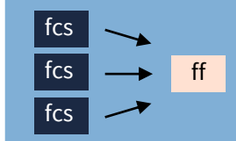
Cytometry analysis with FLOWSOM: : CHEAT SHEET

import FlowSOM as fs

Basics

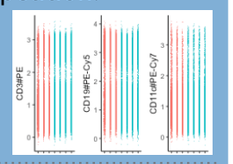
PREPARING DATA

While FlowSOM has some integrated preprocessing functions, it is often easiest to prepare preprocessed files upfront: compensate, transform, normalize …



aggregate_flowFrames(filenames, c_total)
Combine multiple fcs files for training. Uses random sampling without repetition.

plot_file_scatters (files, channels, groups, …) Makes a 2D scatter plot for each subset of interest.



TRAINING A FLOWSOM MODEL

fs.FlowSOM(input, cols_to_use = None, xdim = 10, ydim = 10, n_clus = 10, seed = NULL, …)

input: a file path or an anndata object

cols_to_use: Channels, markers or ids. If NULL, all columns are used

xdim, ydim: xdim * ydim = number of clusters

n_clus: Number of metaclusters

seed: For exact reproducibility

fs.flowsom_clustering(input, cols_to_use, n-clus, x_dim, y_dim, seed) Adds a FlowSOM clustering and metaclustering column as obs in the anndata object

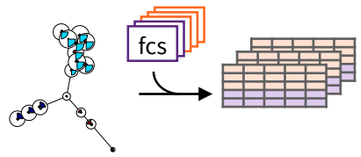
Extra parameters can be found in the documentation.

EXPLORATIVE FUNCTIONS

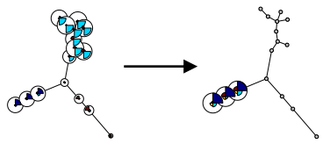
fsom.get_cell_data() – fsom.get_cluster_data()
Returns the cell data or cluster data of the fsom object

fs.pl.FlowSOMmary(fsom, …)
Plots a summary of the FlowSOM object to a pdf file.

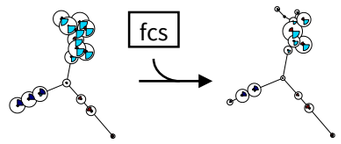
Comparing multiple files



fs.tl.get_features(fsom, files, level = c("clusters", "metaclusters"), type = "counts", MFI, positive_cutoffs, filenames)
Extracts the features specified in type ("counts", "percentages", "MFIs" and/or percentages_positive) from clusters and/or metaclusters.



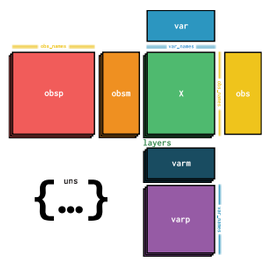
fsom.subset(ids)
Takes a subset from a FlowSOM object, with in "ids" the ids of which cells to keep.



fsom.new_data(inp, mad_allowed = 4, …)
Maps new data to an existing FlowSOM object.

FlowSOM Object

The FlowSOM object is a mudata object consisting of two anndata objects: the cell data and the cluster data.



Cell data:
- X: cell data
- obs: clustering, distance_to_bmu, metaclustering
- var: pretty_colnames, markers, channels, cols_used
- uns: n_nodes, n_metaclusters

Cluster data:
- X: cluster MFIs
- obs: percentages, metaclustering
- uns: xdim, ydim, outliers, graph, metacluster_MFIs
- obsm: cv_values, sd_values, mad_values, codes, grid, layout

Getter functions

fs.tl.get_channels(object, markers, exact = TRUE)
fs.tl.get_markers(object, channels, exact = TRUE)
Gets the channel- or marker names from the markers or channels respectively out of an FCS anndata or fsom object. Returns a dictionary

fs.tl.get_counts(fsom, level = "metaclusters")
Gets counts of number of cells in (meta-)clusters.

fs.tl.get_percentages(fsom, level = "metaclusters")
Gets percentages of number of cells in (meta-)clusters.

fs.tl.get_cluster_percentages_positive(fsom, cutoffs, …) – **fs.tl.get_metacluster_percentages_positive**(fsom, cutoffs, …)
Gets percentage-positive values per marker for all (meta-)clusters.

fsom.test_outliers(fsom, mad_allowed = 4, fsom_reference, plot_file)
Tests if any cells are further than expected from their cluster centers.

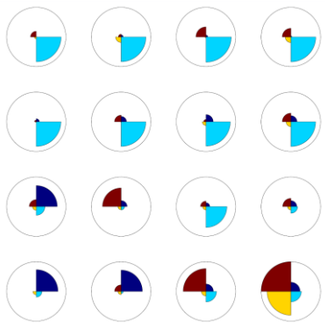
SaeysLab
Data mining and Modelling for Biomedicine

Cytometry analysis with FLOWSOM: : CHEAT SHEET

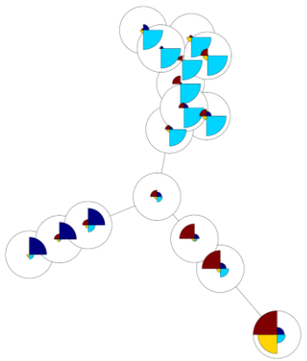
FlowSOM plotting

GRID VIEW AND MINIMUM SPANNING TREE (MST)

The grid structure is built during the SOM clustering. The data is presented cell by cell to the grid and is attached to the closest cluster center. Then this cluster center and the surrounding centers get updated. In the end the clusters close to each other will be similar. view = “grid”

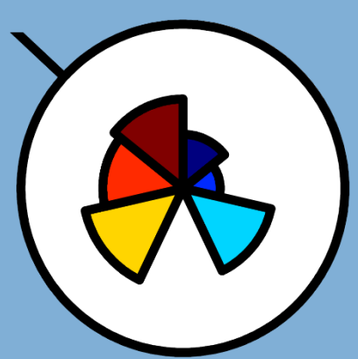


MST connects the nodes of a graph in such a way that the sum of the weights of the branches is minimal. By doing this, nodes will get connected to the ones they are the most similar too, taking the multi-dimensional topology of the data in account. Similarity inferences can only be made on close nodes. view = “MST”



FLOWSOM STARCHART PRESENTATION

The height of each part indicates the expression of that marker: if the part reaches the border of the circle, the cluster's median intensity value is the highest of all FlowSOM clusters.



The size of the node indicates the relative number of cells assigned to each node.

Plotting FlowSOM objects

GENERAL PLOTTING FUNCTIONS

fs.pl.plot_stars(fsom, markers = None, cmap, …)
Shows MFI expression of multiple markers.

fs.pl.plot_numbers(fsom, level = “clusters”, …)
Shows (meta-)cluster numbers.

fs.pl.plot_marker(fsom, marker, ref_markers =None, cmap, lim, …)
Shows the median value of every cluster for this marker.

PLOTTING WITH EXTERNAL INFORMATION ON CLUSTER LEVEL

fs.pl.plot_labels(fsom, labels, text_color, text_size, text_color, …)
Shows a label for every cluster.

fs.pl.plot_variable(fsom, variable, cmap, lim, labels, …)
Shows a variable for every cluster.

PLOTTING WITH EXTERNAL INFORMATION ON CELL LEVEL

fs.plot_pies(fsom, cell_types, cmap, …)
Shows the percentage of cells belonging to each cell type.

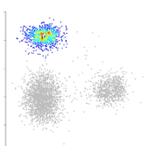
PARAMETERS AVAILABLE IN THE ABOVE FlowSOM PLOTTING FUNCTIONS

view:	‘MST’ (default), ‘grid’ or coordinates
node_sizes:	Node sizes. Default = fsom.get_cluster_data().obs.percentages
max_node_size:	Maximum node size. Default = 1
ref_node_size:	Reference for node size against which node sizes will be scaled. Default = max(node_sizes).

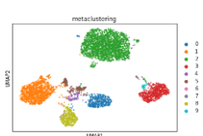
$$node\_sizes_{scaled} = \sqrt{\frac{node\_sizes}{ref\_node\_size}} * \max\_node\_size$$

equal_node_size:	Logical, should nodes be equal sized or not. If TRUE, nodesizes will be equal to max_node_size. Default = FALSE
background_values:	Values used for background coloring
background_colors:	Colors used for background coloring
background_size:	The size of the background relative to the node.
equal_background_size	Logical, if background size should be equal
background_lim:	Limits for numerical background values
title:	Title for plot

Other plotting functions



fs.pl.plot_2D_scatters(fsom, channelpairs, clusters, metaclusters, …)
Shows a scatterplot of (meta-)clusters of interest.



import scanpy as sc
sc.pp.neighbors(fsom)
sc.tl.umap(fsom)sc.pl.umap(fsom, color= “metaclustering”)
Shows a UMAP dimensionality reduction plot colored by the metaclustering.

SaeysLab
Data mining and Modelling for Biomedicine