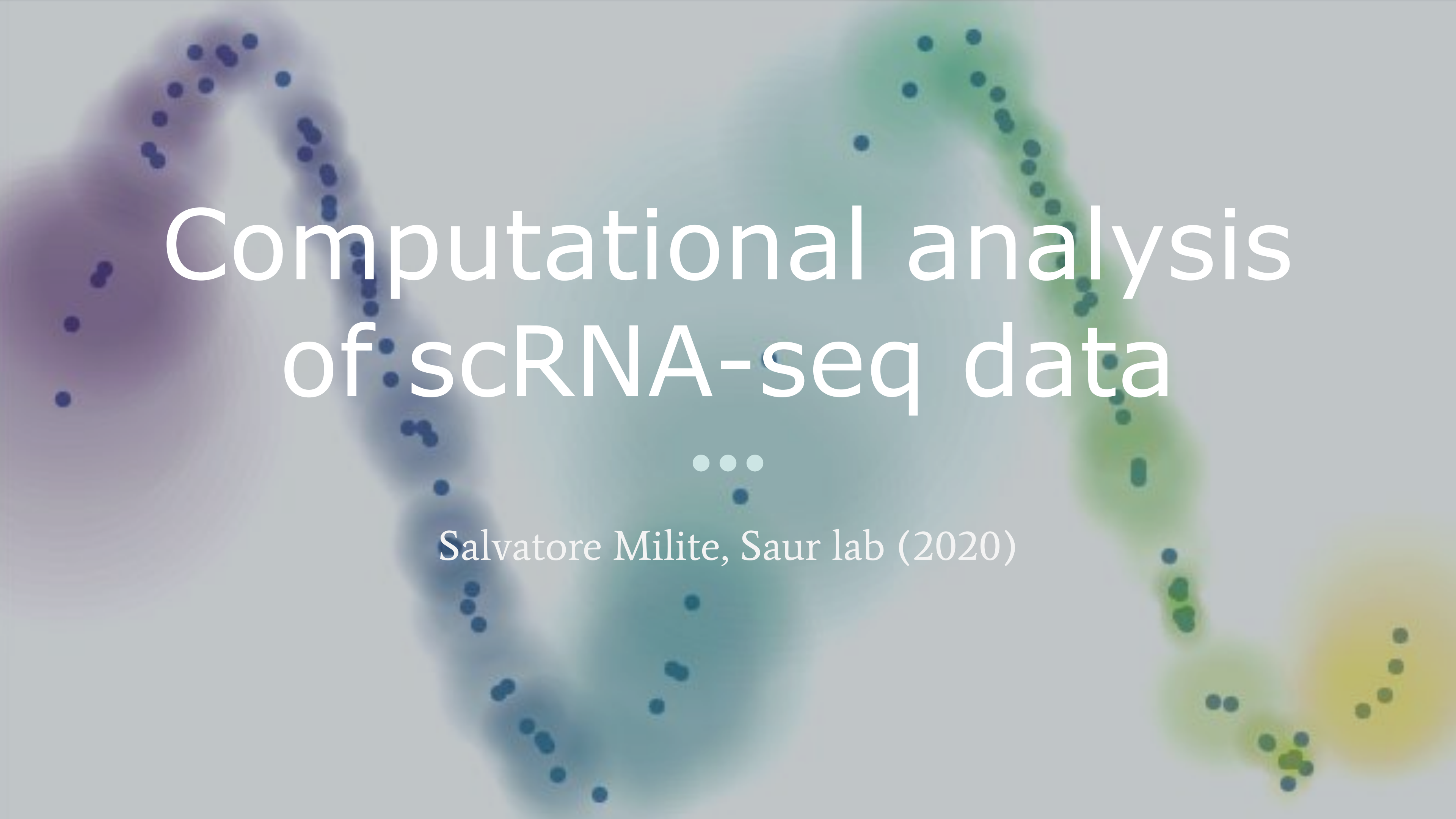


Computational analysis of scRNA-seq data

Salvatore Milite, Saur lab (2020)

Who I am, what I do

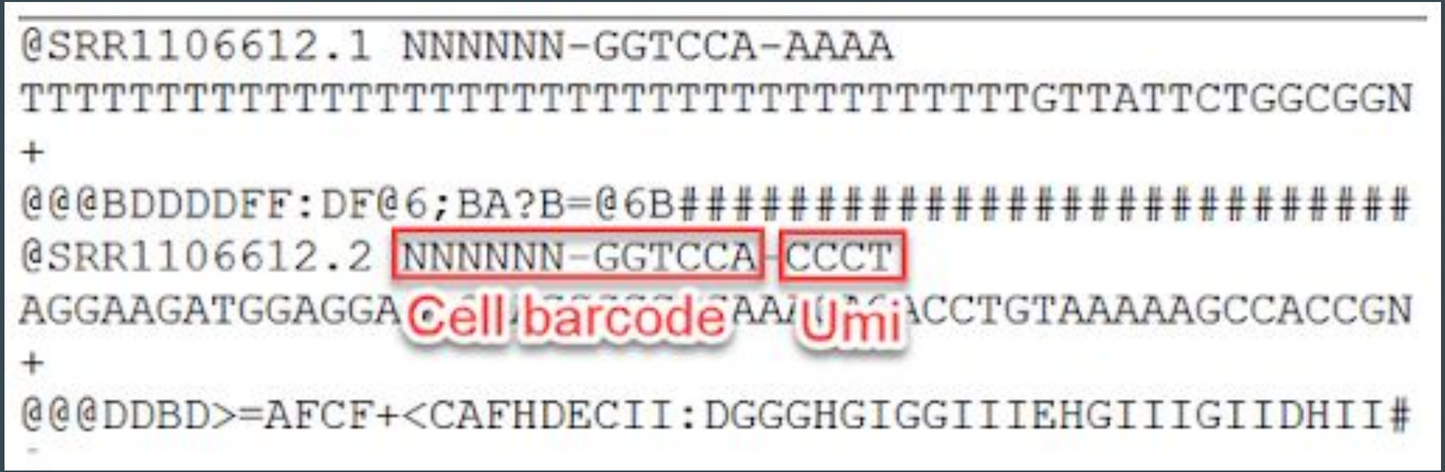
- Background in Data Science and Scientific Calculus
- Statistical learning and modelling of omics data (keen Bayesian supporter)
- Mostly data integration among different platforms
- Interested in cancer evolution, immuno-oncology and cellular signaling
- Just started my PhD here, where I am working on TEM characterization and ligand-receptor interactions in PDAC
- Hopefully by developing new methods for ligand-receptor interaction with spatial and temporal integration

Your path to single cell analysis

- A very informal discussion about methods and pipelines today
- A 2 hours practical on Thursday where we are going to see an end by-end analysis of a single cell datasets
- If you don’t already have it, install RStudio together with a new version of R (>3.6)
- All the material and prerequisites can be found on https://github.com/Militeee/lc_2020
- If you have any problem with your pc let me know asap, so that I can arrange a Google colab notebook to let you run the analysis in cloud

Where everything begins

As most of the sequencing data you will encounter, the starting point is a file with the .fastq extension, which contains all the read you have sequenced together with their quality score



```
@SRR1106612.1 NNNNNN-GGTCCA-AAAA
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTGTTATTCTGGCGGN
+
@@@BDDDDFF:DF@6;BA?B=@6B#############################
@SRR1106612.2 NNNNNN-GGTCCA-CCCT
AGGAAGATGGAGGA              AA        ACCTGTAAAAAGCCACCGN
+
@@@DDBD>=AFCF+<CAFHDECII:DGGGHGIGGIIIEHGIIIGIIDHII#
```

Absence of standardization

- The huge number of tools being developed each year for scRNA-seq analysis and the increasing in the number of cells sequenced has prevented the establishment of a fully standardized workflow
- Some tools provide integrated analysis environment like Seurat, SCANPY and Scater. Their sample tutorial analysis can be considered a de facto-standard way of analyzing a scRNA-seq dataset
- Sometimes limited by the language: R vs Python.

scRNA-seq pre-processing steps

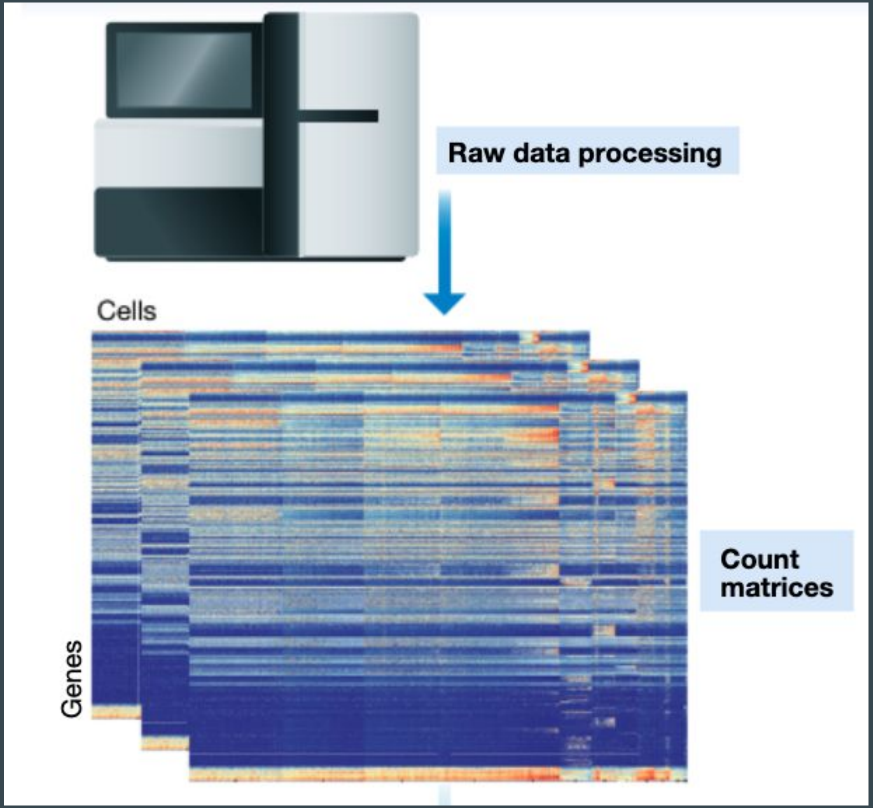
- Alignment (usually you don’t care too much)
    - Kallisto, STAR, Cell Ranger (most used, fully automatic)
- Quality control (QC)
- Data normalization and scaling
- Technical covariates regression
- Cell cycle regression
- Batch effects removal
- Data Inputation/ Denoising
- Feature selection
- Dimensionality reduction

scRNA-seq downstream analysis steps

- Clustering
- Trajectory inference
- Differential expression
- GSEA
- Ligand-receptor interaction
- CNV estimation
- Multi-Omics Data integration

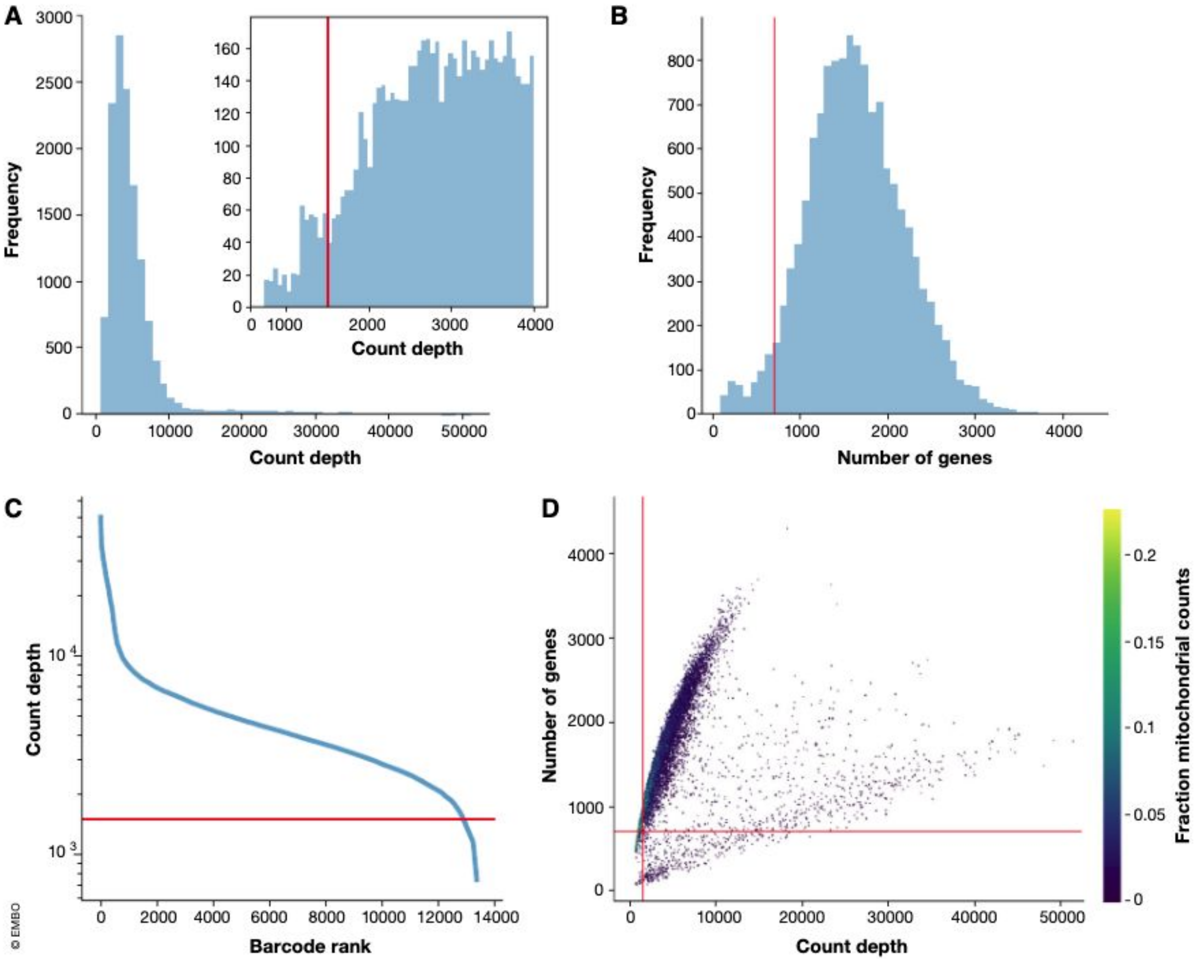
Alignment and feature count

- Tools as Cell Ranger (standard on 10x) take care of aligning genes to the reference genome and performing UMI or read count for each cell
- For some analysis (RNA velocity) it might be necessary to redo this step with other aligning softwares as Kallisto or STAR.



Quality Control

- As the cells are living organism sensitive to mechanical stress, there is the possibility of sequencing cells with broken membrane
- Moreover, the process of cell separation is not always perfect,
- Usually we look at 3 parameters total number of genes captured, total number of counts and percentage of reads/UMI coming from mitochondrial genes
- Consider all those 3 covariates together rather than separately
- Thresholds should be adjusted over downstream analysis


A
3000
2500
2000
1500
1000
500
0
Frequency
0
10000
20000
30000
40000
50000
Count depth
160
140
120
100
80
60
40
20
0
0 1000
2000
3000
4000
Count depth
B
800
700
600
500
400
300
200
100
0
Frequency
0
1000
2000
3000
4000
Number of genes
C
10 4
10 3
Count depth
0
2000
4000
6000
8000
10000
12000
14000
Barcode rank
D
4000
3000
2000
1000
0
Number of genes
0
10000
20000
30000
40000
50000
Count depth
0.2
0.15
0.1
0.05
0
Fraction mitochondrial counts

© EMBO

Data normalization and scaling

- Difference in the total number of molecules sequenced (mainly due to technical reasons) for each cell prevents the direct comparison of the counts
- Counts scaled by quantities called size factors
- Usually CPM normalization is applied (note the assumption that all cells have the same number of counts prior to sequencing)

$$\text{RPM or CPM} = \frac{\text{Number of reads mapped to gene} \times 10^6}{\text{Total number of mapped reads}}$$

Data normalization and scaling

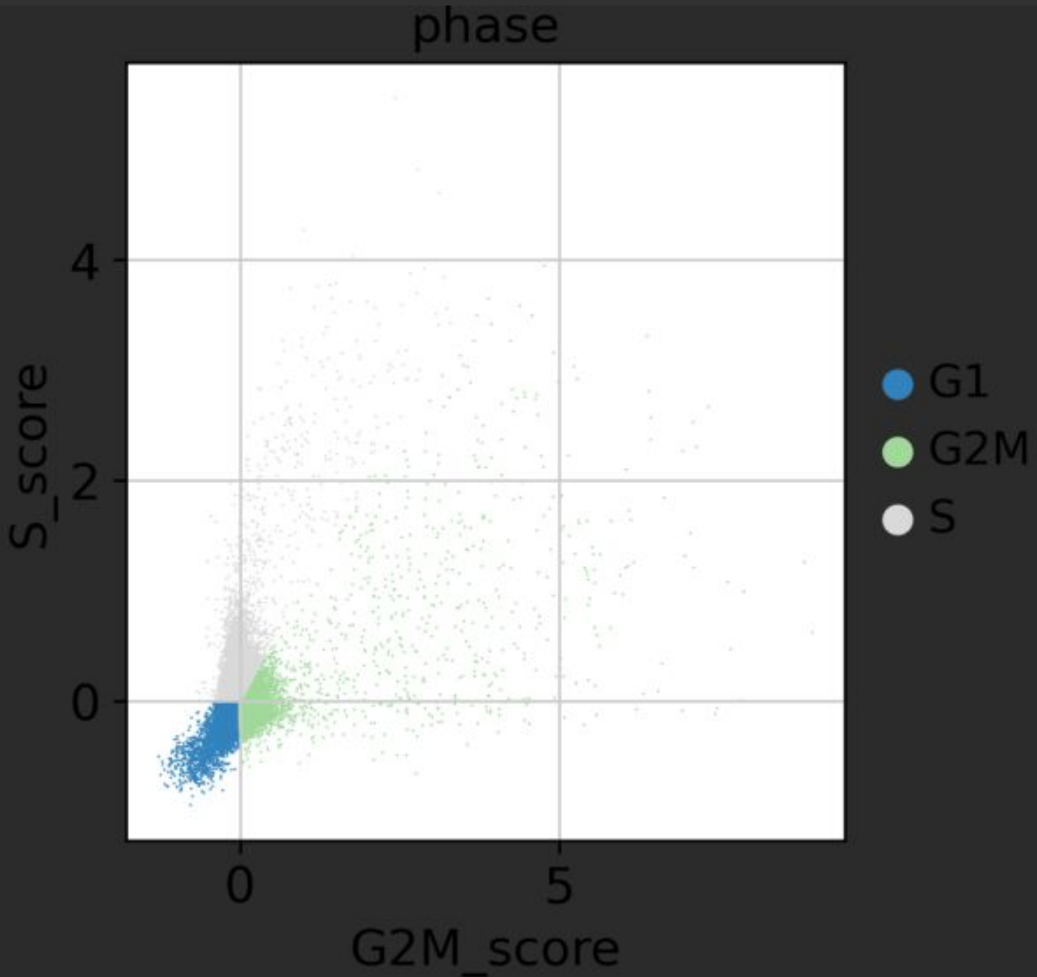
- More sophisticated methods such as Scran can estimate size factor with less assumptions
- Negative Binomial GLM based tools can be a good choice in case of strong effects between samples or more sophisticated experimental design (ex. in plate based sequencing)
- After normalization matrices are usually log(x+1) transformed
- They may be also scaled to zero mean and unit variance

Regressing out biological or technical effects

- Simplest approach is to simply perform a linear regression against the confounders and remove just get the residuals (as implemented in Scanpy and Seurat)
- Non-linear model or mixture models can provide more sophisticated ways of performing this step (scLVM)
- Usually performing correction for biological factor is different from doing the same with technical covariates

Cell cycle regression

- The most common type of biological effect one wants to remove
- Usually what is done is assign a score to each cell based on some known maker genes
- A very simple way of scoring is by averaging the expression of all the genes in a marker set
- Than simple linear regression is performed and the residual are taken


phase
S_score
4
2
0
G2M_score
0
5
G1
G2M
S

Technical effect removal

- In theory is it possible to use standard regression methods
- Sometimes we want to regress out also mitochondrial gene fractions and total number of genes/counts captured (trajectory estimation)
- However there are some effects due to sampling that are impossible to correct (usually referred as drop-out)
- More complicated normalization methods can improve and let you avoid this part

A few notes about dropout and sparsity

- Dropout can be a misleading term
- Useful to distinguish between technical and biological dropout
- Biological dropout arises from the observational process and it is usually not something we want to correct (but we may want to take it into account)

$$x_{ij} \mid x_{i+}, \lambda_{ij} \sim \text{Poisson}\ (x_{i+} \lambda_{ij})$$

$$\lambda_{ij} \sim g_j(\cdot) = \text{Gamma}(.\,)$$

Batch effect and data integration

- Here the nomeclature can be misleading
- Batch effects (within the same experimental unit) are usually based on simple linear regressions, but
- Batch effects (among different experimental units) generally uses non-linear method and to try to preserve (MNN, BBKNN, Scanorama)
- Risk of overcorrection

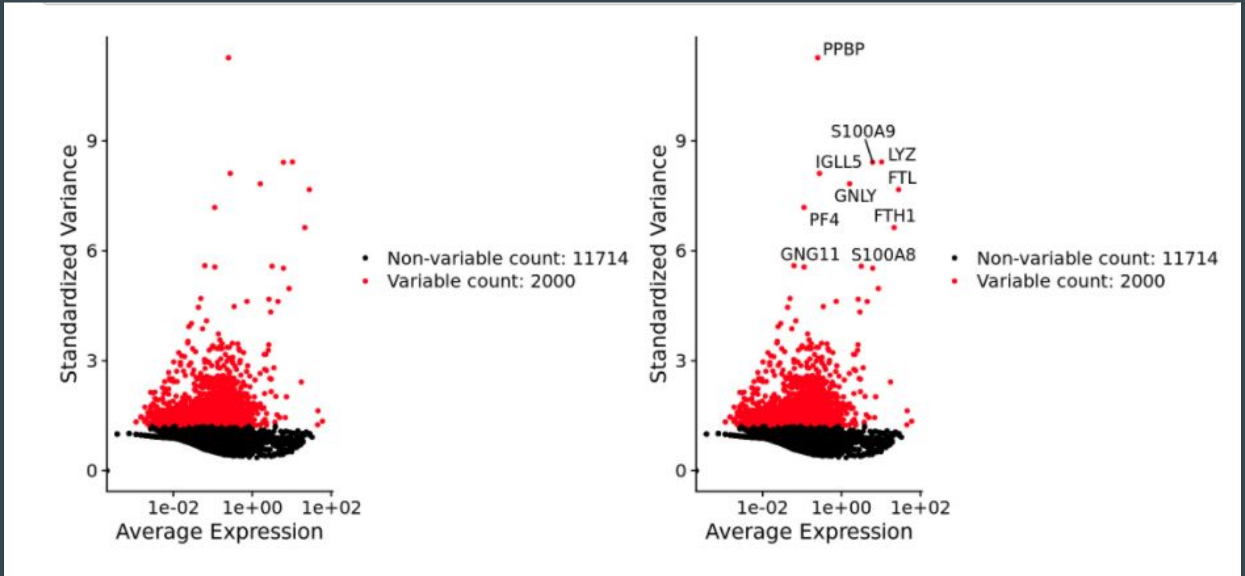
A GOOD EXPERIMENTAL DESIGN CAN IMPROVE AND EVEN REMOVE THE NEGATIVE EFFECTS OF THOSE ALGORITHMS!

Data inputation or expression recovery

- Technical drop out correction
- Can be also think in general as a denoising system
- Usually based on some formulation of a generative model
- Can introduce spurious or iper correlations in the dataset
- Can be nice for visualization purposes
- I suggest you to generally not use them (in some pathological cases they might be useful though)

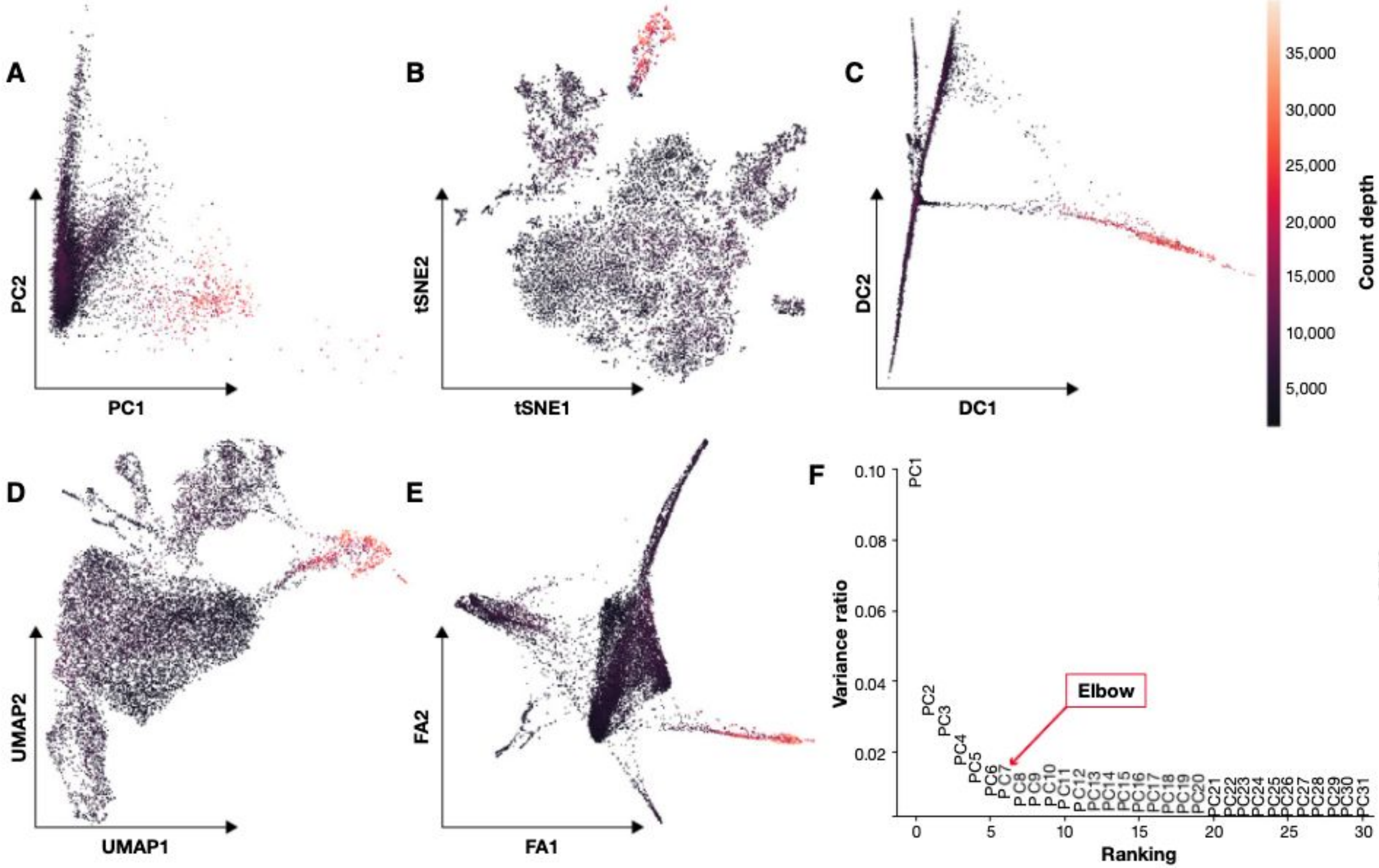
Feature selection

- Most of the genes are not informative, choose just some of them can reduce computation time and denoise the data
- Usually one choose from 1k and 5k genes



Dimensionality reduction

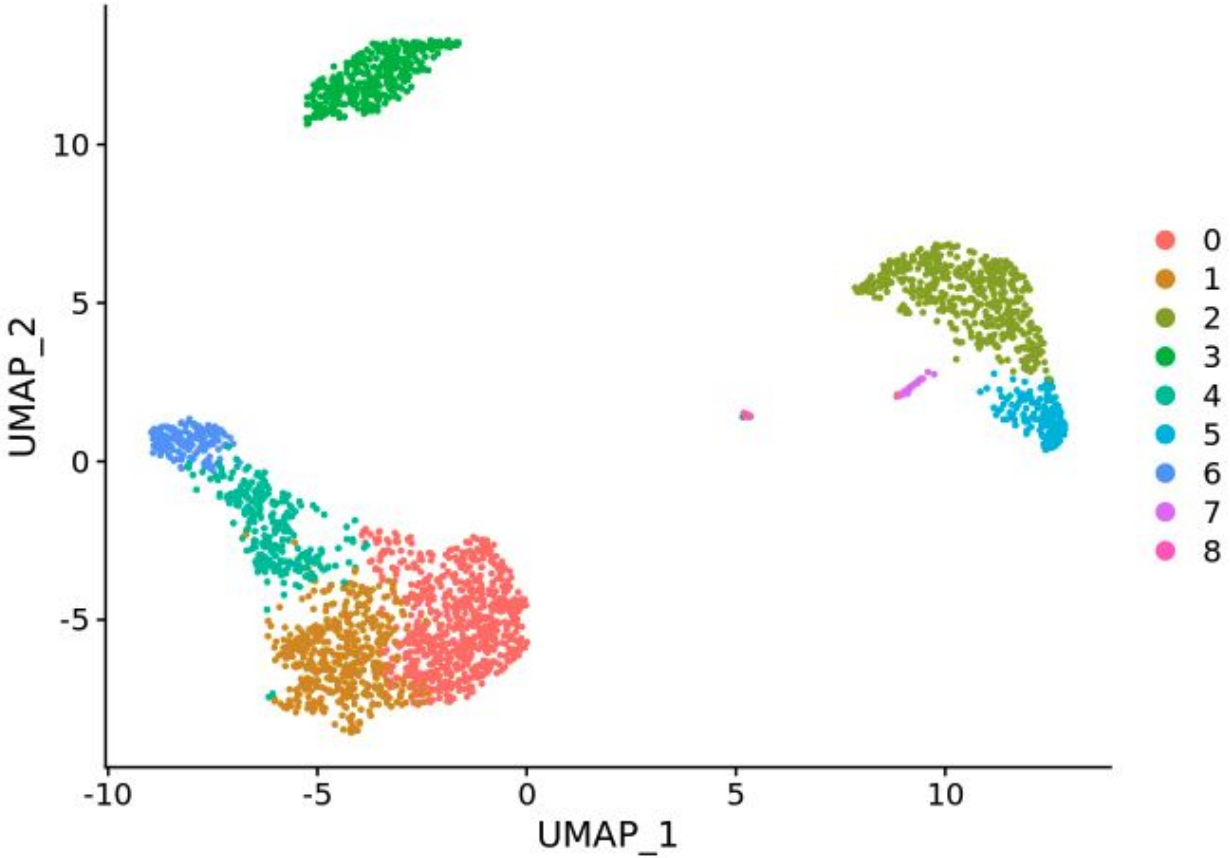
- Usually for visualization purpose
- These algorithms embed the expression matrix into a low-dimensional space
- Some of those methods can be used to try to find the intrinsic dimensionality of the dataset
- scRNA-seq datasets are usually intrinsically low-dimensional
- More to come in the next days
- I generally recommend UMAP

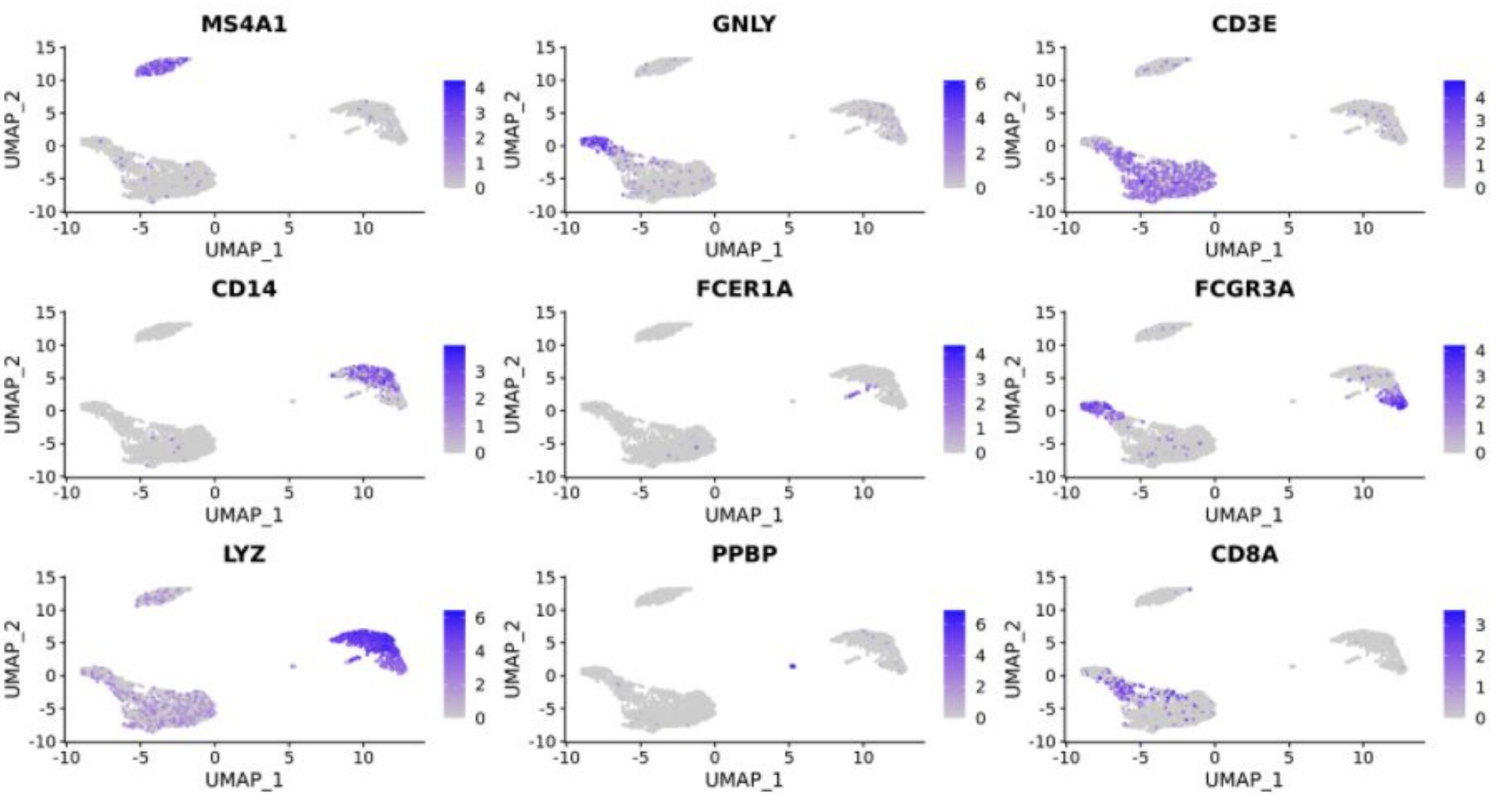

A
PC2
PC1
B
tSNE2
tSNE1
C
DC2
DC1
35,000
30,000
25,000
20,000
15,000
10,000
5,000
Count depth
D
UMAP2
UMAP1
E
FA2
FA1
F
Variance ratio
0.10
0.08
0.06
0.04
0.02
Ranking
0
5
10
15
20
25
30
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
PC19
PC20
PC21
PC22
PC23
PC24
PC25
PC26
PC27
PC28
PC29
PC30
PC31
Elbow

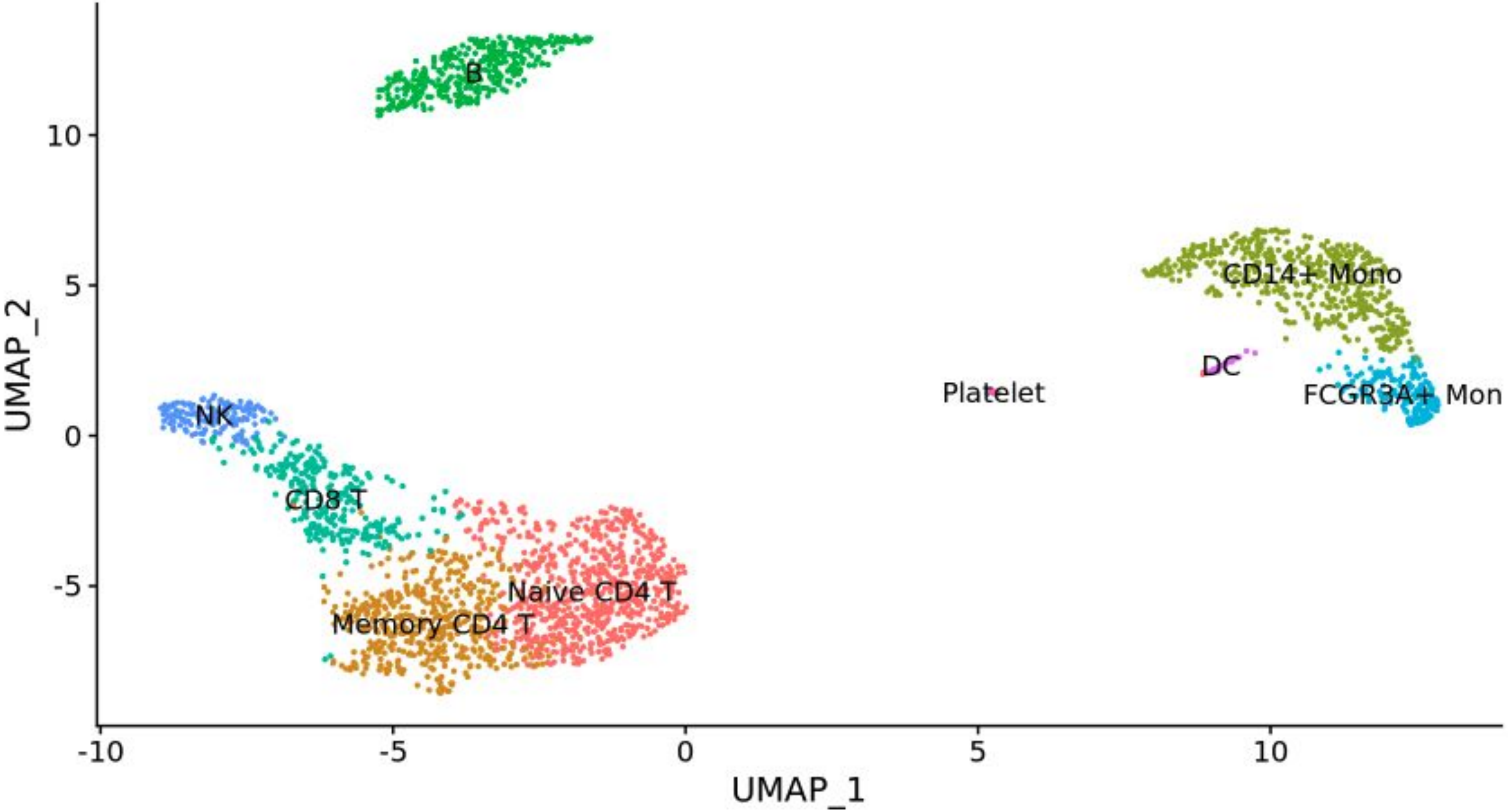
© EMBO

Clustering

- Clusters are obtained by grouping cells based on the similarity of their gene expression profiles
- Expression profile similarity is determined via distance metrics
- Here the state of the art is represented by KNN graph clustering approac, in particular the Louvain clustering algorithm
- Then we usually characterize each cluster calculating their marker genes or using a set of reference marker genes

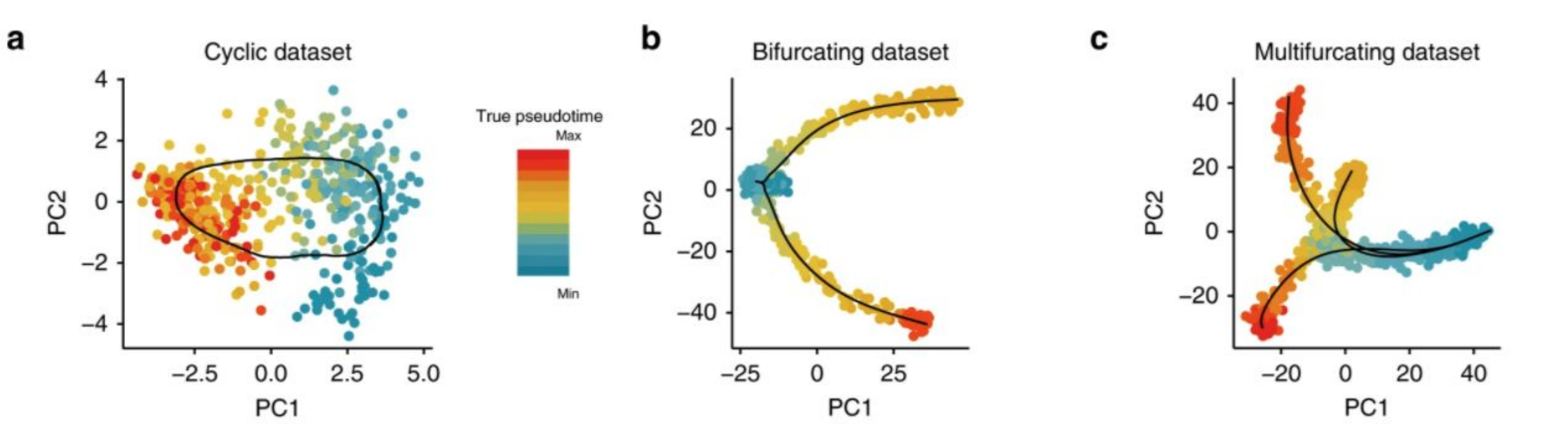

UMAP_2
10
5
0
-5
-10
-5
0
5
10
UMAP_1
0
1
2
3
4
5
6
7
8


MS4A1
GNLY
CD3E
CD14
FCER1A
FCGR3A
LYZ
PPBP
CD8A
UMAP_1
UMAP_2
15
10
5
0
-5
-10
-10
-5
0
5
10
4
3
2
1
0
6
4
2
0

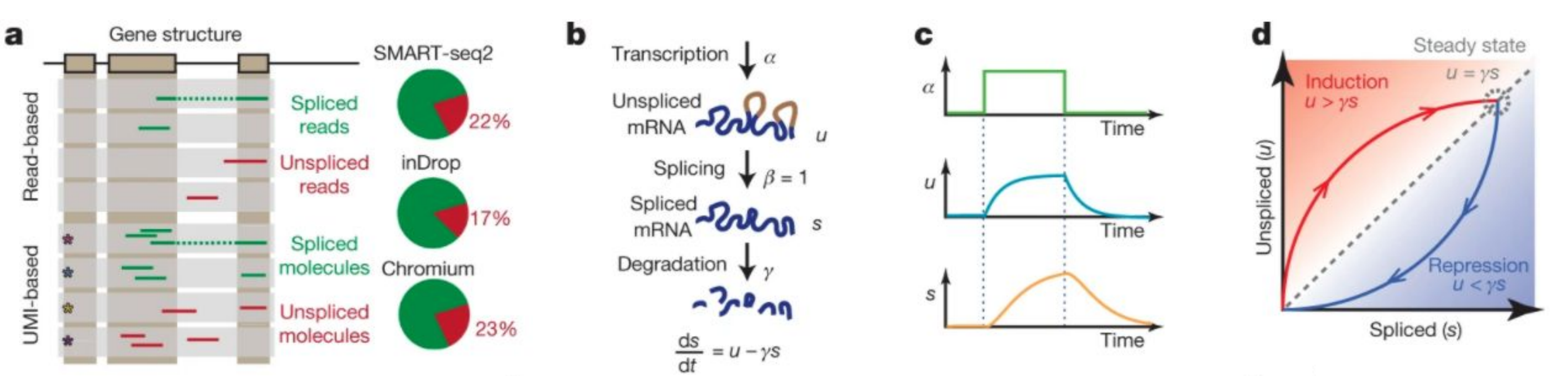

B
CD14+ Mono
DC
Platelet
FCGR3A+ Mon
NK
CD8 T
Naive CD4 T
Memory CD4 T
UMAP_2
10
5
0
-5
-10
-5
0
5
10
UMAP_1

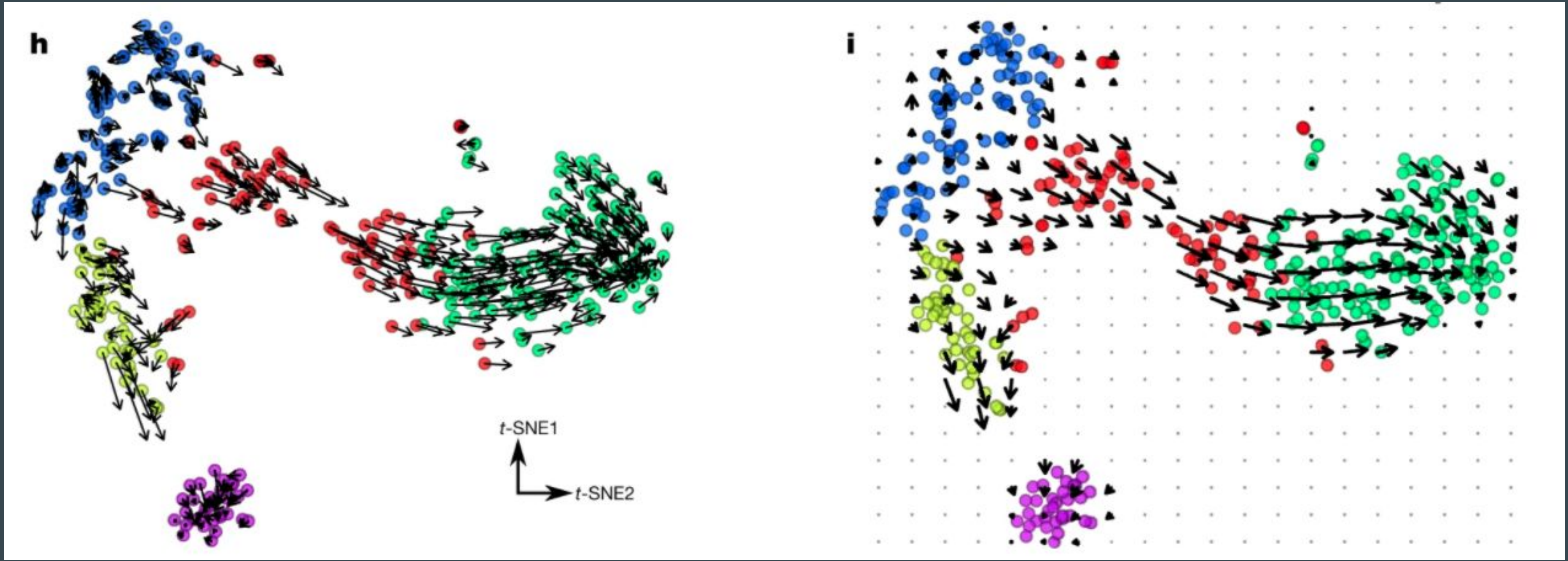
Trajectory inference

- Most biological processes can be described as continuous phenomena
- We can infer “developmental” trajectories over low dimensionality spaces
- The trajectories have to be validated experimentally
- Concept of pseudotime



RNA velocity




h
t-SNE1
t-SNE2
i

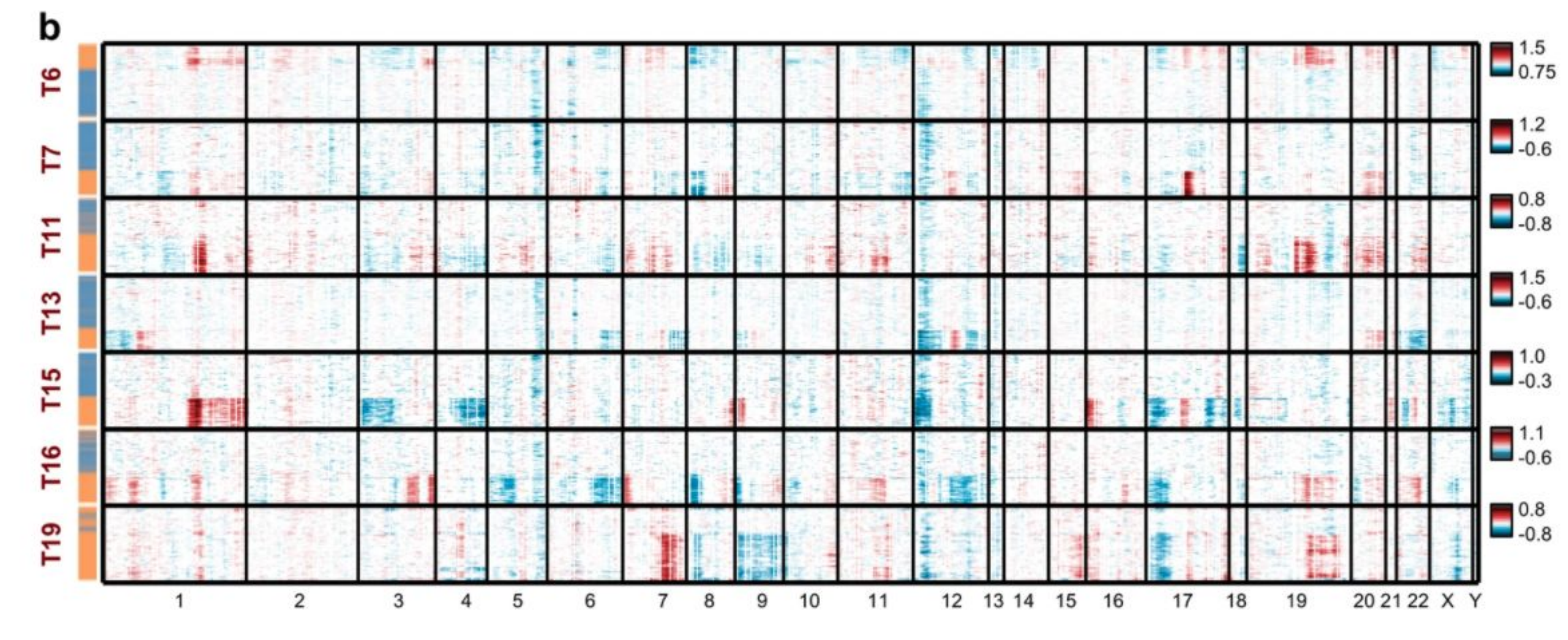
Differential expression

- Generally it has been showed that scRNA-seq specific methods does not consistently perform better than classical bulkRNA tools
- For very simple experimental designs simple approaches such as wilcoxon rank test or t-test have good results
- In a recent benchmark, MAST has been shown to be the best performer over a consistent set of samples, it also allow for complex designs

Gene Set Enrichment Analysis (GSEA)

- Huge set of differentially expressed genes are usually too big to be interpreted
- GSEA is a procedure (there exist different implementations) for calculating the enrichment of DE features against a known list of genes
- Can be pretty generic, is important to choose the right databases

CNV putative inference



That's all Folks!