Lecture 4

Identifying cell populations

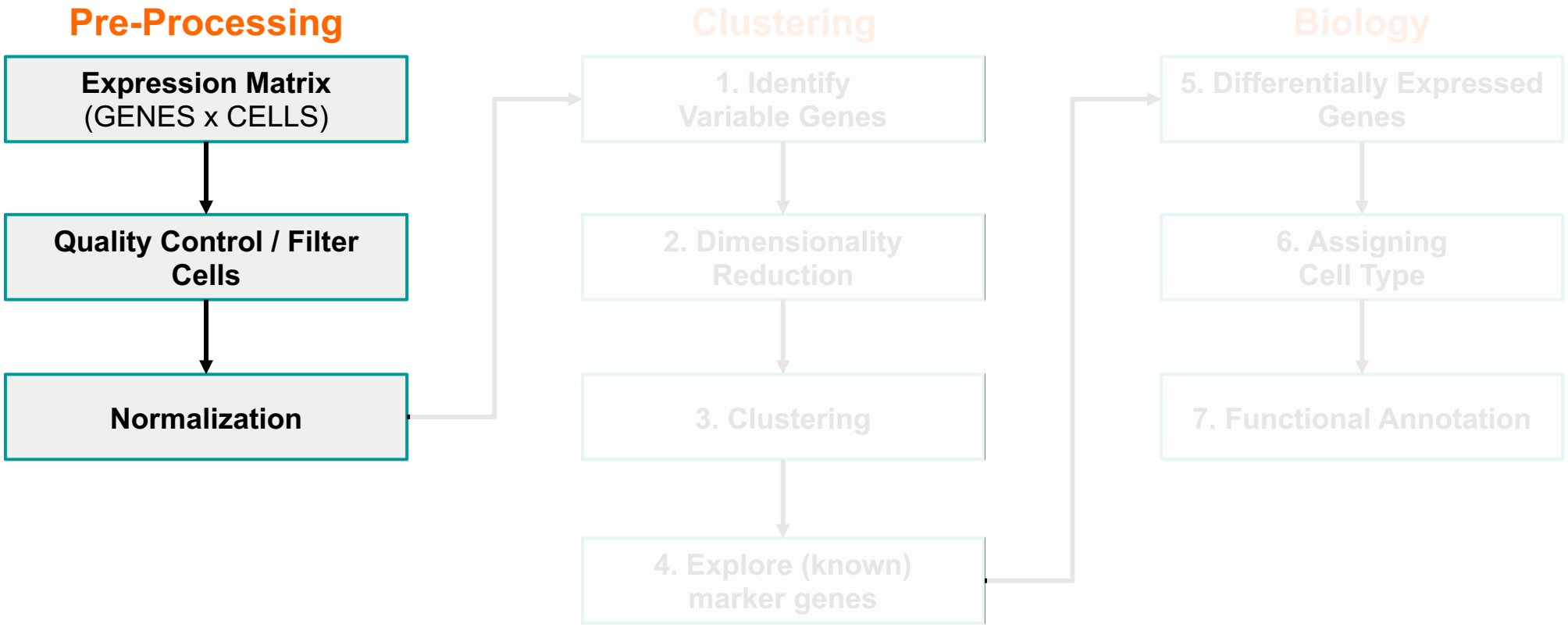
Physalia course 2025

—

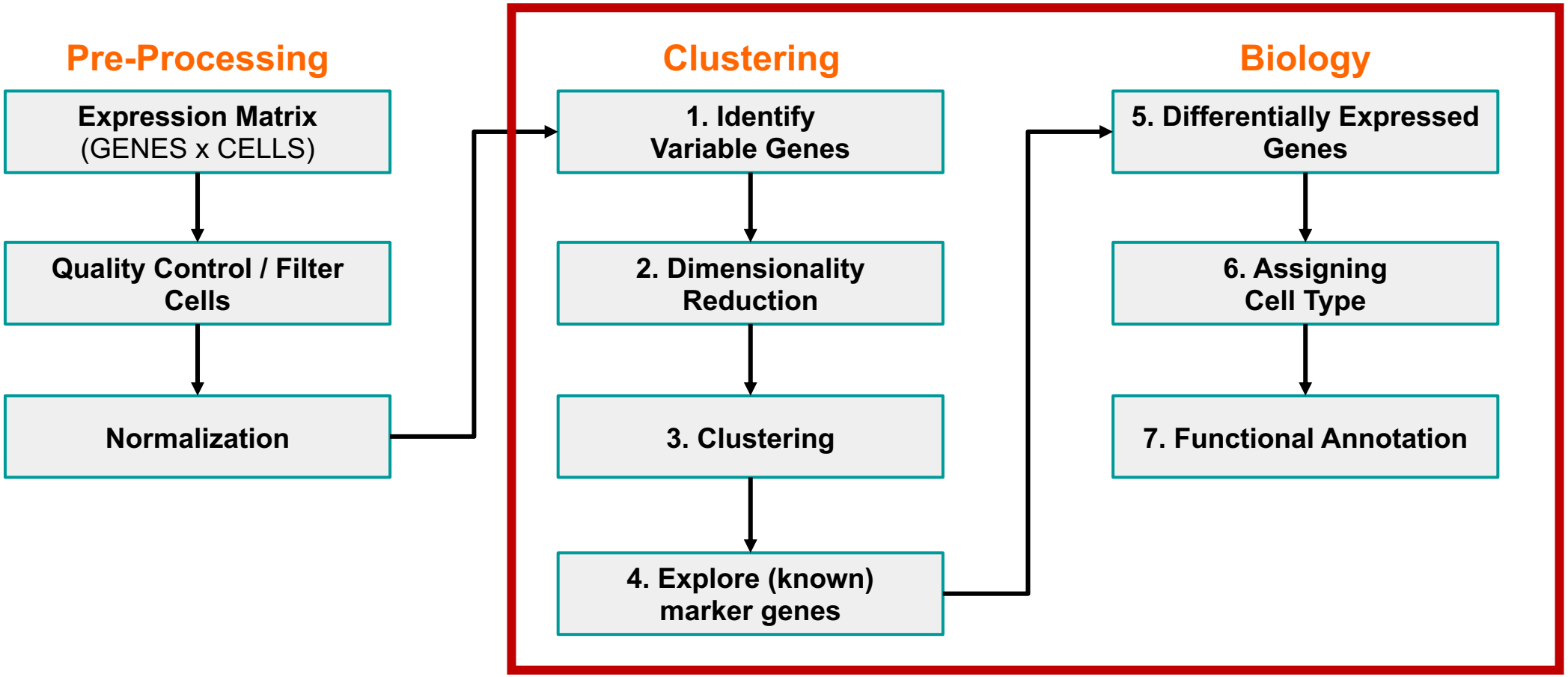
Single-cell RNA-seq with R/Bioconductor

Instructors: Orr Ashenberg, Jacques Serizay, Fabrício Almeida-Silva

Analysis workflow



Analysis workflow



Making sense of variation

For something to be informative, it needs to exhibit variation

Making sense of variation

For something to be informative, it needs to exhibit variation

QUESTION: Can I group these emoticons by their facial hair style?

Making sense of variation

For something to be informative, it needs to exhibit variation

QUESTION: Can I group these emoticons by their facial hair style?

Not everything that exhibits variation is informative. It depends on the question!

QUESTION: Which ones of these emoticons are smiling?

Making sense of variation

Cells are in ~20,000 dimensional space (one dimension per gene)

- Many genes are lowly detected or noisy measurements
- Variable genes contain the biological signal we are interested in

The most used approach is to find genes that have **high variance**

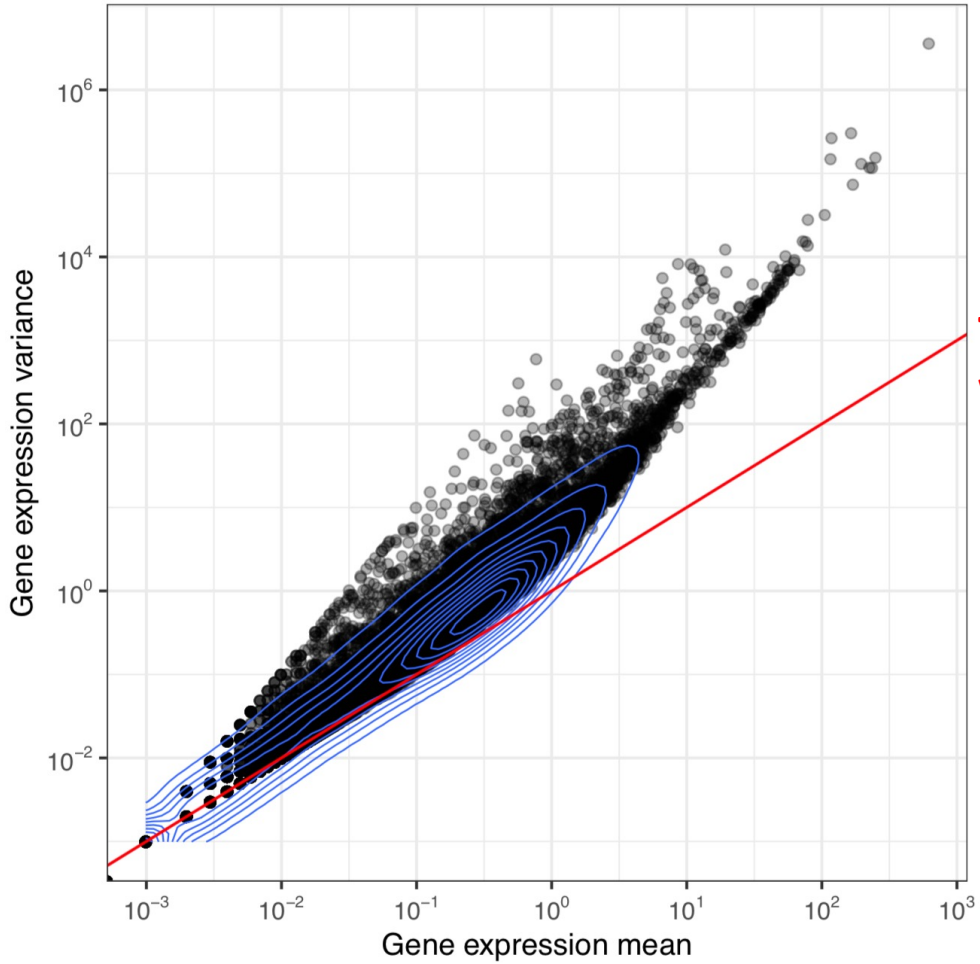
Sources of variations in gene expression

In (single-cell) RNA-seq, the variance in gene expression depends on the level of expression of this gene: A gene highly expressed will have a high variation of expression across cells.

Poisson distribution is generally viewed as a good way to model counts, such as those in a scRNAseq count matrix.

According to Poisson distribution, the variance of counts is equal to the mean of the counts.

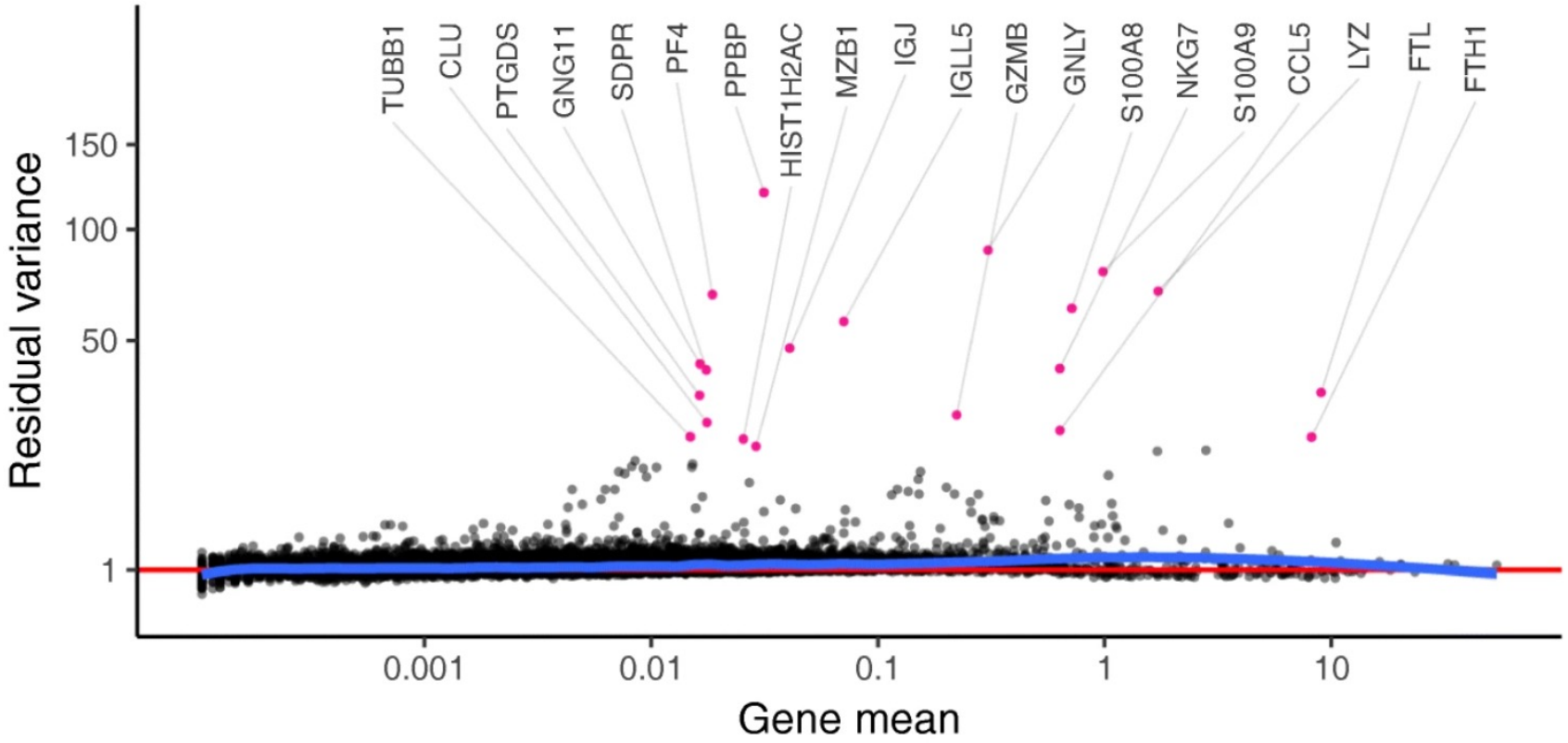
Sources of variations in gene expression



This is the *identity* curve
variance = mean

Variance residuals

The computed variance residuals can be used to identify highly variable genes.



Hafemeister & Sajita, Genome Biology 2019

Making sense of variation

The computed variance residuals can be used to identify highly variable genes.

→ Either taking the first *n* genes (ranked by decreasing residual variance)

e.g. 200-2000 genes

Making sense of variation

The computed variance residuals can be used to identify highly variable genes.

→ Either taking the first *n* genes (ranked by decreasing residual variance)

e.g. 200-2000 genes

→ Or by taking the first % genes (ranked by decreasing residual variance)

e.g. 2-10% genes

Making sense of variation

The computed variance residuals can be used to identify highly variable genes.

→ Either taking the first *n* genes (ranked by decreasing residual variance)

e.g. 200-2000 genes

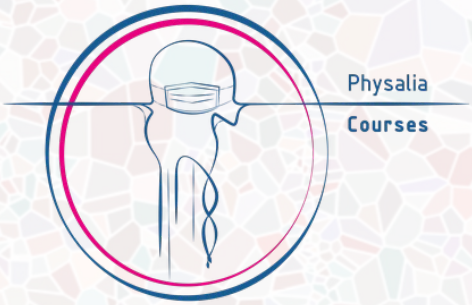
→ Or by taking the first % genes (ranked by decreasing residual variance)

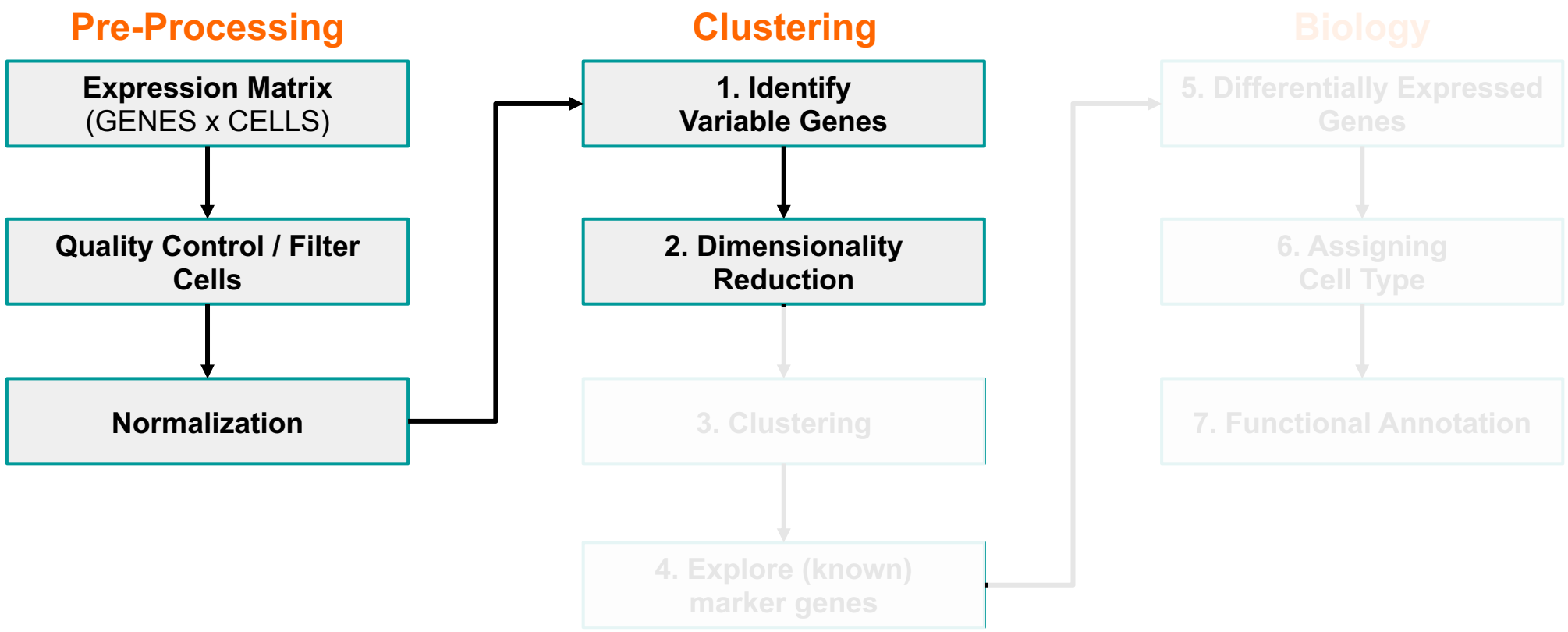
e.g. 2-10% genes

→ Or by taking all genes with a residual variance > threshold

e.g. genes with residual variance > 0.02

Analysis workflow





High-dimensional data can be difficult to interpret.

One approach to simplification is to **assume that the data of interest lies within lower-dimensional space**. If the data of interest is of low enough dimension, the data can be visualised in the low-dimensional space.

- A scRNA seq starts with many measurements (features, genes).
- We want to reduce it to fewer informative dimensions.
- We have starting doing this by using only highly variable genes.
- We can further reduce dimension with linear or non-linear approaches.

High-dimensional data can be difficult to interpret.

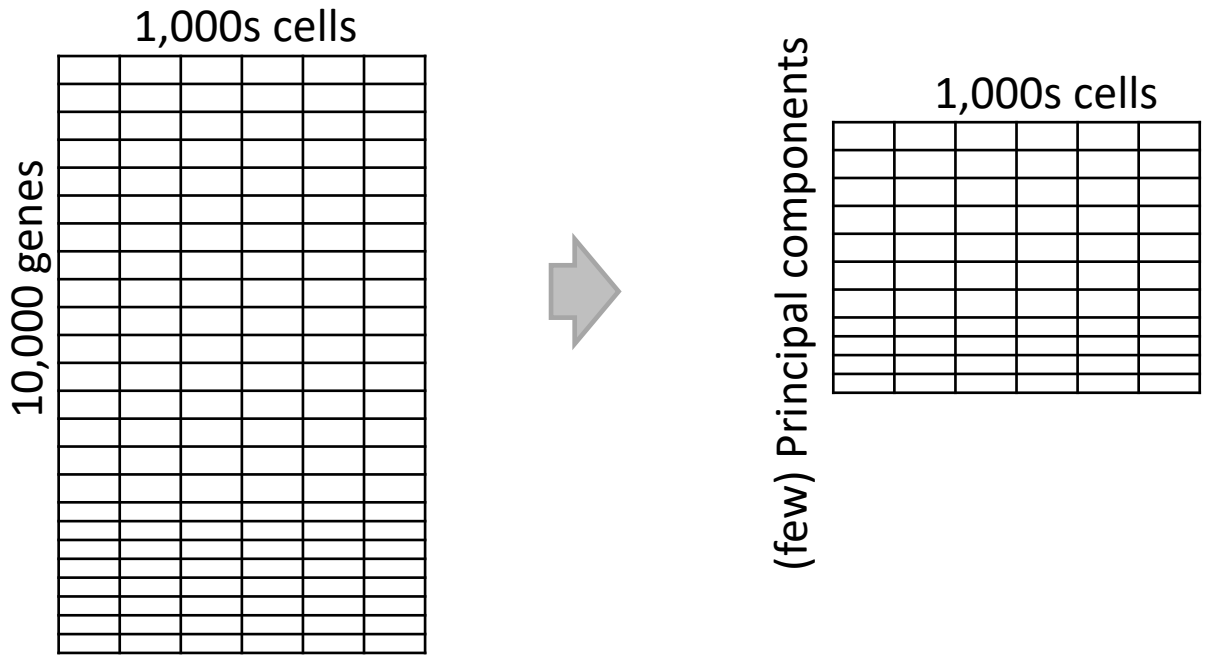
One approach to simplification is to **assume that the data of interest lies within lower-dimensional space**. If the data of interest is of low enough dimension, the data can be visualised in the low-dimensional space.

Common Techniques

- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Multidimensional Scaling (MDS)
- Non-negative Matrix Factorization (NMF)
- Probabilistic Modeling (e.g. Latent Dirichlet Allocation - LDA)

Principal Component Analysis

- PCA is a dimensionality reduction method that transforms a **set of features** into a set of **linearly uncorrelated variables** called principal components

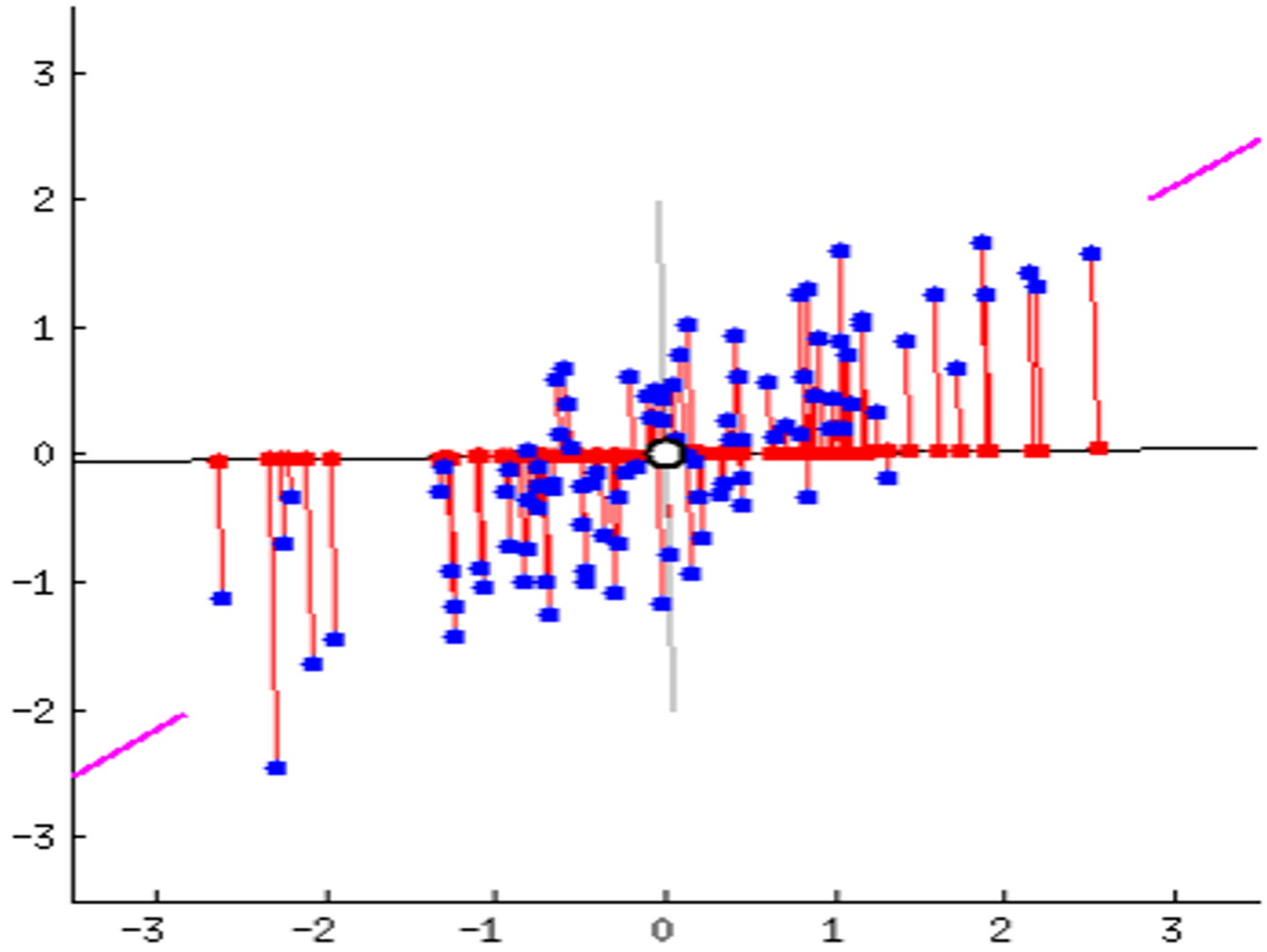


https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

Principal Component Analysis

- PCA is a dimensionality reduction method that transforms a **set of features** into a set of **linearly uncorrelated variables** called principal components

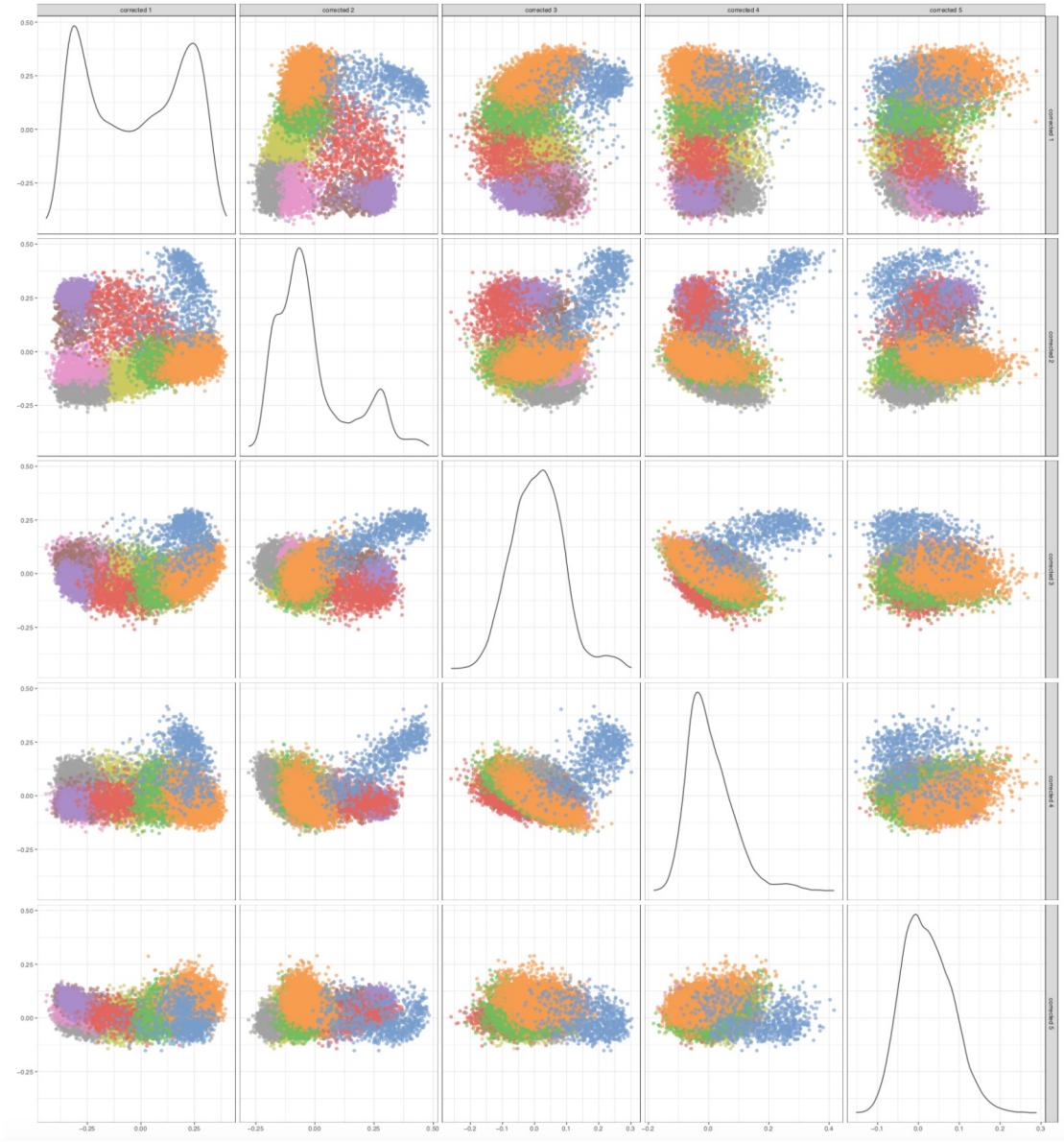
- The first principal component contains the most variance, and each component after contains as much variance while still being orthogonal to other components



https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

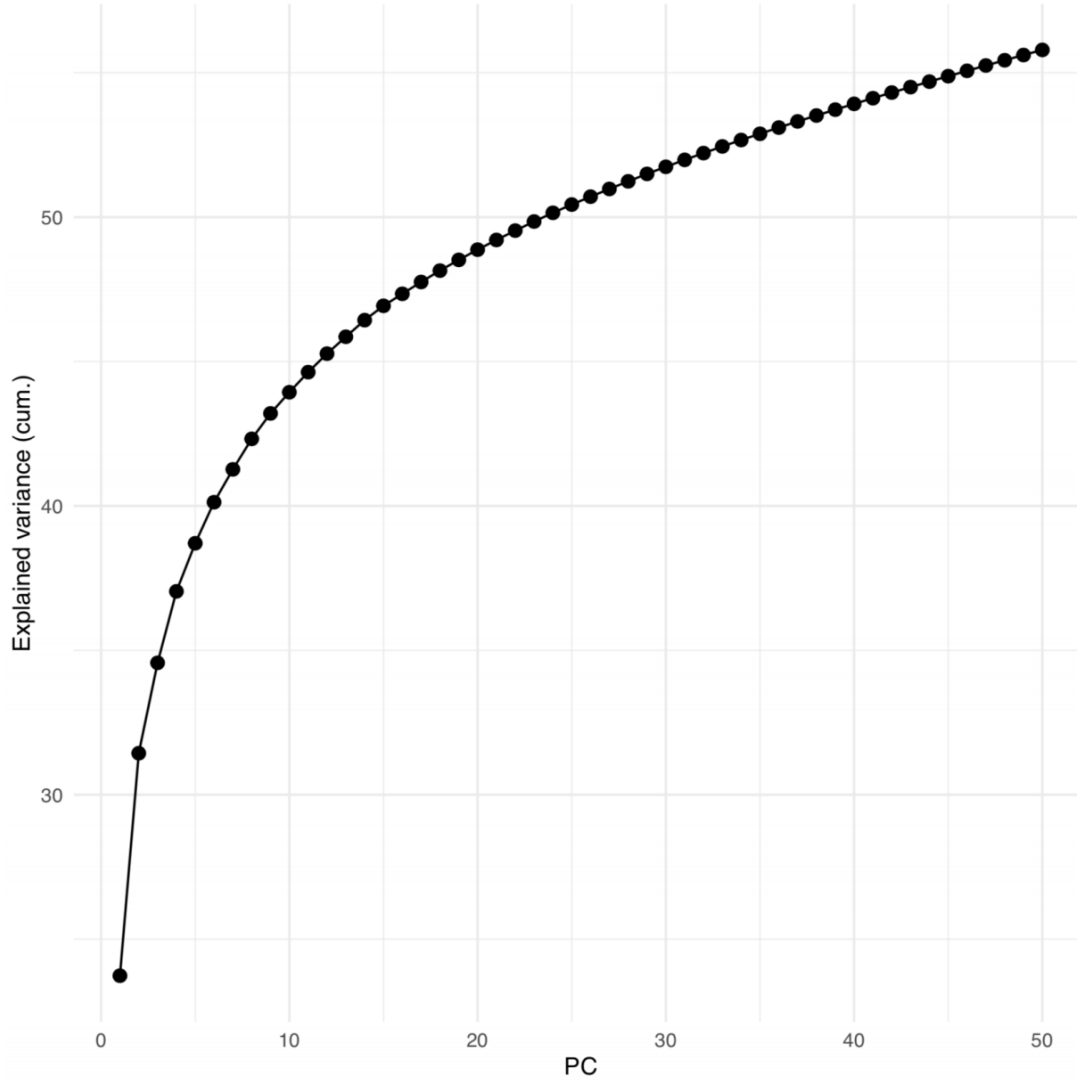
Principal Component Analysis: assessing lower dimensions

Notice how lower PCs look more and more "spherical" - this loss of structure indicates that the variation captured by these PCs mostly reflects noise.



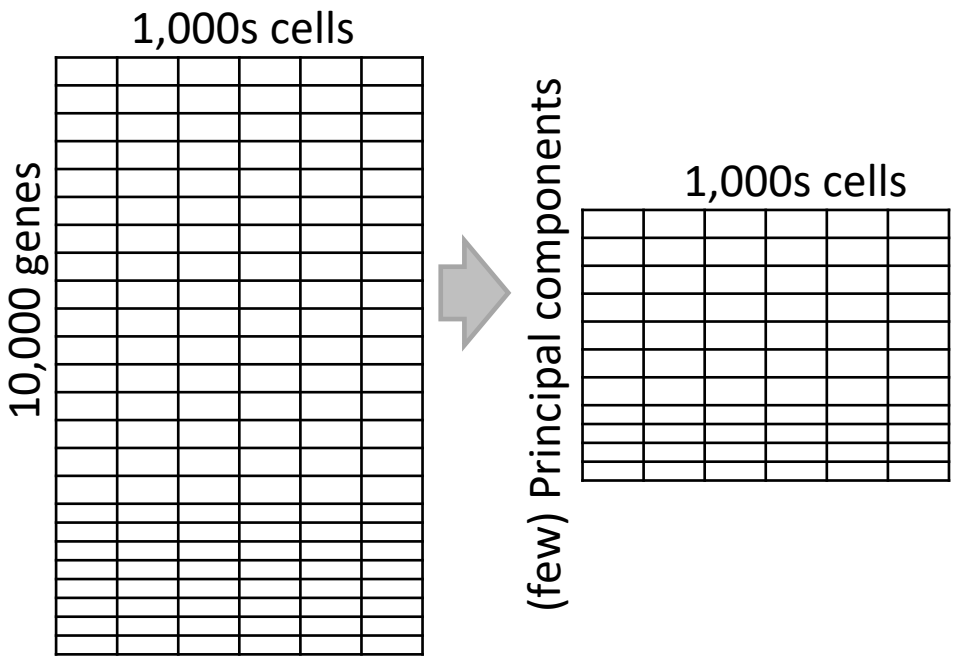
Principal Component Analysis: assessing lower dimensions

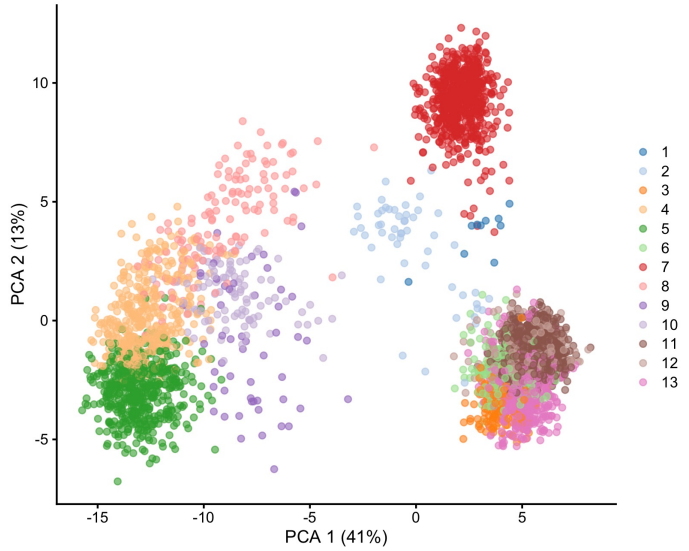
Notice how lower PCs look more and more "spherical" - this loss of structure indicates that the variation captured by these PCs mostly reflects noise.

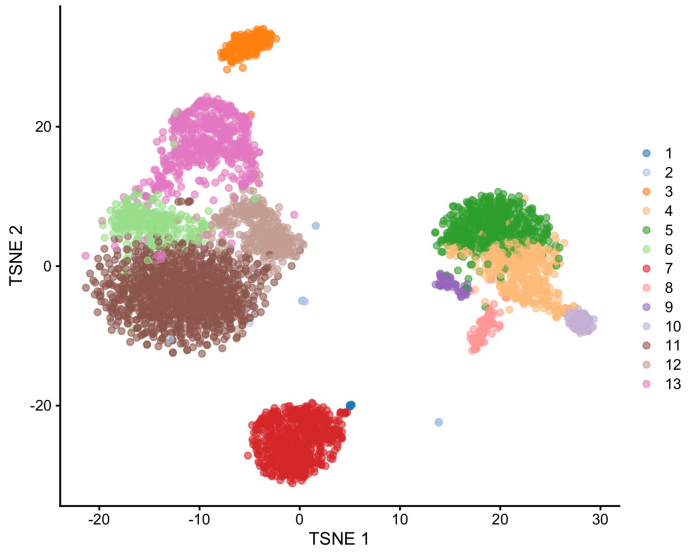


Going further: non-linear dimensional reductions

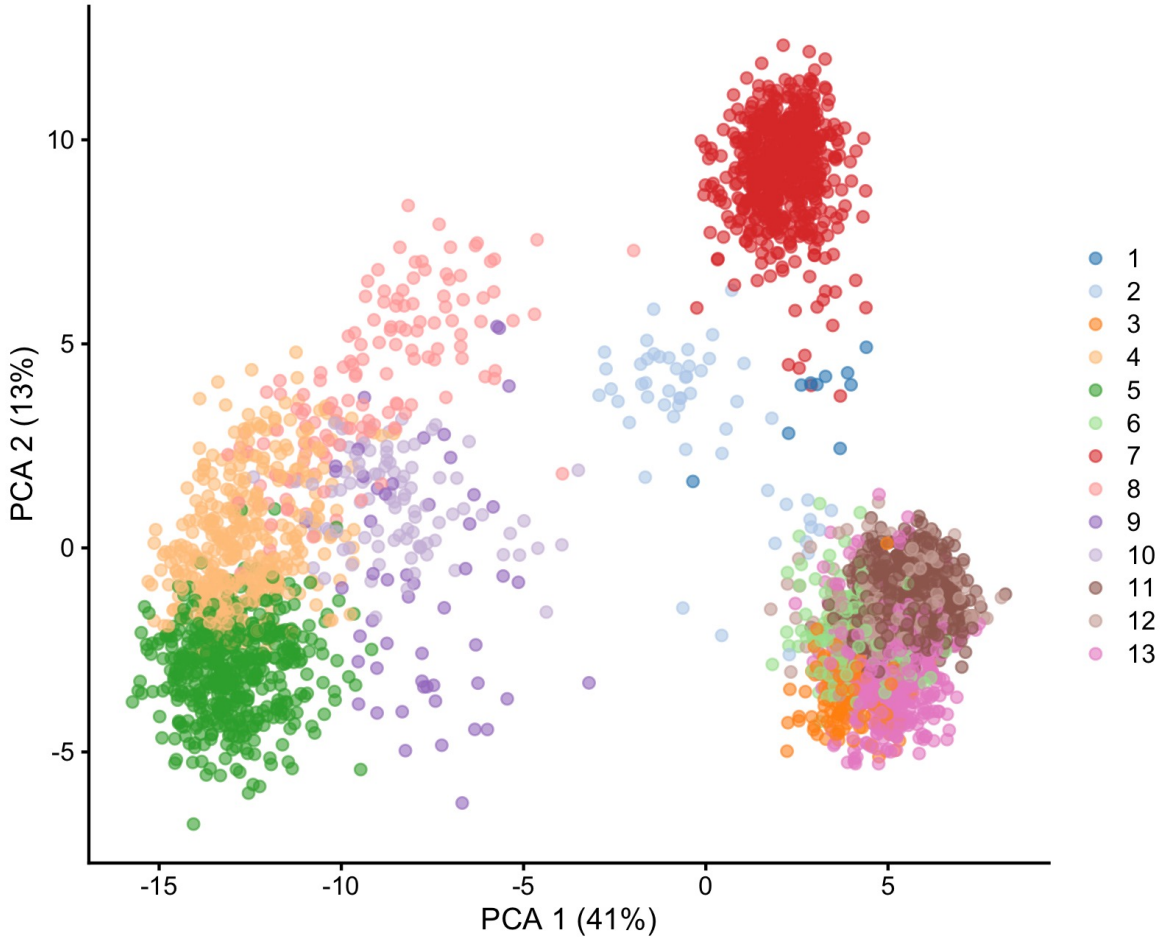
In a t-SNE projection, similar objects (cells) are modeled by nearby two-(three)dimensional points and dissimilar objects are modeled by distant points with high probability.

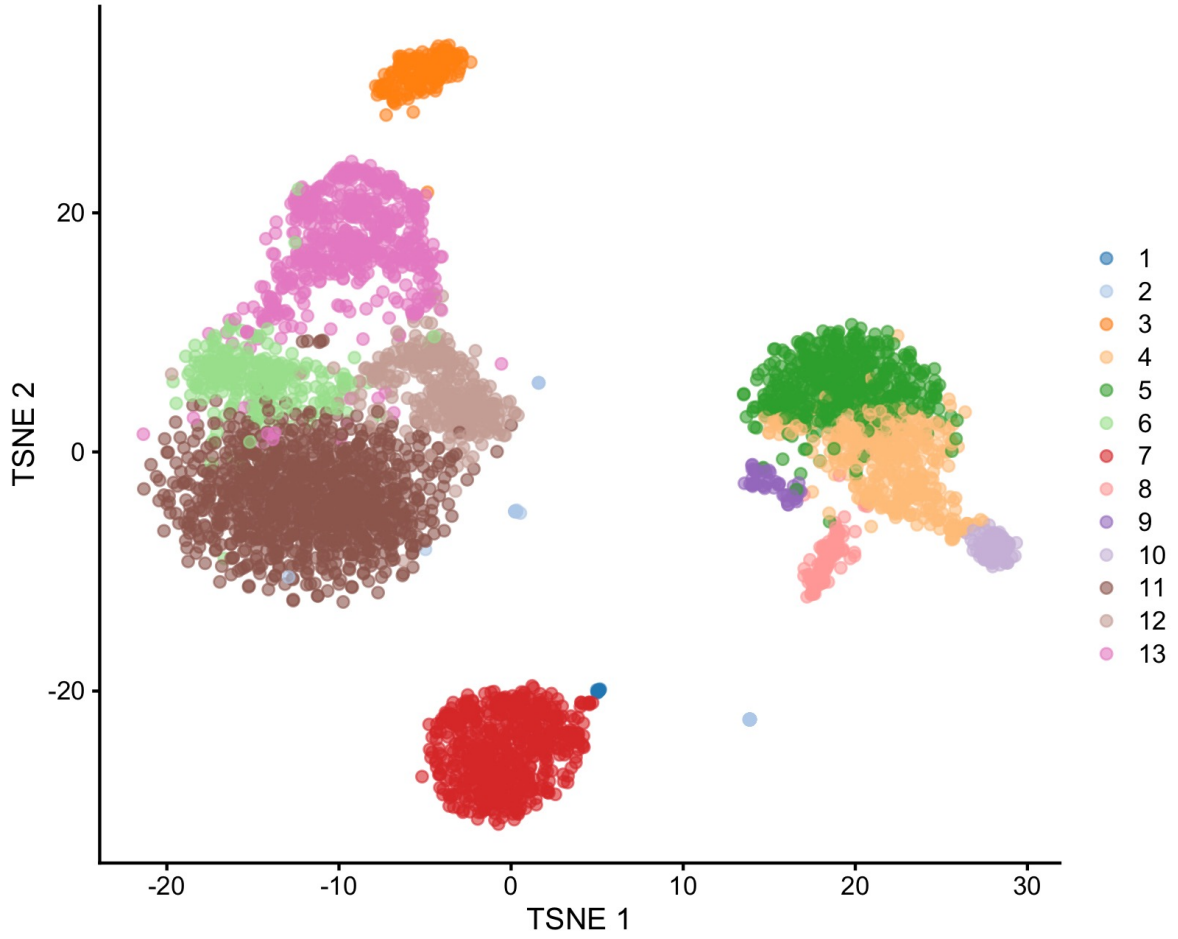






Caution with tSNE visualization





A great tSNE resource! https://distill.pub/2016/misread-tsne/

Other non-linear dimensional reduction approaches

- Force-directed graph embedding
- UMAP
- Diffusion Maps
- Non-negative Matrix Factorization
- Probabilistic (topic models/Latent Dirichlet Allocation (LDA))

BE AWARE!!

- **Some are linear, some other are not.**
- **While PCA is a general "one-size-fits-all" approach, others will yield more specific outputs, targeting a particular question.**

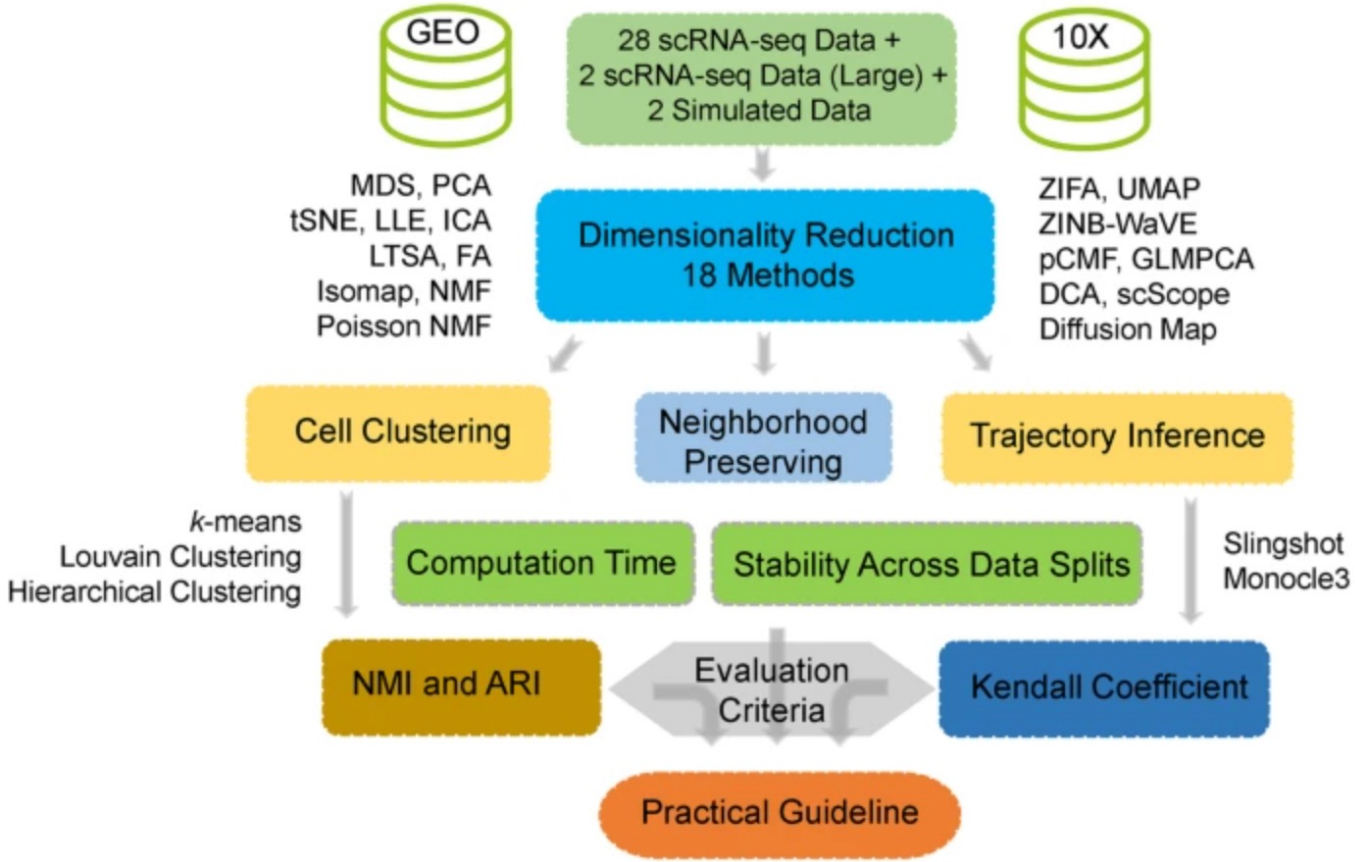
Other non-linear dimensional reduction approaches

Research | Open Access | Published: 10 December 2019

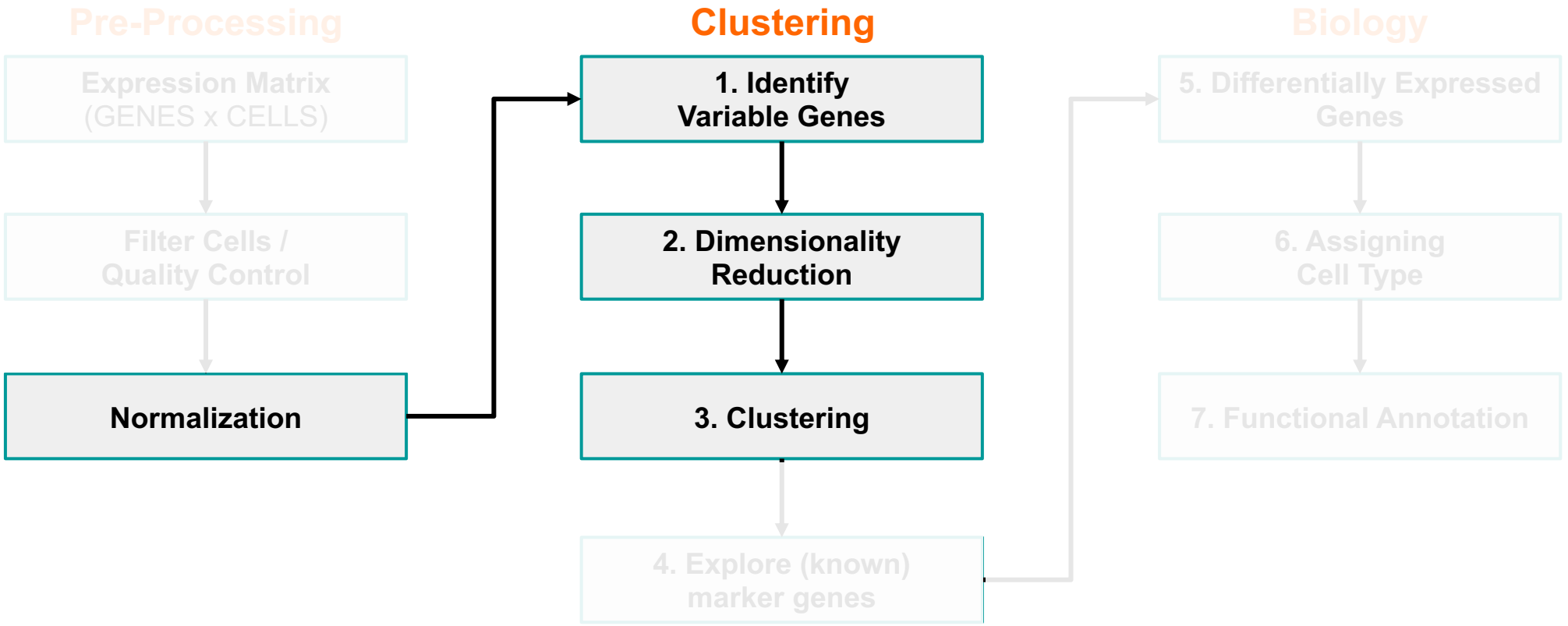
Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis

Shiquan Sun, Jiaqiang Zhu, Ying Ma & Xiang Zhou

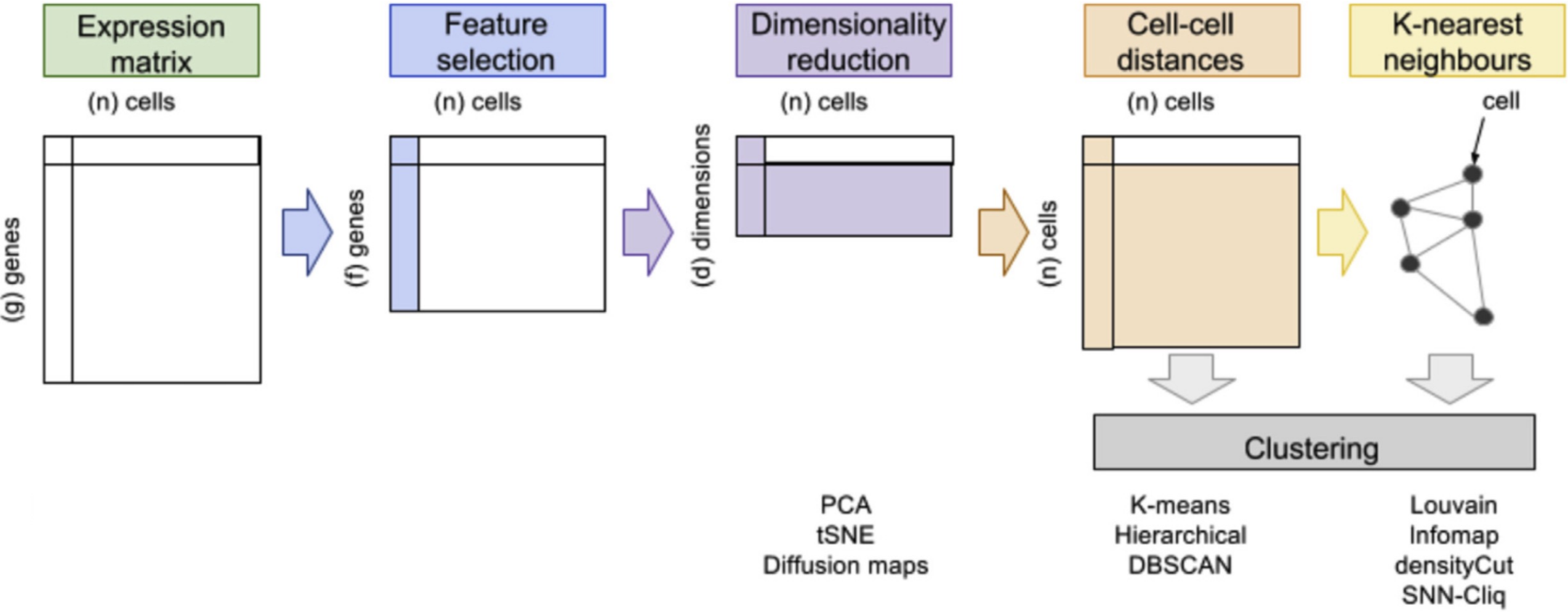
Genome Biology **20**, Article number: 269 (2019)



Analysis workflow

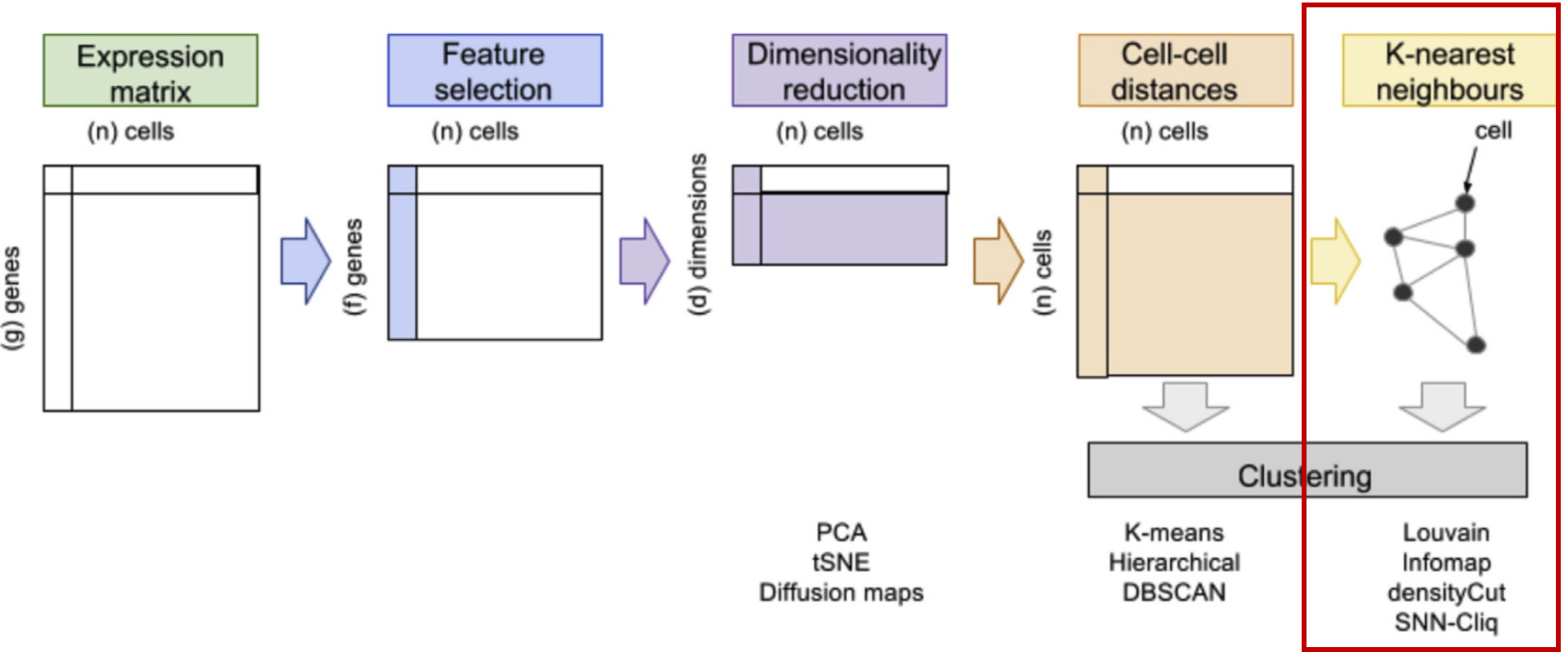


Different methodologies for clustering



Andrews et al., 2018

Graph-based clustering



Andrews et al., 2018

Why do we need graph-based clustering?

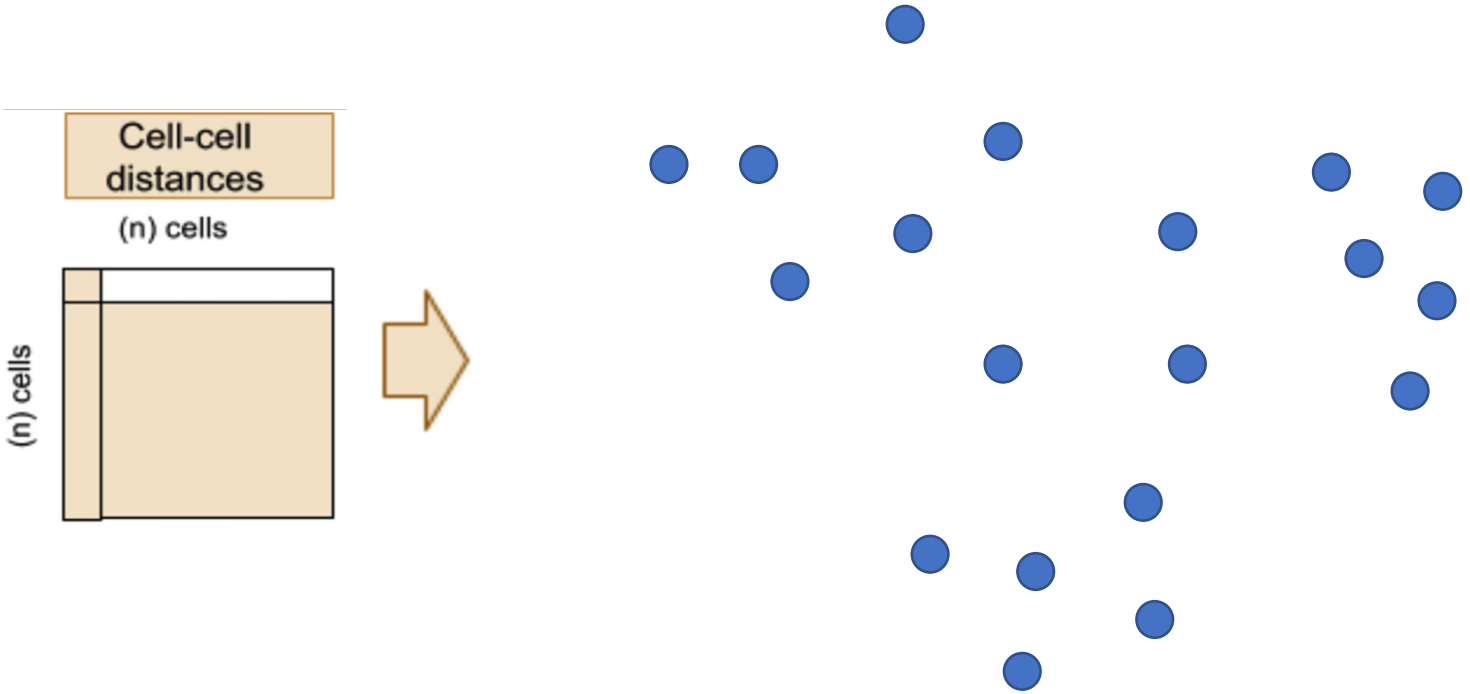
Curse of dimensionality:

"All data become sparse in high-dimensional space and therefore similarities measured by Euclidean distances etc are generally low between all objects."

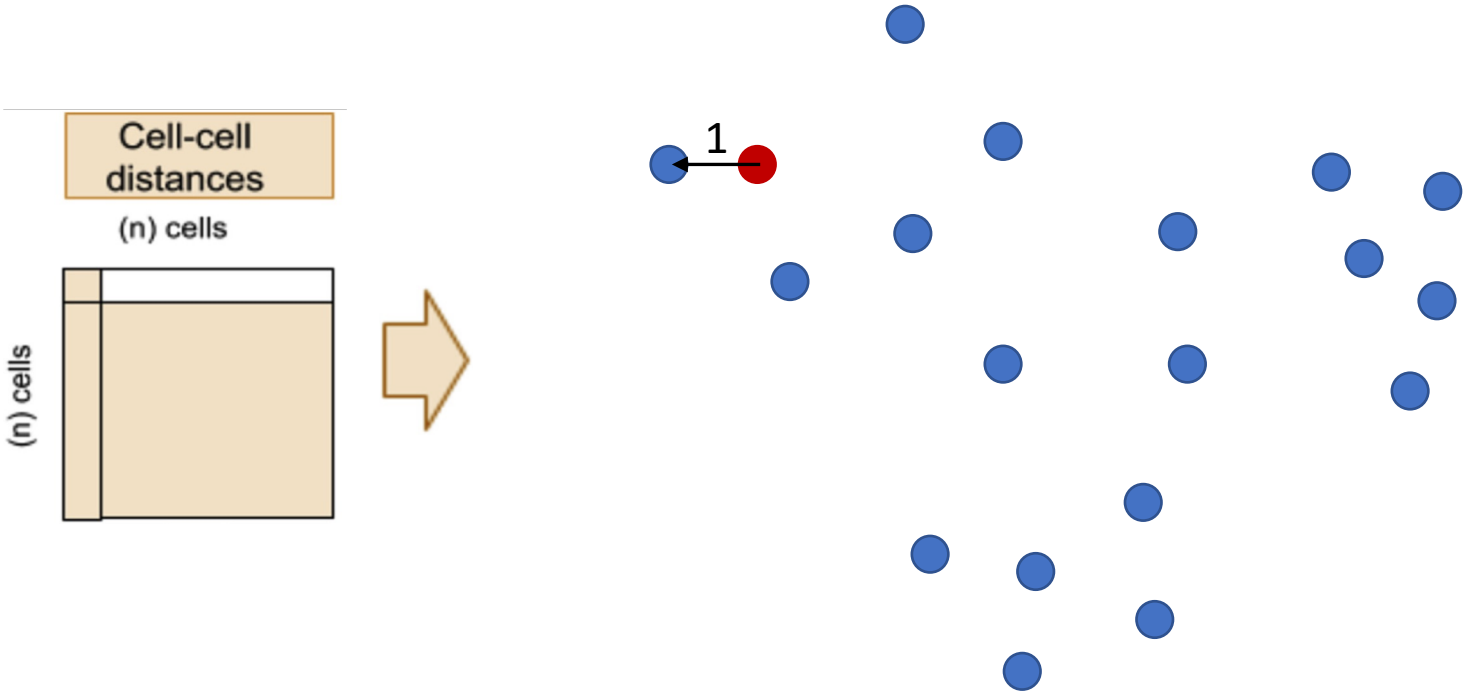
There is no point performing a hierarchical clustering of 10,000 cells if 90% of the pairwise distances are null !!

Andrews et al., 2018

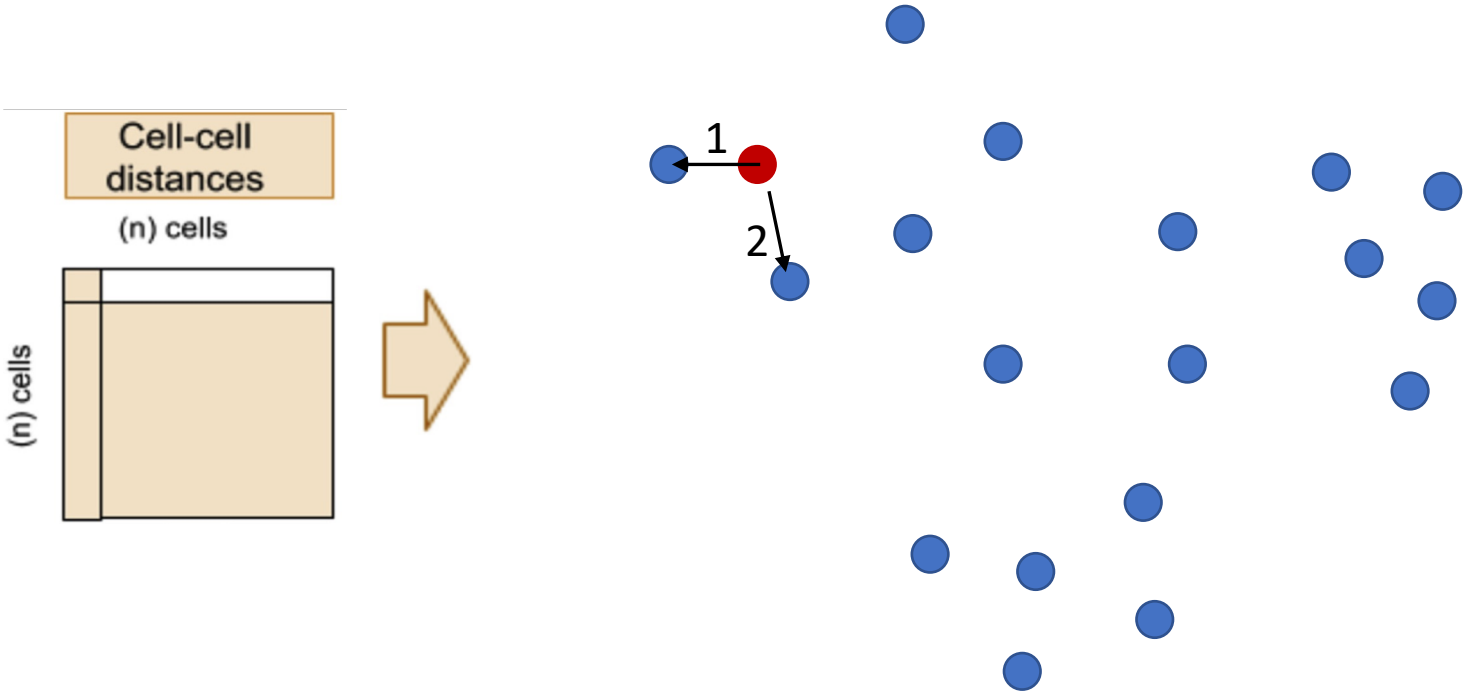
Building a k-Nearest Neighbors graph (with k = 4)



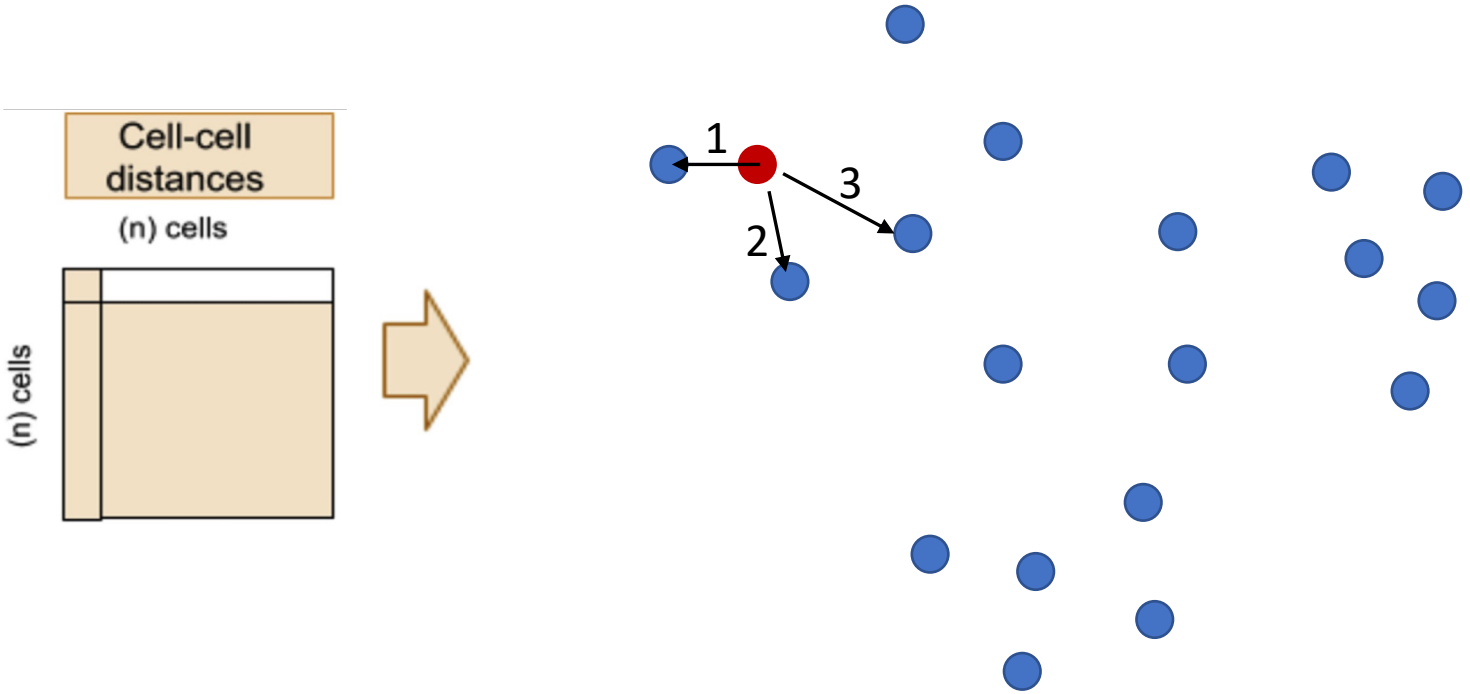
Building a k-Nearest Neighbors graph (with k = 4)



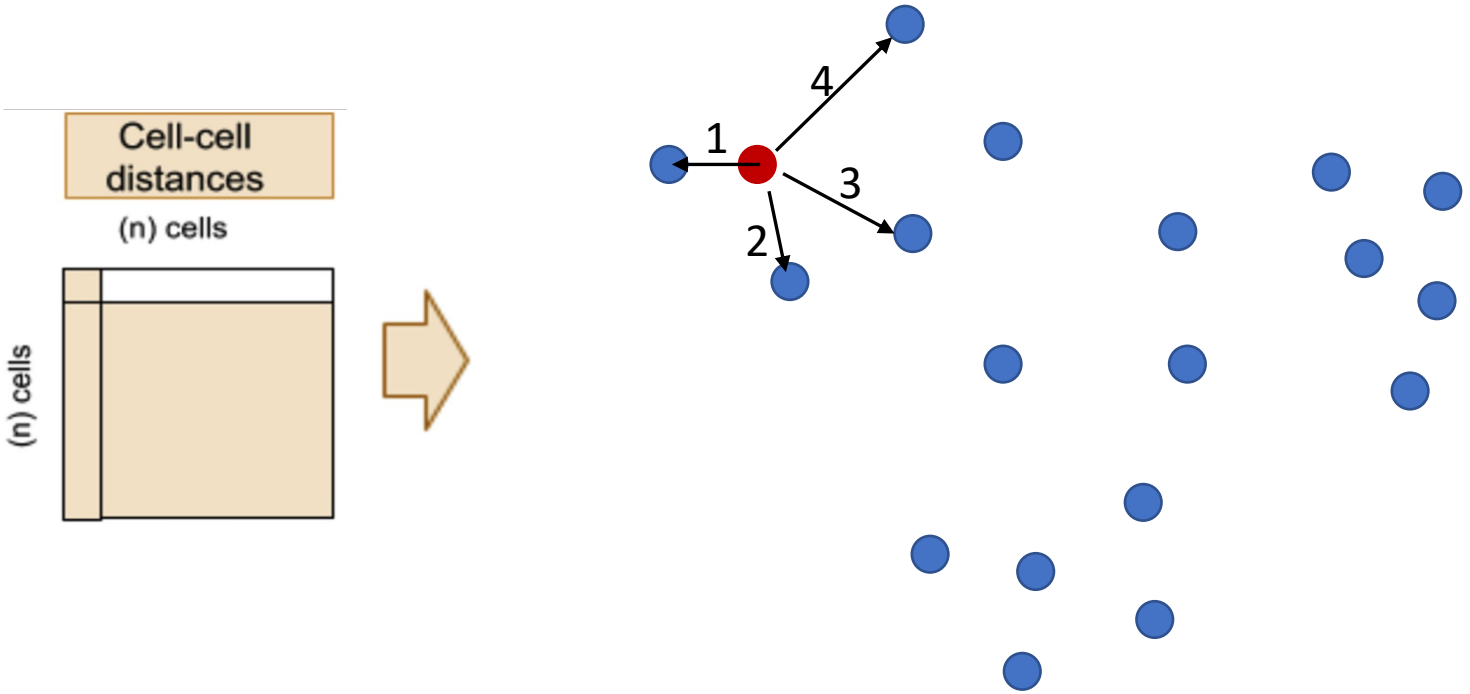
Building a k-Nearest Neighbors graph (with k = 4)



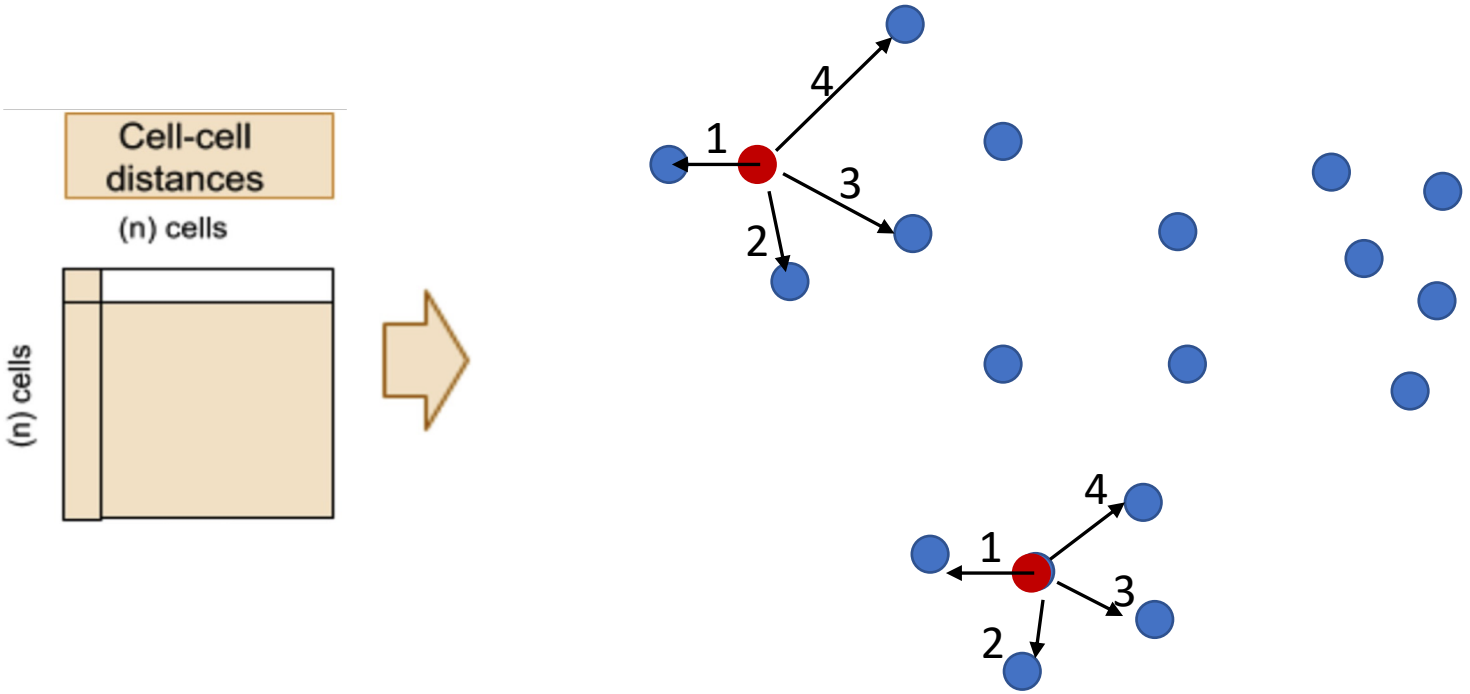
Building a k-Nearest Neighbors graph (with k = 4)



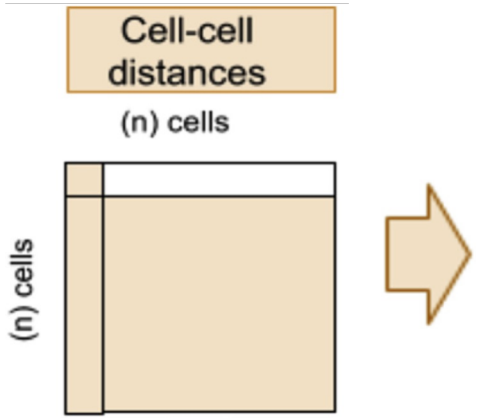
Building a k-Nearest Neighbors graph (with k = 4)

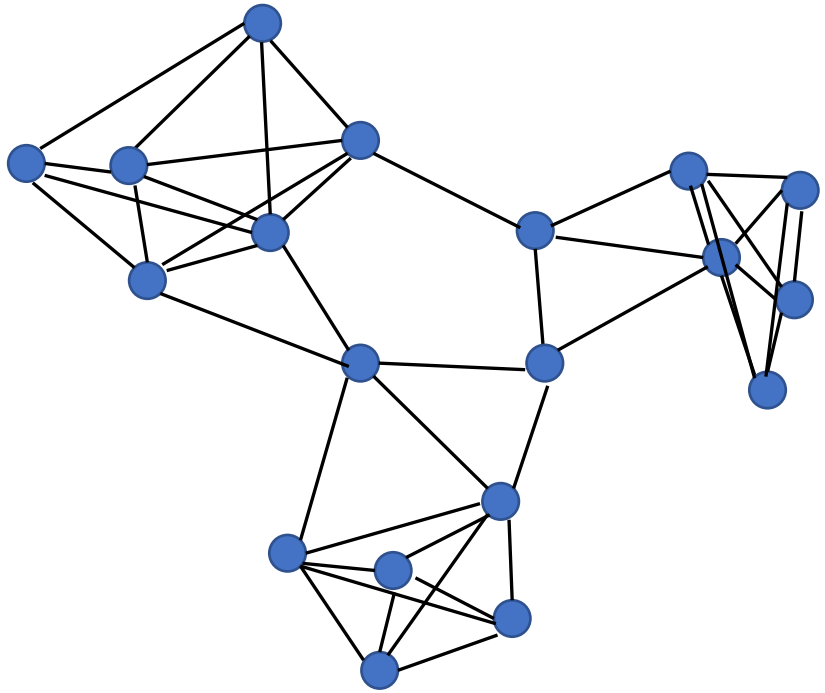


Building a k-Nearest Neighbors graph (with k = 4)

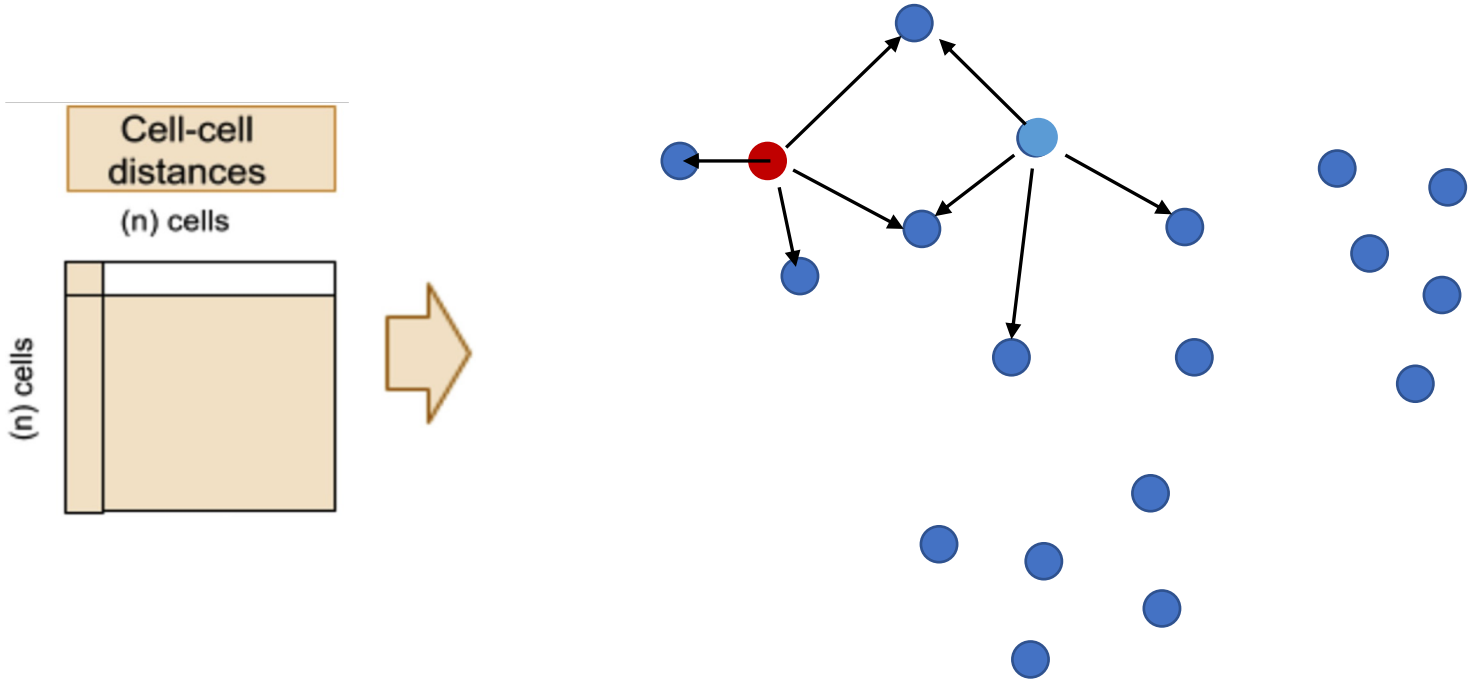


Building a k-Nearest Neighbors graph (with k = 4)



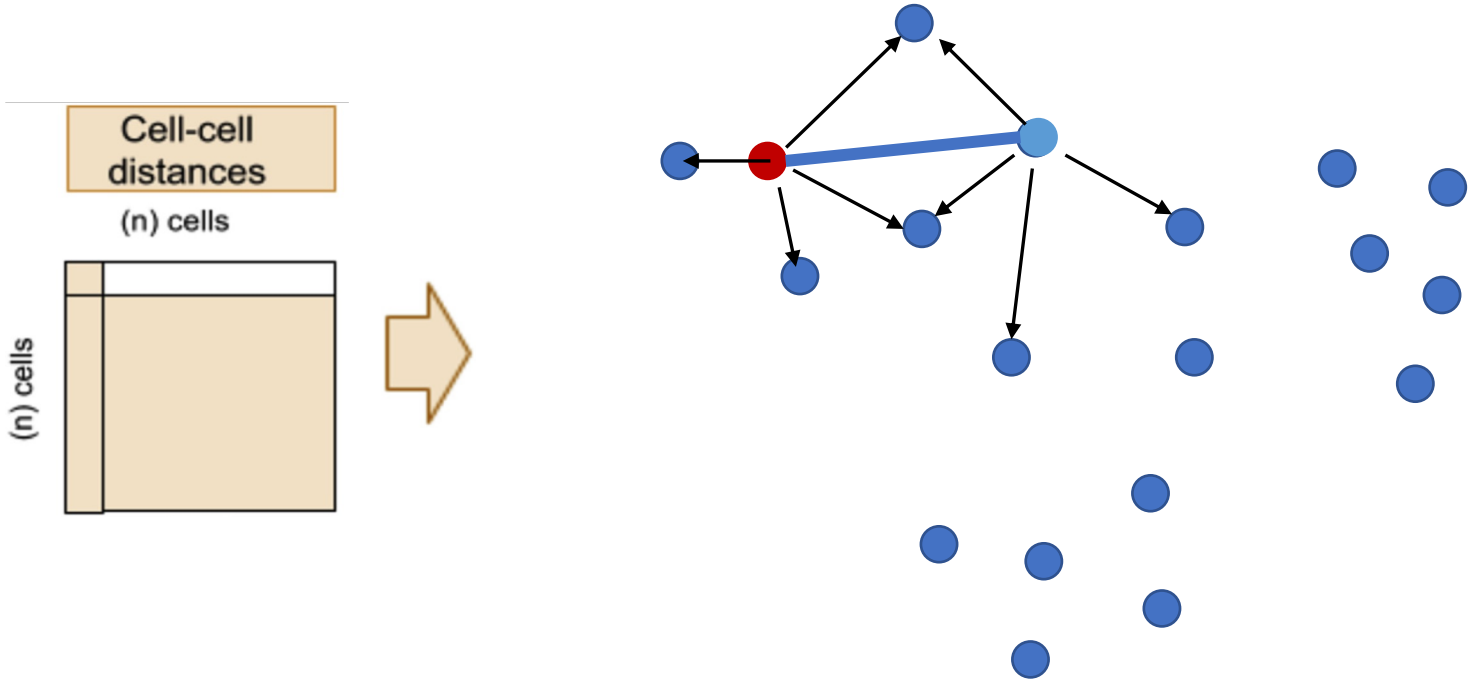


Extending KNN to SNN graphs (Shared Nearest Neighbors) (still with k = 4)



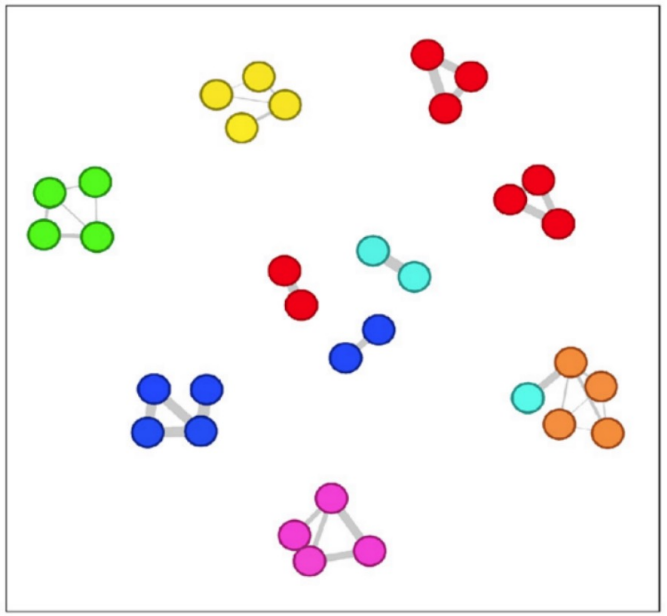
Extending KNN to SNN graphs (Shared Nearest Neighbors) (still with k = 4)

Two cells are connected by an edge if any of their nearest neighbors are shared.

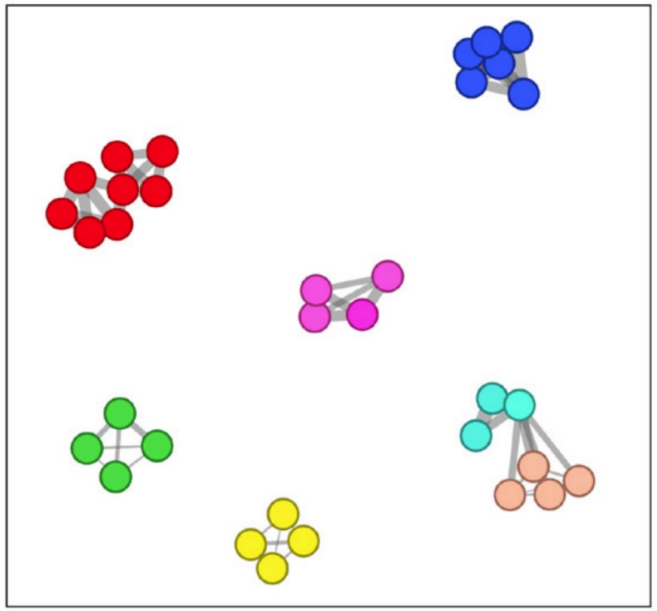


Extending KNN to SNN graphs (Shared Nearest Neighbors) (still with k = 4)

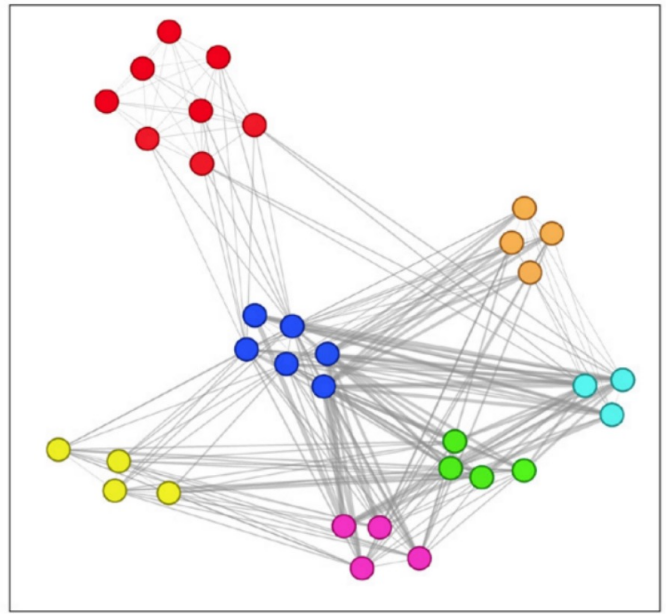
K is important when building KNN or SNN graphs !



(a) Parameter $K = 2$

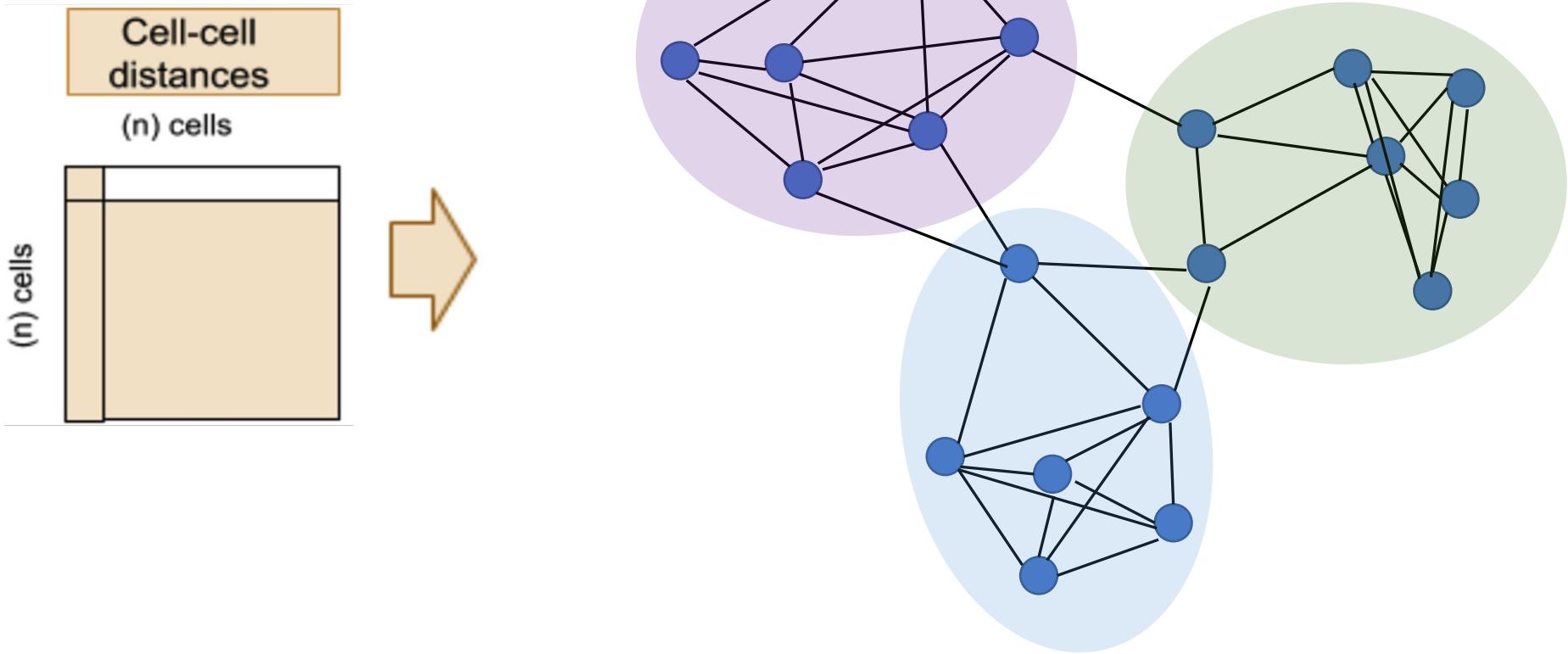


(b) Parameter $K = 3$



(c) Parameter $K = 6$

Graph-based clustering is nothing more than **community detection** (an “old” field from ’00s).

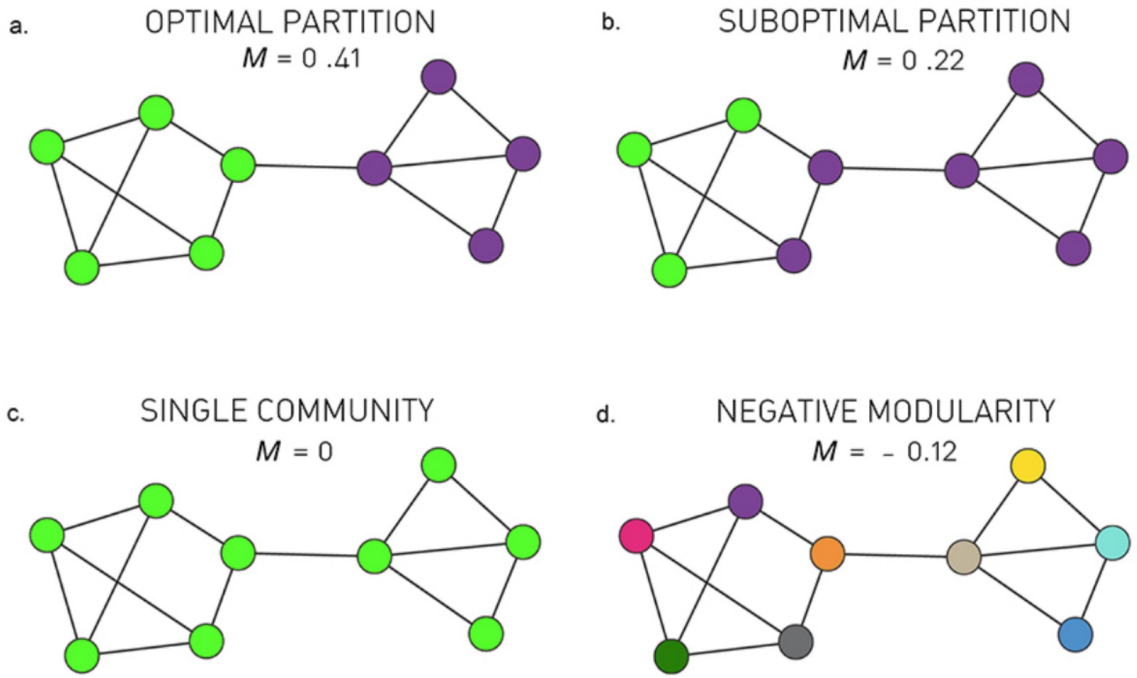


Graph-based clustering is nothing more than **community detection** (an “old” field from ’00s).

Many different algorithms for community detection:

- Louvain (heuristic)
- Infomap
- Walktrap
- …

Most of them are based on modularity maximization

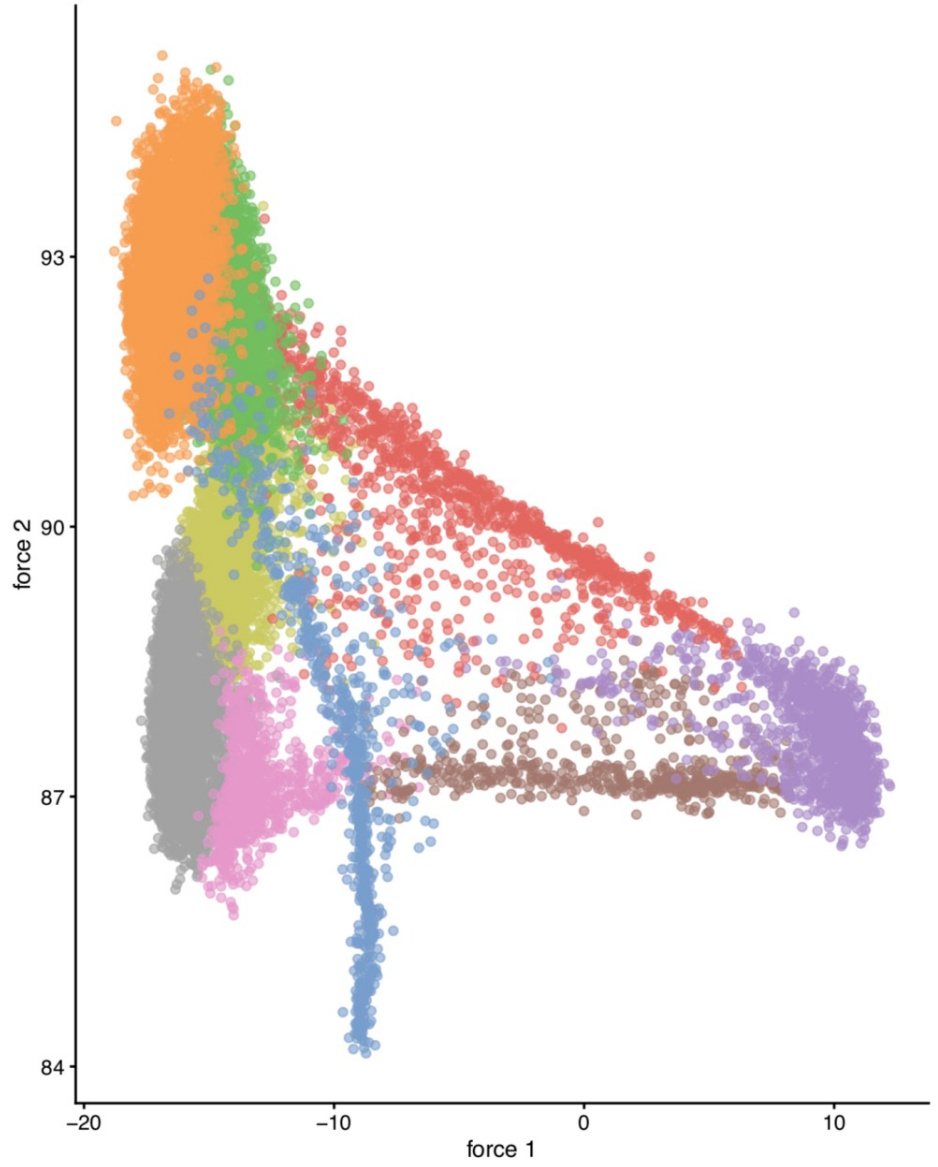


CAREFUL!

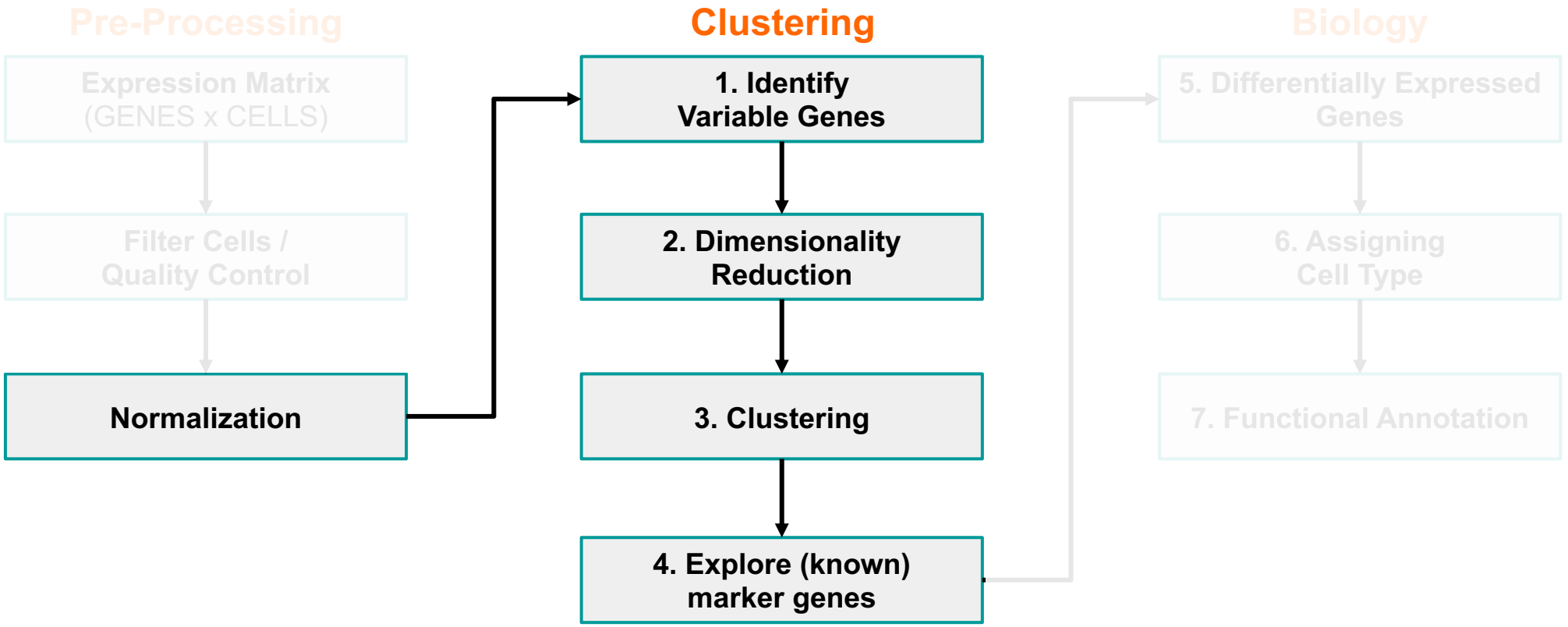
A graph can be **visualized** (i.e. embedded) in 2D, but the graph-based clustering step (i.e. **community finding**) is not done on its 2D embedding!!

"Do not let the tail (of visualization) wag the dog (of quantitative analysis)"

-- A. Lun

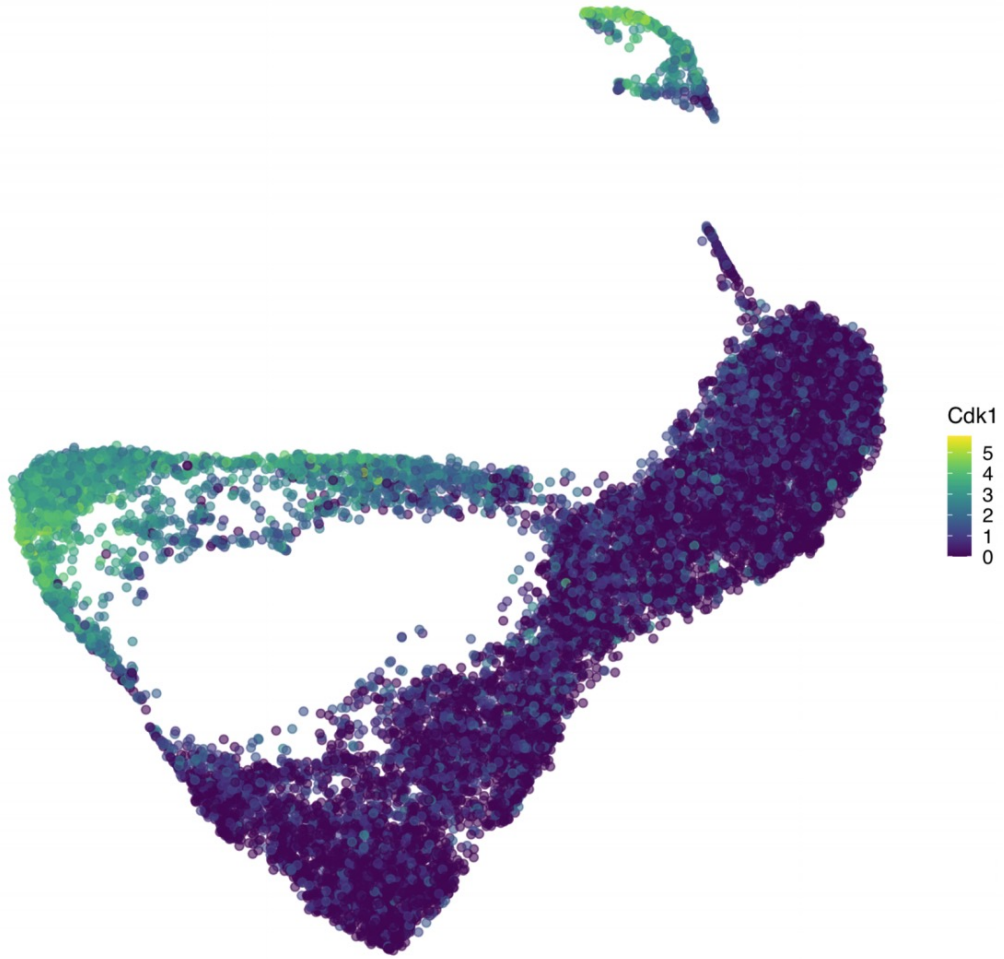


Analysis workflow



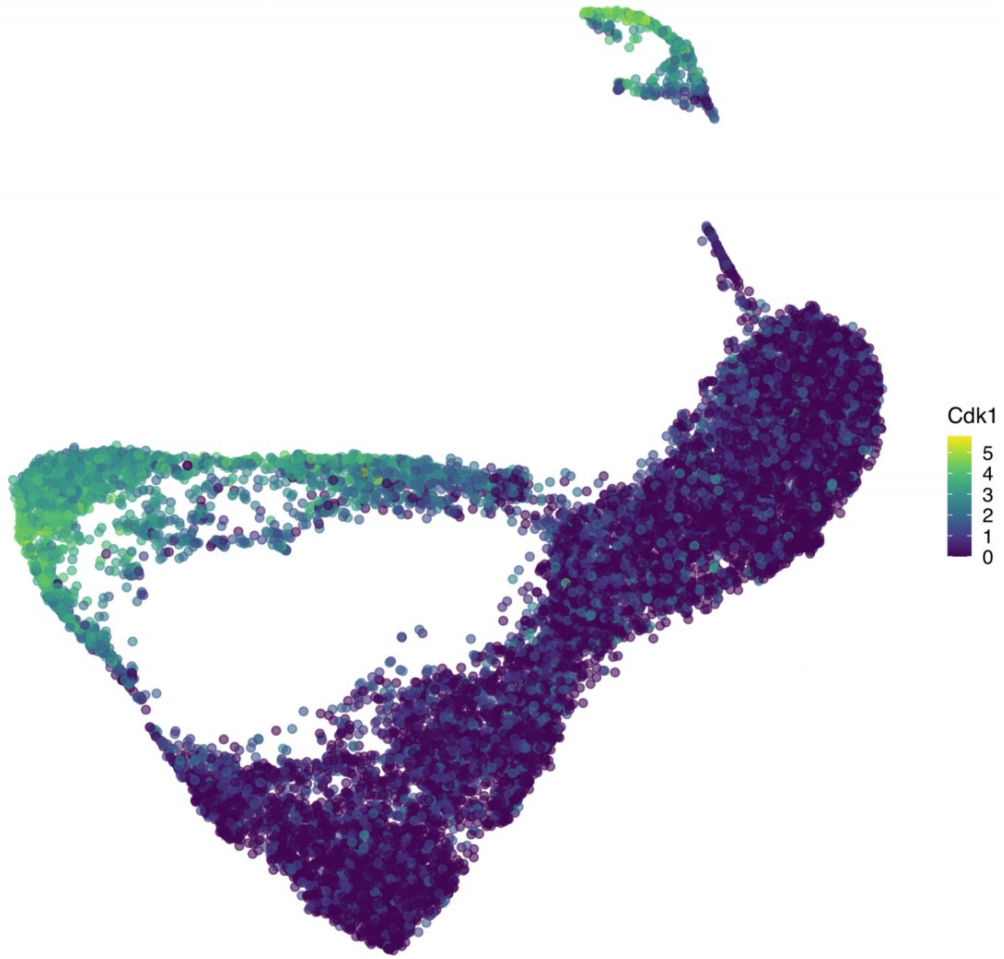
Visualizing expression of a gene of interest

On the dataset embedding:

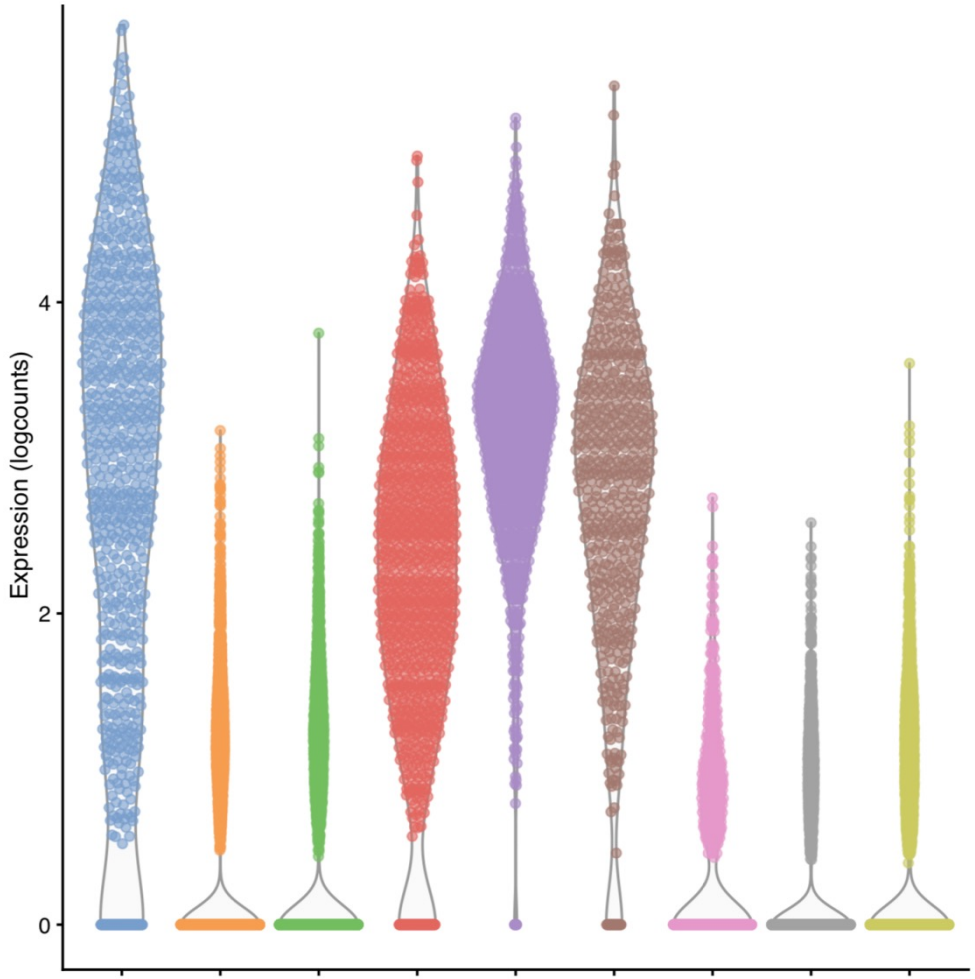


Visualizing expression of a gene of interest

On the dataset embedding:

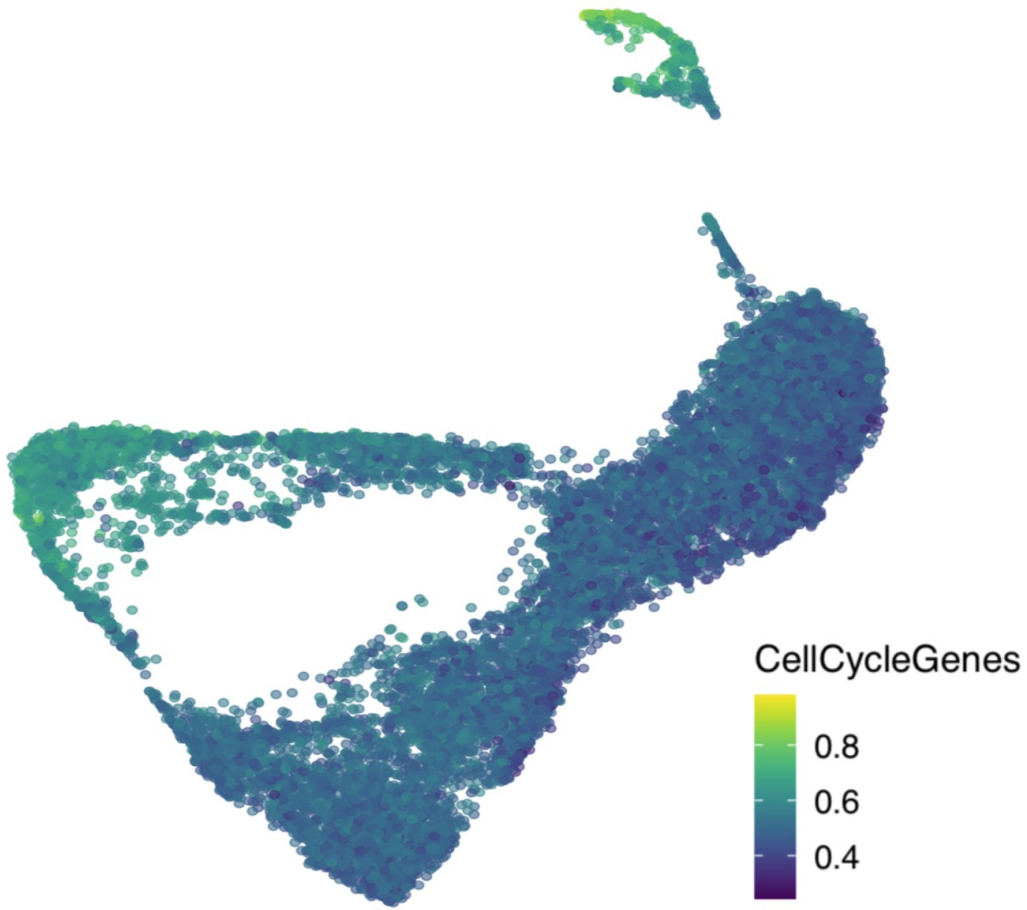


By clusters:

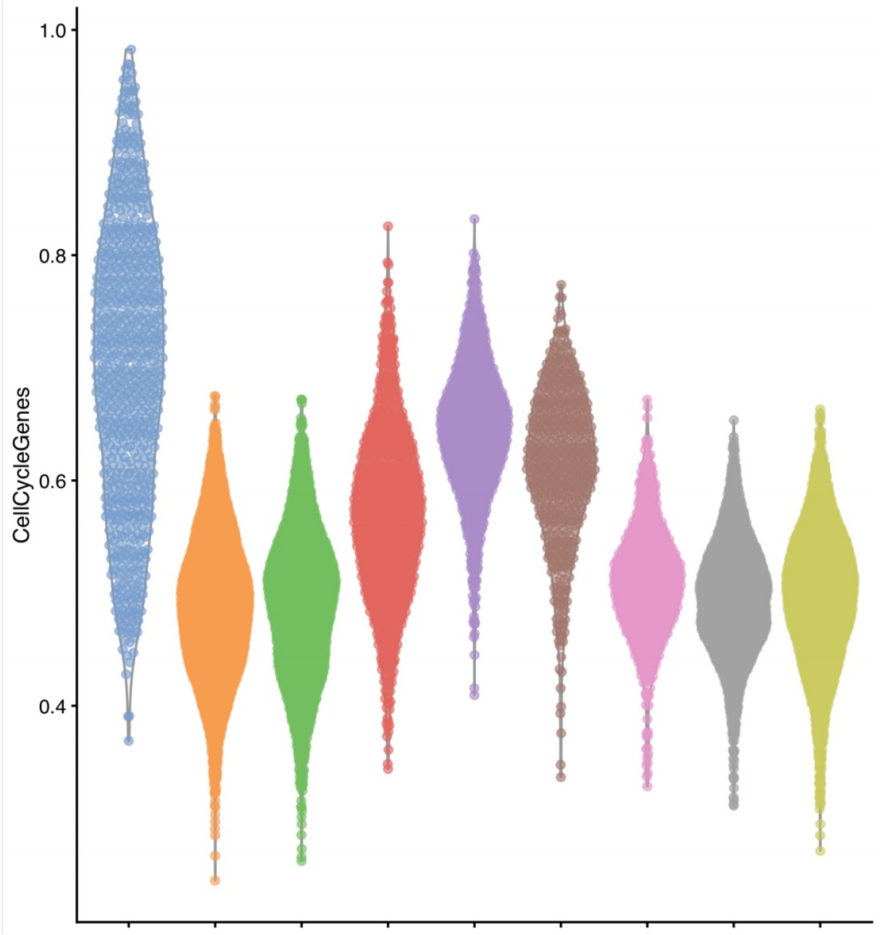


Visualizing expression of a module of interest

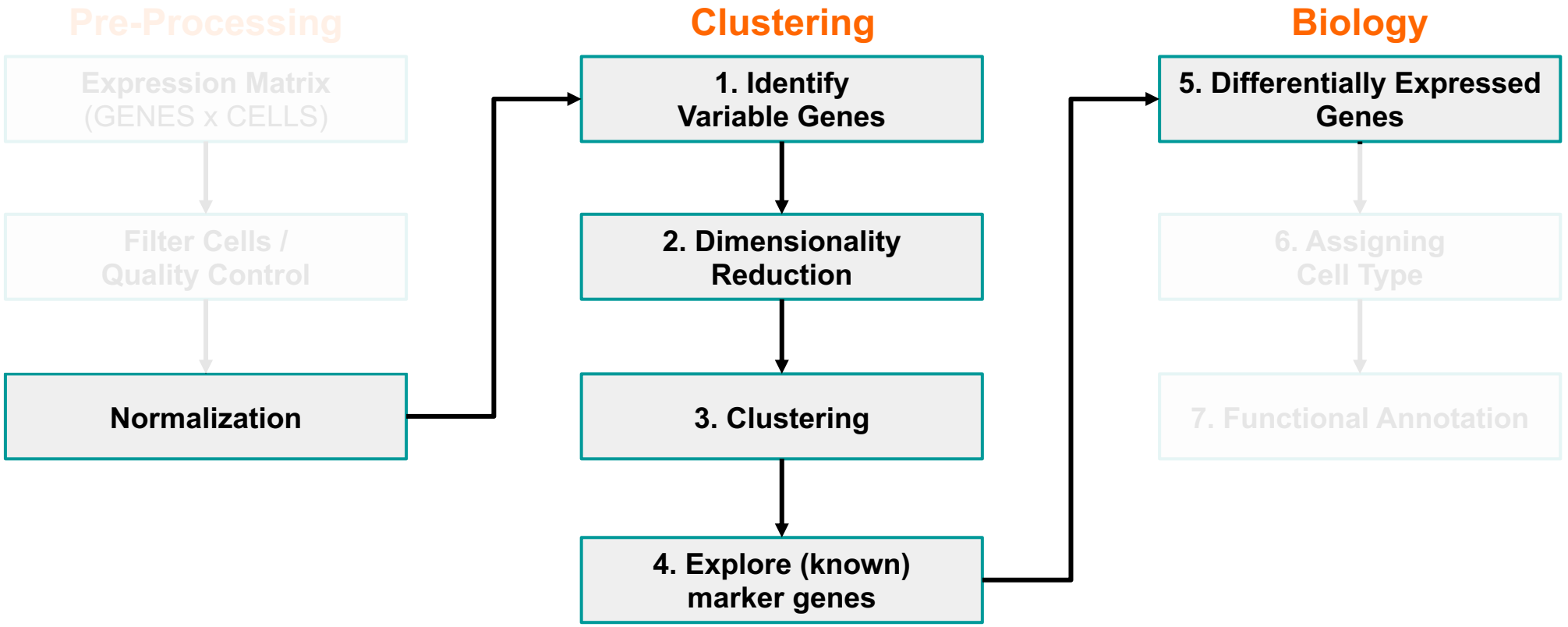
On the dataset embedding:



By clusters:



Analysis workflow



In scRNA-seq we often do not have a defined set of experimental conditions.

Instead, we can perform **pairwise comparisons** of gene expression, **between pairs of cell clusters**, using some of the following tests:

- "wilcox" : Wilcoxon rank sum test (default)
- t" : Student's t-test
- "poisson" : Likelihood ratio test assuming an underlying poisson distribution. Use only for UMI-based datasets
- "negbinom" : Likelihood ratio test assuming an underlying negative binomial distribution. Use only for UMI-based datasets
- Others…

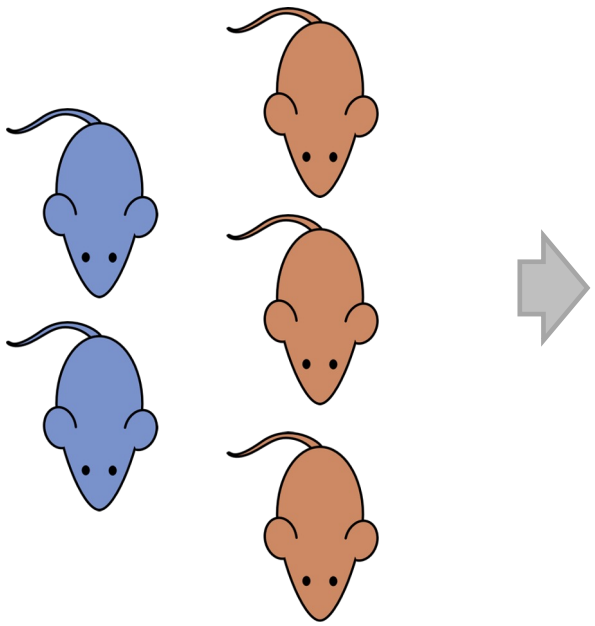
In scRNA-seq we often do not have a defined set of experimental conditions.

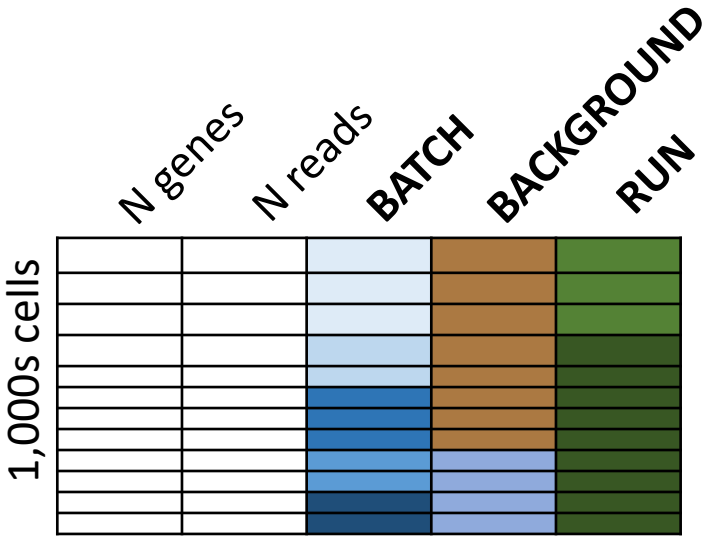
Instead, we can perform **pairwise comparisons** of gene expression, **between pairs of cell clusters**, using some of the following tests:

See Seurat::FindMarkers() and scran::findMarkers() for more info…

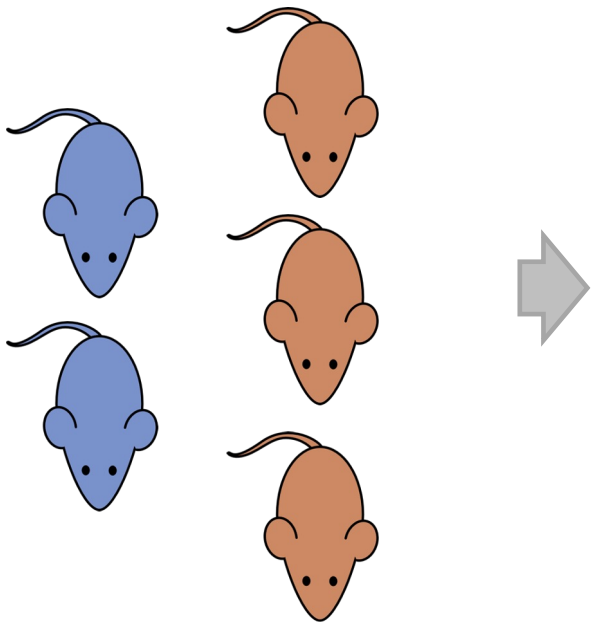
```
markers <- scran::findMarkers(
  sce,
  groups = sce$cluster,
  test.type = "t"
)
```

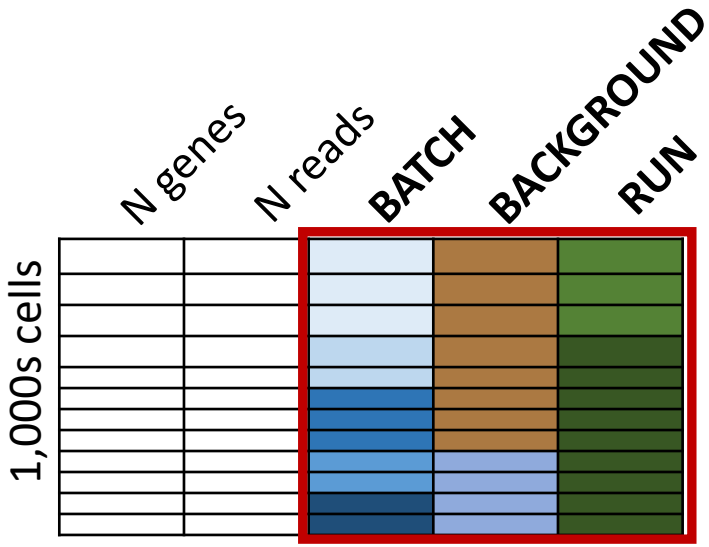
Think about your experimental design!!!





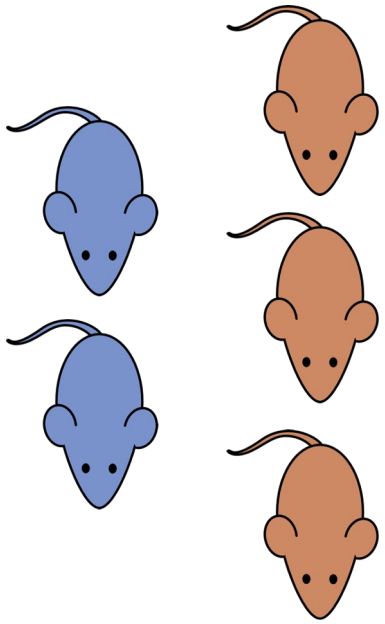
Think about your experimental design!!!

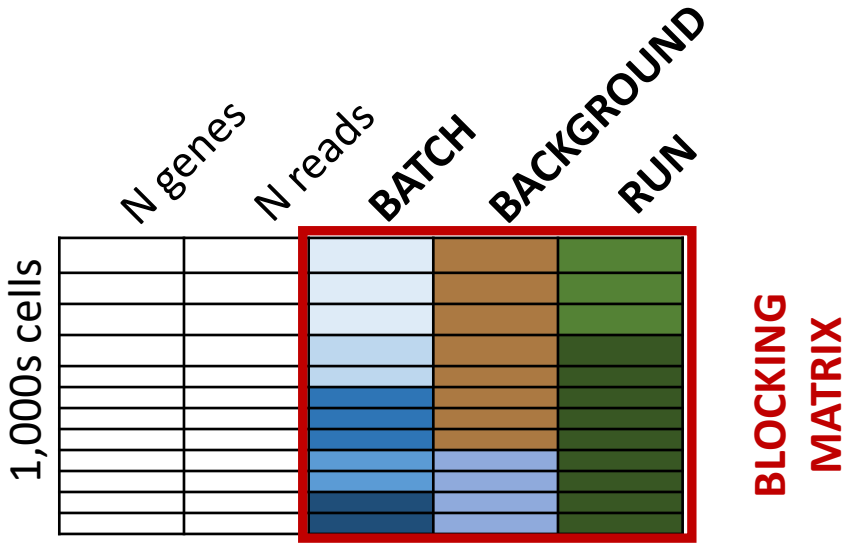




Not all the cells are the same: there are confounding variables.

Think about your experimental design!!!

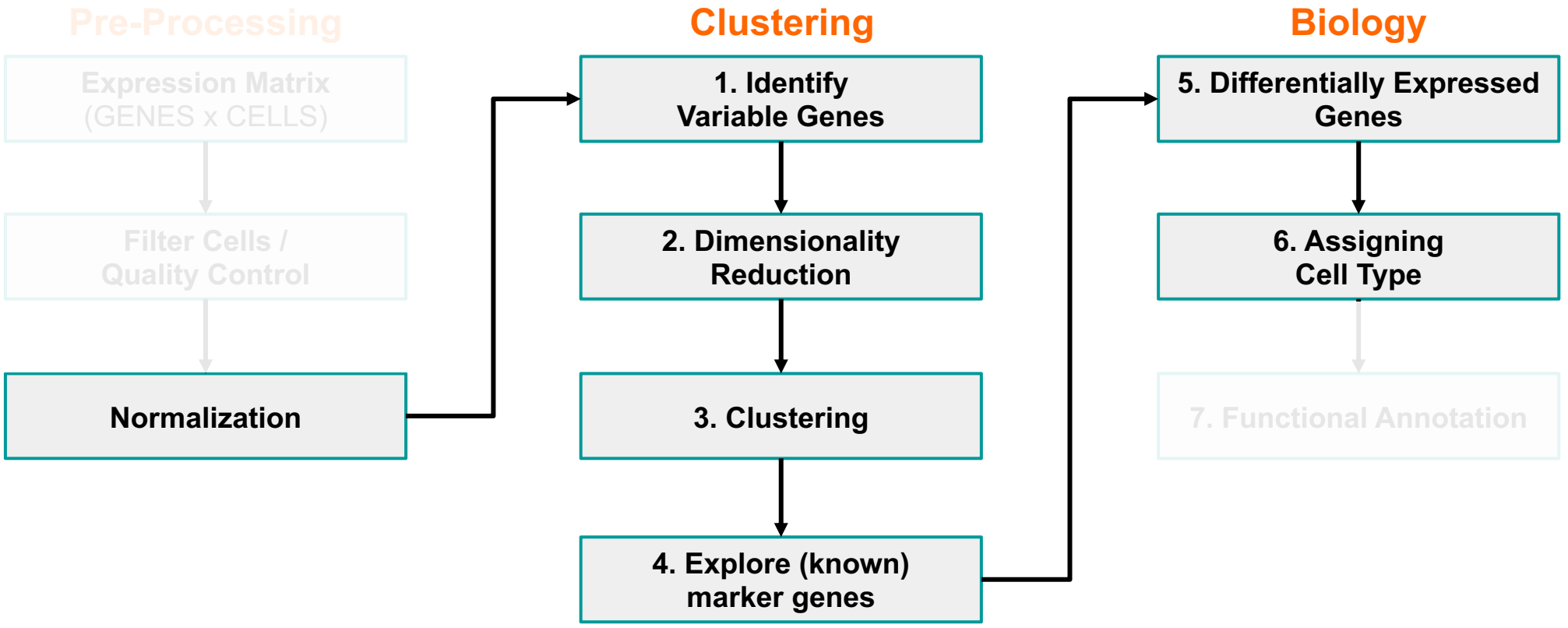




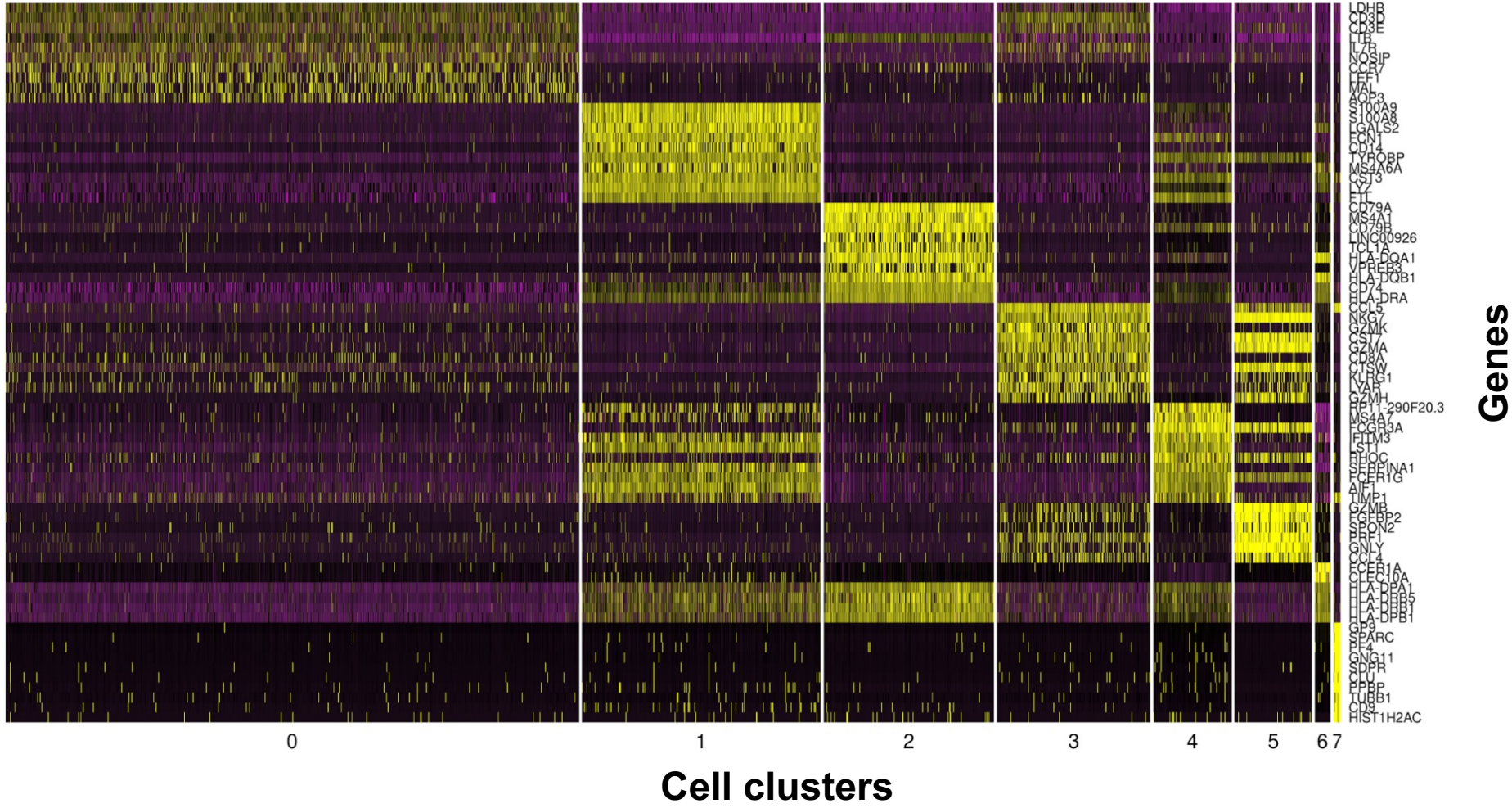
Not all the cells are the same: there are confounding variables.

```
markers <- scran::findMarkers(
  sce,
  groups = sce$cluster,
  test.type = "t",
  block = <BLOCKING MATRIX>
)
```

Analysis workflow



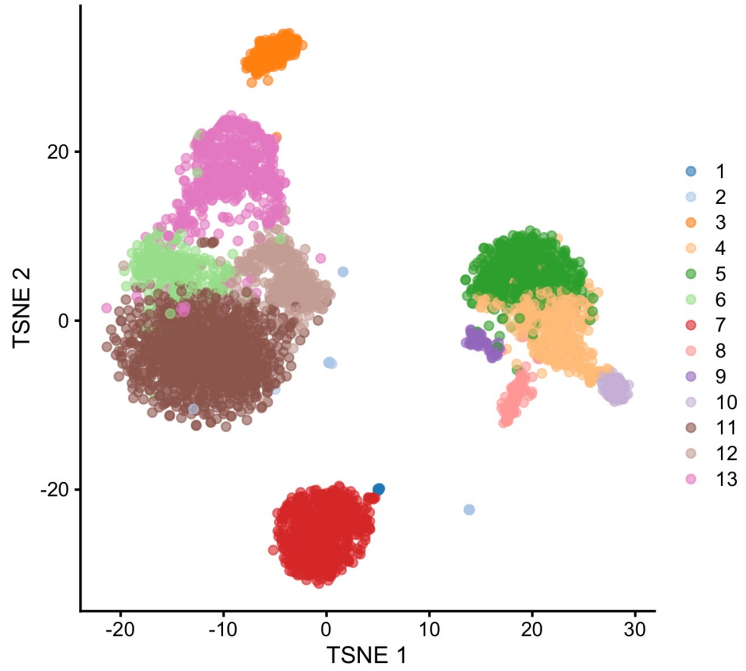
Cell type annotation using identified markers per cluster

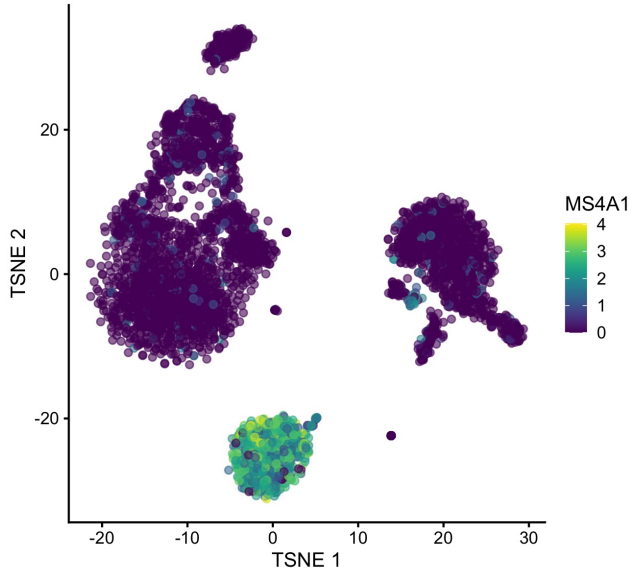


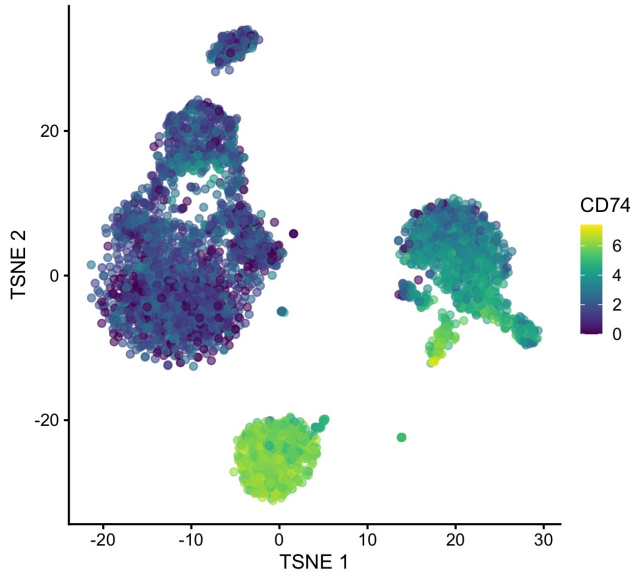
Manual cell type annotation using identified markers per cluster

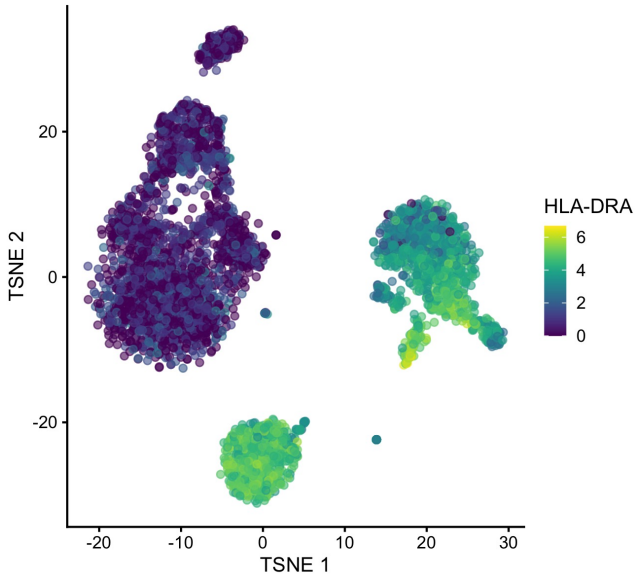
Top markers of cluster #7 in PBMCs:

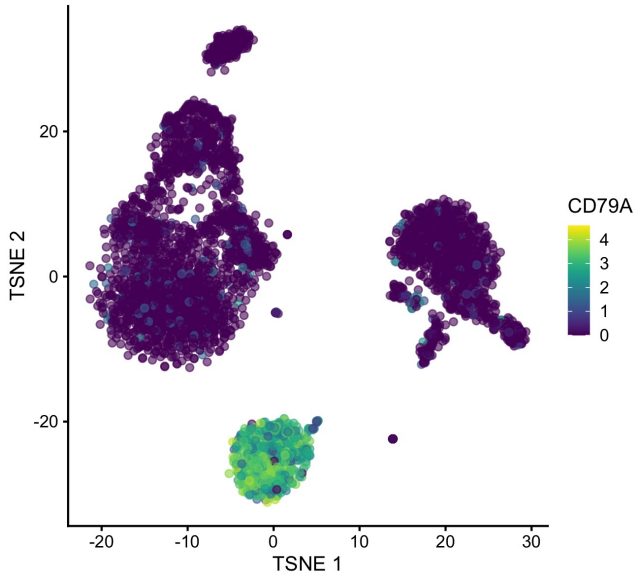
CD74
HLA-DRA
MS4A1
CD79A
HLA-DRB1
HLA-DPA1
CD79B
LTB
HLA-DQB1
TCL1A
CD52
HLA-DPB1
CD37





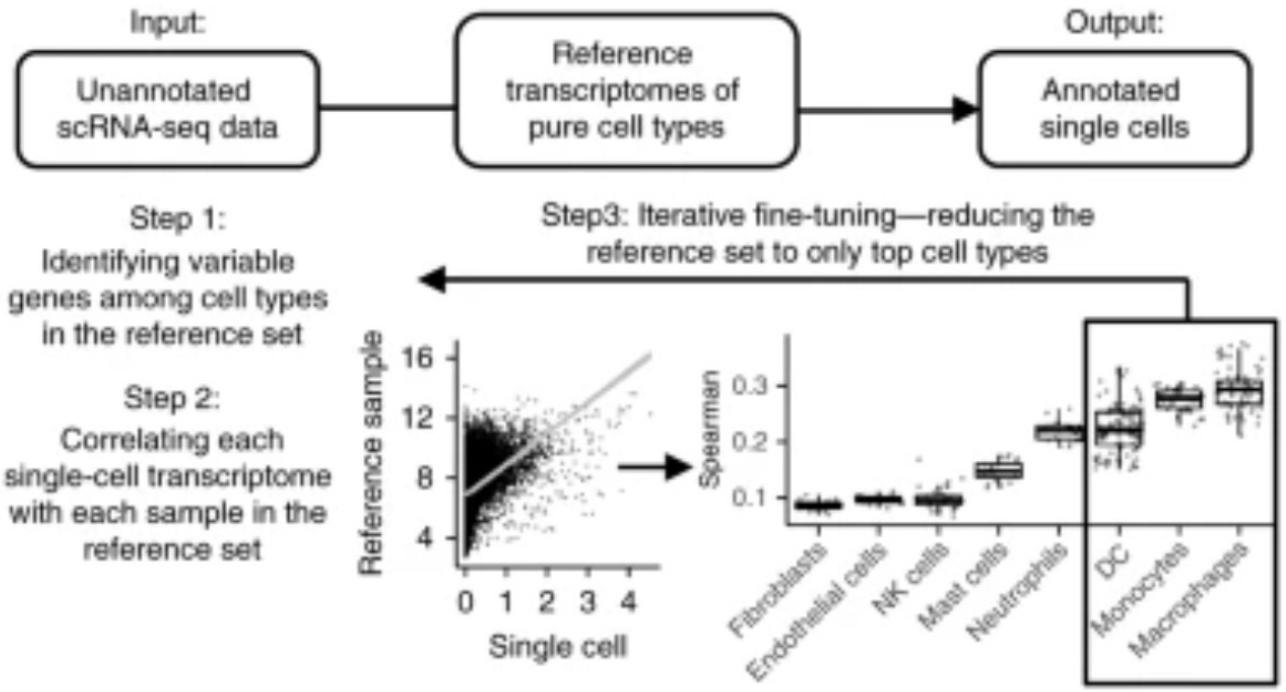






Automated cell type annotation using public marker databases

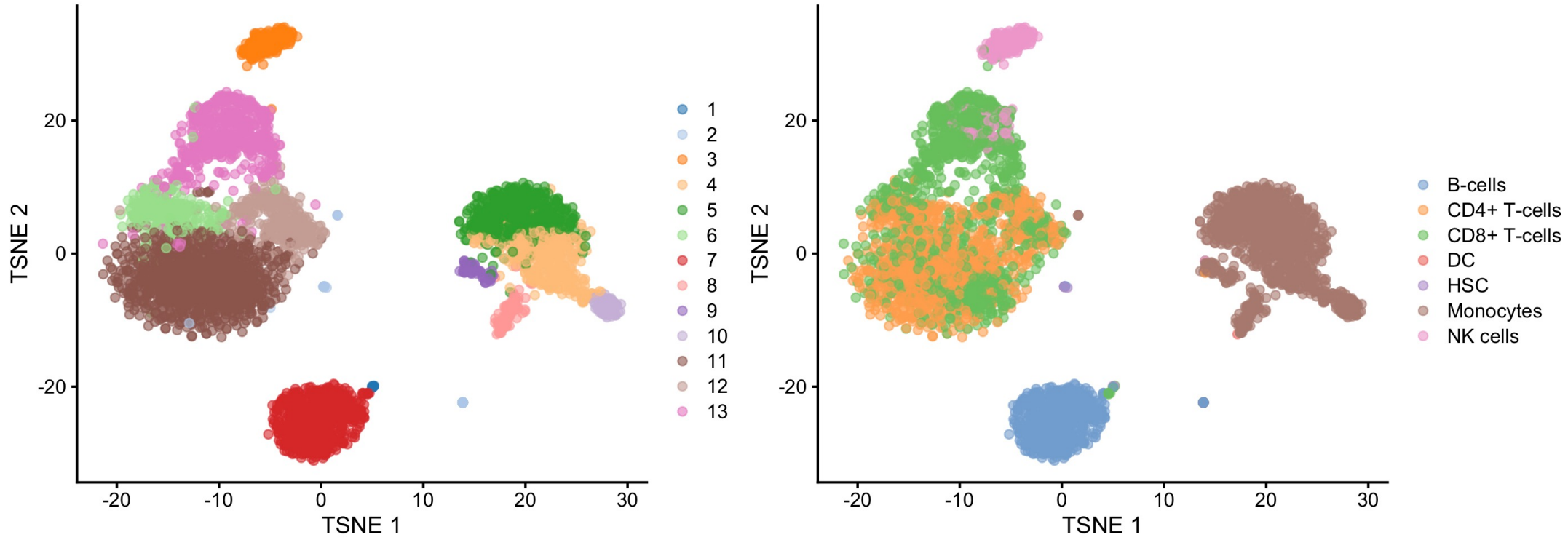
SingleR can rely on references of pure cell types to annotate individual cells within a scRNAseq dataset.



Aran et al., Nat. Immuno. 2019

Automated cell type annotation using public marker databases

SingleR can rely on references of pure cell types to annotate individual cells within a scRNAseq dataset.



However, it is limited in sensitivity, as it can only identify cells based on the references used.

Automated cell type annotation using public marker databases

Huge (and growing!) collection of tools for automated cell annotation…

Name	Version	Language	Underlying classifier	Prior knowledge	Rejection option
Garnett	0.1.4	R	Generalized linear model	Yes	Yes
Moana	0.1.1	Python	SVM with linear kernel	Yes	No
DigitalCellSorter	GitHub version: e369a34	Python	Voting based on cell type markers	Yes	No
SCINA	1.1.0	R	Bimodal distribution fitting for marker genes	Yes	No
scVI	0.3.0	Python	Neural network	No	No
Cell-BLAST	0.1.2	Python	Cell-to-cell similarity	No	Yes
ACTINN	GitHub version: 563bcc1	Python	Neural network	No	No
LAmbDA	GitHub version: 3891d72	Python	Random forest	No	No
scmapcluster	1.5.1	R	Nearest median classifier	No	Yes
scmapcell	1.5.1	R	kNN	No	Yes
scPred	0.0.0.9000	R	SVM with radial kernel	No	Yes
CHETAH	0.99.5	R	Correlation to training set	No	Yes
CaSTLe	GitHub version: 258b278	R	Random forest	No	No
SingleR	0.2.2	R	Correlation to training set	No	No
scID	0.0.0.9000	R	LDA	No	Yes
singleCellNet	0.1.0	R	Random forest	No	No
LDA	0.19.2	Python	LDA	No	No
NMC	0.19.2	Python	NMC	No	No
RF	0.19.2	Python	RF (50 trees)	No	No
SVM	0.19.2	Python	SVM (linear kernel)	No	No
$SVM_{rejection}$	0.19.2	Python	SVM (linear kernel)	No	Yes
kNN	0.19.2	Python	kNN ($k = 9$)	No	No

Abdelaal et al., Genome Biol. 2019