Visium Spatial Transcriptomics

(*near single-neuron resolution*)

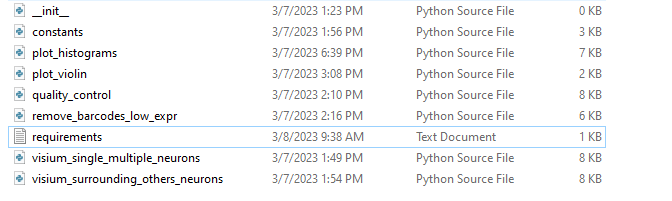
Application User Documentation

This document demonstrates how to analyze Visium spatially-resolved RNA-seq data to obtain ***near*** **single-neuron resolution**. It requires a basic understanding of python and R language(s). The document revolves around how to process the outputs of the 10X spaceranger pipeline and make it downstream-ready. In-order to run through the file(s), **PyCharm** or **VSCode** is a must.

The **Visium** application has approximately six steps for processing data and generating quality controls that will be visualized as **tSNE** and **histogram** charts.

## Installing packages:

There are certain python packages that are required to run the following visium pipeline, these packages are saved in **requirements.txt** file as shown in the below screenshot;



Whenever one develops an application, a new python environment is created with only the packages that are required by the application. Hence, once the development process is completed one can easily capture all the packages by running “**pip freeze > requirements.txt**”. This enables the users to directly install packages in one go, using “pip install” command.

Command to install all the packages: “**pip install -r requirements.txt**”

**Note:** The file path varies in windows and mac/linux environments. It is as shown below;

* **Windows**: **r**"/path/to/file" [Windows treats raw string as \\ in a file path]
* **Mac**/**Linux**: "/path/to/file"

Table of Contents

[Visium Spatial Transcriptomics 2](#_Toc129107406)

[Step 1: Fetching the Single and Multiple Neurons 2](#_Toc129107407)

[Run**:** visium\_single\_multiple\_neurons.py 2](#_Toc129107408)

[Function**:** fetch\_gene\_expr\_selected\_neuronal\_barcodes() 2](#_Toc129107409)

[Params: 2](#_Toc129107410)

[Script: 2](#_Toc129107411)

[Output: 3](#_Toc129107412)

[Step 2: Fetching the Surrounding and Other Neurons 3](#_Toc129107413)

[Run**:** visium\_surrounding\_others\_neurons.py 3](#_Toc129107414)

[Function**:** fetch\_gene\_expr\_surrounding\_and\_other\_barcodes() 3](#_Toc129107415)

[Params: 3](#_Toc129107416)

[Script: 3](#_Toc129107417)

[Output: 3](#_Toc129107418)

[Step 3: Quality Control and tSNE Plots 4](#_Toc129107419)

[Run: quality\_control.py 4](#_Toc129107420)

[Function: quality\_control\_for\_visium\_data() 4](#_Toc129107421)

[Params: 4](#_Toc129107422)

[Script: 4](#_Toc129107423)

[Output: 4](#_Toc129107424)

[Plot: tSNE plot for dataset – 3 of multiple, surrounding and other data 5](#_Toc129107425)

[Step 4: Plotting Histograms (to determine threshold for step 5) 5](#_Toc129107426)

[Run: plot\_histograms.py 5](#_Toc129107427)

[Function: generate\_hist\_plot() 5](#_Toc129107428)

[Params: 6](#_Toc129107429)

[Script: 6](#_Toc129107430)

[Output: 6](#_Toc129107431)

[Plot: Histogram plot for dataset -7 of Single and Multiple neurons; 6](#_Toc129107432)

[Step 5: Removing the Barcodes with low expression and SNAP25=0 6](#_Toc129107433)

[Run: remove\_barcodes\_low\_expr.py 6](#_Toc129107434)

[Function: remove\_barcodes\_with\_low\_expression() 6](#_Toc129107435)

[Params: 7](#_Toc129107436)

[Script: 7](#_Toc129107437)

[Output: 7](#_Toc129107438)

# 

# Visium Spatial Transcriptomics

Flow diagram of the steps being performed in the tool.

Histogram plots

Quality Control and tSNE plots

Generating surrounding and other neuron datasets

Generating single and multiple neuron datasets

Removing barcodes and making the single neuron dataset cluster-ready

Flow-diagram of the Visium spatial transcriptomics automated pipeline

## Step 1: Fetching the Single and Multiple Neurons

Here we are trying to fetch the Single and Multiple/All neurons into separate folder(s) under “**Processed\_files**”. The folder “**Processed\_files**” will be automatically created during the run-time and save all the files generated to folders: **Single** and **Multiple**.

### Run**:** visium\_single\_multiple\_neurons.py

### Function**:** fetch\_gene\_expr\_selected\_neuronal\_barcodes()

The function **fetch\_gene\_expr\_selected\_neuronal\_barcodes()** needs 4 parameters: 3 file-path(s) and a run\_param(run parameter: to check if single or multiple needs to be triggered). Upon running this function, “**Single**” and “**Multiple**” folders are created in the “**Processed\_files**” directory and all the files being processed will be saved to the respective file locations, for reference check the output section.

### Params:

1. final\_matrix\_fp: /path/to/file/**Final\_matrix**/
2. neuronal\_barcodes\_fp: /path/to/file/**Neuronal\_barcodes**/
3. processed\_files: /path/to/file/ [folder where **Processed\_files** need to be created]
4. run\_param: **‘single’/’multiple’**

### Script:

file\_path = "/path/to/file"

# Single  
response\_template = fetch\_gene\_expr\_selected\_neuronal\_barcodes(file\_path + os.sep + "Final\_matrix", file\_path + os.sep + "Neuronal\_barcodes", file\_path, 'single')  
# Multiple  
response\_template = fetch\_gene\_expr\_selected\_neuronal\_barcodes(file\_path + os.sep + "Final\_matrix", file\_path + os.sep + "Neuronal\_barcodes", file\_path, 'multiple')

### Output:

Running: final\_matrix\_drg1.csv -- neuronal\_barcodes\_drg1.csv

Running: final\_matrix\_drg2.csv -- neuronal\_barcodes\_drg2.csv

Running: final\_matrix\_drg3.csv -- neuronal\_barcodes\_drg3.csv

Running: final\_matrix\_drg4.csv -- neuronal\_barcodes\_drg4.csv

Running: final\_matrix\_drg5.csv -- neuronal\_barcodes\_drg5.csv

Running: final\_matrix\_drg6.csv -- neuronal\_barcodes\_drg6.csv

Running: final\_matrix\_drg7.csv -- neuronal\_barcodes\_drg7.csv

Running: final\_matrix\_drg8.csv -- neuronal\_barcodes\_drg8.csv

{'Status': 'Success', 'Response': 'Files generated are In the directories: /path/to/file//Processed\_files/Single & /path/to/file/Processed\_files/Multiple'}

## Step 2: Fetching the Surrounding and Other Neurons

In this step, we are trying to fetch the barcodes that are surrounding neuronal barcodes and other barcodes (barcodes that do not fall in any other category). The data generated in this step will be saved to “**Surrounding**” and “**Other**“ folder(s) in “**Processed\_files**” directory.

### Run**:** visium\_surrounding\_others\_neurons.py

### Function**:** fetch\_gene\_expr\_surrounding\_and\_other\_barcodes()

The function **fetch\_gene\_expr\_surrounding\_and\_other\_barcodes()** in the above-mentioned file needs 3 file-path parameters i.e., Final\_matrix, Neuronal\_barcodes and Processed\_files file-path(s). Upon running this method, two directories “Surrounding” and “Others” are generated, where the files being processed are stored for further analysis in the QC step.

In this step, all the Processed\_files under Multiple, Surrounding and Other are ***merged*** to a single dataframe and stored under “**Seurat\_input**” folder.

### Params:

1. final\_matrix\_fp: /path/to/file/**Final\_matrix**/
2. neuronal\_barcodes\_fp: /path/to/file/**Neuronal\_barcodes**/
3. processed\_files: /path/to/file/ [folder where **Processed\_files** is created in the Step-1]

### Script:

file\_path = "/path/to/file"  
fetch\_gene\_expr\_surrounding\_and\_other\_barcodes(file\_path + os.sep + "Final\_matrix",  
 file\_path + os.sep + "Neuronal\_barcodes",  
 file\_path + os.sep + "Processed\_files")

### Output:

Running: final\_matrix\_drg1.csv -- neuronal\_barcodes\_drg1.csv

Running: final\_matrix\_drg2.csv -- neuronal\_barcodes\_drg2.csv

Running: final\_matrix\_drg3.csv -- neuronal\_barcodes\_drg3.csv

Running: final\_matrix\_drg4.csv -- neuronal\_barcodes\_drg4.csv

Running: final\_matrix\_drg5.csv -- neuronal\_barcodes\_drg5.csv

Running: final\_matrix\_drg6.csv -- neuronal\_barcodes\_drg6.csv

Running: final\_matrix\_drg7.csv -- neuronal\_barcodes\_drg7.csv

Running: final\_matrix\_drg8.csv -- neuronal\_barcodes\_drg8.csv

{'Status': 'Success', 'Response': 'Files created in /path/to/file//Processed\_files/Other & /path/to/file//Processed\_files/Surrounding'}

## Step 3: Quality Control and tSNE Plots

In this step, we are performing the PCA and generating the TSNE plots. These plots generated are saved to “Plots” folder in the given file-path, however the file-path should contain all the folders generated from the previous steps: **Multiple**, **Surrounding** and **Other**.

### Run: quality\_control.py

### Function: quality\_control\_for\_visium\_data()

The function quality\_control\_for\_visium\_data() applies PCA and generated tSNE plots to a “Plots” directory where the plots are generated are unique and have an extra 10 char-length string along with the file names as shown below;

A screenshot of a computer

Description automatically generated with medium confidence

This method expects one parameter as an input, which is the file-path for where the folders Final\_matrix, Neuronal\_barcodes and **Processed\_files** are present. Rest of the file-paths are autogenerated by the function.

### Params:

1. file\_path: /path/to/file/ [path there **Final\_matrix** and **Neuronal\_barcodes** directories are present]
2. plot\_type: 'pdf'/'png' [format that the generated tSNE plot needs to be stored]

### Script:

file\_path = "/path/to/file"

quality\_control\_for\_visium\_data(file\_path)

### Output:

Running for all\_neurons\_id1.csv, all\_surrounding\_neurons\_id1.csv and other\_id1.csv

Running for all\_neurons\_id2.csv, all\_surrounding\_neurons\_id2.csv and other\_id2.csv

Running for all\_neurons\_id3.csv, all\_surrounding\_neurons\_id3.csv and other\_id3.csv

Running for all\_neurons\_id4.csv, all\_surrounding\_neurons\_id4.csv and other\_id4.csv

Running for all\_neurons\_id5.csv, all\_surrounding\_neurons\_id5.csv and other\_id5.csv

Running for all\_neurons\_id6.csv, all\_surrounding\_neurons\_id6.csv and other\_id6.csv

Running for all\_neurons\_id7.csv, all\_surrounding\_neurons\_id7.csv and other\_id7.csv

Running for all\_neurons\_id8.csv, all\_surrounding\_neurons\_id8.csv and other\_id8.csv

### Plot: tSNE plot for dataset – 3 of multiple, surrounding and other data

Chart, scatter chart

Description automatically generated

## Step 4: Plotting Histograms

### Run: plot\_histograms.py

Here we are plotting histograms for both Single and Multiple neurons and it is as explained in the below function **generate\_hist\_plot()**.

### Function: generate\_hist\_plot()

generate\_hist\_plot() will generate histograms and this method expects 3 parameters: file-path where the directories “**Single**” and “**Multiple**” are located(generally in “**Processed**\_files”), **upper\_threshold** and **lower\_threshold**. The threshold values are used to filter the data on both end; ceil and floor.

**Note:** lower\_threshold is by default set to 0, if no value is provided as we are having counts in this particular step and there would be no chance of encountering any negative values.

### Params:

1. file\_path: /path/to/file/ [path there **Final\_matrix** and **Neuronal\_barcodes** directories are present]
2. upper\_threshold: 5000 [upper\_threshold for the histogram]
3. lower\_threshold: By default, 0 however, it can be set of any value less than the upper\_threshold.

### Script:

Generate\_hist\_plot(“/Users/Nikhil/Downloads/Visium\_near\_single\_neuron\_workflow’”, 5000, 2000)

### Output:

{'Status': 'Success', 'Response': "The plot is saved here: ['/path/to/file//Processed\_files/Plots/hist\_plot\_1\_lFaIUXmKup.png', '/path/to/file//Processed\_files/Plots/hist\_plot\_2\_58x2ivTpJu.png', '/path/to/file//Processed\_files/Plots/hist\_plot\_3\_tGVB0VvcQL.png', '/path/to/file//Processed\_files/Plots/hist\_plot\_4\_S47qMYPLKl.png', '/path/to/file//Processed\_files/Plots/hist\_plot\_5\_sUb0LJkKX6.png', '/path/to/file//Processed\_files/Plots/hist\_plot\_6\_FiLu2qneyc.png', '/path/to/file//Processed\_files/Plots/hist\_plot\_7\_iRLdvnlTtx.png', '/path/to/file//Processed\_files/Plots/hist\_plot\_8\_O7J3ZH9cbF.png']"}

### Plot: Histogram plot for dataset -7 of Single and Multiple neurons;

Chart, histogram

Description automatically generated

## Step 5: Removing the Barcodes with low expression

Removing barcodes with low expression is basically filtering out the genes that are expressed low in the Single neuron dataset. Also, in this step we clean(**clean\_unique\_and\_gene\_names()** the data by removing all the duplicates and replacing few gene names.

### Run: remove\_barcodes\_low\_expr.py

### Function: remove\_barcodes\_with\_low\_expression()

This function expects 4 parameters; **file-path**, **threshold**, **filters** and **filter\_threshold**. File-path here is where the **Final\_matrix** and **Processed\_files** are saved, from which **/Processed\_files/Single** is used in this particular step as we are trying to make the Single datasets downstream analysis ready. Based on the parameters available the data is processed, cleaned for duplicated and is stored in the **Cluster** folder in the **file-path/Processed\_files** location.

In this step, all the Processed\_files under Cluster are ***merged*** to a single dataframe and stored under “**Seurat\_input**” folder.

### Params:

1. file\_path: /path/to/file/ [path there **Final\_matrix** and **Neuronal\_barcodes** directories are present]
2. threshold: By default, None, however can take numeric values Ex.1000, threshold Is used to filter the single neuron dataframe.
3. filter\_threshold: By default, 0 however filter\_threshold is only applied when filters are not None. [Ex. 1, 2 or numeric value]
4. filters: By default, None however filters can be as shown below;
   * 'SNAP25'
   * 'SNAP25,RELL1,..,..,..,'

**Note:** By default, the **threshold** and **filters** are **None** and **filter\_threshold** is **0**.

Params:

1. file\_path = “/path/to/file”

### Script:

file\_path = "/path/to/file"

remove\_barcodes\_with\_low\_expression(file\_path=file\_path, threshold=1000, filters='SNAP25', filter\_threshold=1)

### Output:

Running: single\_neurons\_id1.csv

Running: single\_neurons\_id2.csv

Running: single\_neurons\_id3.csv

Running: single\_neurons\_id4.csv

Running: single\_neurons\_id5.csv

Running: single\_neurons\_id6.csv

Running: single\_neurons\_id7.csv

Running: single\_neurons\_id8.csv

{'Status': 'Success', 'Response': 'Barcodes with low expression are removed successfully.'}

Notes: (see R\_instructions file)

1. Use merged neuronal files, surrounding and other barcodes to generate violin plots in R.
2. Use the output files of step 5 as input for Seurat’s integration and clustering workflow in R.

The End