

05

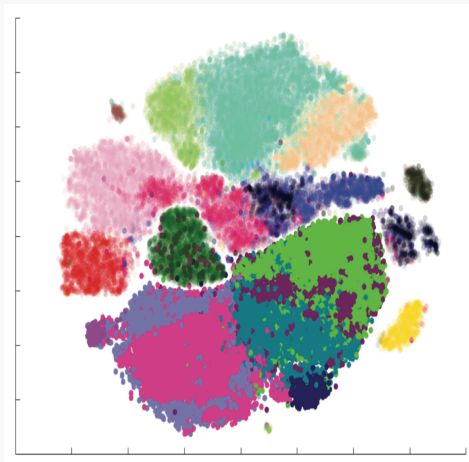
Dimensionality reduction

Dimensionality reduction (DR)

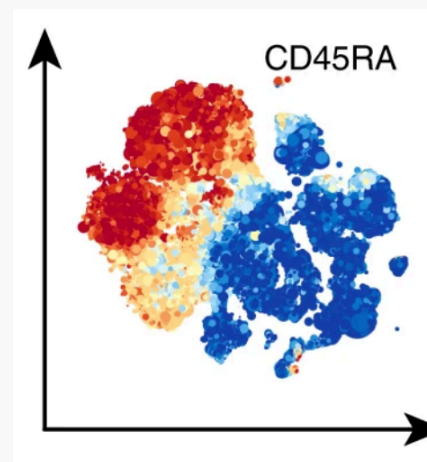
- Represent the cellular heterogeneity assessed by many parameters into a two-dimensional scatterplot
- Commonly applied DR methods:
 - t-stochastic neighbor embedding (tSNE)
 - embedding hierarchical stochastic neighbor embedding (HSNE)
 - uniform manifold approximation and projection (UMAP)

Do you really need DR? Remember: there is no "correct" DR plot.

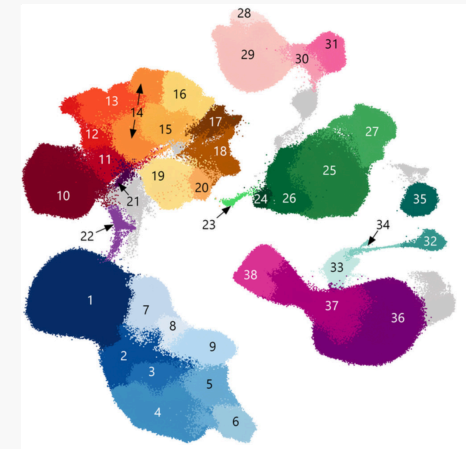
tSNE



HSNE



UMAP



Scalability of HSNE compared to tSNE:

<https://www.nature.com/articles/s41467-017-01689-9>

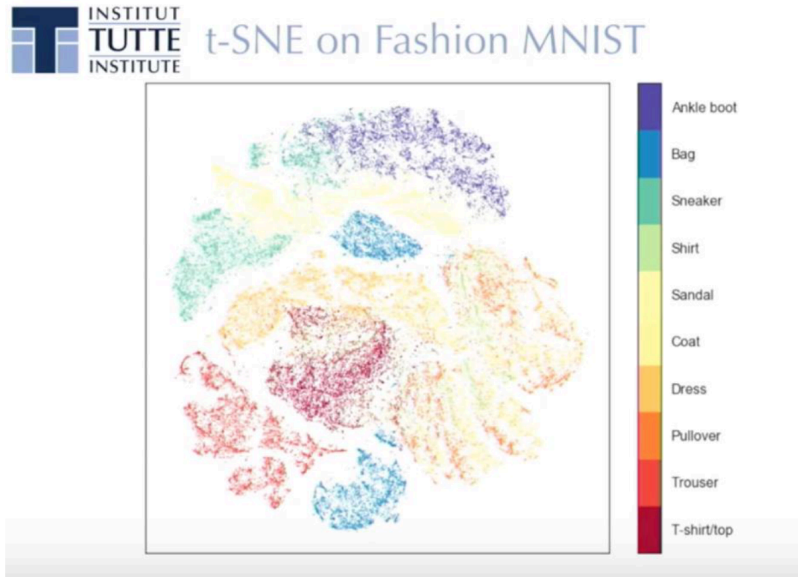
Mathematical explanation of PCA, tSNE and UMAP (starting at 8'25'')

https://sib-swiss.github.io/single-cell-training/day2/day2-1_dimensionality_reduction.html

Comparative analysis of dimension reduction methods for cytometry by time-of-flight data

<https://www.nature.com/articles/s41467-023-37478-w>

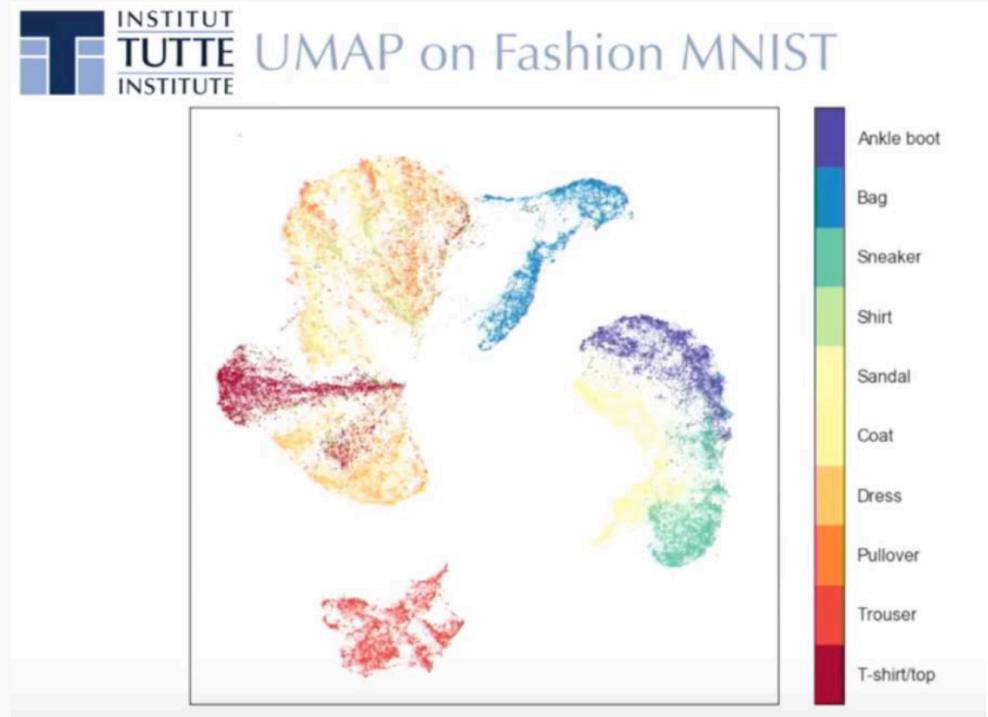
tSNE vs UMAP



From L.McInnes, SciPy 2018

TSNE preserves local similarity only

UMAP also preserves some of the global similarities



CATALYST

<https://bioconductor.org/packages/release/bioc/html/CATALYST.html>

- Tools for preprocessing of and differential discovery in cytometry data, including:
 - Dimensional reduction
 - Clustering
 - Visualization for exploratory data analysis and exploration of results from differential abundance (DA) and state (DS) analysis

*CATALYST operates with
SingleCellExperiment (sce)
class of objects*

Example dataset from the CATALYST package

- 8 PBMC samples from 4 patients, mass cytometry
- 2 conditions: before (REF) and upon BCR/FcR-XL stimulation (BCRXL) with B cell receptor/Fc receptor crosslinking for 30'
- Expression of 10 cell surface proteins and 14 signaling markers

```
> data(PBMC_fs, PBMC_panel, PBMC_md)
```

```
> PBMC_fs
```

```
A flowSet with 8 experiments.
```

```
column names(24): CD3(110:114)Dd CD45(In115)Dd ... HLA-DR(Yb174)Dd  
CD7(Yb176)Dd
```

Example dataset from CATALYST

- 8 PBMC samples from 4 patients, mass cytometry
- 2 conditions: before (REF) and upon BCR/FcR-XL stimulation (BCRXL) with B cell receptor/Fc receptor crosslinking for 30'
- Expression of 10 cell surface proteins and 14 signaling markers

> View(PBMC_md)

	file_name	sample_id	condition	patient_id
1	PBMC_patient1_BCRXL.fcs	BCRXL1	BCRXL	Patient1
2	PBMC_patient1_Ref.fcs	Ref1	Ref	Patient1
3	PBMC_patient2_BCRXL.fcs	BCRXL2	BCRXL	Patient2
4	PBMC_patient2_Ref.fcs	Ref2	Ref	Patient2
5	PBMC_patient3_BCRXL.fcs	BCRXL3	BCRXL	Patient3
6	PBMC_patient3_Ref.fcs	Ref3	Ref	Patient3
7	PBMC_patient4_BCRXL.fcs	BCRXL4	BCRXL	Patient4
8	PBMC_patient4_Ref.fcs	Ref4	Ref	Patient4

Example dataset from CATALYST

- 8 PBMC samples from 4 patients, mass cytometry
- 2 conditions: before (REF) and upon BCR/FcR-XL stimulation (BCRXL) with B cell receptor/ Fc receptor crosslinking for 30'
- Expression of 10 cell surface and 14 signaling markers

> View(PBMC_panel)

	fcs_colname	antigen	marker_class
1	CD3(110:114)Dd	CD3	type
2	CD45(In115)Dd	CD45	type
3	pNFkB(Nd142)Dd	pNFkB	state
4	pp38(Nd144)Dd	pp38	state
5	CD4(Nd145)Dd	CD4	type
6	CD20(Sm147)Dd	CD20	type
7	CD33(Nd148)Dd	CD33	type
8	pStat5(Nd150)Dd	pStat5	state
9	CD123(Eu151)Dd	CD123	type
10	pAkt(Sm152)Dd	pAkt	state
11	pStat1(Eu153)Dd	pStat1	state
12	pSHP2(Sm154)Dd	pSHP2	state
13	pZap70(Gd156)Dd	pZap70	state
14	pStat3(Gd158)Dd	pStat3	state
15	CD14(Gd160)Dd	CD14	type
16	pSlp76(Dy164)Dd	pSlp76	state
17	pBtk(Er166)Dd	pBtk	state
18	pPlcg2(Er167)Dd	pPlcg2	state
19	pErk(Er168)Dd	pErk	state
20	pLat(Er170)Dd	pLat	state
21	IgM(Yb171)Dd	IgM	type
22	pS6(Yb172)Dd	pS6	state
23	HLA-DR(Yb174)Dd	HLA-DR	type
24	CD7(Yb176)Dd	CD7	type

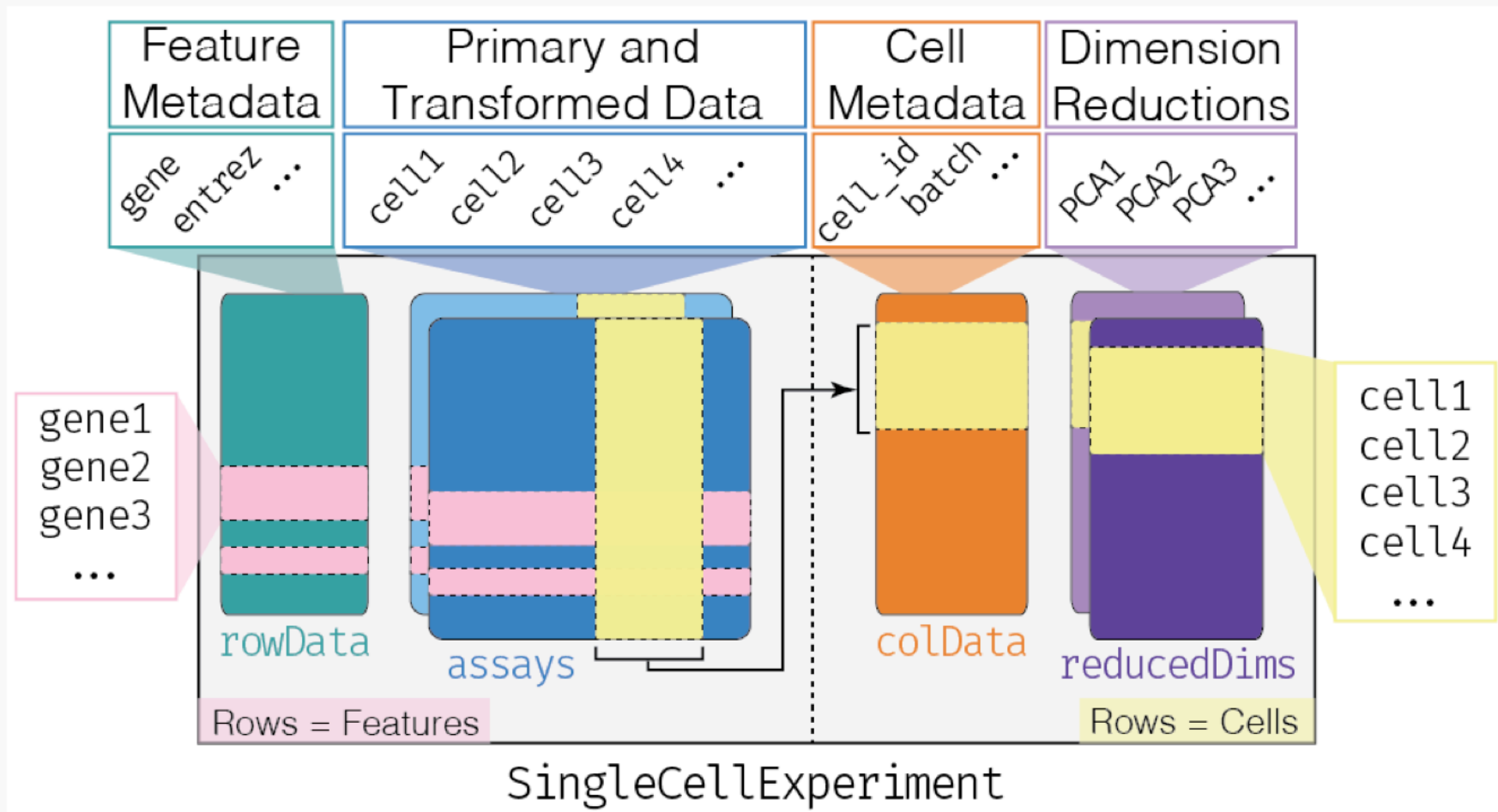
SingleCellExperiment (sce) class of objects

<https://bioconductor.org/packages/devel/bioc/vignettes/SingleCellExperiment/inst/doc/intro.html>

- Lightweight Bioconductor container for storing and manipulating single-cell (genomics) data.
- Rows contain features (proteins) and columns contain cells
- Provides methods for storing dimensionality reduction results
- It is the central data structure for Bioconductor single-cell packages

SingleCellExperiment class - Bioconductor

```
> class(sce)  
[1] "SingleCellExperiment"
```



Creating the SingleCellExperiment object

```
> sce_PBMC <- prepData(PBMC_fs,  
  md=PBMC_md,  
  panel= PBMC_panel,  
  transform = TRUE,  
  FACS = FALSE,  
  features = NULL)
```

Is this FACS / flow cytometry data ?

a flowSet holding all samples or a path to a set of FCS files

A data frame with data about samples.
For example: file_name, sample_id, patient_id and condition

A data.frame containing, for each channel, its column name in the input data, targeted protein marker, and (optionally) class ("type", "state", or "none").

Should arcsinh transformation be performed ?

List of markers to be used

Check name of the matrix with transformed values

```
> assayNames(sce_PBMC)
```

«raw» (input) values

```
[1] "counts" "exprs"
```

transformed values (if transform = TRUE)

UMAP with the CATALYST package

```
> sce_PBMC <- runDR(sce_PBMC,  
  assay = "exprs",  
  dr = "UMAP",  
  cells = NULL,  
  features="type",  
  n_neighbors = 10,  
  min_dist = 0.05)
```

Possibilities are "UMAP", "TSNE",
"PCA", "MDS" and "DiffusionMap"

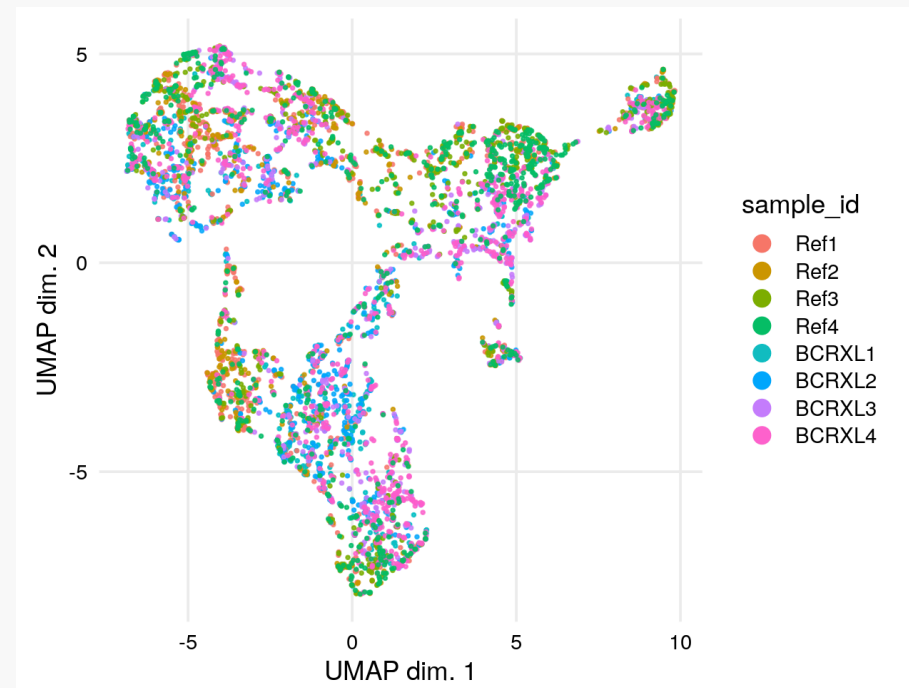
Try them out! TSNE is slow though

n_neighbors = how many neighbors to include in similarity estimation
min_dist = controls the spread of the points in the projection

Plot the UMAP with the CATALYST package

CATALYST provides the **plotDR function**, specifically to allow for coloring cells by the various grouping variables available, and to support facetting by metadata factors (e.g., experimental condition, sample IDs):

```
> plotDR(sce_PBMC,  
  dr = "UMAP",  
  color_by = "sample_id")
```

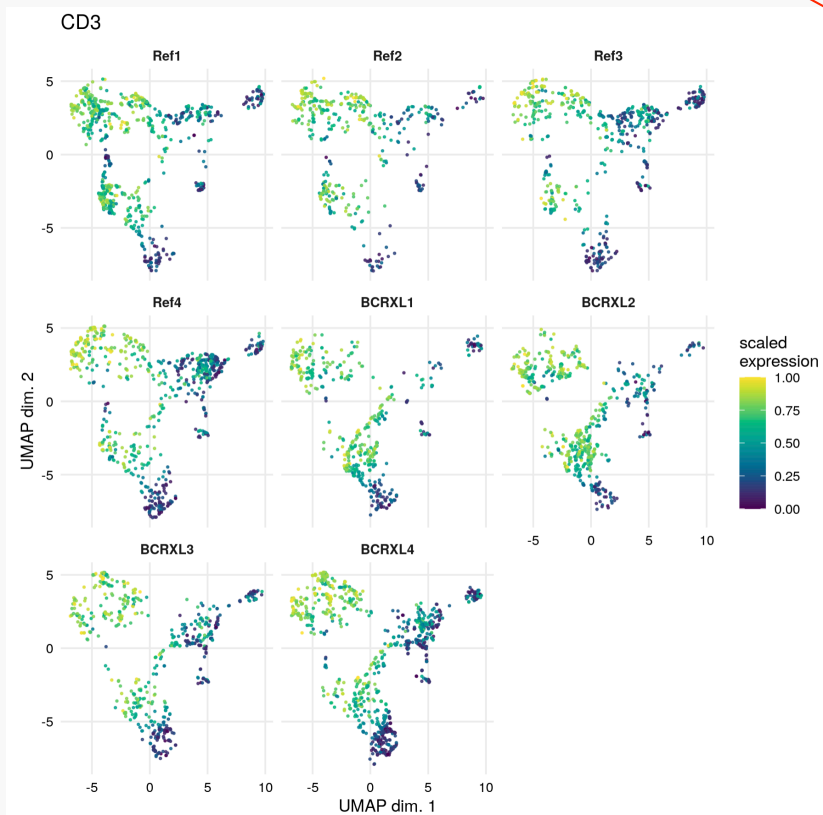


Plot the UMAP with the CATALYST package

```
> plotDR(sce_PBMc,  
  dr = "UMAP",  
  assay = "exprs",  
  color_by = "CD3",  
  facet_by = "sample_id")
```

marker (names(sce))
or
column of the experimental info
(sce@metadata\$experiment_info)

column of the experimental info
(sce@metadata\$experiment_info)



UMAP with the uwot R package

<https://cran.r-project.org/web/packages/uwot/index.html>

An R implementation of the Uniform Manifold Approximation and Projection (UMAP) method for dimensionality reduction

Extract the expression matrix and transpose

```
> exprs_PBMC <- assay(sce_PBMC, "exprs")  
> exprs_PBMC <- t(exprs_PBMC)
```

Subset to markers you want to use for clustering

```
> marker_type <- PBMC_panel$antigen[PBMC_panel$marker_class=="type"]  
> exprs_PBMC <- exprs_PBMC[,c(marker_type)]
```

Compute the UMAP

```
> set.seed(1234)  
> umap_PBMC <- umap(exprs_PBMC)
```

set a “seed” so that the results are reproducible

Add UMAP coordinates to sce object

```
> reducedDim(sce_PBMC, "UMAP") <- umap_PBMC
```

Plot using the plotDR function of CATALYST

```
> plotDR(sce_PBMC, dr = "UMAP", color_by="sample_id")
```

Let's practice – 4

In this exercise we will continue with the clean flowSet from the last exercise. We will use the CATALYST package to create a SingleCellExperiment (sce) object, perform dimensionality reduction (UMAP) and use the UMAP to plot the expression of markers.

Create a new script in which you will

- 1) Load the clean flowSet from last exercise («fcs_clean.Rdata»)
- 2) Downsample the flowSet to 2'000 cells per flowFrame (source the file «function_for_downsampling_flowSets.R»)
- 3) Create a sce object from the downsampled flowSet
- 4) Create a UMAP with default parameters, based on the expression of the «type» markers. Show the expression of CD3 by time point.
- 5) Check the effect of changing parameters «min_dist» and «n_neighbors» from the default values.



Clustering & Annotation

Unsupervised clustering vs Gating

- Flow cytometry data are traditionally analyzed by subjective gating of sub-populations on two-dimensional plots.
- This approach is highly dependent on the user's interpretation and knowledge and is time-consuming
- The increasing number of parameters measured by conventional and spectral flow cytometry reinforces the need to apply many of the recently developed tools for single-cell analysis on flow cytometry data

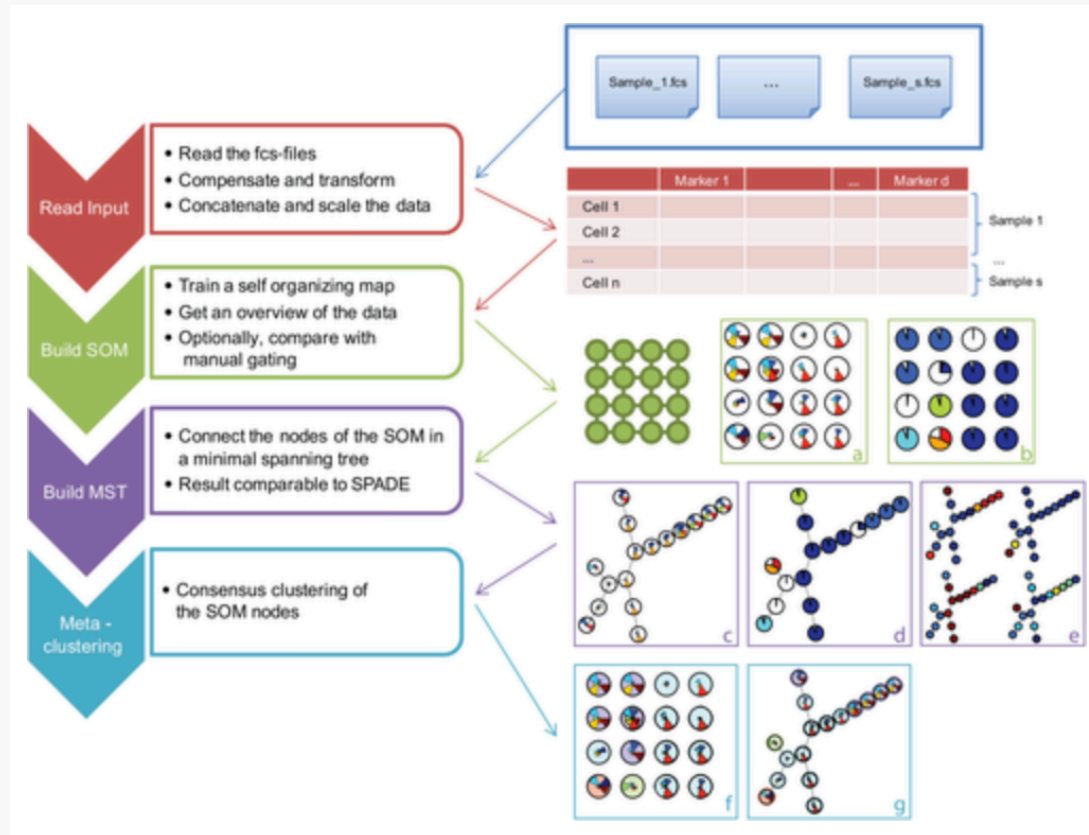
The *CATALYST* package provides a function to first cluster data with **FlowSOM** clustering and then apply **ConsensusClusterPlus** metaclustering

FlowSOM for clustering

<https://bioconductor.org/packages/release/bioc/html/FlowSOM.html>

- Generates **Self-organizing maps (SOM)** for visualization and interpretation of cytometry data
- A self-organizing map (SOM) is an unsupervised technique for clustering and dimensionality reduction, in which a discretized representation of the input space is trained
- The advantage of FlowSOM clustering is the speed of the algorithm
- SOM can be used to distinguish cell populations in an unsupervised way
- However, FlowSOM generates a much larger amount of clusters than the expected number of cell types -> **metaclustering**

FlowSOM for clustering



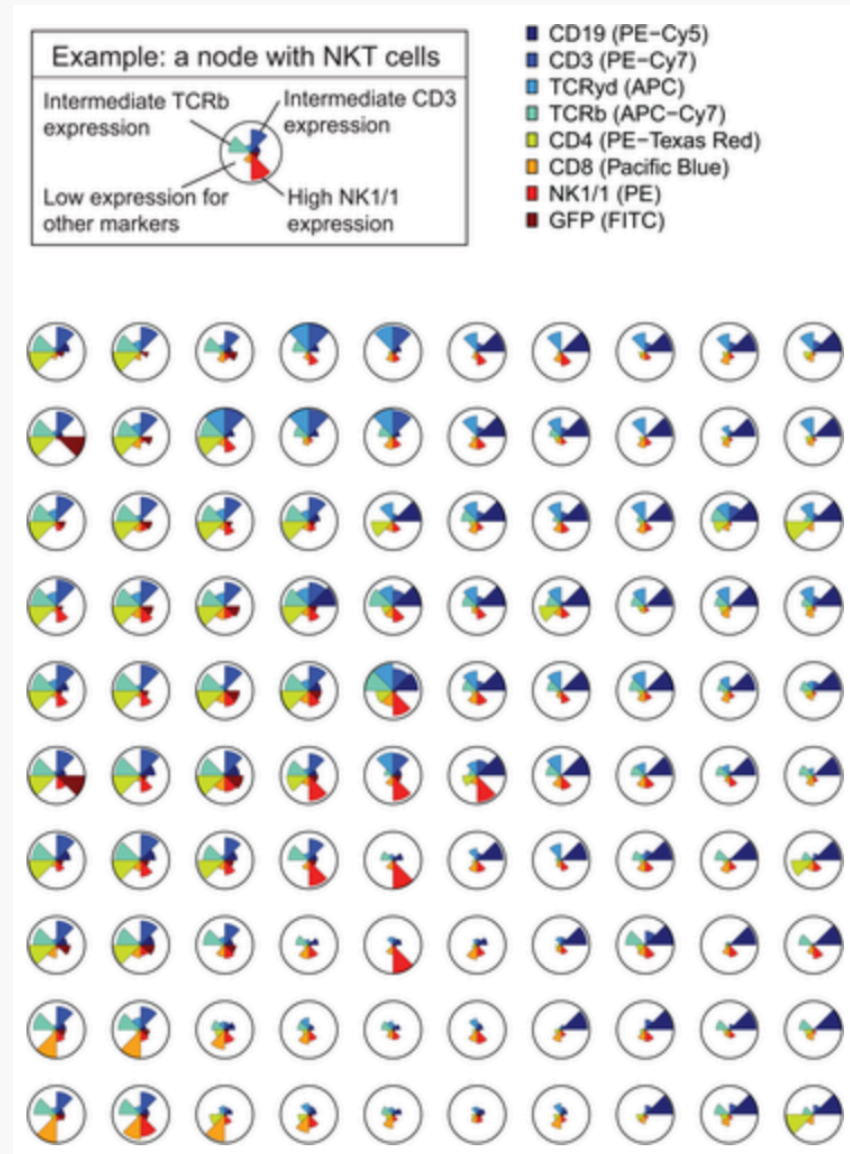
Train the SOM on the matrix. The result is a **grid of nodes, corresponding to cell clusters**

Build a **minimal spanning tree** for visualization

Gasse et al., Cytometry (2015)

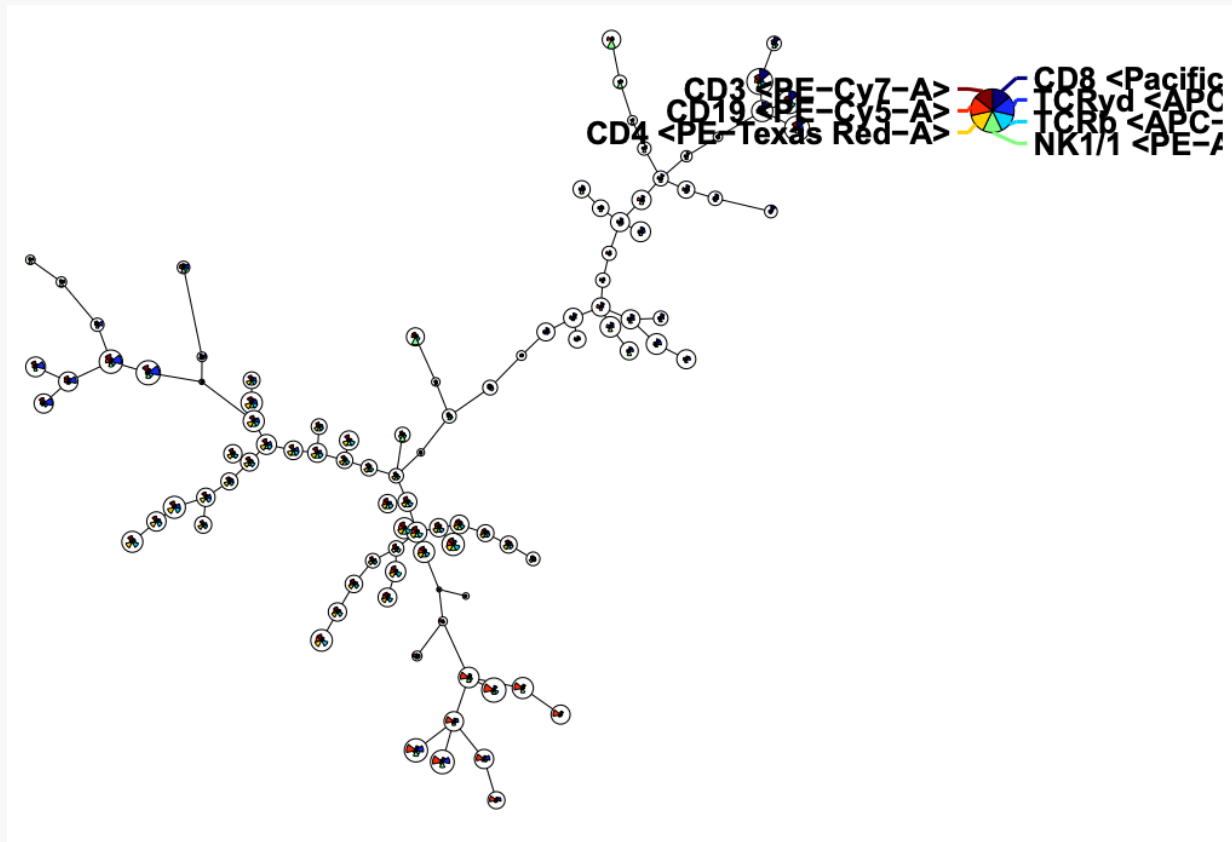
Self-organizing map (SOM)

- A SOM consists of a **grid of nodes** (points in the multidimensional input space)
- When clustering, cells are classified with the node that is its nearest neighbour
- The grid is trained in such a way that the nodes closely connected to each other resemble each other more than nodes that are only connected through a long path
- In the end, each cell of the dataset is assigned to the node that resembles it the best, resulting in the final clustering



Minimum spanning tree

- The resulting clustering of the SOM can be visualized in a **minimal spanning tree**



<https://bioconductor.org/packages/release/bioc/vignettes/FlowSOM/inst/doc/FlowSOM.pdf>

ConsensusClusterPlus metaclustering

- SOMs can be used to get an immediate clustering
- However, it is advantageous to include more nodes than the expected number of clusters: cells that are in between cell types can also get a place in the grid and smaller changes in the cell types can be noticed

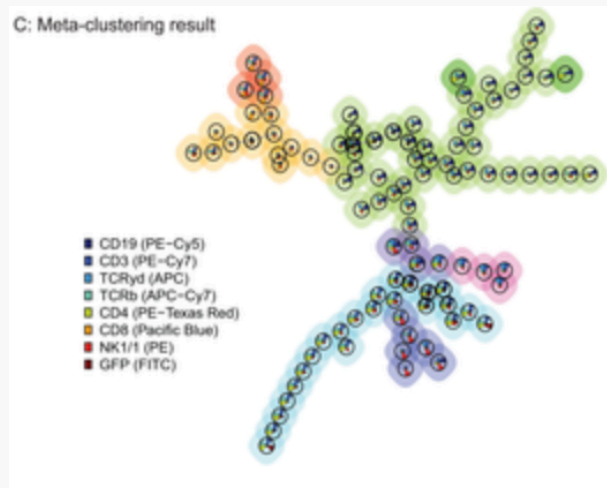


Metaclustering

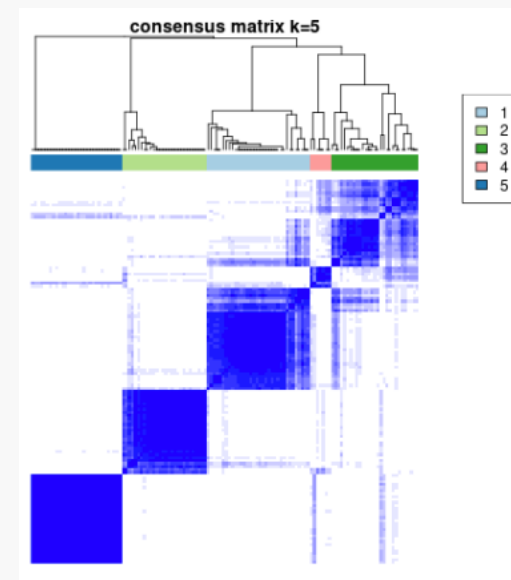
ConsensusClusterPlus metaclustering

<https://bioconductor.org/packages/release/bioc/html/ConsensusClusterPlus.html>

- **Consensus hierarchical clustering**: subsampling of the points several times, and making a **hierarchical clustering** for each subsampling. Based on how often the same points are clustered together or not, a final clustering is made



Gasse et al., Cytometry (2015)



<https://bioconductor.org/packages/release/bioc/vignettes/ConsensusClusterPlus/inst/doc/ConsensusClusterPlus.pdf>

Clustering with CATALYST

The CATALYST package provides a wrapper function to first cluster data with FlowSOM clustering and then apply ConsensusClusterPlus metaclustering

```
> sce_PBMC <- cluster(sce_PBMC,  
                      features="type",  
                      xdim = 10,  
                      ydim = 10,  
                      maxK=20,  
                      seed=5024)
```

```
> names(cluster_codes(sce_PBMC))
```

```
[1] "som100" "meta2" "meta3" "meta4" "meta5" "meta6" "meta7" "meta8" "meta9"  
"meta10" "meta11" "meta12" "meta13" "meta14" "meta15" "meta16" "meta17"  
"meta18" "meta19" "meta20"
```


Clustering with CATALYST

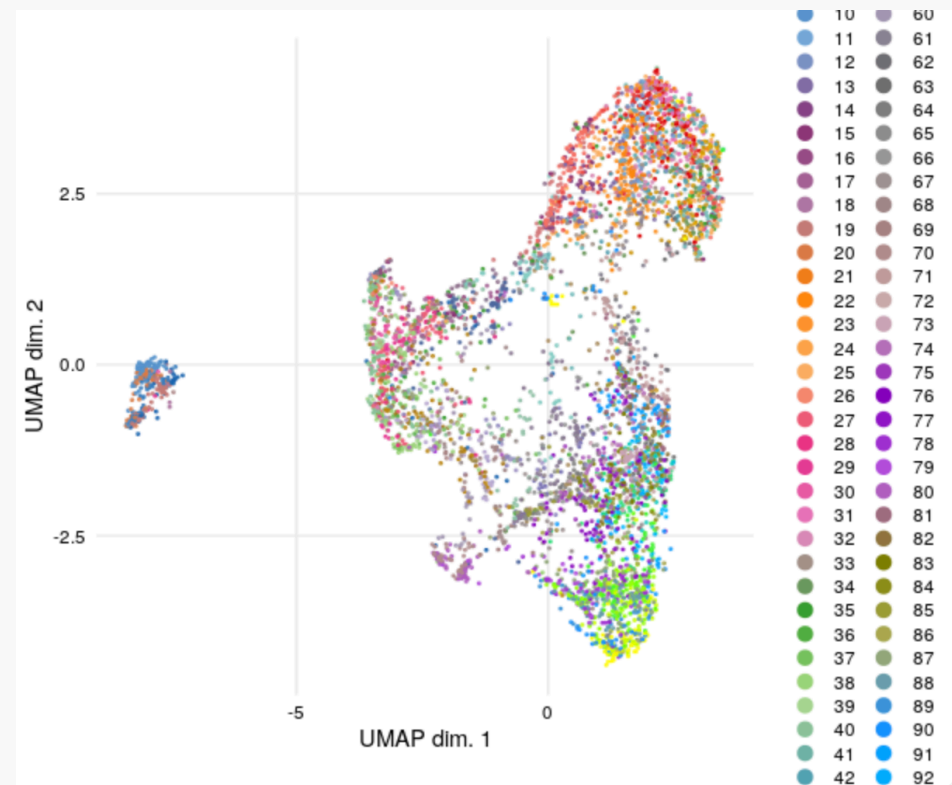
The CATALYST package provides a wrapper function to first cluster data with FlowSOM clustering and then apply ConsensusClusterPlus metaclustering

```
> sce_PBMC <- cluster(sce_PBMC,  
                      features="type",  
                      xdim = 10,  
                      ydim = 10,  
                      maxK=20,  
                      seed=5024)
```

```
> names(cluster_codes(sce_PBMC))
```

Plot UMAP with SOM clusters

```
> plotDR(sce_PBMC, "UMAP",  
         color_by="som100")
```



Clustering with CATALYST

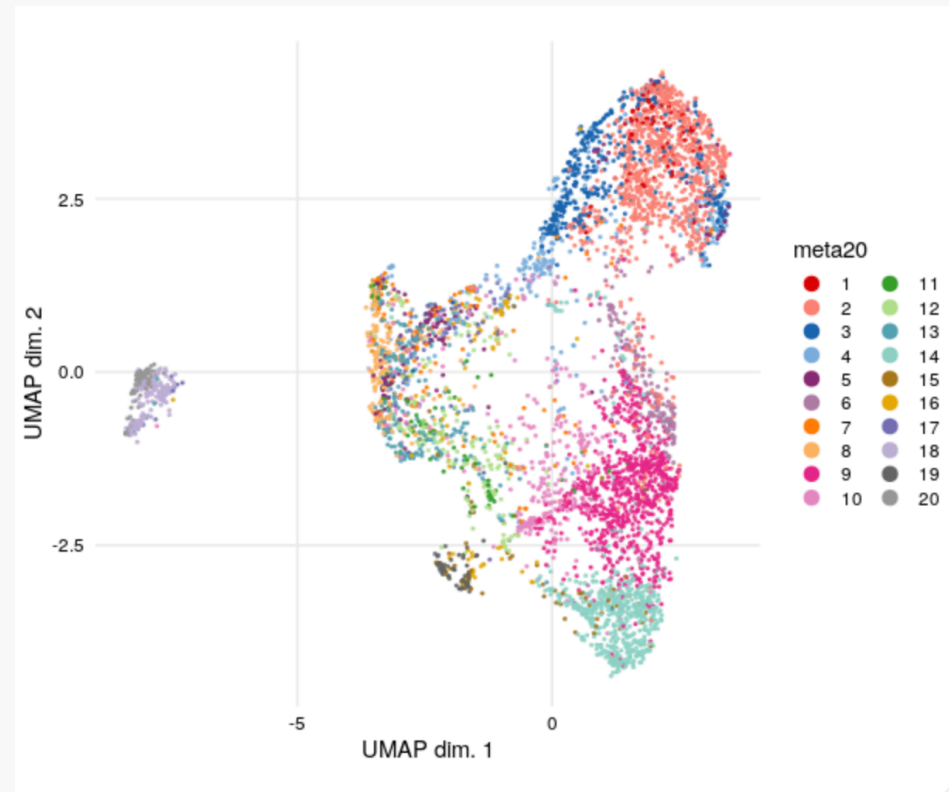
The CATALYST package provides a wrapper function to first cluster data with FlowSOM clustering and then apply ConsensusClusterPlus metaclustering

```
> sce_PBMC <- cluster(sce_PBMC,  
                      features="type",  
                      xdim = 10,  
                      ydim = 10,  
                      maxK=20,  
                      seed=5024)
```

```
> names(cluster_codes(sce_PBMC))
```

Plot UMAP with 20 metaclusters

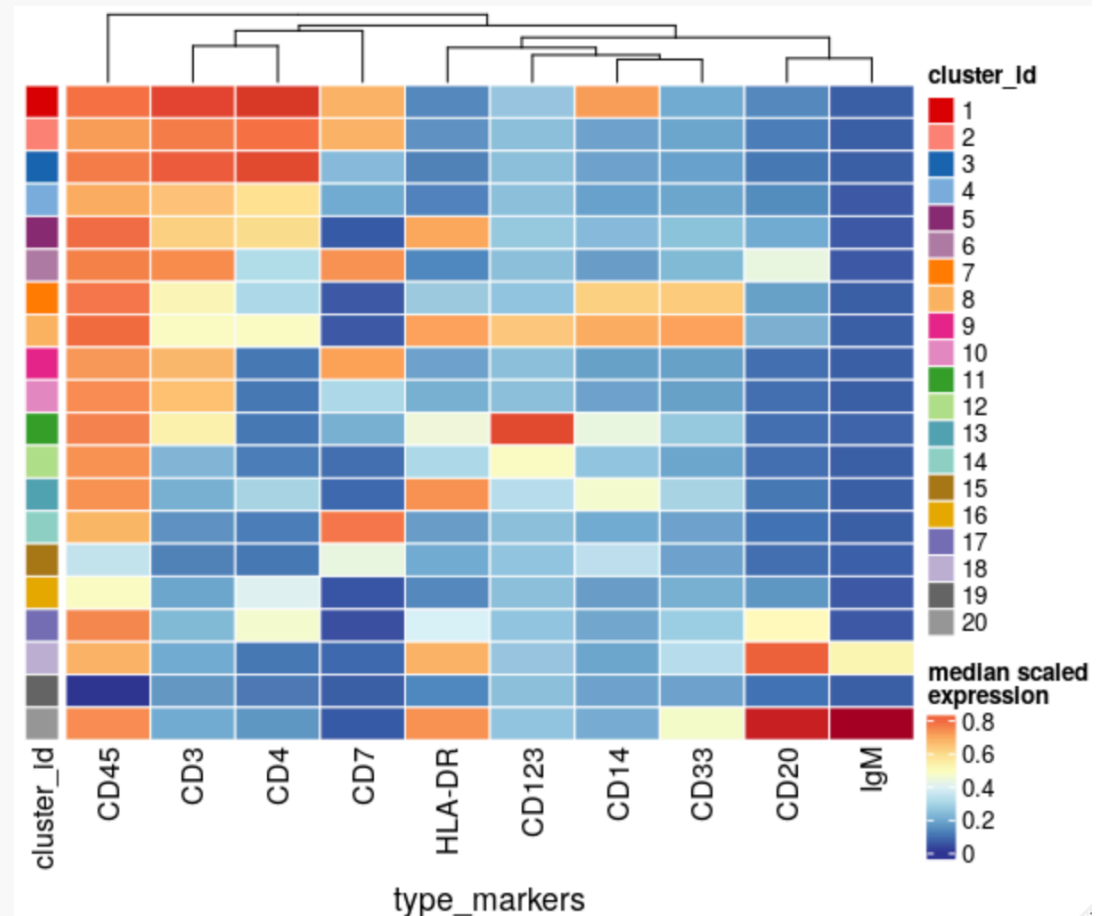
```
> plotDR(sce_PBMC, "UMAP",  
         color_by="meta20")
```



Clustering with CATALYST

Heatmap of the median expression per marker and metacluster

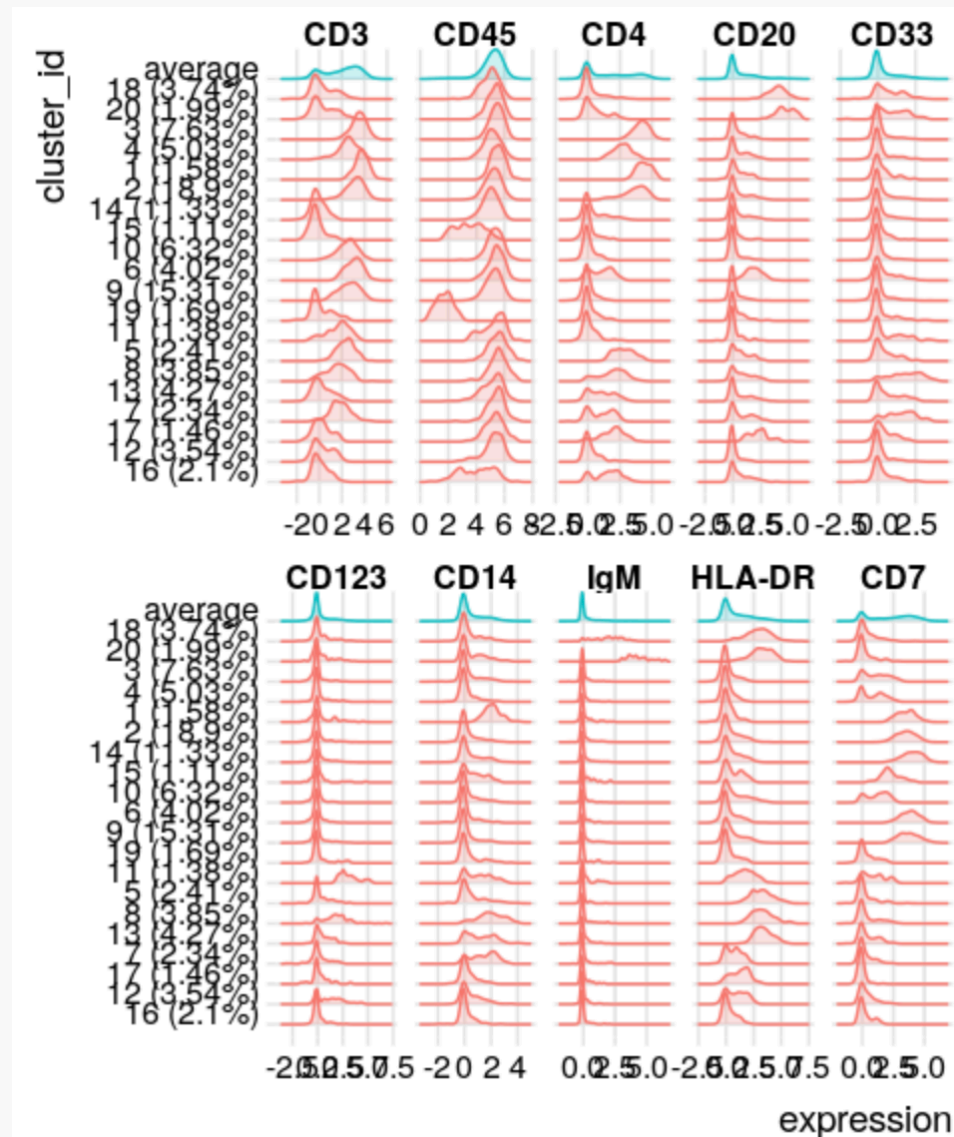
```
> plotExprHeatmap(sce_PBMC,  
  features = "type",  
  by = "cluster_id",  
  k = "meta20",  
  scale = "first",  
  q = 0.01,  
  perc = TRUE,  
  row_clust = FALSE,  
  col_dend = TRUE)
```



Clustering with CATALYST

Ridge plots of the expression per marker and metacluster

```
> plotClusterExprs(sce_PBMC,  
  k = "meta10",  
  features = "type")
```



Manual cluster merging (and renaming)

Create a 2 column data.frame containing old_cluster and new_cluster IDs

```
> merging_table <- data.frame(old_cluster = 1:20,  
                              new_cluster = c("B-cells IgM+", "surface-", "NK cells",  
                                              "CD8 T-cells", "B-cells IgM-", "monocytes",  
                                              "monocytes", "CD8 T-cells", "CD8 T-cells",  
                                              "monocytes", "monocytes", "CD4 T-cells",  
                                              "DC", "CD8 T-cells", "CD4 T-cells", "DC",  
                                              "CD4 T-cells", "CD4 T-cells", "CD4 T-cells",  
                                              "CD4 T-cells"))
```

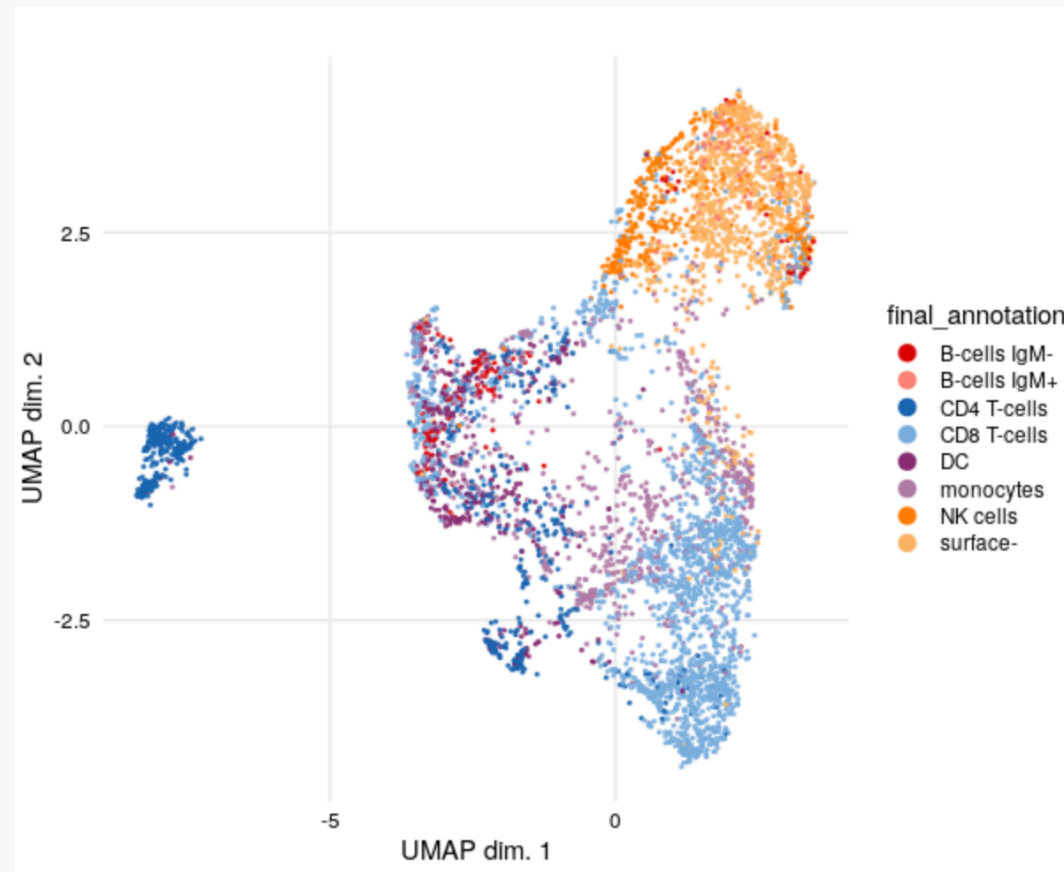
Merge / rename clusters

```
> sce_PBMC <- mergeClusters(sce_PBMC,  
                           k = "meta20",  
                           table = merging_table,  
                           id = "final_annotation")
```

Manual cluster merging and renaming

Plot UMAP with final annotation

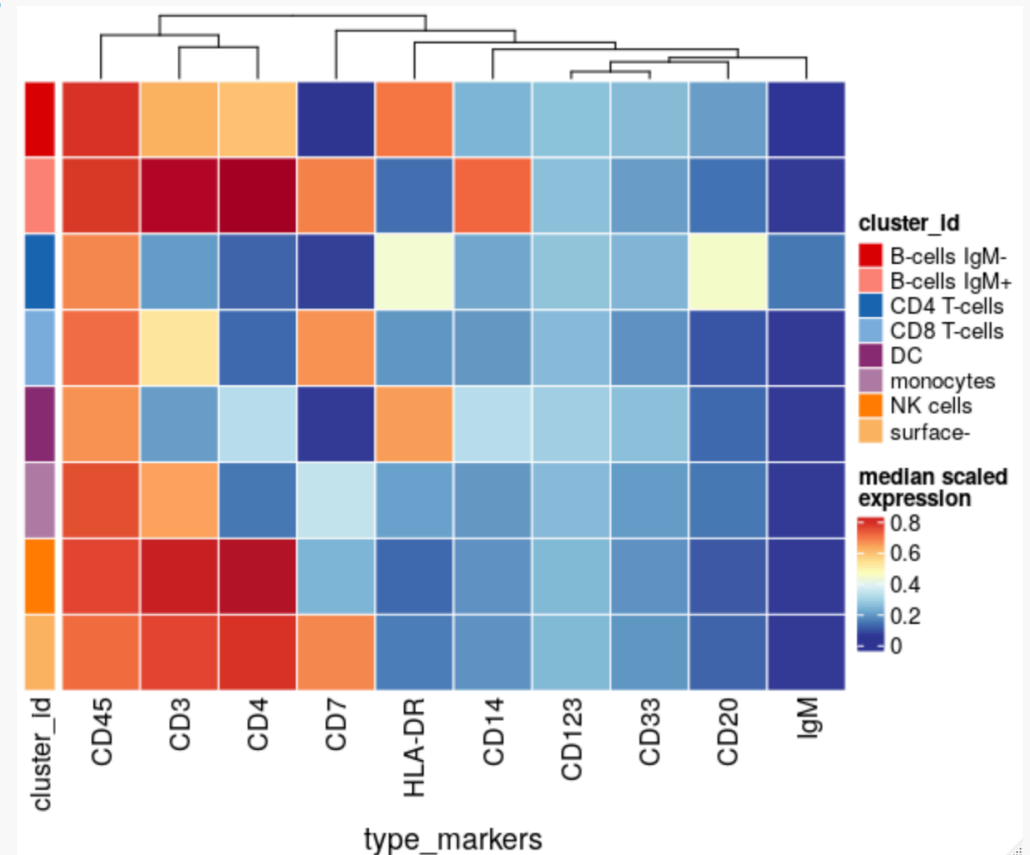
```
> plotDR(sce_PBMCM, "UMAP",  
         color_by="final_annotation")
```



Manual cluster merging and renaming

Heatmap of the median expression per marker and metacluster

```
> plotExprHeatmap(sce_PBMC,  
  features = "type",  
  by = "cluster_id",  
  k = "final_annotation",  
  scale = "first",  
  q = 0.01,  
  perc = TRUE,  
  row_clust = FALSE,  
  col_dend = TRUE)
```



Let's practice – 5

In this exercise we will apply the FlowSom method for unsupervised clustering of cells, followed by ConsensusClusterPlus metaclustering. We then check the expression of markers by metacluster. Finally, we will rename / merge the metaclusters to annotate major cell populations.

Create a new script in which you will

- 1) Load the sce object with UMAP from the previous exercise ("course_datasets/FR_FCM_Z3WR/sce_UMAP.RData")
- 2) Apply FlowSOM clustering + ConsensusClusterPlus metaclustering.
- 3) Plot a UMAP showing the location of metaclusters; marker expression heatmap and ridge plots. Use 8 metaclusters.
- 4) Rename / merge metaclusters as major cell populations according to the expression of markers.
- 5) Plot a UMAP showing the major cell populations.



Differential testing

Differential testing with *diffcyt*

<https://bioconductor.org/packages/release/bioc/html/diffcyt.html>



- Statistical methods for differential discovery analyses in high-dimensional cytometry data (including flow cytometry and mass cytometry)
- Based on:
 - High-resolution clustering
 - Empirical Bayes moderated tests adapted from transcriptomics
- The input to the *diffcyt* pipeline can either be raw data loaded from *.fcs* files, or a pre-prepared SingleCellExperiment object from **CATALYST**

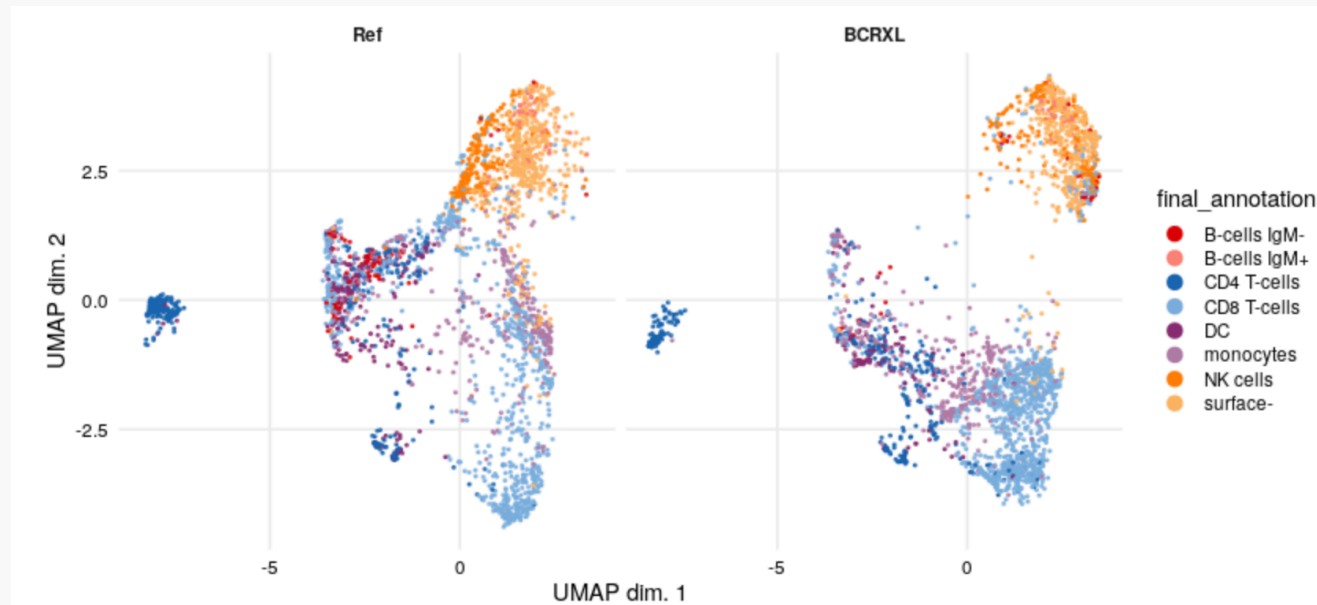
Differential testing with *diffcyt*

Example of a pre-processed dataset

- 8 PBMCs samples from 4 patients
- 2 conditions: before (REF) and upon BCR/FcR-XL stimulation (BCRXL) with B cell receptor/Fc receptor crosslinking for 30'
- Expression of 10 cell surface and 14 signaling markers

```
> load("./datasets/DA_example_sce_PBMC.RData")
```

```
> plotDR(sce_PBMC, color_by = "final_annotation", facet_by = "condition")
```



Differential testing with *diffcyt*

- The **design matrix** describes the experimental design
- Flexible experimental designs are possible, including blocking (e.g. batch effects or paired designs) and continuous covariates.

```
> design <- createDesignMatrix(ei(sce_PBMC),  
                               cols_design = "condition")
```



Accessor for the experimental information

```
> ei(sce_PBMC)
```

	sample_id	condition	patient_id	n_cells
1	BCRXL1	BCRXL	Patient1	528
2	Ref1	Ref	Patient1	881
3	BCRXL2	BCRXL	Patient2	665
4	Ref2	Ref	Patient2	438
5	BCRXL3	BCRXL	Patient3	563
6	Ref3	Ref	Patient3	660
7	BCRXL4	BCRXL	Patient4	934
8	Ref4	Ref	Patient4	759

```
> design
```

	(Intercept)	conditionBCRXL
1	1	1
2	1	0
3	1	1
4	1	0
5	1	1
6	1	0
7	1	1
8	1	0


Differential testing with diffcyt

- The **contrast matrix** specifies the comparison of interest, i.e. the combination of model parameters assumed to equal zero under the null hypothesis

```
> contrast <- createContrast(c(0, 1))
```

```
> contrast
```

	[,1]
[1,]	0
[2,]	1



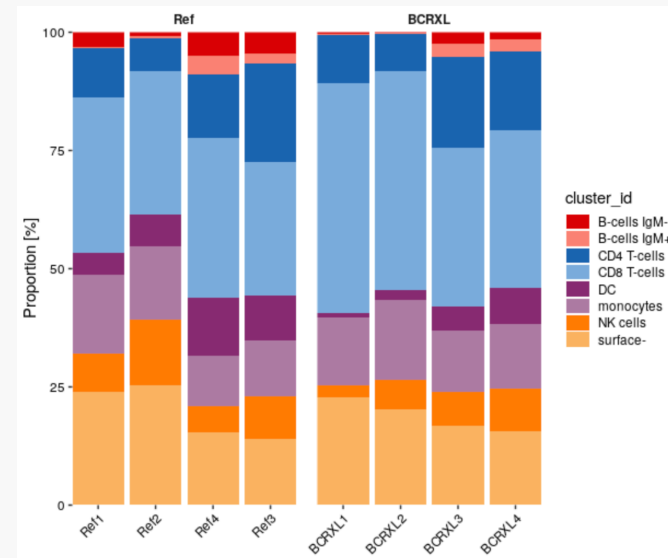
Vector of zeros and a single entry equal to one, corresponding to the columns of the design matrix.

Test whether a single parameter is equal to zero.

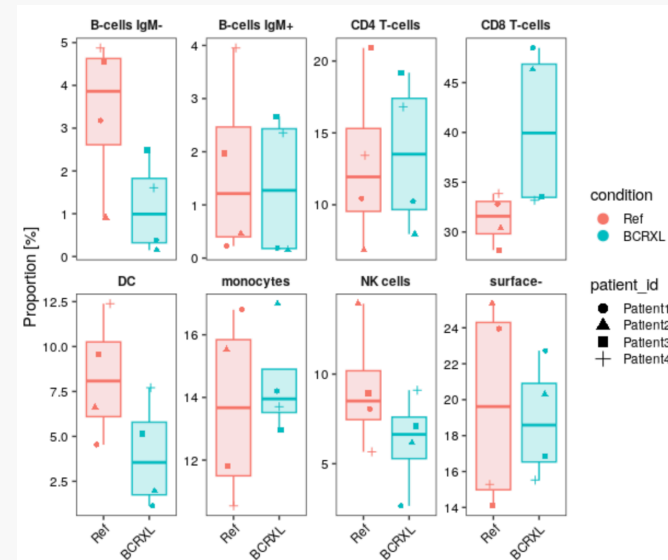
Differential abundance (DA) analysis

Plot relative population abundances

```
> plotAbundances(sce_PBMC,  
  k = "final_annotation",  
  by = "sample_id",  
  group_by = "condition")
```




```
> plotAbundances(sce_PBMC,  
  k = "final_annotation",  
  by = "cluster_id",  
  group_by = "condition",  
  shape_by = "patient_id")
```



Differential abundance analysis

```
> res_DA <- diffcyt(sce_PBMBC,  
  clustering_to_use = "final_annotation",  
  analysis_type = "DA",  
  method_DA = "diffcyt-DA-edgeR",  
  design = design,  
  contrast = contrast)
```



Methods for DA: functions from
the [edgeR](#) package and [limma](#)
packages

Differential abundance analysis

Extract results table

```
> tbl_DA <- rowData(res_DA$res)
> tbl_DA
```

```
DataFrame with 8 rows and 6 columns
      cluster_id      logFC      logCPM      LR      p_val      p_adj
      <factor> <numeric> <numeric> <numeric> <numeric> <numeric>
B-cells IgM- B-cells IgM- -1.5382727  14.6562 3.8721267 0.0490943 0.247282
B-cells IgM+ B-cells IgM+ -0.2978413  14.1418 0.0802018 0.7770241 0.881058
CD4 T-cells  CD4 T-cells  0.0665281  17.0386 0.0223887 0.8810577 0.881058
CD8 T-cells  CD8 T-cells  0.3635269  18.4547 3.0665023 0.0799213 0.247282
DC           DC           -1.0357401  15.9682 2.8263188 0.0927307 0.247282
monocytes   monocytes    0.0843930  17.1233 0.1242922 0.7244250 0.881058
NK cells    NK cells     -0.5245515  16.2760 1.4026678 0.2362774 0.472555
surface-    surface-     -0.0597664  17.5676 0.0425953 0.8364893 0.881058
```


Differential abundance analysis

Plot results

```
> plotDiffHeatmap(sce_PBMC,  
tbl_DA,  
fdr = 0.01,  
lfc = 1,  
top_n = 20,  
all = TRUE,  
normalize = TRUE,  
col_anno = "condition")
```

Options

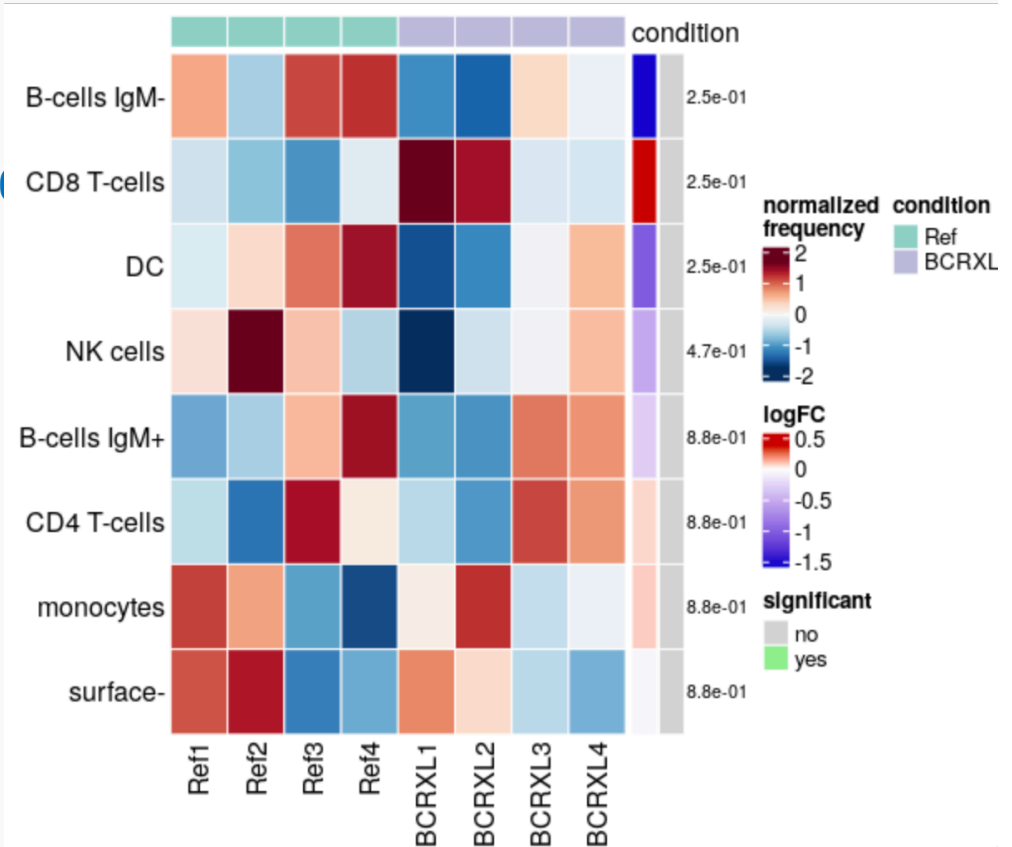
fdr: threshold on adjusted p-values *below* which to keep a result

lfc: threshold on absolute logFCs *above* which to keep a result

top_n: number of top clusters to display

all: if all top_n results should be displayed

normalize: if frequencies should be scaled



Let's practice – 6

In this exercise we will test if cell populations have significantly different abundances between two time points (D14 compared to D0)

Create a new script in which you will

- 1) Load the sce object from the previous exercise ("sce_annotated.RData").
- 2) Plot relative cell population abundances by sample and time point.
- 3) Set up the design and contrast matrices.
- 4) Test for differences in abundances between D14 and D0.
- 5) View table of results

Differential state (DS) analysis

Differential expression of cell state markers within clusters

```
> res_DS <- diffcyt(sce_PBMC,  
  clustering_to_use = "final_annotation",  
  analysis_type = "DS",  
  method_DS = "diffcyt-DS-limma",  
  design = design,  
  contrast = contrast)
```



Methods for DS: uses the [limma](#) package

Differential state analysis

Extract results table

```
> tbl_DS <- rowData(res_DS$res)
> tbl_DS
```

DataFrame with 112 rows and 9 columns

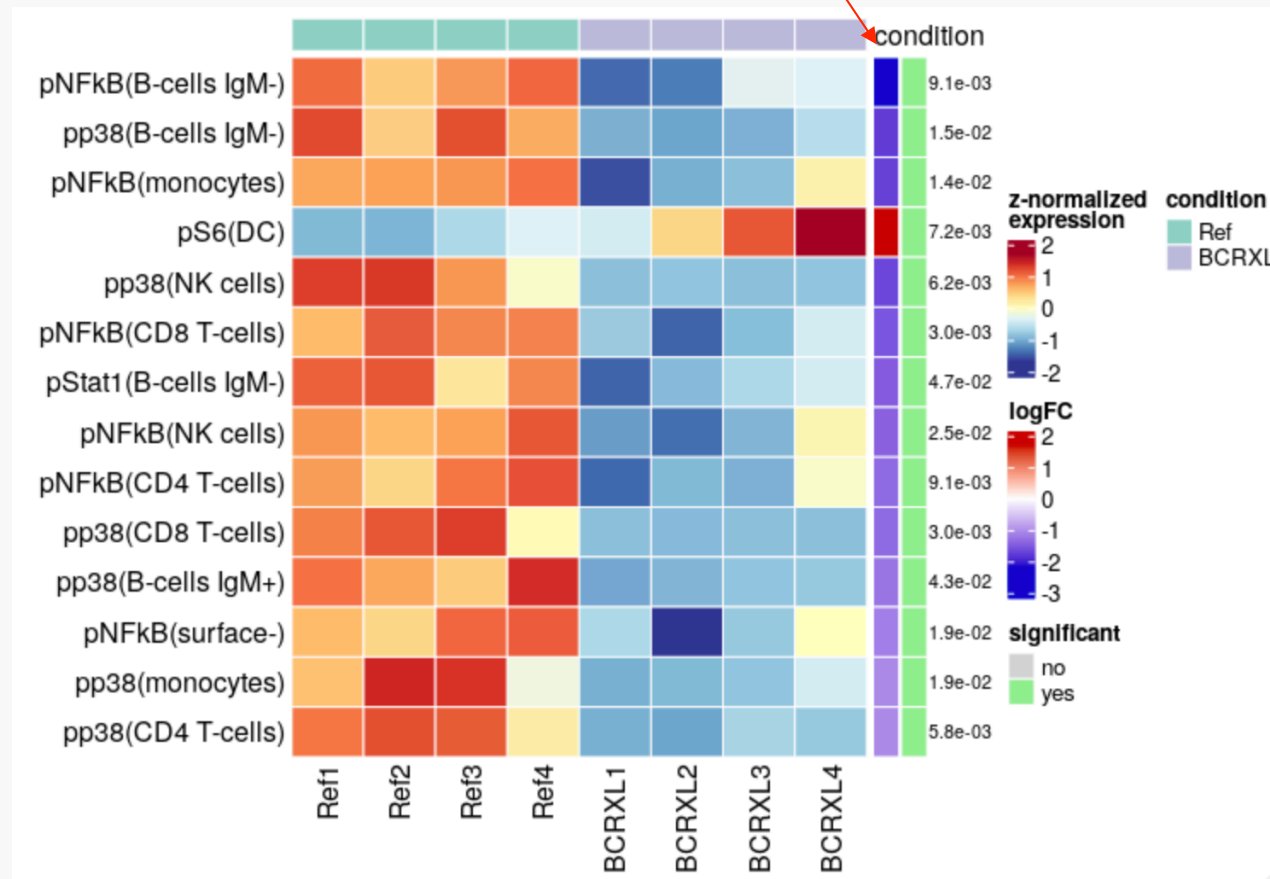
	cluster_id	marker_id	ID	logFC	AveExpr	t	p_val	p_adj	B
	<factor>	<factor>	<character>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
1	B-cells IgM-	pNFkB	B-cells IgM-	-2.21572	2.31036	-5.11031	7.46675e-04	0.00905117	-0.751794
2	B-cells IgM+	pNFkB	B-cells IgM+	-0.83796	2.03630	-1.97808	8.09829e-02	0.21111216	-4.029912
3	CD4 T-cells	pNFkB	CD4 T-cells	-1.36276	1.51960	-5.04956	8.08140e-04	0.00905117	-0.183536
4	CD8 T-cells	pNFkB	CD8 T-cells	-1.54604	1.88391	-7.04762	7.75061e-05	0.00301564	1.988404
5	DC	pNFkB	DC	-1.15503	2.25842	-3.06479	1.42961e-02	0.05718454	-2.778137
...
108	CD8 T-cells	pS6	CD8 T-cells	0.1247776	0.0676208	1.83165	0.101976280	0.24541379	-5.34420
109	DC	pS6	DC	1.7136612	0.7474637	5.40764	0.000510802	0.00715123	0.21896
110	monocytes	pS6	monocytes	0.2544115	0.1385676	1.44022	0.185399053	0.37079811	-5.44622
111	NK cells	pS6	NK cells	0.1159033	-0.0217907	3.80112	0.004624448	0.02524031	-1.77275
112	surface-	pS6	surface-	0.0748209	0.0237420	1.37291	0.204707087	0.38211990	-5.66771

Differential state analysis

Plot results

Sorts results by absolute value of logFoldChange

```
> plotDiffHeatmap(sce_PBMC, tbl_DS, fdr = 0.05, sort_by = "lfc", col_anno = "condition")
```



Let's practice – 7

In this exercise we will test if markers were differentially expressed between two time points (D14 compared to D0)

Create a new script in which you will

- 1) Load the sce object from the previous exercise ("sce_annotated.RData").
- 2) Set up the design and contrast matrices.
- 3) Test for differences in marker expression between D14 and D0.
- 4) View table of results

Thank you for your attention!

<https://agora-cancer.ch/scientific-platforms/translational-data-science-facility/>

Any questions? Contact us !

tds-facility@sib.swiss