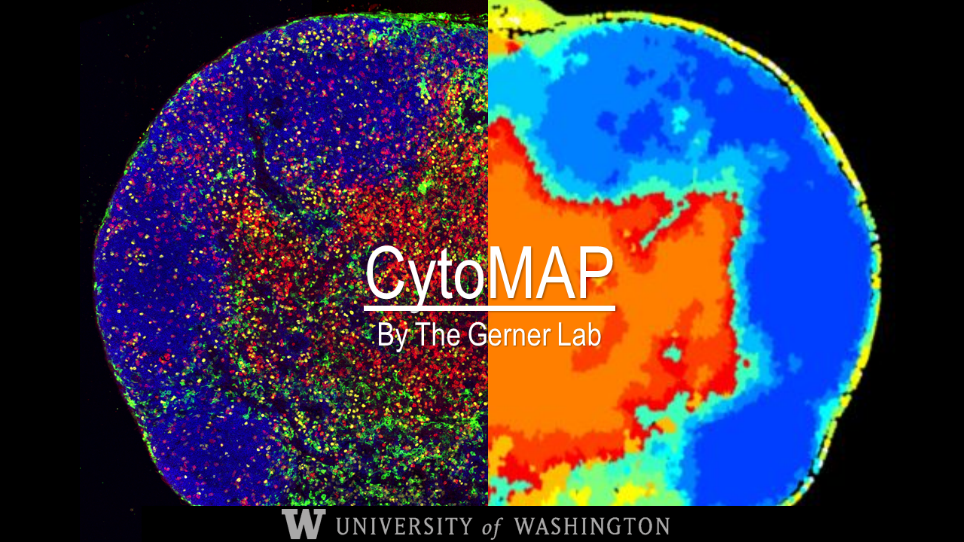
CytoMAP User Manual

**Authors**

*Dr. Caleb Stoltzfus, Jakub Filipek*

**Contributors**

*Dr. Michael Gerner, Brandy Olin, Joseph Leal, Miranda Lyons-Cohen, Jessica Huang, Dr. Ben Gern, Dr. Michelle Messmer*



**

 Gerner Lab, University of Washington, School of Medicine,

Department of Immunology

SLU E Building Box 358059

750 Republican Street, 4th floor

Seattle, WA 98109

# Installing CytoMAP

Download and run the CytoMAP\_V0.exe file. Follow the prompts to install a Matlab 2018 runtime and CytoMAP.

# Importing Data

This section discusses the various data types and formats that can currently be imported into CytoMAP.

## Importing FlowJO Populations

### Exporting From FlowJo

To export populations from FlowJo to analyze in CytoMAP, select the populations of interest, right-click on the selected populations and select; “Concatenate, export populations”.

For output choose the **Scale** values. Strangely, “channels” correspond to values which have been scaled by FlowJo, and “scale” corresponds to the raw data.

Each sample must be exported separately. Selecting a parent population will not automatically export all of the children populations. However, you can select multiple populations (‘ctrl’ + click) from the same sample and FlowJo will export them each to their individual .csv file.

Currently, CytoMAP only sees the immediate cell phenotype name. Thus, to avoid errors when importing cells into CytoMAP, name each phenotype to with a unique name, avoiding repeats like those shown in Figure 1.

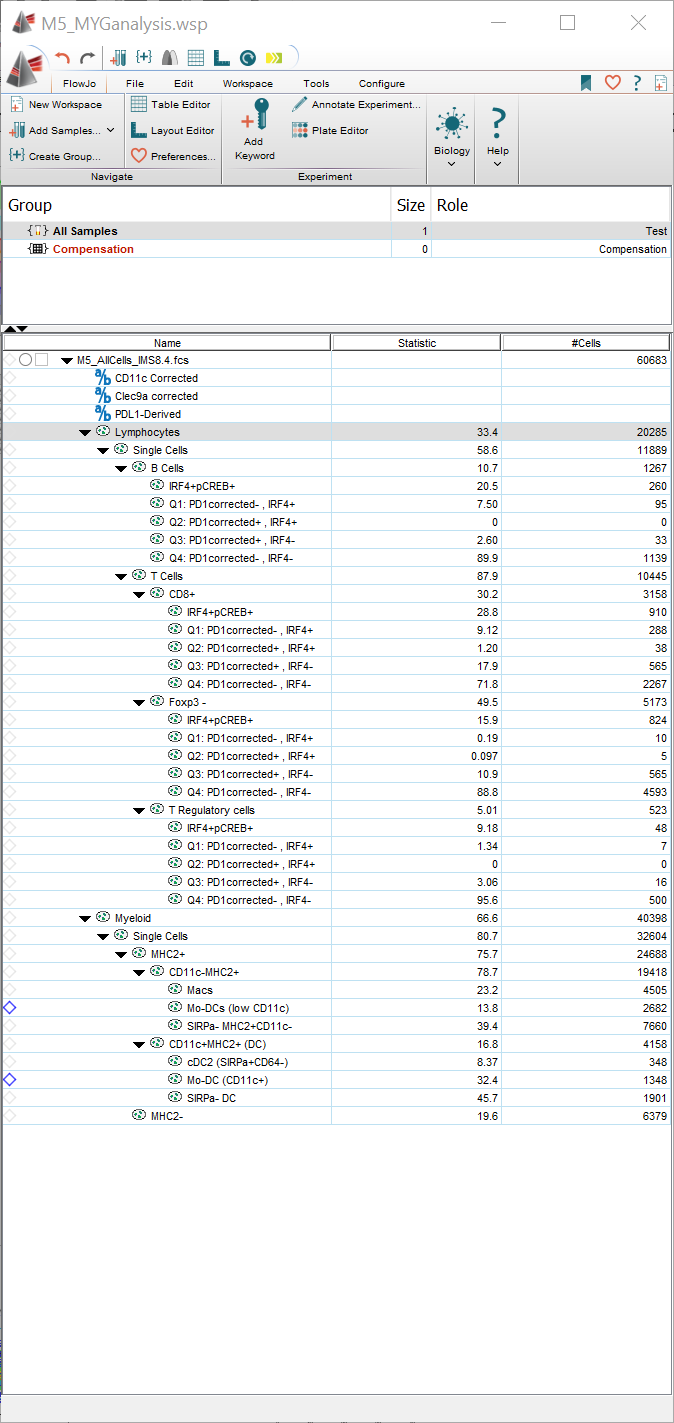


Figure 1| FlowJo populations to be exported should each have unique names

### Importing .csv from FlowJO

To import cell populations for a single sample into CytoMAP, select;

File > “Load .csv or .mat”

Next, select all of the phenotypes you wish to import. There should be a separate .csv file, exported from FlowJo, for each phenotype you are importing. CytoMAP will look at all of the file names, and use the string that is common to all files as the sample name. The unique string for each filename will be used as the phenotype name\*.

Once you have selected the files you wish to import, the window shown in Figure 2 will open. In this window, you can rename your samples, phenotypes, or channels.

**For CytoMAP to work smoothly, all of the phenotypes imported should have at least one cell, and all of the phenotypes should have the same channels, in the same order. Additionally, there should be at least two spatial channels labeled as “X” and “Y”. There can also be a “Z” dimension, but this is not required for analysis.**

For convenience, using the “String to replace” and “Replace with” fields the user can rename/remove common things in the cell names or channel names.

To do this type the string you want to replace in the box below the “string to be replaced” label. Next, in the field below the “Replace with” label, type the string you want to replace this string with, or leave this field blank if you want to remove the string. Finally, click the “Rename” button in the bottom left of the file import window.

When finished renaming channels etc. click the load button on the bottom left of the window to import your selected cells. CytoMAP will warn you if any sample names/channel names/ or Phenotype names still have symbols, or can otherwise not be imported as variable names.

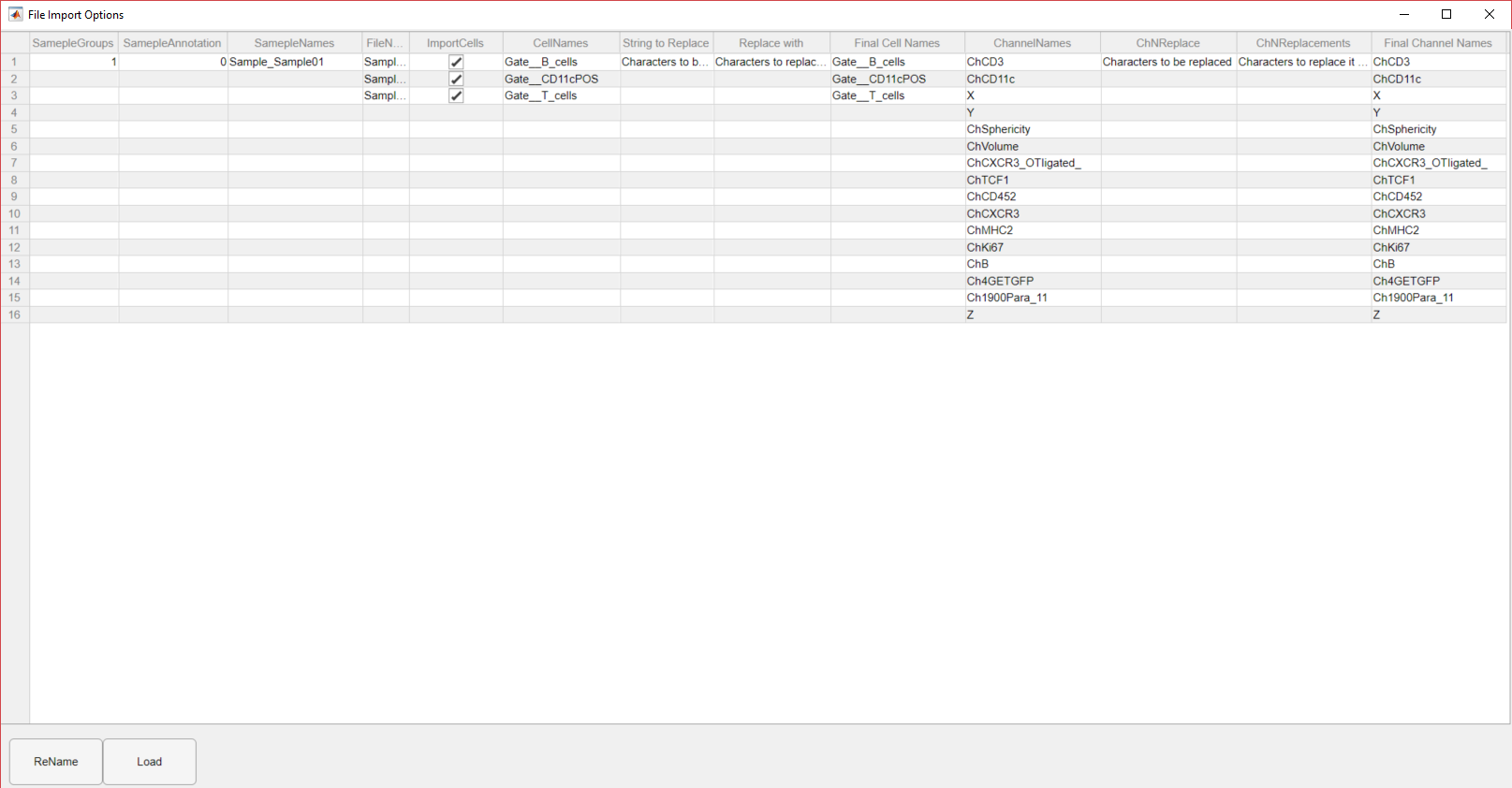


Figure 2| Load samples window

\*Due to limitations within MALAB all symbols and spaces will be replaced/removed. Additionally, since MATLAB variable names cannot start with a number, strings are added to the beginning of all of the sample/phenotype/channel names.

## Importing Multiple Samples

To simultaneously import multiple samples into CytoMAP, first save all of the individual phenotypes for each sample into a unique folder containing only that sample’s .csv files. Next select;

File > Import Multiple Samples

This will bring up a dialog box asking the user to select the folders where the .csv files are for each sample. Select each sample’s folder and click “Open”. This will scan each folder and open all of the .csv files contained in those folders, again using the file names to populate the sample and phenotype fields in the resulting import dialog box. Rename any sample names or phenotype names as desired. When finished renaming things, click the load button in the bottom left of the window.

Note:

This multi-sample import assumes all samples have the same number and type of cells, as well as the same channels. If there are empty .csv files representing empty cell gates exported from FlowJo, or if some samples have additional channels, CytoMAP will run into errors.

The sample group and sample annotation currently have no functionality other than annotation.

## Importing .wsp files to CytoMAP

In order to ease user workload when working with both CytoMAP and FlowJO, the user can import a .wsp file directly. This is a preferred method when working with data from one sample but organized into multiple subfolders, which makes it cumbersome to load with csvs.

This option can be found under:

1. **File > “Import data from wsp”**
2. Then select your desired .wsp file. This will load all of the data and the calculated parameters.

## Importing .mat CytoMAP workspace files

When working in CytoMAP you can save your current workspace to use later. This saves any defined distances, neighborhoods, regions, and other calculations to a single .mat file. To pick up where you left off select:

1. File > “Load .csv or .mat”
2. Then select your desired .mat file. This will load all of the data and the calculated parameters.

# Neighborhoods

Neighborhood analysis finds the local composition of cells within a circular (spherical in 3D samples) area in the tissue. The number of cells and the MFI of each channel averaged over all cells in each neighborhood, are calculated. This information can then be used to find local cellular densities, or look at patterns, tissue architecture, and cell-cell associations across your sample.

## Raster Scanned Neighborhoods

The raster-scanned neighborhoods button creates a table of neighborhoods, where the position of each neighborhood is evenly distributed across the tissue in a grid pattern. The distance between neighborhoods is *r*/2, where *r* is the user-defined radius of the neighborhoods (field 2 in Figure 3).

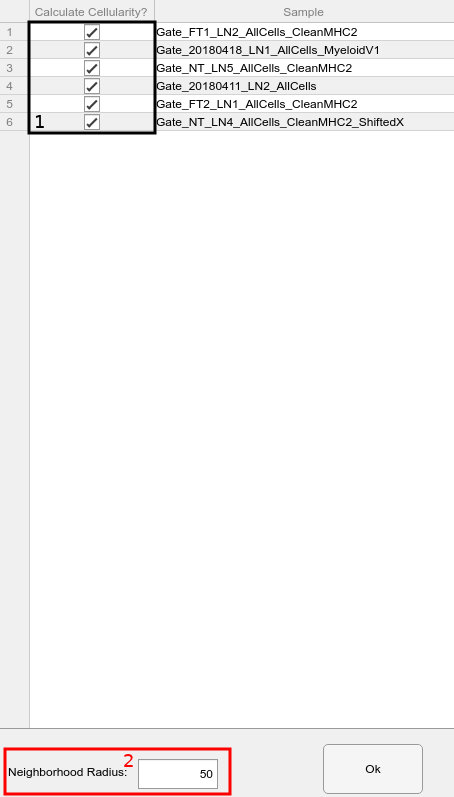
To Calculate Raster Scanned Neighborhoods for your samples, click the “Raster Scan Neighborhoods” button in the main user interface window.

In the resulting window, select which samples you wish to calculate neighborhoods for (field 1 in Figure 3).

On the bottom of this interface define the radius of the neighborhoods you wish to calculate, 50 is the default. The units of *r* are whatever units your imported data are in. Assuming your microscope software exports spatial data in micrometers, this definition of *r* will also be in micrometers.

Note:

If you wish to define smaller features you will need to use a smaller neighborhood radius. However, the smaller the neighborhoods the longer this, and resulting calculations will take. Additionally, if the radius of the neighborhood is smaller than the distance between cells, you will effectively be doing analysis on individual cells, and no longer considering local neighborhoods of cells.

Figure 3| User interface for calculating neighborhoods

## Cell Centered

For Cell Centered Neighborhoods a Table is created where the position of each neighborhood is centered on a user selected cell type. This is useful for looking at the local microenvironment of a specific cell type.

The interface, in this case, is similar to one for Raster Scanned Neighborhoods, but the option of selecting only cells from certain samples is added.

# Clustering; Classifying Neighborhoods into Regions

Once you have defined neighborhoods, you can use various clustering algorithms to cluster those neighborhoods into groups, or “regions”. To do this click the “Classify Neighborhoods into Regions” button on the main CytoMAP window. This will open the interface shown in Figure 4. There are many options when clustering neighborhoods, including how you prepare your data, which algorithm to use, and how many regions to cluster your neighborhoods into. They are all described below.

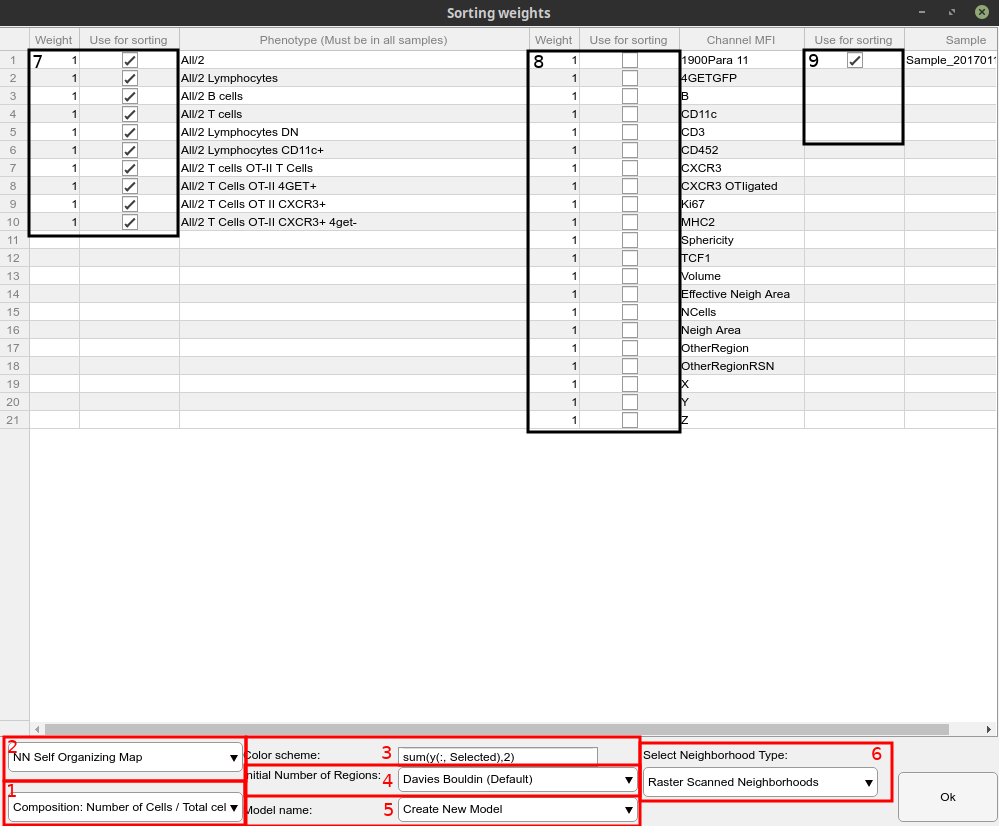


Figure 4| Classify Neighborhoods into Regions interface

## Data Preparation

By scaling your data in different ways, you can highlight different types of regions (Field 7 in Fig. 4), and find interesting patterns across your tissue. In the bottom left of the interface window there is a drop-down menu (Field 1 on Fig. 4) containing the following data preparation options;

### Composition (default)

* This option takes the number of cells in each neighborhood and divides by the total number of cells in that neighborhood, yielding the local composition of cells. This helps deal with edge effects from when the neighborhood is only partially on the tissue.

*Global Composition*

* This option divides the number of cells in each neighborhood by the maximum number of cells in all neighborhoods across the tissue. This normalizes each cell type to its own maximum number, which should make all of the cell types closer to equal weights for clustering.

### Cellularity Number

* This option feeds the un-modified number of cells in each neighborhood multiplied by the user-selected weights (default = 1; field 7 on Fig. 4) into the chosen clustering algorithm.

*Cell Density*

* This divides the number of cells by the area/volume of the neighborhoods. Most clustering algorithms don’t change if you multiply your data by a constant, so this yields the same results as using just cellularity number.

### Binary

* TODO

*Standardize*

* TODO

### Corrected Density

* This attempts to estimate the volume of the neighborhood taken up by cells and divides the number of cells by that estimated volume. The volume estimation runs into issues when there are only a few cells in a neighborhood.

### Composition + Density

* This option takes the cell numbers, divided by the total cells for each neighborhood, and adds a new column containing the total cell numbers for each neighborhood so that the total cellular density is also considered by the clustering algorithm. Currently, there isn’t an option to weight this additional column differently.

## Clustering Algorithms (K = number of clusters)

Different clustering algorithms use different methods of deciding how to group the neighborhood data into different regions. Each algorithm makes different assumptions about how your data was prepared and what the natural distributions within your data are. If the tissue has an unknown structure it is best to try different algorithms and explore the results to find which patterns are the most relevant/useful. Different algorithms can be selected via the drop-down menu in the bottom left of the user interface (Field 2 on Fig. 4).

### NN Self-Organizing Map (Default) (Jiang et al., 2010; Vesanto et al., 2000)

* This algorithm starts with K “neurons” positioned throughout the data. It then iteratively moves the position of the neurons closer to the data to match the landscape of the data. The neighborhoods are then clustered by finding the closest neuron for each neighborhood. This yields very similar results to K-means clustering for small numbers of cell types.

### K-Means

* This algorithm attempts to find K unique clusters based on the multidimensional means of the data, attempting to minimize the within-cluster sum of squares. It is prone to misrepresenting more advanced data (such as overlapping clusters, and clusters with much different size)

### Gaussian Distribution Model

* This algorithm assumes the data has a normal distribution and fits the data with K Gaussian functions, assigning the neighborhoods to clusters based on which cluster’s distribution they fall under. In case of small samples, with clear differences between clusters it tends to yield similar results to K-means.

## Initial Number of Regions, K

For each clustering algorithm, the user must specify how many groups/regions to break the data into. This problem can be very difficult since often we do not know how many different types of neighborhoods are present in the tissue. Furthermore, smoothing effects from the raster scan can blend two adjacent regions together making a non-biological intermediate region. There are multiple algorithms that attempt to automatically find the number of groups present within your data. Each of these algorithms uses different metrics to define how many regions the chosen clustering algorithm should use. The desired number of regions can be set using the options in the bottom middle of the interaction window (Field 4 on Figure 4). (Descriptions of algorithms copied from MATLAB)

### Calinski-Harabasz

The Calinski-Harabasz criterion is sometimes called the variance ratio criterion (VRC). The Calinski-Harabasz index is defined as

,

where *SS*B is the overall between-cluster variance, *SS*W is the overall within-cluster variance, *k* is the number of clusters, and *N* is the number of observations. The overall between-cluster variance *SS*B is defined as

,

where *k* is the number of clusters, *n*i is the number of observations in cluster *i*, *mi* is the centroid of cluster *i*, *m* is the overall mean of the sample data, and  is the *L2* norm (Euclidean distance) between the two vectors. The overall within-cluster variance *SS*W is defined as

,

where *k* is the number of clusters, *x* is a data point, *ci* is the *i*th cluster, *mi* is the centroid of cluster *i*, and  is the *L2* norm (Euclidean distance) between the two vectors. Well-defined clusters have a large between-cluster variance (*SS*B) and a small within-cluster variance (*SS*W). The larger the VRCk ratio, the better the data partition. To determine the optimal number of clusters, maximize VRCk with respect to *k*. The **optimal number of clusters** is the solution with the **highest Calinski-Harabasz index value**. The Calinski-Harabasz criterion is best suited for *k*-means clustering solutions with squared Euclidean distances.

* Calinski, T., and J. Harabasz. “A dendrite method for cluster analysis.” *Communications in Statistics*. Vol. 3, No. 1, 1974, pp. 1–27.

### Davies Bouldin

The Davies-Bouldin criterion is based on a ratio of within-cluster and between-cluster distances. The Davies-Bouldin index is defined as

,

where *D*i,j is the within-to-between cluster distance ratio for the *i*th and *j*th clusters. In mathematical terms,

.

is the average distance between each point in the *i*th cluster and the centroid of the *i*th cluster.  is the average distance between each point in the *j*th cluster and the centroid of the *j*th cluster. *di*,*j* is the Euclidean distance between the centroids of the *i*th and *j*th clusters. The maximum value of *D*i,j represents the worst-case within-to-between cluster ratio for cluster *i*. The **optimal clustering** solution has the **smallest Davies-Bouldin index value**.

* Davies, D. L., and D. W. Bouldin. “A Cluster Separation Measure.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. PAMI-1, No. 2, 1979, pp. 224–227.

### Gap

A common graphical approach to cluster evaluation involves plotting an error measurement versus several proposed numbers of clusters and locating the “elbow” of this plot. The “elbow” occurs at the most dramatic decrease in error measurement. The gap criterion formalizes this approach by estimating the “elbow” location as the number of clusters with the largest gap value. Therefore, under the gap criterion, the **optimal number of clusters** occurs at the solution with the **largest local or global gap value** within a tolerance range. The gap value is defined as

*Gapn*(*k*)=*En*{log(*Wk*)}−log(*Wk*),

where *n* is the sample size, *k* is the number of clusters being evaluated, and *W*k is the pooled within-cluster dispersion measurement

,

where *nr* is the number of data points in cluster *r*, and *D*r is the sum of the pairwise distances for all points in cluster *r*. The expected value *En*{log(*Wk*)} is determined by Monte Carlo sampling from a reference distribution, and log(*W*k) is computed from the sample data. The gap value is defined even for clustering solutions that contain only one cluster and can be used with any distance metric. However, the gap criterion is **more computationally expensive than other cluster evaluation criteria**, because the clustering algorithm must be applied to the reference data for each proposed clustering solution.

* Tibshirani, R., G. Walther, and T. Hastie. “Estimating the number of clusters in a data set via the gap statistic.” *Journal of the Royal Statistical Society: Series B*. Vol. 63, Part 2, 2001, pp. 411–423.

### Manual Choice of number of regions

With this option, the user can manually decide how many regions to group the neighborhoods into. This is the fastest method of clustering and is useful if you are only looking for a few broad features.

## Model Name

Additionally, the user is able to create multiple models of same/different types. This can be helpful for comparison of different methods with the same number of clusters, or the same method, with different number of clusters. This can be done through field in the bottom center of the window (Field 5 on Figure 4).

In order to use this feature, user needs to name model which is to be trained, and it will be saved under that name. If the user wants to overwrite the model already existing, then he needs to put that models name in the text box. This will give the user to either retrain the model, or reuse it on the currently chosen data.

## Color Scheme (Advanced Feature)

TODO (Field 3 on Figure 4)

# Plotting/Visualization

## New Figure

Clicking the new figure button in the main CytoMAP window will open a new plot. Using this interface the user can plot any cell type, use the colormap to color-code the cells by MFI or other calculated parameters, explore the makeup of the neighbors. Hover the mouse over the buttons on the tool-strip at the top of the new figure window for descriptions of their function.

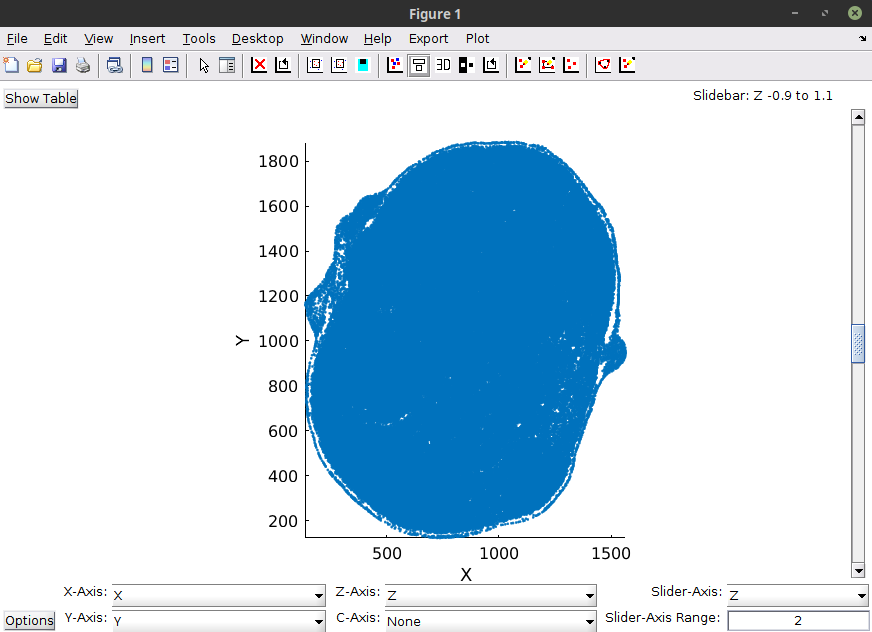


Figure 5a| New figure, by default the x-y positions of all cells are plotted.

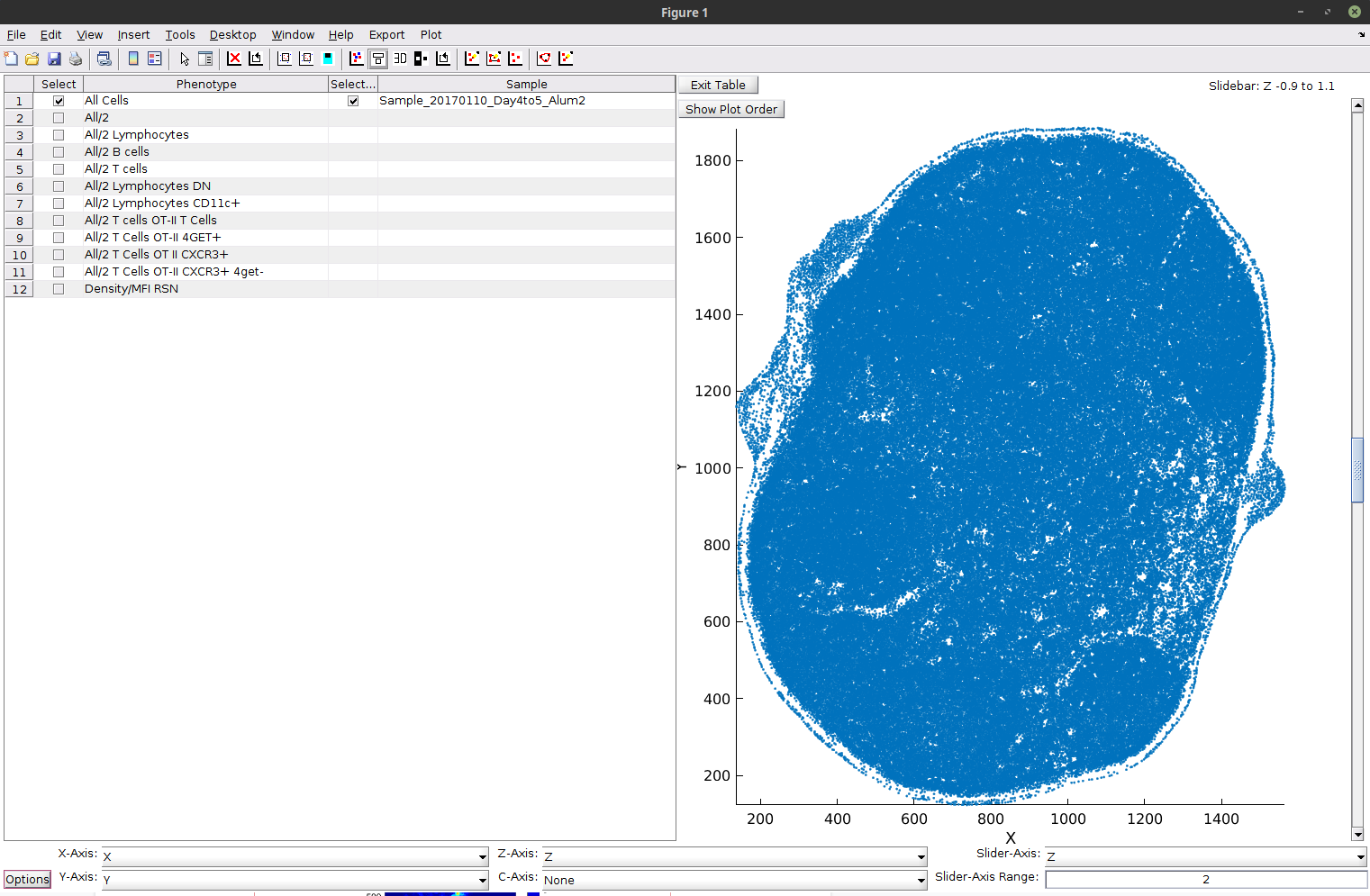


Figure 5b| New figure, by default the x-y positions of all cells are plotted, with interface of choosing different phenotypes, and samples.

### Phenotype/Sample Table

The user has the ability of choosing samples and phenotypes to be shown on the plot. Phenotypes available to be chosen are dependent on the samples chosen. This is because not all samples have to be have all phenotypes, and in order to be able to visualize all of the chosen samples equally every sample has to have all of the phenotypes chosen.

To open the table click “Show Table” in the upper right corner of the plot. To close it choose “Exit Table in the upper center. To choose a phenotype/sample click a corresponding tick mark in the “Select” column to the left of the thing you want to display.

*Plot Ordering*

This function is only available when the Phenotype/Sample is opened. Then under “Exit Table”, there is a button “Show Plot Order”. Clicking it will open a new window, assigned to the specific figure, which allows user to manipulate the order in which samples and phenotypes are plotted.

The most important effect of this feature is when Color Axis is set to “None,” and multiple, overlapping samples are chosen. Then, each Phenotype/Sample combination will be assigned a different color, and if there will be multiple cells from different points, the top-most in the plot ordering menu will be shown. The same effect happens, when Color Axis is not set to “None”, but it might be less noticable, since colors are defined the same for each Phenotype/Sample combination.

*Point Density Plot*

Point Density Plot is an option unique for Color Axis in 2D view. It divides the plot into a 100x100 grid, in which every rectangle is colored in color corresponding to number of elements within this square. It can be thought of as a raster scan in 2 dimensions, and it should yield equivalent results, if done on the same 2 axes.

### Density

Using the “number of cells/neighborhood” option in the new figure window’s c-axis drop-down menu, the user can plot the number of cells per neighborhood, which helps visualize the local density of the various cell types. To do this, raster-scanned neighborhoods must be calculated first. The local resolution of this density depends on the user-defined radius of the raster-scanned neighborhoods.

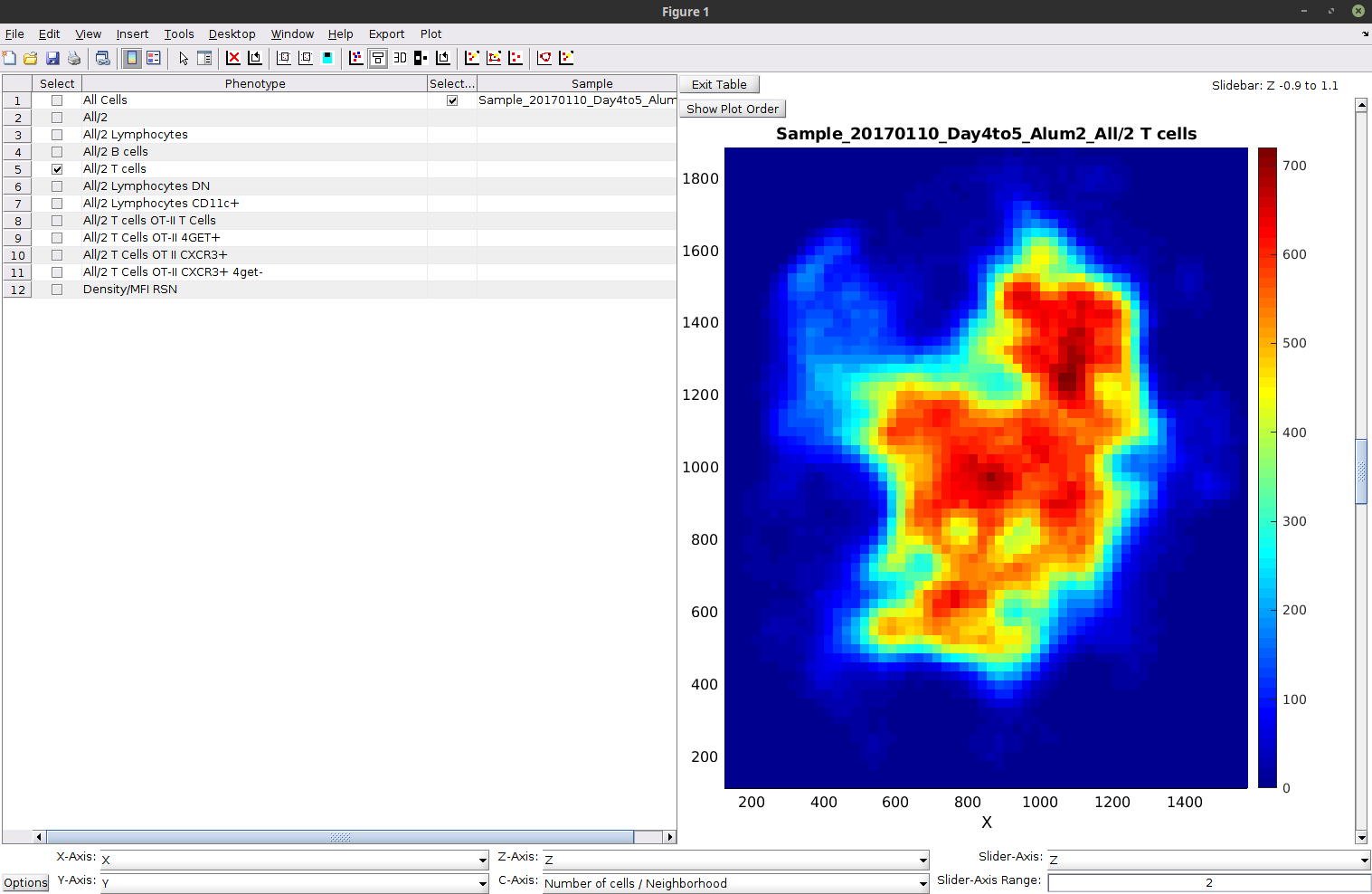


Figure 6| Example plot the cell density.

### Histograms

By selecting the histogram option under the Y-Axis parameter the user can plot histograms of the data.

Note:

The “Number of Cells/Neighborhood” selection on Color Axis overrides the either of the histogram selections on Y Axis. This is because they cannot exist simultaneously, i.e. there is no way of coloring the histogram in a meaningful way.

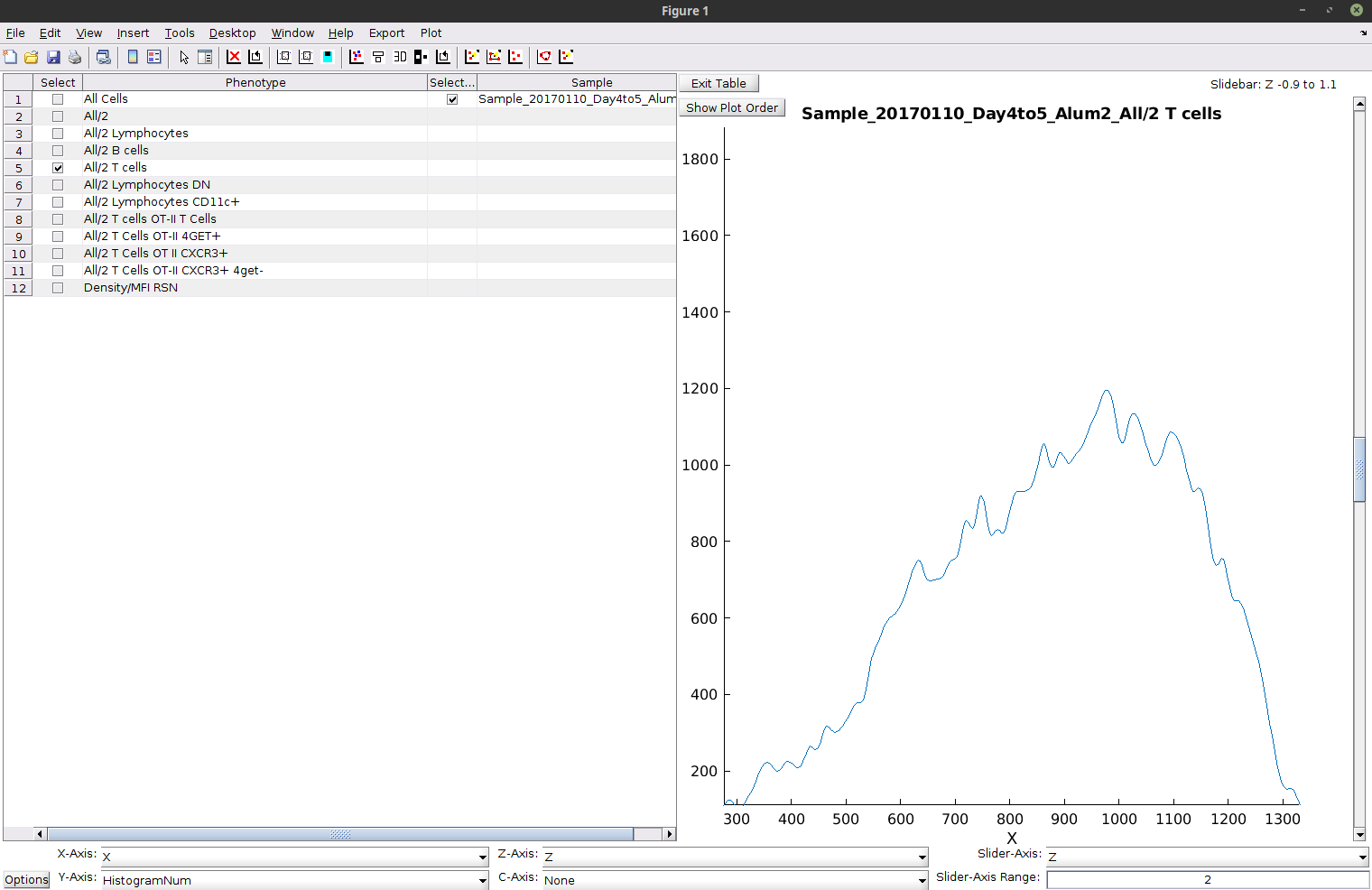


Figure 7| Example histogram plot, showing the distribution of X coordinate in the tissue.

## Heatmap

This plots the makeup of each region, in terms of the average number/percentage of cells. After classifying neighborhoods into regions click the “Plot Heatmaps” button on the main window. The user can change what is plotted on the color axis to match the type of data preparation used for clustering. Any cell type, or MFI can be added to the heatmap, even if it wasn’t used for clustering, as long as it was calculated for the neighborhoods. Multiple samples or groups of samples can be combined on one heatmap, and the color scale can be set to log or linear.

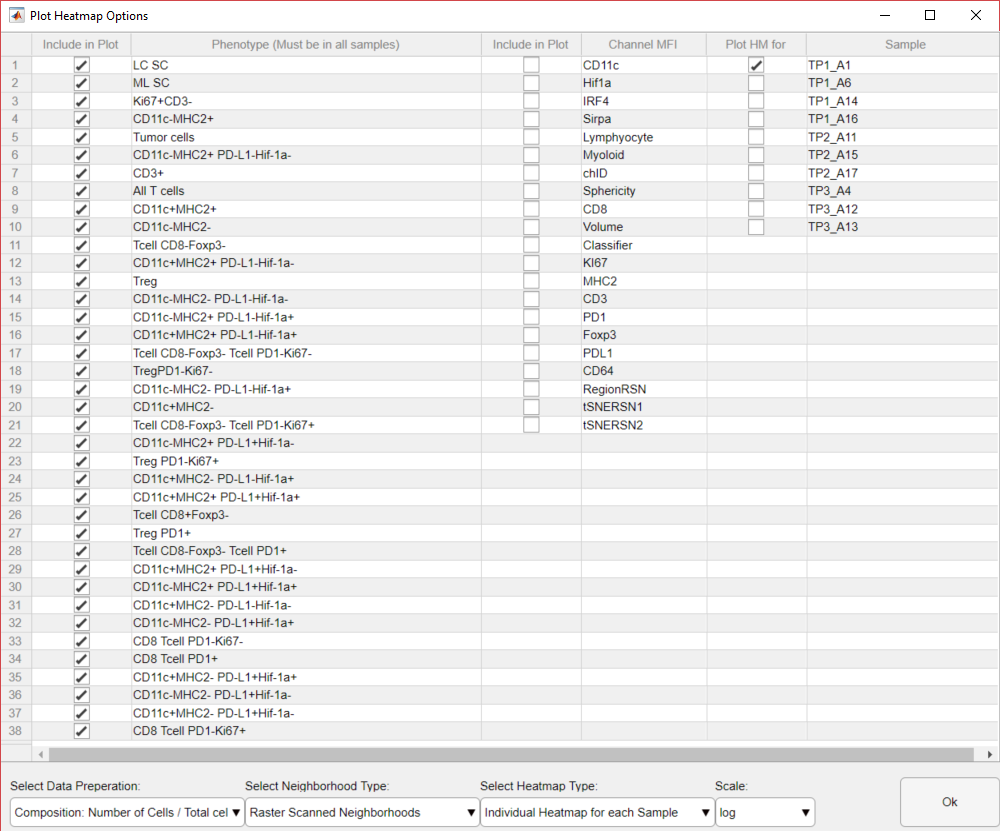


Figure 8| Options for plotting heatmaps of the regions compositions.

## Region Statistics

* This button produces multiple plots, which help investigate what the composition of the different regions are. The plots include line graphs showing the number of cells in each region and the fold change of the cellular composition of the regions compared to the averages across all of the tissues. This also outputs a plot showing what percentage of the total image is made up of each region.

## Cellularity

* This button produces plots of the number and percentage of cells across multiple samples and groups of samples, as both line graphs and heatmaps.

## t-SNE

* The t-SNE plot reduces the neighborhood data onto two axes, which can be used to visualize how the neighborhood data is distributed. This can be color-coded by cellular density, MFI, sample number, or regions. This is useful for comparing groups of samples. The t-sne axes are added to your data once calculated and can be plotted in a new figure.

## Box Plot

* This can be used to compare the expression/distances of different cell types.

# Calculations: Distance

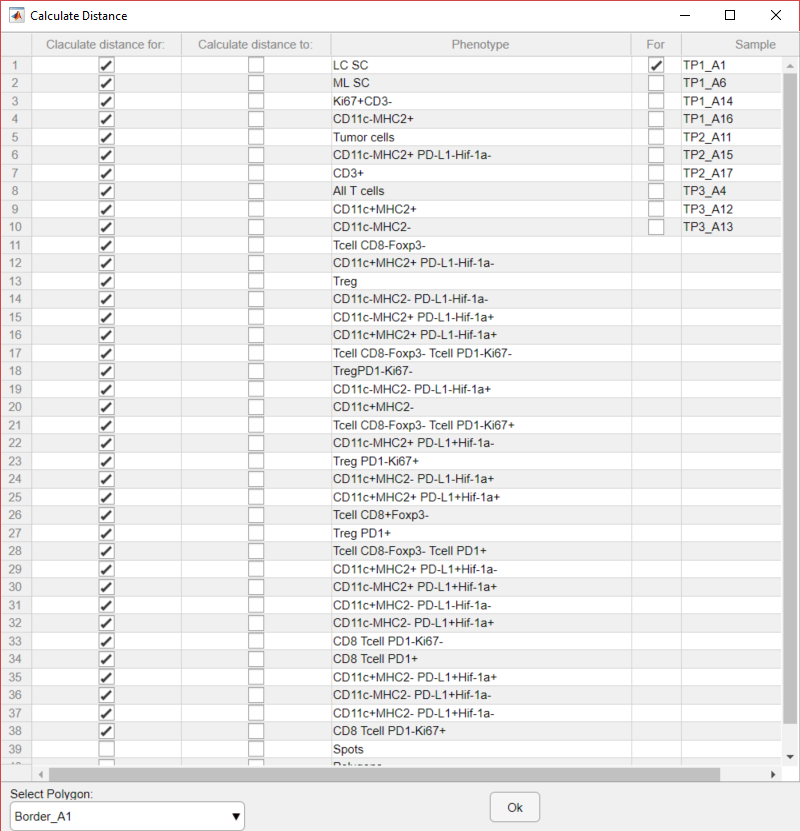


Figure 9| Options for calculating distances.

## Cell-Cell

* The user can calculate the distance from anyone cell type to the nearest cell of any other cell type.

## Polygon

* The user can manually define a polygon (on an x-y plot in a new figure), and calculate the distance from cells to this polygon. Additionally, this calculation differentiates between cells inside of the polygon and those outside of the polygon by making all “outside” distances negative. This works with either one polygon or multiple polygons.

## Points

* Calculate distance to manually defined points. This is useful for quickly calculating the distance to the center of the tissue, etc.

## Regions

* Calculate the distance to the various regions defined when classifying neighborhoods into regions. This is useful for showing where cells are located in relation to the regions defined using clustering.