

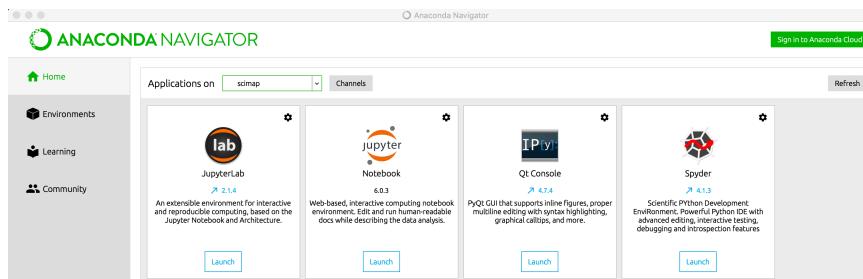
# Prerequisite

## Install Anaconda

<https://www.anaconda.com/products/individual>

## Install Jupyter notebook from the anaconda navigator.

Make sure you have Jupyter notebook or at least spyder installed. We would need to spin out napari instances which requires an interactive viewer. It is possible to work with any IDE I guess but I have only tested Jupyter notebooks and spyder and will be able to help you with troubleshooting if you run into errors.



## Create a new conda environment with python 3.7

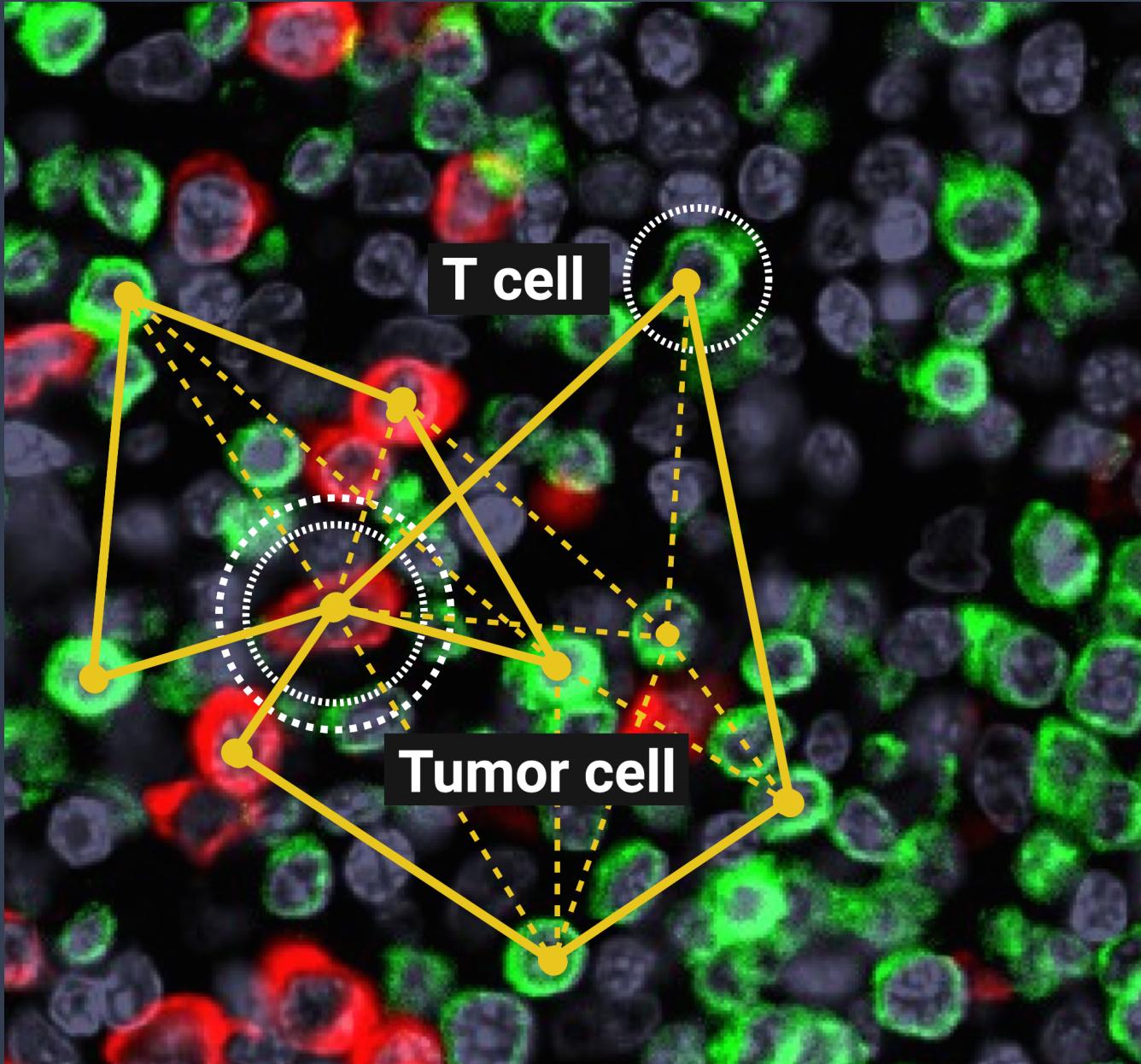
<https://docs.anaconda.com/anaconda/navigator/getting-started/#:~:text=Navigator%20uses%20conda%20to%20create,then%20click%20the%20Create%20button.>

## Activate the environment and within it install the following python packages

*pip install scimap*

*pip install scanpy*

*pip install leidenalg*



# SCIMAP

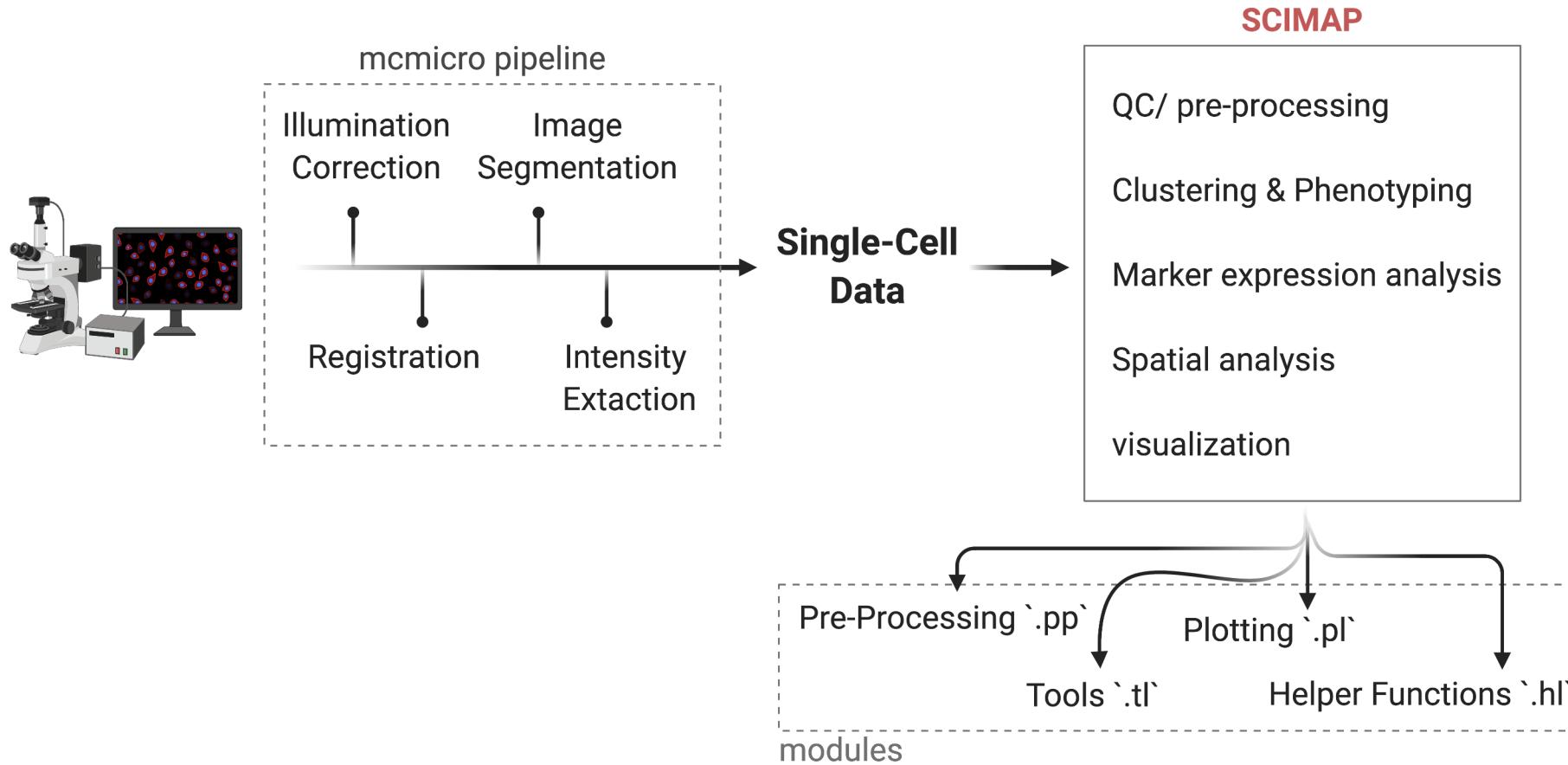
## Single-Cell Image Analysis Package

Ajit Johnson Nirmal

<https://github.com/ajitjohnson/scimap>

<https://scimap-doc.readthedocs.io/>

# Background



# Single-Cell Data

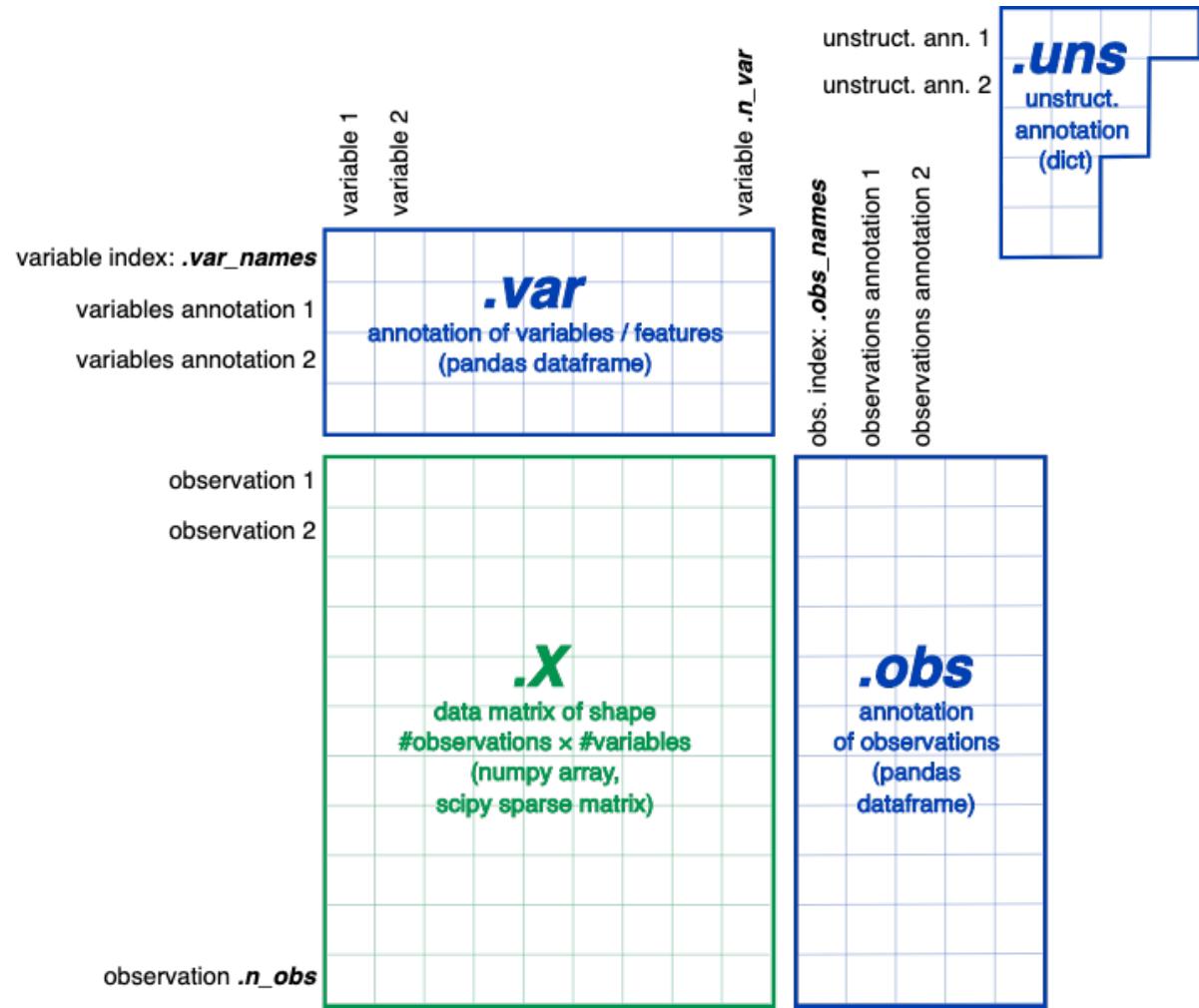
Count's Data

CellID	DNA1	CD25	CD2	CD10	DNA3	CD163
1	19114.07	914.47	543.76	1759.28	6909.67	554.22
2	19244.78	1060.69	685.41	2136.28	6857.63	618.68
3	15447.87	991.50	754.65	2122.12	5815.90	588.42
4	11846.93	919.16	580.85	2003.80	4664.97	572.86
5	10419.45	926.58	674.55	2069.19	4480.33	579.51
6	19156.65	1060.11	829.84	2158.96	7052.54	618.80
7	19940.20	1028.07	675.84	2091.51	7004.79	595.04
8	19095.52	1063.01	688.37	2537.75	7148.63	606.37
9	16921.73	1030.07	677.80	2036.20	5856.99	615.51
10	28127.88	1137.15	576.58	1953.47	10876.20	641.03
11	15183.16	959.78	701.74	2061.97	5614.13	571.39
12	17283.85	998.11	946.83	2125.95	6433.99	601.51
13	20210.02	1074.98	753.31	2108.33	7317.61	621.30
14	18553.24	1020.70	771.97	2243.26	6953.07	619.37

Meta Data

X_position	Y_position	Area	MajorAxisLen	MinorAxisLen
160.841121	669.925234	107	14.818204	9.71569875
160.324074	686.453704	108	14.8142724	9.45204597
165.448819	577.716535	127	18.3658425	9.69364271
164.290909	557.678788	165	16.5788006	12.9504534
162.505155	569.443299	97	13.63834	9.31939017
165.756579	630.894737	152	19.593052	10.1637963
163.614173	709.614173	127	13.7107977	12.0055121
166.906475	611.94964	139	20.7149849	9.67104005
169.187755	513.779592	245	19.1418644	17.5955342
171.846154	661.882784	273	20.6359746	18.0545916
168.375	592.511364	88	13.2489933	8.49777085
170.049505	619.891089	101	13.2046997	9.87751609
168.96875	640.708333	96	13.4906325	9.099053
171.485714	726.971429	140	15.4395819	11.6080077

# AnnData Object



- Similar to ExpressionSet in R
- Allows for tools developed by other developers to be directly applied to the dataset (e.g. scanpy)
- Think of this is as a excel sheet, with additional tabs storing unstructured data.

# Counts table stored in:

`adata.X`

# Meta data stored in:

`adata.obs`

# Subset data

```
subset = adata [ adata.obs["column_name"] == "batch1", : ]
```

## Getting Started

Step 1: Create an AnnData object with your data

```
# Import Libraries
import sys
import os
import anndata as ad
import pandas as pd

# Set Working Directory
os.chdir ("/path/to/your/data/")

# Import your data
data = pd.read_csv ('something.csv') # counts matrix
meta = pd.read_csv ('something.csv') # Meta data like x and y coordinates

# combine them to the AnnData object
adata = ad.AnnData (data)
adata.obs = meta
```

## Helper Function to create AnnData object (mcmicro output)

```
# Import
import scimap as sm

# Function
mcmicro_to_scimap (image_path, # path to the CSV file as a list
                    remove_dna=True, # Remove the DNA channels
                    remove_string_from_name=None, # Remove unwanted string from column names e.g image_72_CD45
                    log=True, # Log the data (log1p transformation will be applied)
                    drop_markers=None, # Drop unwanted markers like background
                    random_sample=None, # Randomly sub-sample the data with the desired number of cells
                    CellId='CellID', # Column Name of the cell ID
                    split='X_position', # Split the counts table and meta data file
                    custom_imageid=None, # Pass a user defined Image ID
                    min_cells=None # If these many cells are not in the image, the image will be dropped)

# Apply Function
image_path = ['/path/to/csv/file/']
adata = sm.pp.mcmicro_to_scimap (image_path)
```

## Cell-Phenotyping using Scimap

**The probability distribution-based algorithm involves three steps:**

1. Identify the gates using `sm.pl.gate_finder`
2. Rescale the data based on the identified gates using `sm.pp.rescale`
3. Run the phenotyping algorithm on the rescaled data using `sm.tl.phenotype`

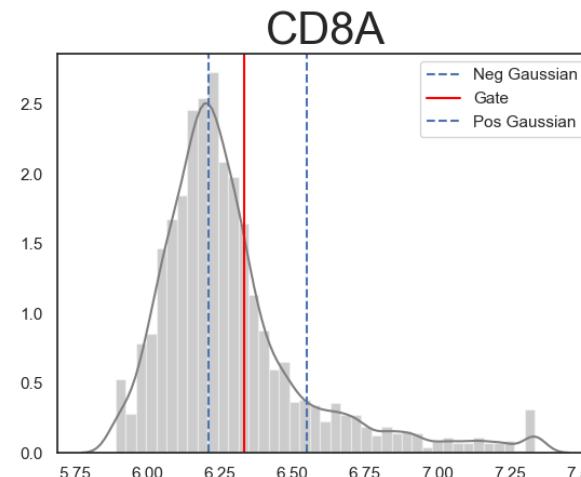
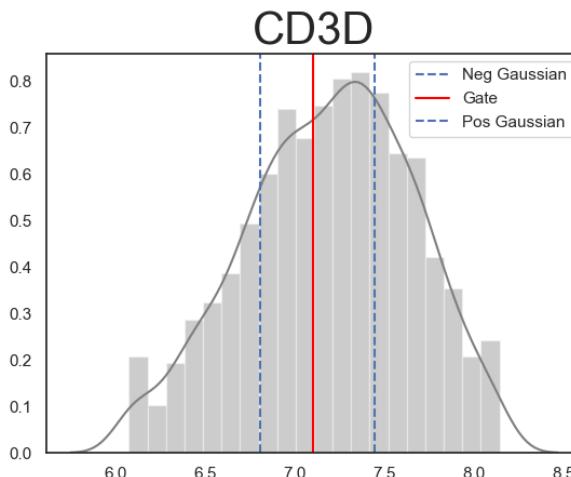
# Function to identify the manual gates

```
scimap.pl.gate_finder (
    image_path, # path to image
    adata, # AnnData object
    marker_of_interest, # Marker you would like to identify the gate for
    from_gate = 6, to_gate = 8, increment = 0.1, # Gating works on the log expression of the
    data
    markers = None, # Any additional markers you would like to add to the image
    channel_names = 'default', # By default it looks into the adata.uns['markers']
    x='X_centroid', y='Y_centroid', # x and y coordinates
    point_size=10,
    image_id=None, # imageid, if you have multiple images in the dataset
    seg_mask=None)
```

## Step 2: Phenotyping

```
# Rescale the data based on the identified gates
```

```
scimap.pp.rescale (  
    adata, # AnnData Object  
    gate=None, # Supply the manual gats  
    failed_markers=None, # If you think some markers did not work, you can pass it here  
    method='all', # You can scale each image independently or all images together  
    save_fig=False) # Save the gaussian fits
```



## Step 3: Phenotype Workflow

		ASMA	CD163	CD206	CD68	CD20	CD21	CD3D	CD45	CD56	CD8A	FOXP3	CD11B	CD11C	CD15	CD4	PD1	HLADR	CD25
all	Other Immune cells								pos										
all	ASMA+ cells	pos																	
Other Immune cells	T cells							pos											
Other Immune cells	B cells						pos												
Other Immune cells	Myeloid Lineage		anypos	anypos	anypos							anypos	anypos						
Other Immune cells	NK cells									pos									
Other Immune cells	Granulocytes															pos			
Myeloid Lineage	T cells							pos											
Myeloid Lineage	B cells						pos												
Myeloid Lineage	NK cells									pos									
Myeloid Lineage	Granulocytes														pos				
T cells	CD4 T cells															pos			
T cells	CD8 T cells									pos									
CD4 T cells	Regulatory T cells											pos							
CD4 T cells	Follicular Helper T cells															pos			
CD8 T cells	PD1+ T cells															pos			
Myeloid Lineage	CD68+ Macrophages				pos														
Myeloid Lineage	M2 Macrophages		anypos	anypos															
Myeloid Lineage	Myeloid Dendritic cells															pos			
Myeloid Lineage	Follicular Dendritic cells						pos												
CD68+ Macrophages	M1 Macrophages															pos			
Myeloid Dendritic cells	CD25+ Dendritic cells																pos		

\* pos, neg, anypos, anyneg, allpos, allneg

## Step 3: Run the phenotyping algorithm

```
scimap.tl.phenotype_cells ( adata,  
                            phenotype, # The phenotype workflow CSV  
                            label="phenotype", # Column name to store the results  
                            unique_id='imageid', # Only necessary if you use one of the next two parameters  
                            pheno_threshold_percent=None, # remove small populations of cells  
                            pheno_threshold_abs=None)
```

Once the cells are phenotyped, verify them by overlaying the phenotypes back on to the image by using the following function.

```
scimap.pl.image_viewer ( image_path, adata,  
                        overlay=None, overlay_category=None, # column name of the column to overlay  
                        markers=None, # specific markers in the image  
                        channel_names='default',  
                        x='X_centroid', y='Y_centroid',  
                        point_size=10, point_color=None,  
                        image_id=None, seg_mask=None)
```