

# Cytometry analysis with FLOW SOM: : CHEAT SHEET

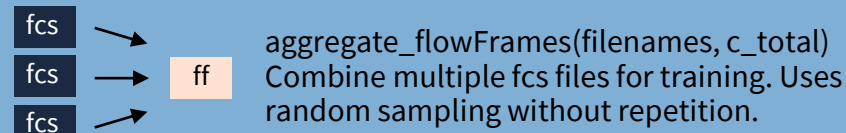


```
import flowsom as fs
```

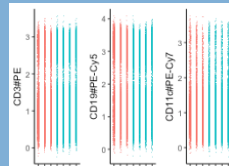
## Basics

### PREPARING DATA

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



**plot\_file\_scatters** (files, channels, groups, ...) Makes a 2D scatter plot for each subset of interest.



### TRAINING A FLOW SOM 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.

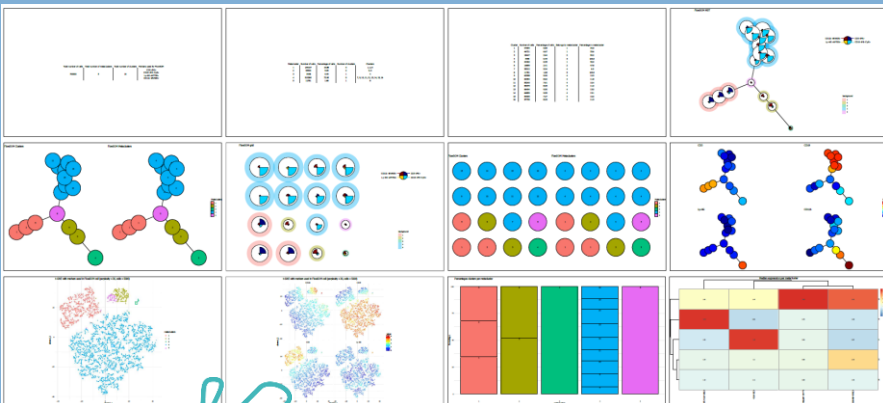
### EXPLORATIVE FUNCTIONS

**fsom.get\_cell\_data()** – **fsom.get\_cluster\_data()**

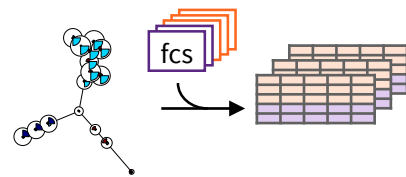
Returns the cell data or cluster data of the fsom object

**fs.pl.FlowSOMmary**(fsom, ...)

Plots a summary of the FlowSOM object to a pdf file.

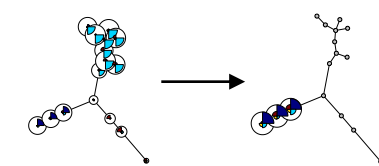


## Comparing multiple files



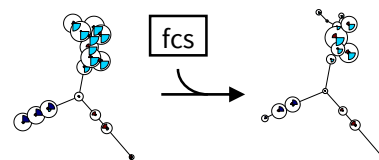
**fs.tl.get\_features**(fsom, files, level = c("clusters", "metaclusters"), type = "counts", MFI, positive\_cutoffs, filenames)

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.

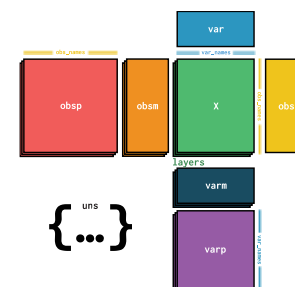


**fsom.new\_data**(inp, mad\_allowed = 4, ...)

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.

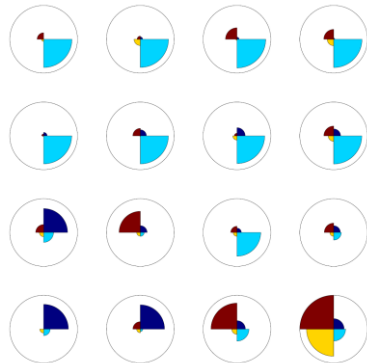
# Cytometry analysis with FLOW SOM: : CHEAT SHEET



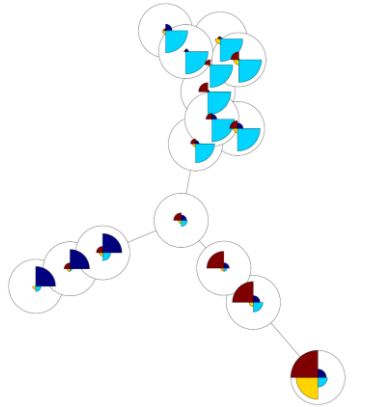
## FlowSOM plotting

### 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.

66

**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.pl.plot\_pies**(fsom, cell\_types, cmap, ...) Shows the percentage of cells belonging to each cell type.

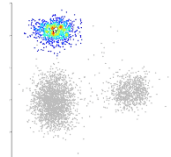
### PARAMETERS AVAILABLE IN THE ABOVE FlowSOM PLOTTING FUNCTIONS

**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).

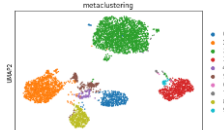
$$node\_sizes_{scaled} = \frac{node\_sizes}{ref\_node\_size} * max\_node\_size$$

**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.