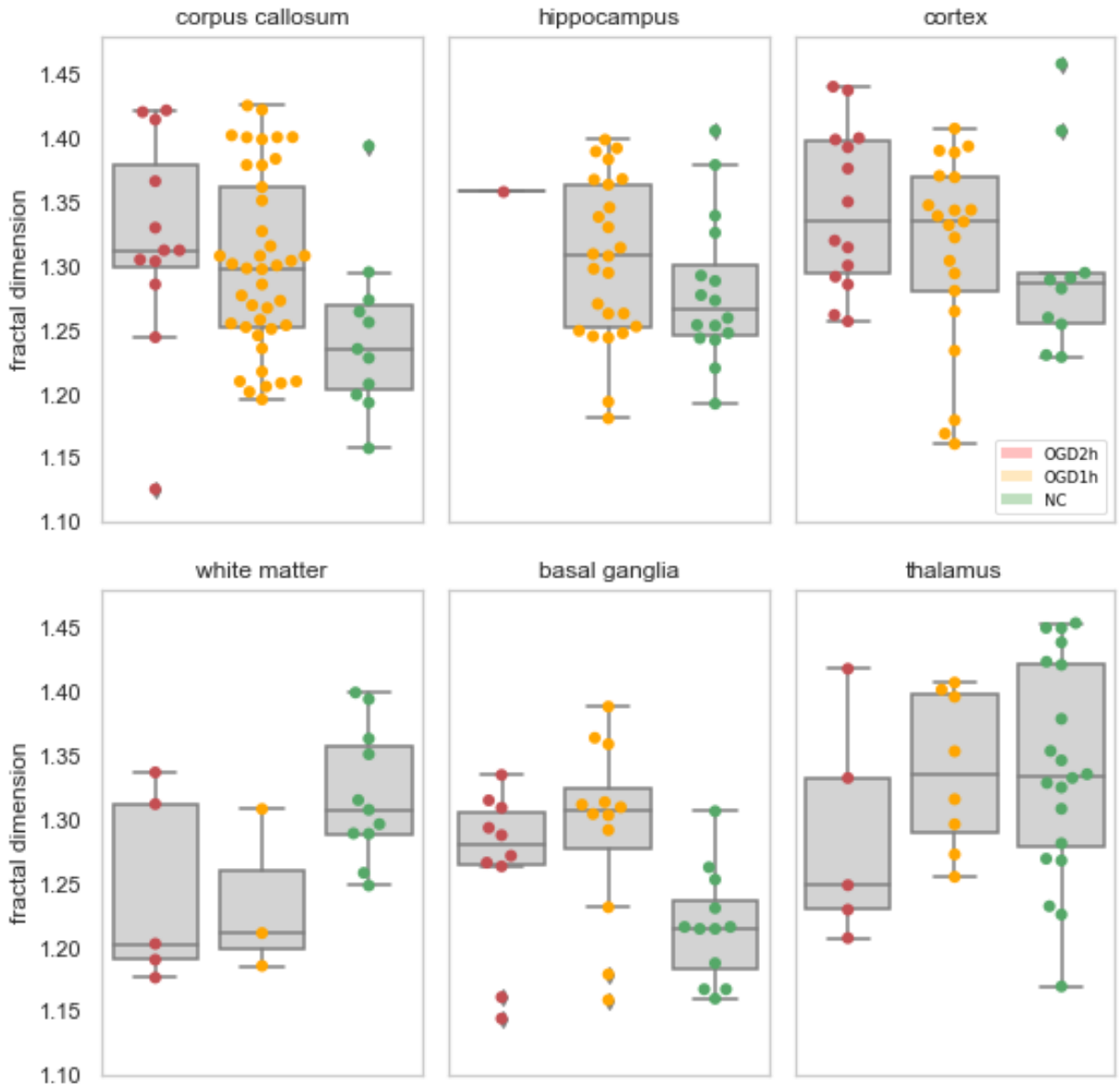


Figure 4-4 Distribution of fractal dimension of different brain regions.

4.4 Discussion

The fractal dimension distribution becomes more aggregated as the OGD exposure time increases. This is possibly due to the higher ratio of microglial cells becoming activated to a more amoeboid morphology, thus showing less morphological heterogeneity. These distributions align with the previous studies that the morphological heterogeneity of microglial cells decreases when the brain is injured or exposed to environmental stimuli and toxins.[9,12] The cell count in the 2hr OGD group is also lower, showing the higher cell death rate induced by longer OGD exposure time.

The difference between regions is also substantial. The heterogeneity of fractal dimension in thalamus is significantly higher than the overall distribution. A narrow distribution in fractal dimensions is observed in white matter. These data show that microglial cells express different morphological heterogeneity in different brain regions in the ferret organotypic brain slice model, findings that align well with previous studies in vivo.[7]

In the skeletonization process, the size of the object greatly affects how well-defined the result is. To prevent the falsely generated objects from biasing the analysis, objects smaller than 300 square pixels are excluded. As a result, 80.4% of skeletonized objects are excluded from this analysis due to their size. A possible solution for this is to increase the magnification at the time of image acquisition. Taking images with higher magnification can capture more cell body details. This can be beneficial for both skeletonization process and downstream fractal analysis, however reduces the overall number of cells because a higher magnification results in a smaller field of view.

4.5 Conclusion

The SCARF package captures different fractal dimension distributions between treatment groups and regions. The distribution of morphological heterogeneity aligns with the previous studies, and supports the use of the SCARF package to skeletonize and extract morphological features correctly in an automated way.

Compared to branch length analysis, the fractal dimension analysis appears to be a better tool in understanding the different microglial cells morphology in different brain region and treatment group. The relationship between branch length and morphological heterogeneity of microglial cells can be hard to define due to the drastic phenotypical difference between different activation states. On the other hand, the fractal dimensions of microglial cells have been proven to be an accurate indicator of their morphological phenotype in the previous studies.[9,15,16] With proper threshold to extract well-defined cells, fractal analysis is a robust way to study the correlation between morphological heterogeneity and brain disease in different brain regions.

CHAPTER 5: Summary and Future Directions

In previous chapters, I have showcased the capability of SCARF package to skeletonize microglial cells properly based on the image condition, and the downstream analysis such as branch feature and fractal dimension analysis that can be achieved with the skeletonized images. There are more image processing packages under development in Nance Lab and can be compiled into a unified work pipeline along with SCARF in the future.

5.1 Current capability and future development of SCARF

SCARF has been proven to binarize and skeletonize the microglial cells microscopy images accurately based on the image condition. The package is built to provide a unified pipeline in Python to flatten the learning curve and increase the image analysis productivity for individual users. However, the downstream analysis using the output skeletonized images are not completely compiled into this work pipeline. In current process to analyze the fractal dimensions, the skeletonized images are saved to local machine before using the FracLac plugin in ImageJ. The result data from FracLac are then import back into Python for further analysis. This back-and-forth process can be avoided if the skeletonized images are analyzed without leaving the Python working environment. For the future development, we can include a fractal analysis function written in Python into SCARF to simplify the work pipeline. This will greatly increase the efficiency of the pipeline, and we can further customize the analysis function for the usage on skeletonized images.

Another future direction is to use SCARF on other brain cells, not just on the microglial cells. SCARF provides a powerful and automated work pipeline to analyze the branch features, yet the various morphological phenotypes of microglial cells make it hard to define a distinct relationship between microglial branch features and their morphological heterogeneity. On the other hand, Astrocytes can be a potential candidate to be processed with the branch feature analysis effectively. Astrocytes shows less drastic changes in cell phenotypes compared to microglial cells in their different activation states, yet the morphological heterogeneity are still highly related to neurodegenerative diseases.[25] This makes astrocytes a potential target to establish a better relationship between branch feature data and the morphological heterogeneity in astrocytes using the SCARF package.

5.2 Integration with existing packages in Nance Lab

There are two packages under development that can work with SCARF in the future image process pipeline. The IfThreshold package, co-developed by former lab member July Zhou and current PhD. candidate Hawley Helmbrecht in Nance lab. The IfThreshold package provides a score system to find

out the best threshold method for a certain image. This package can be introduced into the SCARF workflow to automatically pick the best threshold method for binarizing the images. This can improve the current SCARF binarizing workflow, and the result images can be binarized more precisely.

The other package is Visually Aided Morpho-Phenotyping Image Recognition (VAMPIRE) .[26] The original VAMPIRE package is adapted by Hawley to better accommodate the research in Nance Lab. VAMPIRE can calculate the representative shape modes for cells using machine learning algorithm. VAMPIRE can work parallelly with SCARF to better analyze the cell features from different approaches.

5.3 SCARF in future Nance Lab image process routine

As more image analysis packages are constructed, establishing a universal image process routine compiling all packages can be helpful for researchers in Nance Lab. Fluorescent neuroImage Based Experimental Routines (FIBER), established by Hawley, is an extensive image process routine that integrate all image process packages developed in Nance Lab. SCARF offers functions that can be used in image process pipeline other than microglial cell studies. The end indicator function built in SCARF can mark the end points of object in a binarized image. The skeletonization process can be useful for analyzing cell branches in an automated workflow. With the integration of FIBER, SCARF can provide features to aid future image analysis.

BIBLIOGRAPHY

1. Schitine, C., Nogaroli, L., Costa, M. R., & Hedin-Pereira, C. (2015). Astrocyte heterogeneity in the brain: From development to disease. *Frontiers in Cellular Neuroscience*, 9(March), 1–11.
2. Matias, I., Morgado, J., & Gomes, F. C. A. (2019). Astrocyte Heterogeneity: Impact to Brain Aging and Disease. *Frontiers in Aging Neuroscience*, 11(March), 1–18.
3. Bachstetter, A. D. *et al.* (2015). Disease-related microglia heterogeneity in the hippocampus of Alzheimer's disease, dementia with Lewy bodies, and hippocampal sclerosis of aging. *Acta Neuropathologica Communications*, 3, 32.
4. Ginhoux, F. *et al.* (2013). Origin and differentiation of microglia. *Frontiers in Cellular Neuroscience*, 7(March), 1–14.
5. Ginhoux, F., & Prinz, M. (2015). Origin of microglia: Current concepts and past controversies. *Cold Spring Harbor Perspectives in Biology*, 7(8), 1–16.
6. Mosser, C. A., Baptista, S., Arnoux, I., & Audinat, E. (2017). Microglia in CNS development: Shaping the brain for the future. *Progress in neurobiology*, 149-150, 1–20.
7. Olah, M., Biber, K., Vinet, J., & W.G.M. Boddeke, H. (2011). Microglia Phenotype Diversity. *CNS & Neurological Disorders - Drug Targets*, 10(1), 108–118.
8. Tremblay, M. Ě., Lowery, R. L., & Majewska, A. K. (2010). Microglial interactions with synapses are modulated by visual experience. PLoS Biology, 8(11).
9. Martianova, E., Aniol, V. A., Manolova, A. O., Kvichansky, A. A., & Gulyaeva, N. V. (2019). Activation of microglia associated with lentiviral transduction: A semiautomated method of assessment. *Acta Histochemica*, 121(3), 368–375.
10. Bachiller, S., Jiménez-Ferrer, I., Paulus, A., Yang, Y., Swanberg, M., Deierborg, T., & Boza-Serrano, A. (2018). Microglia in neurological diseases: A road map to brain-disease dependent-inflammatory response. *Frontiers in Cellular Neuroscience*, 12(December), 1–17.
11. Wong, W. T. (2013). Microglial aging in the healthy CNS: Phenotypes, drivers, and rejuvenation. Frontiers in Cellular Neuroscience, 7(MAR), 1–13.
12. Lier, J., Ondruschka, B., Bechmann, I., & Dreßler, J. (2020). Fast microglial activation after severe traumatic brain injuries. *International Journal of Legal Medicine*, 134(6), 2187–2193.
13. Sánchez-Corrales *et al.* (2018). Morphometrics of complex cell shapes: Lobe contribution elliptic Fourier analysis (LOCO-EFA). *Development (Cambridge)*, 145(6), 1–13.

14. Heindl, S., Gesierich, B., Benakis, C., Llovera, G., Duering, M., & Liesz, A. (2018). Automated morphological analysis of microglia after stroke. *Frontiers in Cellular Neuroscience*, 12(April), 1–11.
15. Young, K., & Morrison, H. (2018). Quantifying microglia morphology from photomicrographs of immunohistochemistry prepared tissue using imagej. *Journal of Visualized Experiments*, 2018(136), 1–9.
16. Karperien, A., Ahammer, H., & Jelinek, H. F. (2013). Quantitating the subtleties of microglial morphology with fractal analysis. *Frontiers in Cellular Neuroscience*, 7(JANUARY 2013), 1–34.
17. Harris, C. R. *et al.* (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362.
18. Van Der Walt, S. *et al.* (2014). Scikit-image: Image processing in python. PeerJ, 2014(1), 1–18.
19. Nunez-Iglesias, J. *et al.* (2018). A new Python library to analyse skeleton images confirms malaria parasite remodelling of the red blood cell membrane skeleton. *PeerJ*, 2018(2), 1–10.
20. McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 1(Scipy), 56–61.
21. Karperien, A., FracLac for ImageJ. http://rsb.info.nih.gov/ij/plugins/fraclac/FLHelp/Introduction.htm. 1999-2013.
22. Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7), 671–675.
23. Joseph, A., Liao, R., Zhang, M., Helmbrecht, H., McKenna, M., Filteau, J. R., & Nance, E. (2020). Nanoparticle-microglial interaction in the ischemic brain is modulated by injury duration and treatment. Bioengineering and Translational Medicine, 5(3), 1–14.
24. Morrison, H., Young, K., Qureshi, M., Rowe, R. K., & Lifshitz, J. (2017). Quantitative microglia analyses reveal diverse morphologic responses in the rat cortex after diffuse brain injury. *Scientific Reports*, 7(1), 1–12.
25. Zhou, B., Zuo, Y. X., & Jiang, R. T. (2019). Astrocyte morphology: Diversity, plasticity, and role in neurological diseases. *CNS Neuroscience and Therapeutics,* 25(6), 665–673.
26. Phillip, J. M., Han, K. S., Chen, W. C., Wirtz, D., & Wu, P. H. (2021). A robust unsupervised machine-learning method to quantify the morphological heterogeneity of cells and nuclei. *Nature Protocols*, 16(2), 754–774.

APPENDIX A: Supplementary Figures to Chapter 3

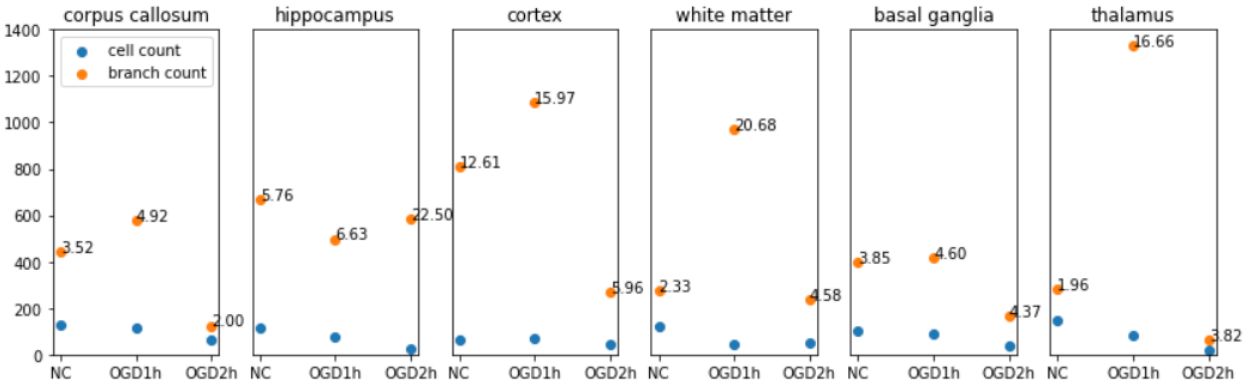


Figure S3-1 Branch count to cell count ratio comparison in different brain regions. The number indicates the branch count to cell count ratio.

Cheng-Hao (Robin) Lin

Tel: (206)591-1700

Email: robin60903@gmail.com

Education

University of Washington Seattle, WA

M.S. in Chemical Engineering: Data Science 2019-2021

National Tsing Hua University Hsinchu City, Taiwan

B.S. in Chemical Engineering 2014-2018

Skills

- **Programming:**
  Python, relative courses:

  Data Science Methods for Clean Energy Research, Software Engineering for Molecular Data Scientists, Data Science Capstone Project

  Java, relative courses:

  Computer Programming II, Data Structures and Algorithms

  SQL database, relative courses:

  Software Engineering for Molecular Data Scientists, Data Science Capstone Project
- **Software:** CADs (SolidWorks, AutoCAD), MATLAB, Aspen
- **Languages**: English, Mandarin

Research and Work Experience

University of Washington Seattle, WA

Graduate Research Assistant 2019-2021

Advisor: Elizabeth Nance, Ph.D.

- Analyzed the relationship between microglial cells morphological heterogeneity and brain disease
- Developed a microglial cells image analysis pipeline adapting from previous work

University of Washington Seattle, WA

DIRECT trainee 2020-2020

- Participated in the DIRECT (Data Intensive Research Enabling Clean Technologies) training program of the Clean Energy Institute at the University of Washington
- Worked in interdisciplinary team projects to solve various scientific problems using data science methods

Project

Thesis project: Skeletonization and fractal analysis of microglial cells in neonatal brain

2020-2021

- Constructed an image analysis package on Python for microscopy cell image morphological studies
- Improved the previously established work pipeline to accommodate background noises for better object recognition
- Performed analysis of cell branch length distribution and fractal dimensions using Python

Capstone project: open battery database on MS Azure 2020-2020

- Created an open source cloud database on Microsoft Azure for academic and industrial battery performance data
- Designed a web page UI for open query to the database

Team project: Flashing light 2020-2020

- Created an image stack processing package for fluorescence microscopy research with teammates
- Constructed a Python function for identifying region of interest based on relative pixel brightness