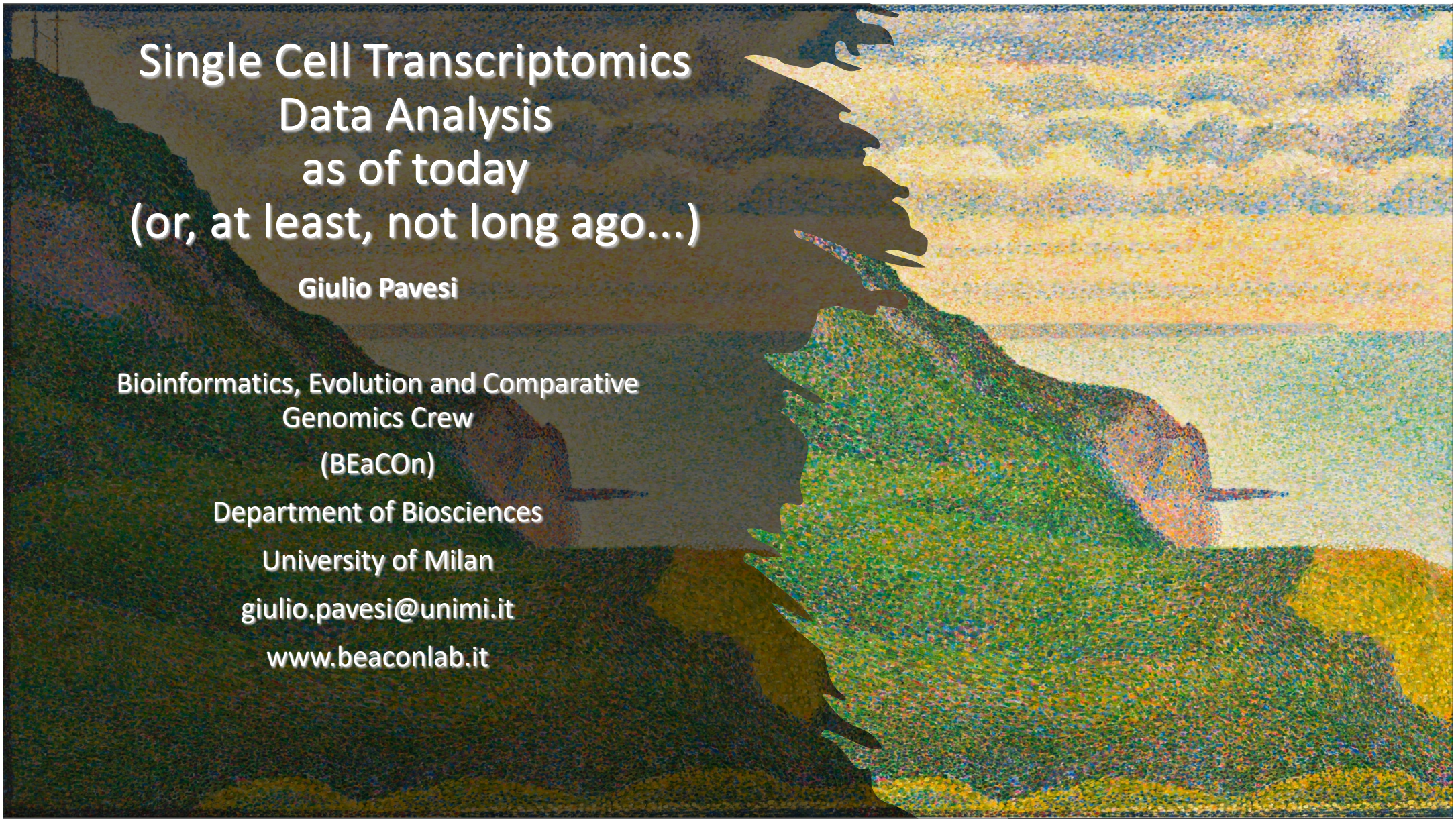
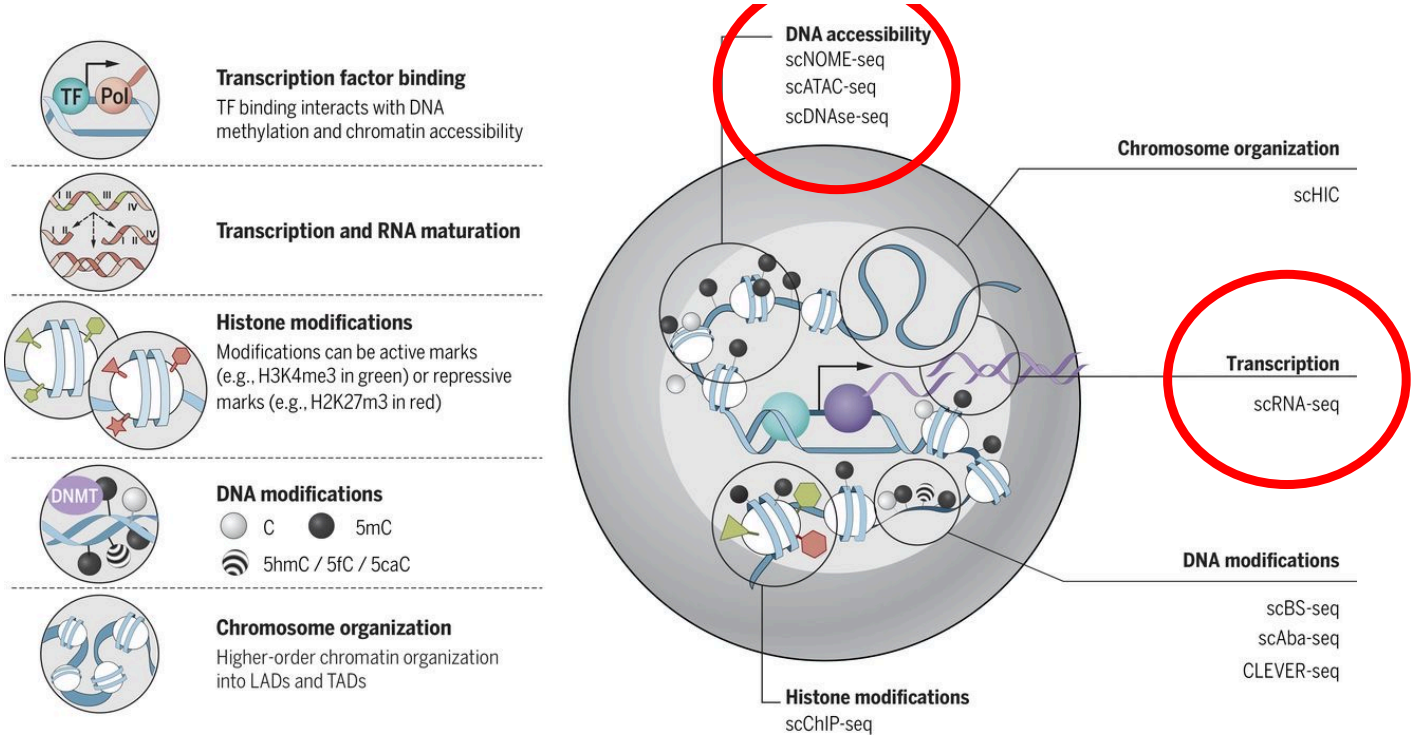
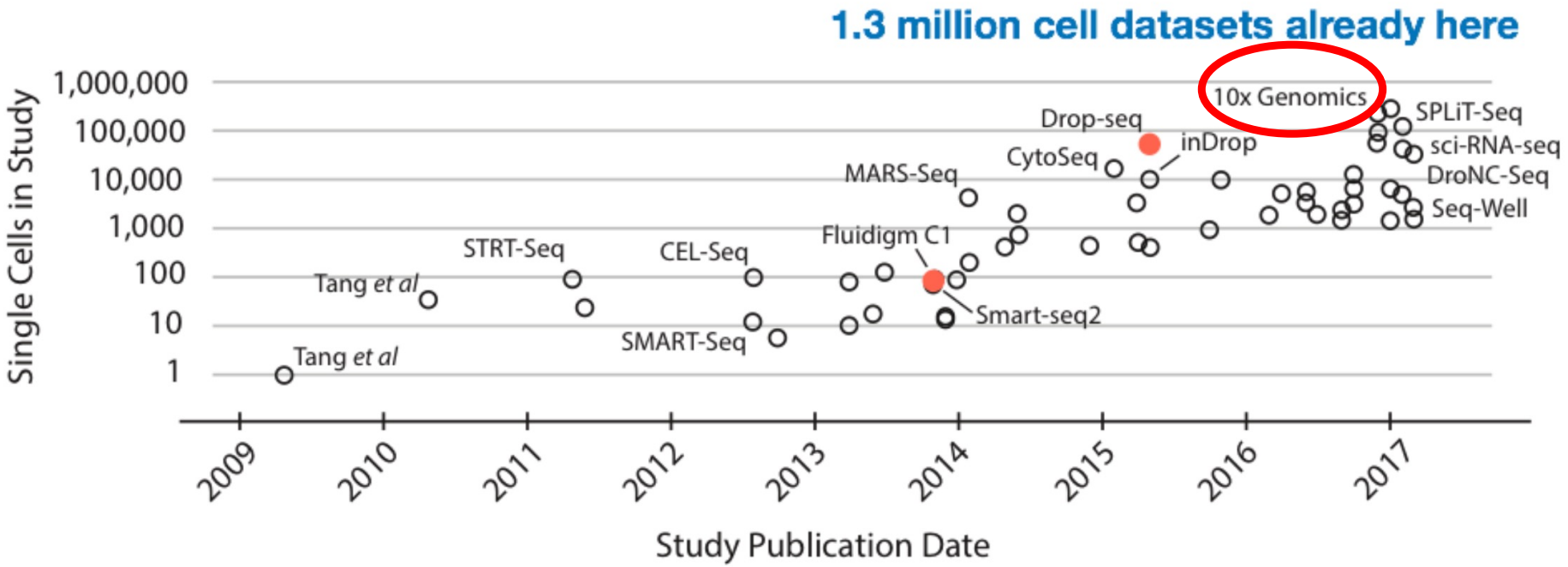

Single Cell Transcriptomics
Data Analysis
as of today
(or, at least, not long ago...)
Giulio Pavesi
Bioinformatics, Evolution and Comparative Genomics Crew
(BEaCOn)
Department of Biosciences
University of Milan
giulio.pavesi@unimi.it
www.beaconlab.it

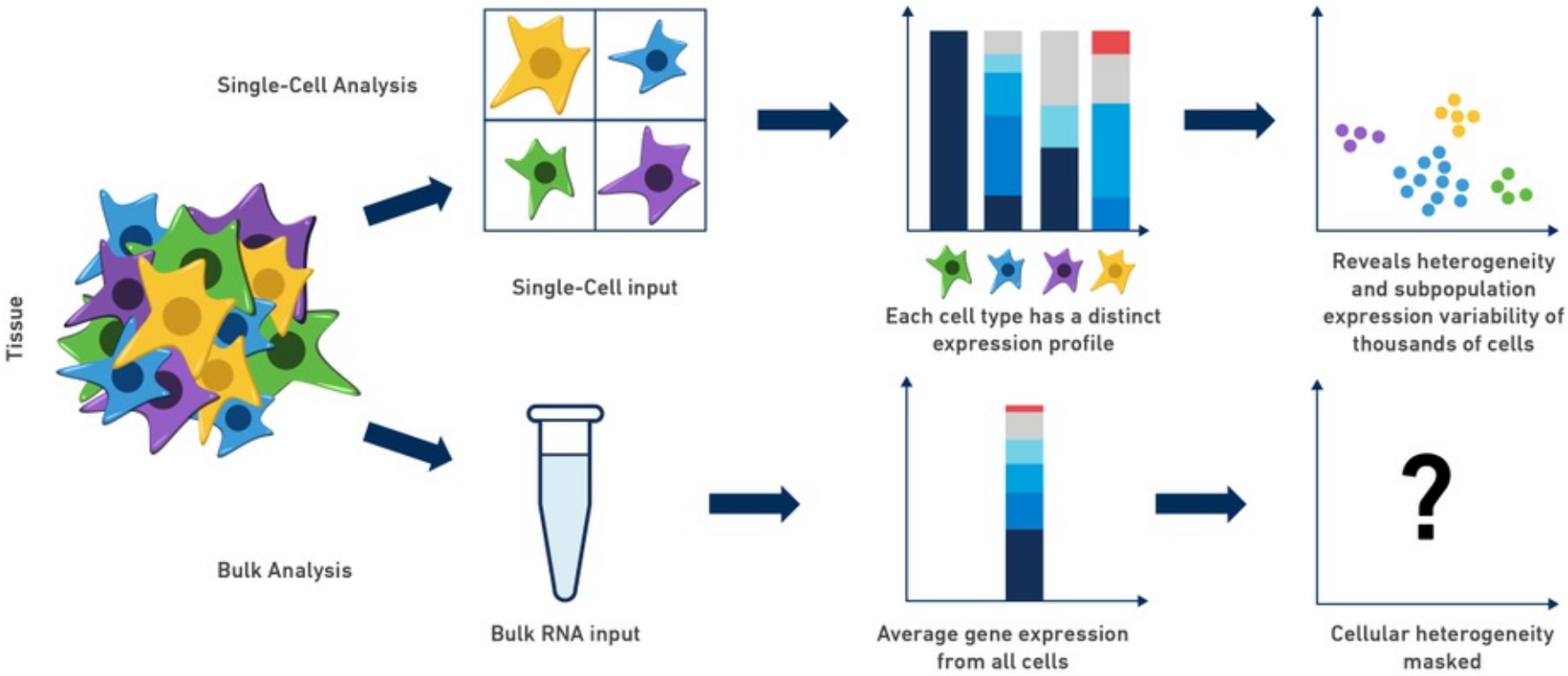


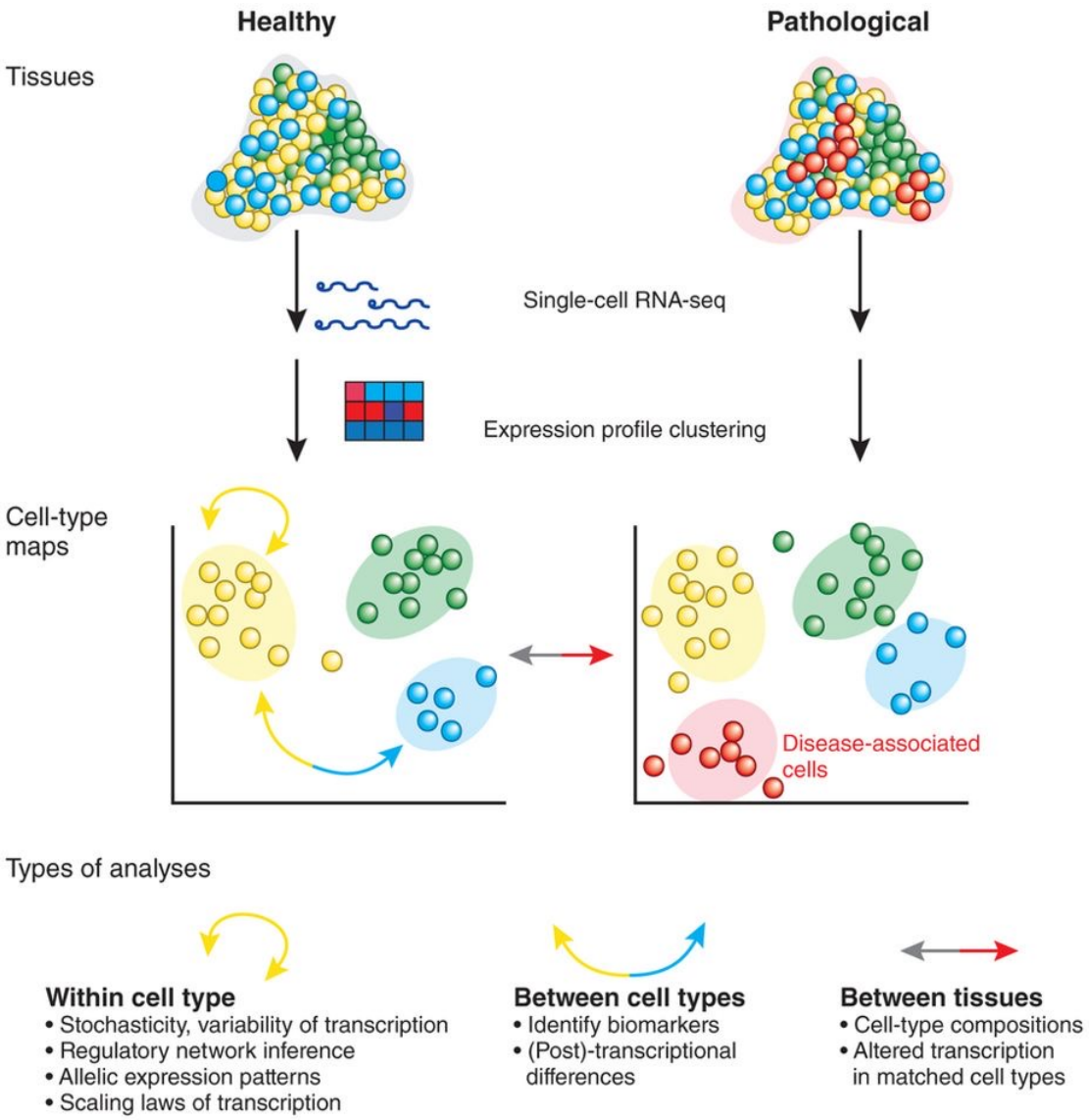
- Single cell genomics and epigenomics
  - Genome and exome sequencing
  - Transcriptome characterization and quantification
  - DNA methylation
  - Histone modifications and chromatin structure
- ... and so on: in principle, what can be done on a "bulk" of cells, can be done on a single cell

scRNA-Seq – different approaches



RNA-Seq: single cell vs bulk





Each cell is a dot in the plot.

The position of each cell depends on the expression of its genes.

“Similar” cells are thus close in the space.

Cells of the same type “cluster” together

The main goal of the analysis is to produce a beautiful colored picture with different cell groups (types) clearly separated

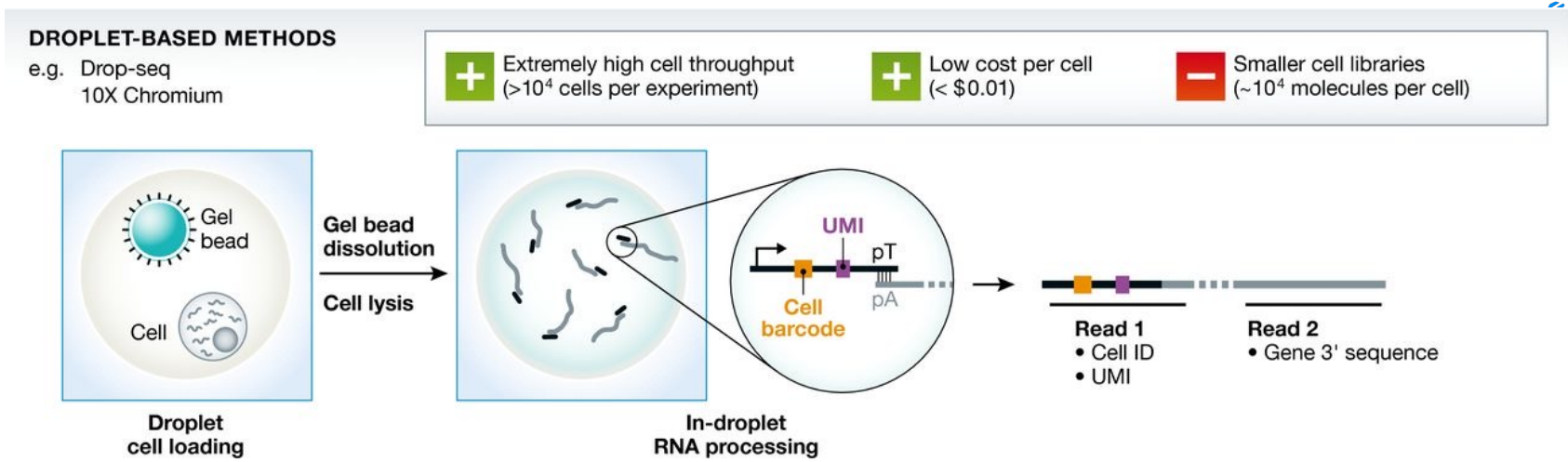
We can then investigate each group/type

Cell processing

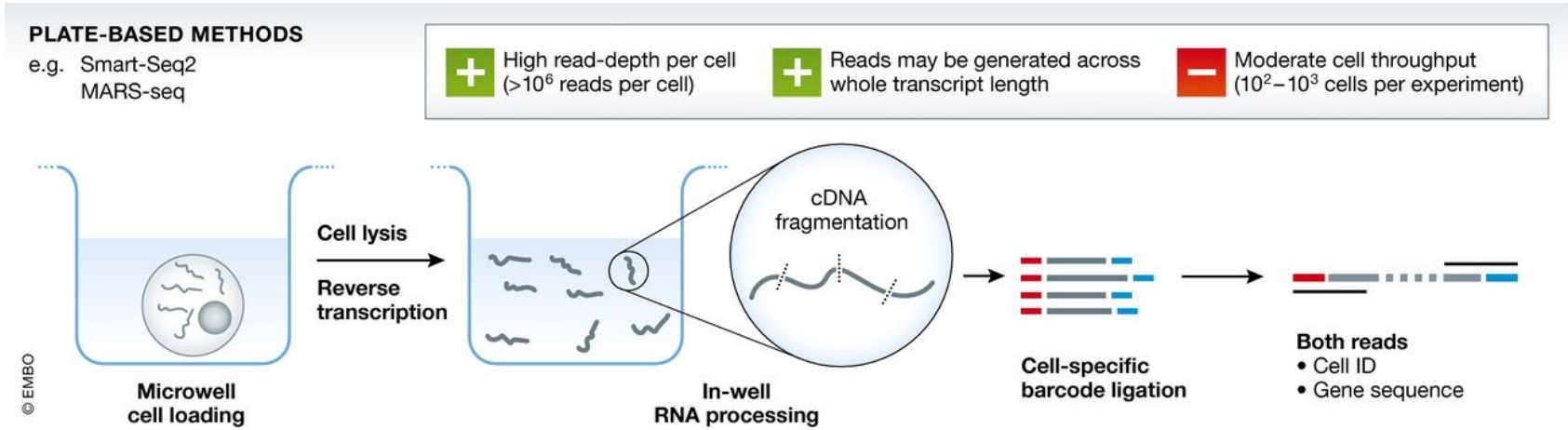
- Several different single cell platforms and protocols have been introduced over the years, that differ in:
  - How single cells are processed
  - How RNAs are processed and converted into cDNAs
  - How cDNAs are fragmented and sequenced
  - How the resulting sequences are processed
- The only thing we can assume to be common is that the sequencing will be Illumina-Solexa again, but with great differences in how the sequencing itself is performed from the original RNAs
- That is, usually **it is not the "random fragmentation + sequencing of fragments"** covering the whole cDNA usually resulting from bulk RNA-Seq

Cell processing

- From a logical point of view, current platforms and protocols can be divided into two main categories:
- "**Few cells/high sequencing depth**"**:** can process up to a **few hundred** of cells in a single run, and the sequencing is quite similar to "bulk RNA-Seq", that is, we can have up to **millions of reads per single cell -> usually "sorted"** (we know what cell types we are looking at)
- **"Many cells/low sequencing depth":** can process up to **tens of millions of cells in a single run**, but with sequencing depth just of **thousands of reads per single cell**, and where reads do not cover the whole RNA as in bulk RNA-Seq -> the most widely used today and the one we will see more in detail -> usually "**unsorted"** (we do not know in advance which cell types are in our sample, or at least the abundance of each one)



Many cells, low depth just one read from each RNA

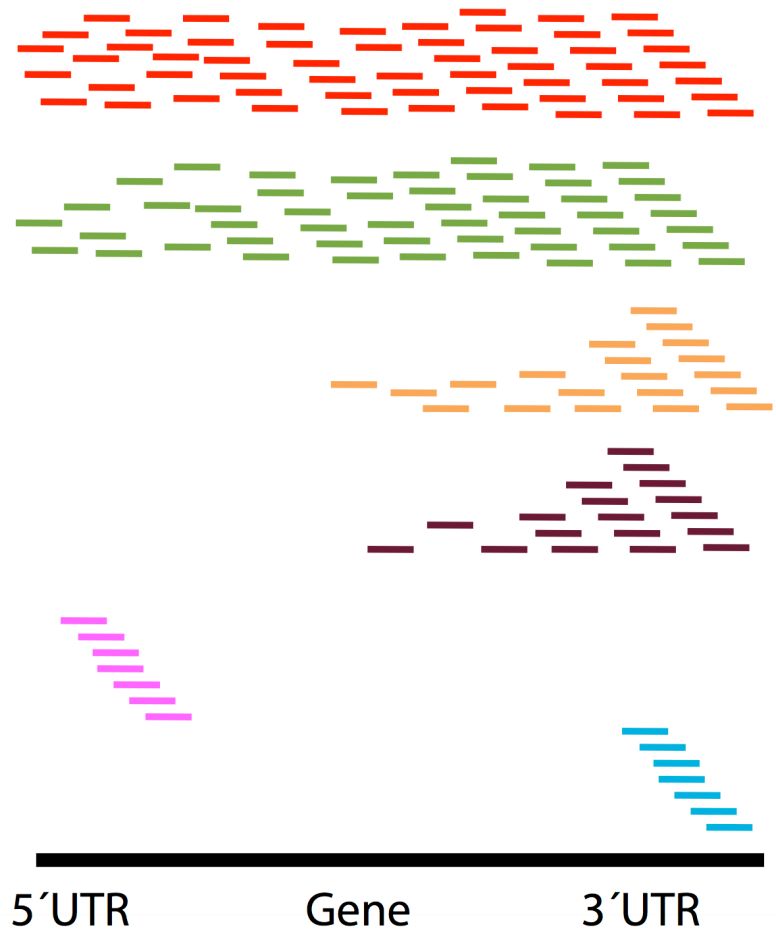


Few cells, high depth, several paired end reads per RNA

In summary

- Regardless of single cells are isolated and processed, from the logical point of view the difference is the one outlined before: **“few cells/high depth” versus “many cells/low depth”**
- But, the above choice also has a substantial impact on how sequencing is performed
  - **High depth:** sequencing is essentially similar to ”bulk” RNA-Seq, and reads will cover the whole transcript
  - **Low depth:** what we sequence are indeed “**tags**”, that is, reads that come from only a small portion of the RNA but are specific for each RNA, and thus we can use them **for quantification only**
- **“Tags”**: nomenclature coming from early studies in RNA sequencing for quantification (e.g. SAGE, digital expression profiling), where only one or more “unique identifiers” were sequenced for each of the RNAs

Different sequencing strategies, different coverage



SmartSeq2
(Picelli et al. Nature Methods 2014)

SmartSeq – SMARTer kit
(Ramsköld et al. Nature Biotech 2012)

Quartz-seq
(Sasagawa et al. Genome Biology 2013)

Tang et al.
(Nature methods 2009)

STRT
(Islam et al. Genome Res 2011)

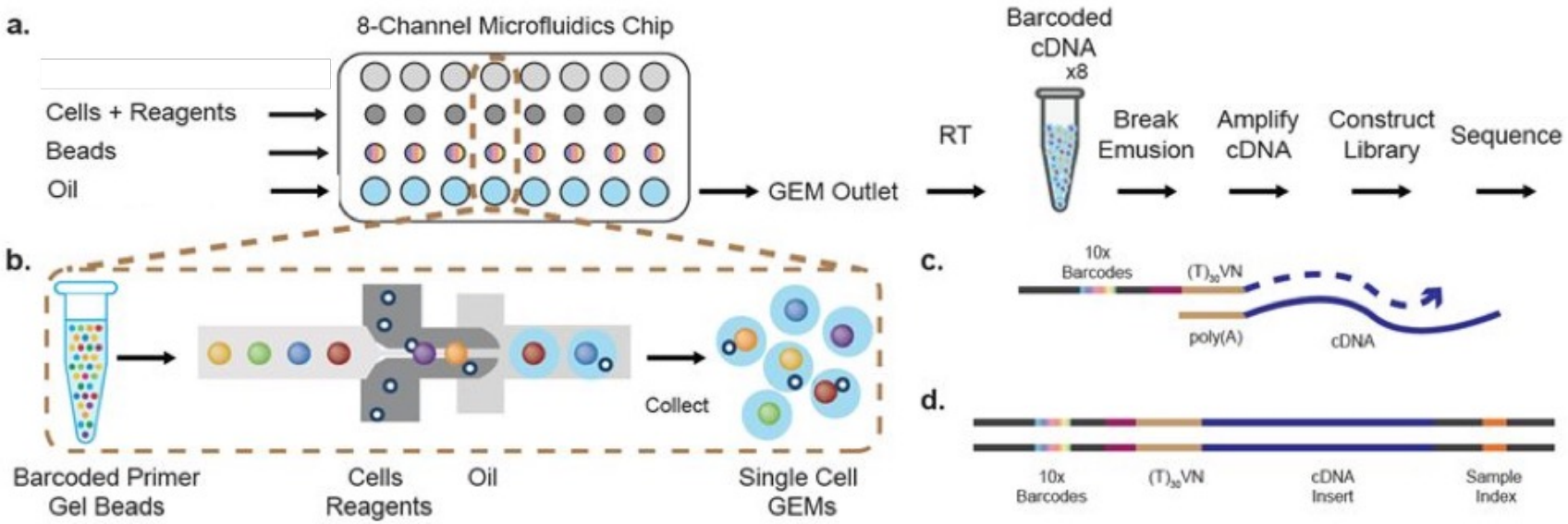
CEL-Seq
(Hashimshony et al. Cell Reports 2012)

SmartSeq(1 and 2): coverage of the whole transcript

All others: bias towards the 3'end
OR: sequencing restricted only to the 5' or 3' end of the RNA

The 10x Genomics Chromium technology

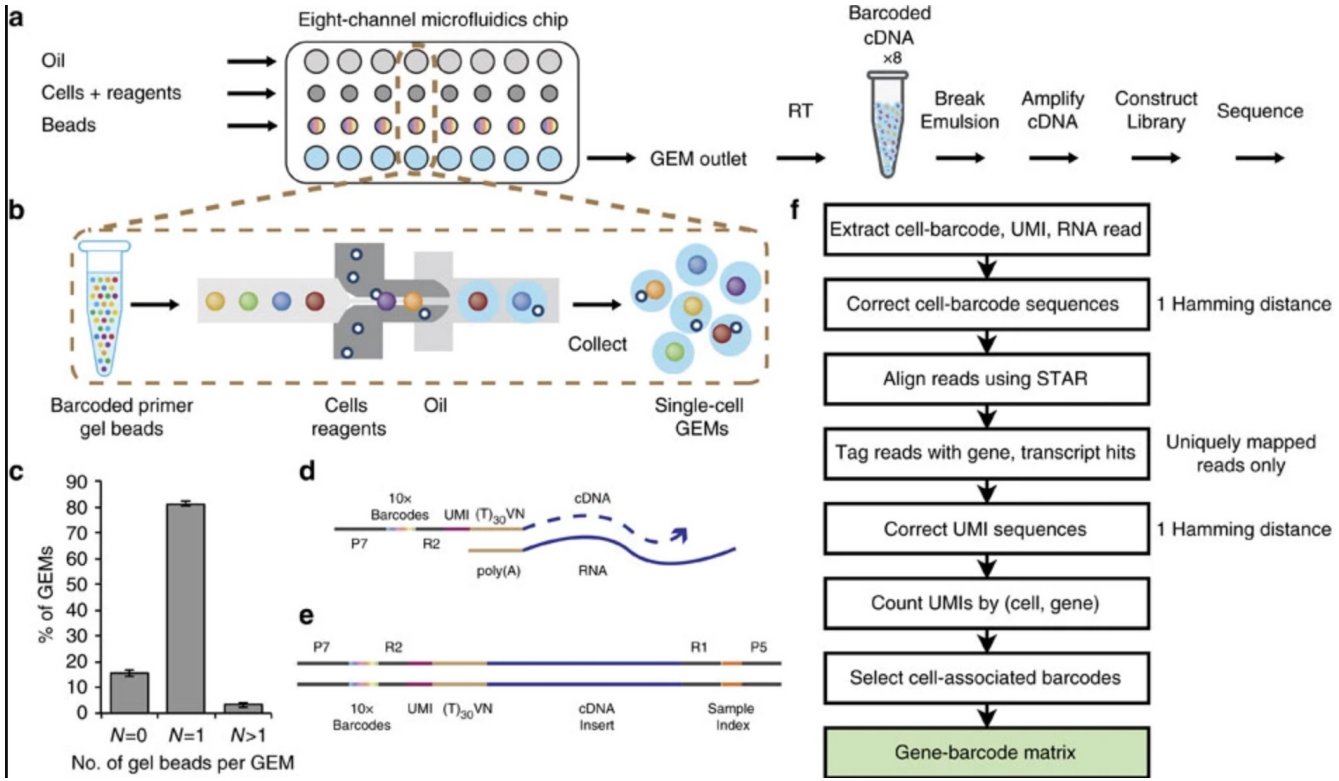
Gel bead in EMulsion (GEM)



As of today, the most likely technology you're going to work with in scRNA-Seq: combines microfluidics with droplet-based techniques (processing of tens of thousands of cells in a single experiment)

https://www.nature.com/articles/ncomms14049

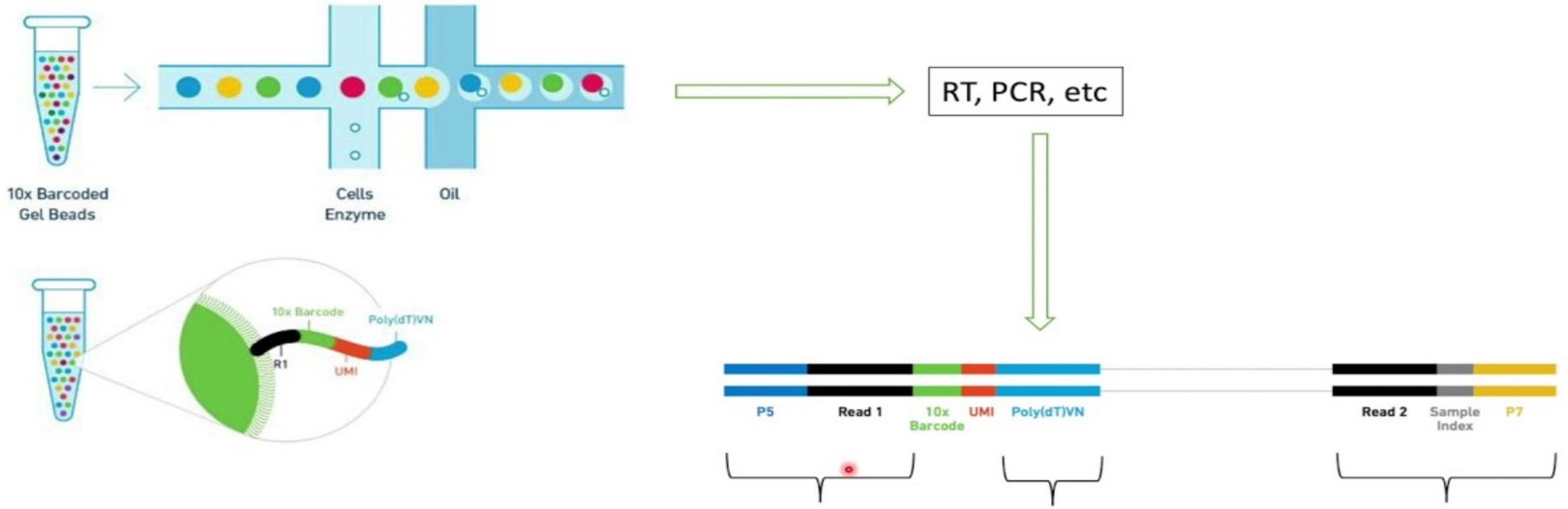
The 10x Genomics approach



https://www.nature.com/articles/ncomms14049

To the right, the different steps that have to be followed in order to process the sequencing reads
Luckily enough they are all encompassed in a single pipeline/software developed ad hoc (**CellRanger**)
However, the different steps are worth being explained in order to better understand how the technology works

The 10x Genomics approach



A closer look to a bead and the probes fixed on it. The poly-T captures the RNAs by hybridizing their poly-A. Then, all the probes of the bead have the same 10x barcode, different from the beads in the other droplets, that permit to identify the RNAs that came from the droplet itself. R1 “marks” the fact that it is the beginning of a 10x read, followed by the cell barcode, followed by a special additional barcode called “UMI”. Sequencing adapters (P5, P7) are finally attached to the cDNA fragment.
We will see what the “UMI” is in a few slides...

Processing RNA (in 10x): the basics

- First thing to be noticed: it **works only for poly-A RNAs**, since the poly-A itself is essential for the processing of the RNAs/cDNAs
- **It can produce at most one fragment and read pair for each RNA molecule**
- The actual region sequenced comes from the 3' of the RNA: hence transcript assembly is not possible (alternative protocols sequence the 5' - the logic remains the same)
- The goal is to sequence a **"tag"** unique for the transcript itself, coming from a restricted portion of its 3' end
- Thus, it is a **"digital expression profile" experiment:** the number of "tags" assigned to each RNA/gene will be used to quantify its expression

How many cells and how many reads?

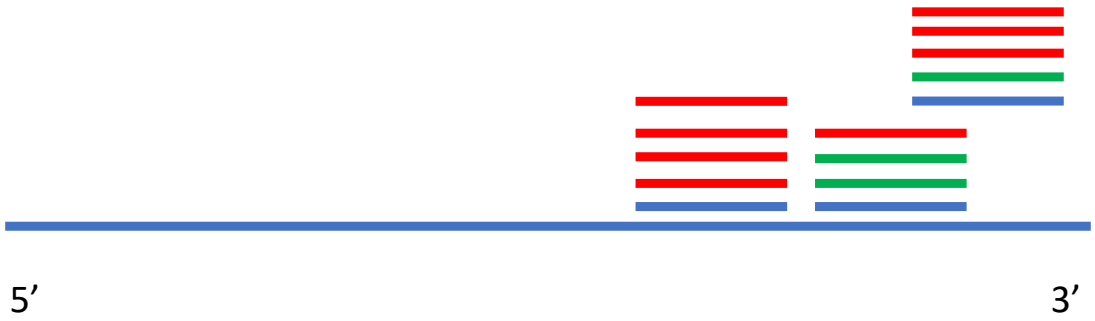
- A "**cell type**" has to be **redundant** in order to be identified and characterized: it can be identified if we have enough cells representing it
- Rule of thumb for 10x: around 20-30 cells from each expected cell type
- **Preselecting cells is possible**, but unbiased cell selection is better
- Individual mammalian cells contain 50,000–300,000 transcripts
- The 10X processing can produce **at most one read per transcript**
- Although up to several hundred thousand transcripts may be expressed per individual cell, up to 85% of these are present at only 1–100 copies.
- Critically important in scRNA-Seq to capture low-abundance mRNA transcripts and **amplify the synthesized cDNAs** to ensure that all transcripts are represented in the library sent to sequencing

Problem: PCR amplification is essential!

- **PCR amplification**: the bane of every read quantification-based NGS experiment
- Unfortunately, **a necessary step** in several protocols, including all those for scRNA-Seq (0.1 pg of "usable" RNA per cell, and most of it will be from ribosomal protein genes)
- **In theory:** every RNA/fragment has the same probability of being amplified
  - **"Duplicate" reads must count as 1** in downstream processing – we do not count PCR amplified fragments
- In practice, not, hence two main choices:
  - **Hope that everything works anyway**, duplicate reads count as 1, and not much is lost
  - **Devise additional strategies** for sequence preparation to cope with this problem

PCR amplification

- 10x- like approach



Green: identical fragments coming from different RNAs

Red: identical fragments due to PCR amplification

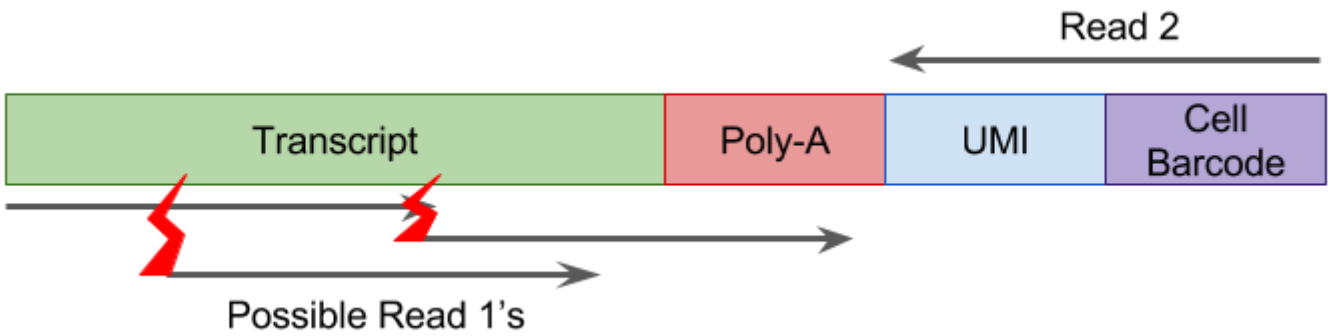
Mapping of reads along a transcript

Since the sequenced region restricted to
the 3’ of the RNA molecule
it is more likely that two identical fragments
are present in the library to be sequenced

Thus, we will find many more reads mapping
at the same position, and in this case
we cannot tell whether they were actually
from different RNAs, or
from PCR amplifications of the same fragment

“Unique molecule identifiers” (UMI)

- Idea: before PCR amplification, attach to each fragment a “Unique molecule identifier” (UMI), which is essentially a random oligonucleotide
- The RNA fragment to be sequenced will look like this:

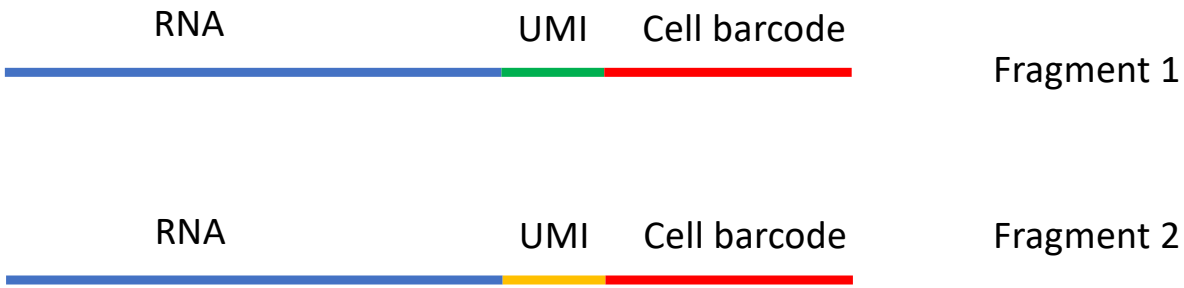


Cell barcode: a unique oligo for all the fragments of a cell
UMI: a random oligo (10 bp) attached to each fragment
For quantification, we will use read 1 after PE sequencing

Read2 will tell us the “UMI” and the cell the read came from

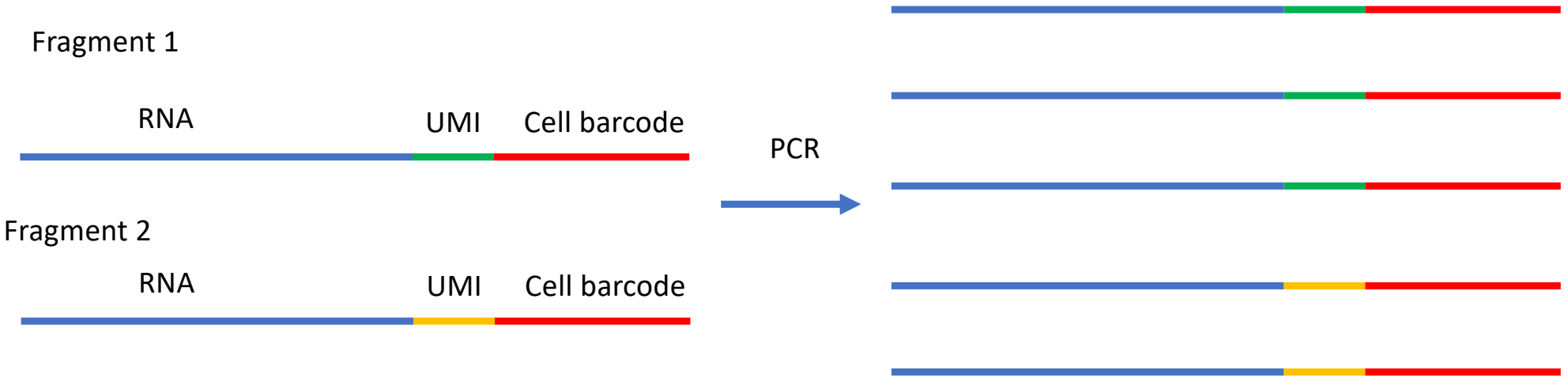
“Unique molecule identifiers” (UMI)

- Suppose that we have two identical fragments in the library coming from the same RNA and not from PCR amplification
- After processing, they will have the same RNA sequence, the same cell barcode, but different UMIs

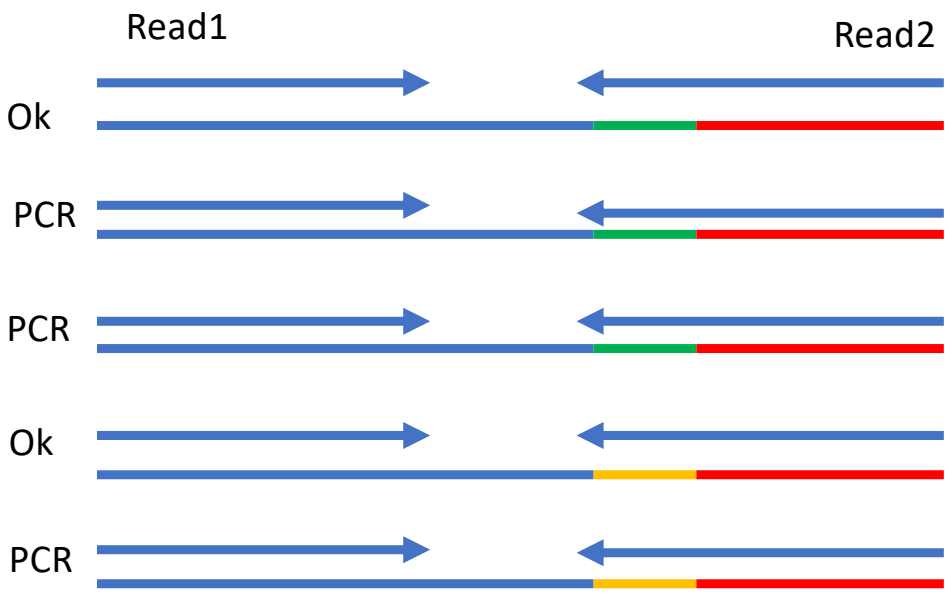


“Unique molecule identifiers” (UMI)

- PCR amplification will make copies of the whole fragment, including the UMI

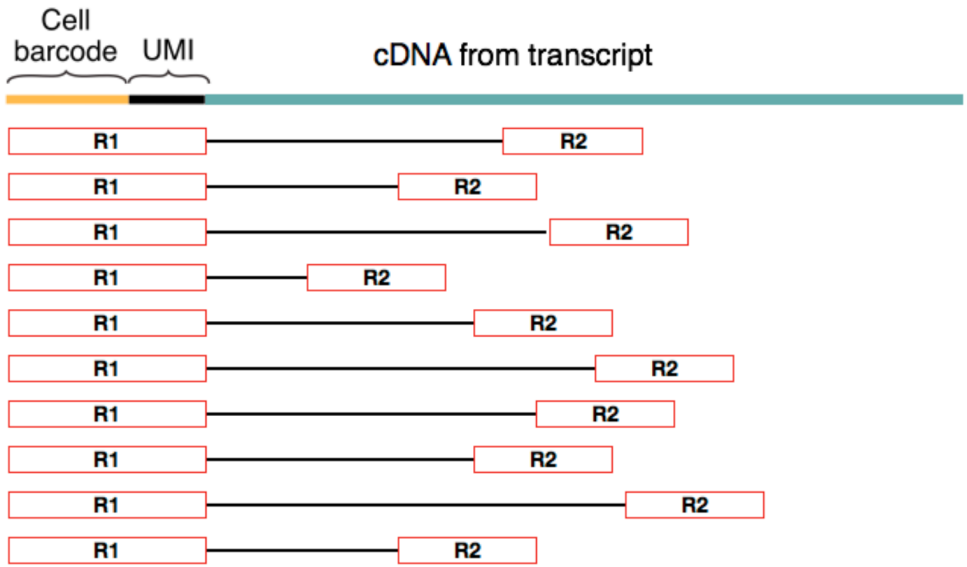


“Unique molecule identifiers” (UMI)

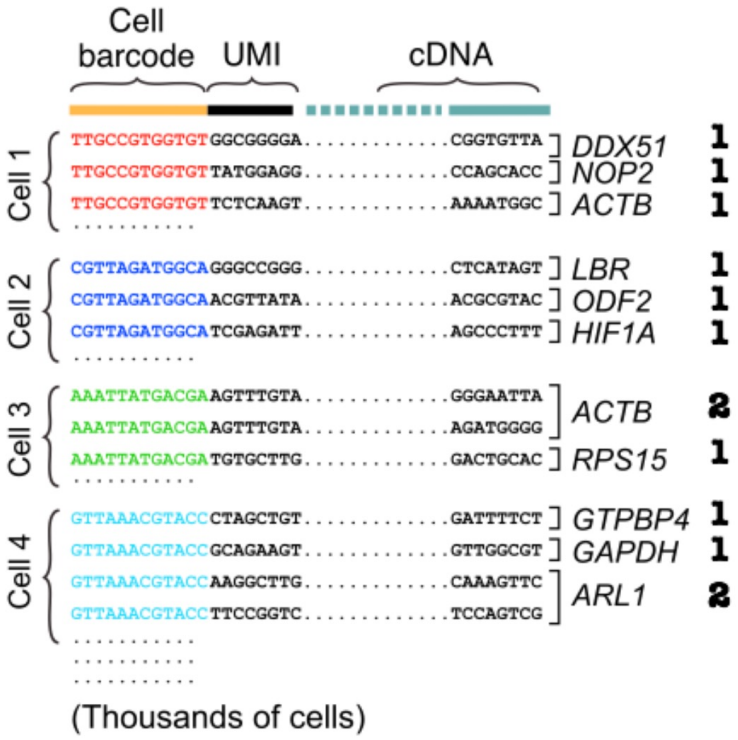


- After sequencing, “genuine” reads coming from different fragments will have
  - Identical read1
  - Different read2, since the UMI is different
- PCR amplicons will have
  - Identical read1
  - Identical read2, since the UMI and cell barcode are identical
- So, in this example, we are able to reduce the 5 reads mapping at the same position to 2 different fragments, and not 1 as in without UMIs

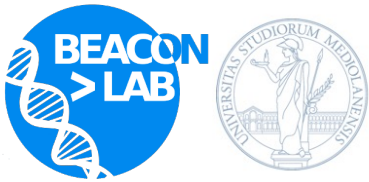
10x Sequencing, in summary



Biased paired-end reads.



The “count table”: each “sample” is a single cell



	Sample1	**Sample2**	**Sample3**	**…**	**SampleN**
Gene1	count(1,1)	count(1,2)	count(1,3)	...	count(1,N)
Gene2	count(2,1)	count(2,2)	count(2,3)	...	count(2,N)
Gene3	count(3,1)	count(3,2)	count(3,3)	...	count(3,N)
…	…				
GeneM	count(M,1)	count(M,2)	count(M,3)	…	count(M,N)
SUM (Lib. Size)	Tot(1)	Tot(2)	Tot(3)	…	Tot(N)

The “count table”

- If a gene in a cell as at least one read, then we can consider it to be expressed in that cell
- But sequencing depth can be very different according to the assay employed, from >1M reads per cell (e.g. Smart-Seq) to a few thousands (10x)
- Keep in mind that even restricted to poly-A transcripts, there will be a small percentage of genes “eating up” most of the reads (ribosomal protein genes, usually)
- Hence, the number of genes actually “visible” to be expressed is in turn influenced by sequencing depth – but **increasing sequencing depth does not help** (we will just keep sequencing PCR replicates of the original fragments)

The “count table”

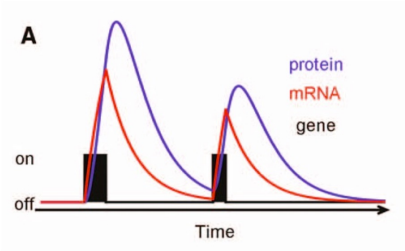
- With **“cell quality” control** we filter from the table all cells (columns) that have:
  - **Too few reads** in general (likely an empty droplet)
  - **Too many reads** mapped on mt genes (dead or suffering cells)
  - **Too few nuclear genes “detected”** as expressed (often for either of the previous two problems)
  - **Too many reads/too many expressed genes**: indications of being not a single cell but a “doublet” (one droplet containing two cells)
- The selection is often relative to the other cells, that is, we filter out those cells that are “outliers” for the above parameters with the overall behavior of the cells
- But we are not ready to work on the data yet – one more issue to be considered

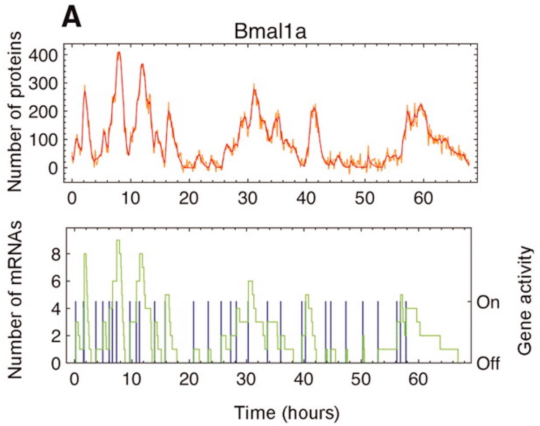
“Dropouts”

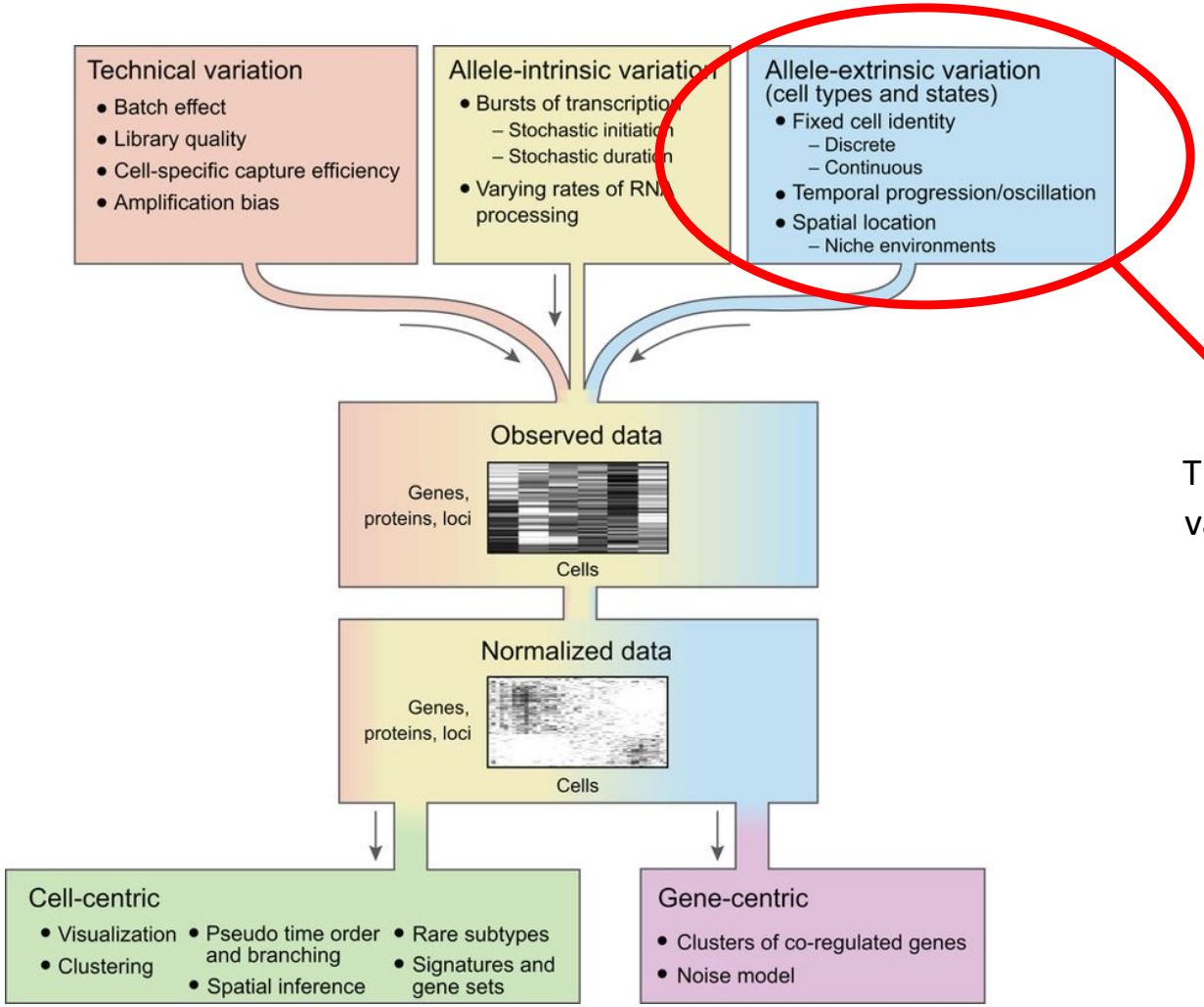
- We can assume in principle that **“the same cells” will have a similar expression profile**
- That is – even before considering normalization - **a gene known to be expressed in a given cell type should have a “sizable” number of reads associated to it in all the cells of that type**
- The surprise is that even a gene that should be “uniformly expressed” by all cells of the same time **disappears altogether**
- That is, in some cells where **we should find reads** associated with it **the count is zero: we have seen a “dropout”**

Dropouts

- The reasons behind these two phenomena are two:
  - technical
  - **biological**
- Technical: sequencing takes place after PCR. If fragments from RNAs of a gene are not “picked up”, the gene disappears
- Biological: transcription is not uniform in single cells (see figure to the right). If we process the cell right when the gene is not being transcribed, we don’t pick up any read from it



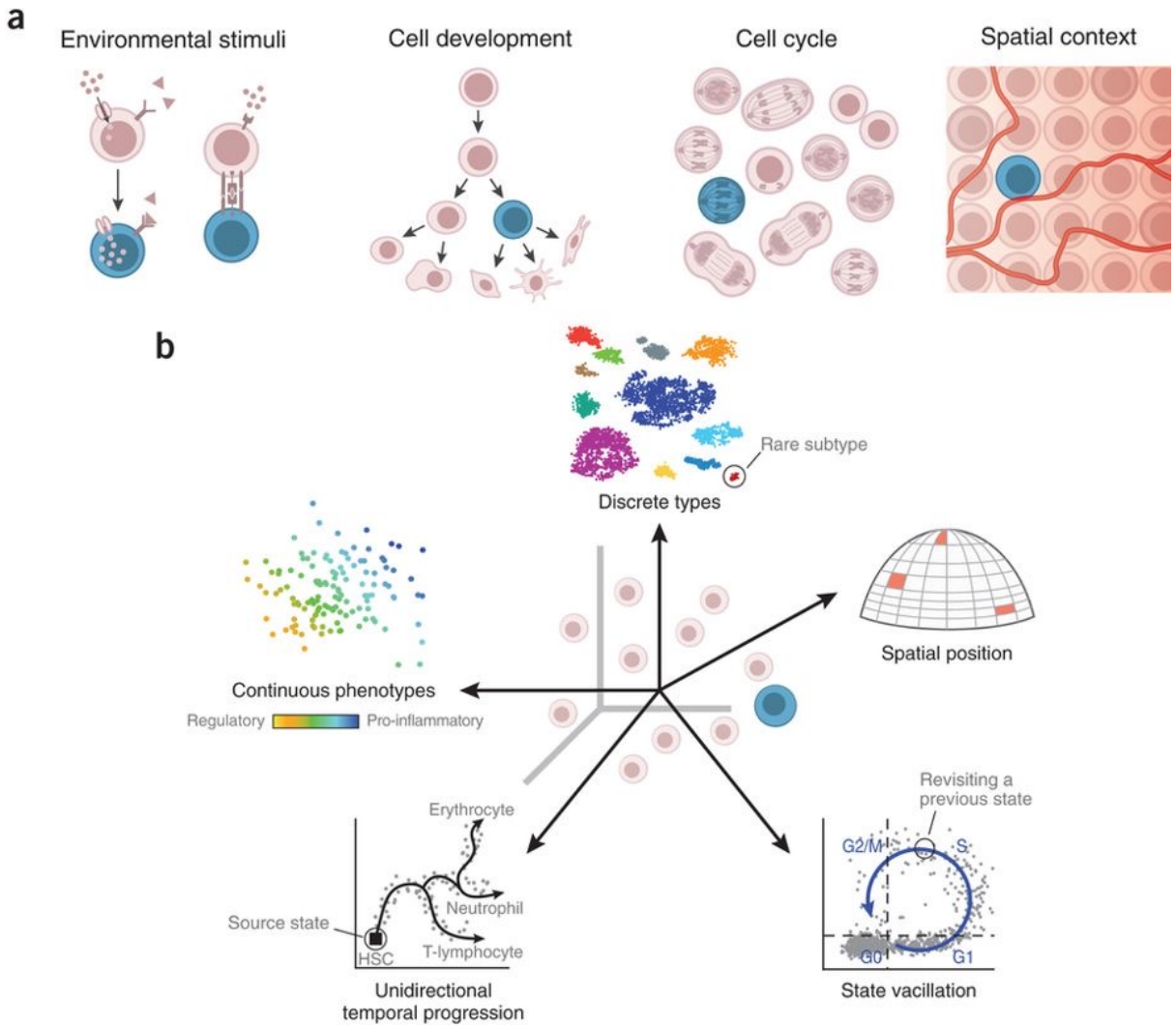




The real biological variation we want to capture!

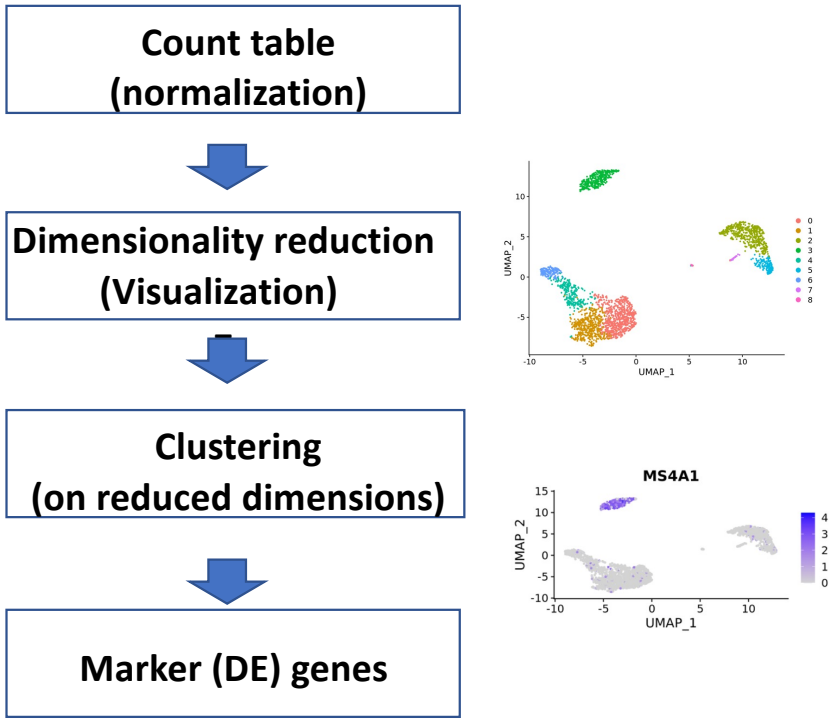
Cell to cell variability

- Different “cell types” have different patterns of gene expression
- Different biological factors contribute to cell-to-cell variability, even within cells that can be considered to be ”of the same type/subtype”
- Different cells of the same “type” can be:
  - at “slightly different” stages of development/differentiation
  - subject to different stimuli
  - at different phases of the cell cycle
  - at different locations in the source tissue
- All the above factors contribute to differences at the transcriptional level also for cells that “look” the same with more traditional methods for cell sorting



Processing the count table

- In general, the main steps are:
  - **normalize** the counts
  - **project and visualize** the data,
  - group cells into "**clusters**" with similar expression profile, corresponding to the same cell (sub)-type
  - find "**marker genes**" characterizing each cluster, e.g. those differentially expressed (over-expressed) in one cluster with respect to the other
  - **further processing**, e.g. finding "sub-clusters" or "trajectories" or integration with other types of data (e.g. scATAC-Seq)



Normalization: do nothing

- For some platforms (like the 10x) the advice is to keep the original counts scaled by the library size of each cell – use the $\log_2$ of counts per million (or counts per 100,000)
- Rationale: these are platforms producing a very low number of reads per cell (e.g. 50-60K – to be split across 20K genes)
- **Hence, the quantification looks more like a "binary signature"** (gene transcribed/not transcribed) rather than a fine-tuned assessment of transcript levels
- Hence, the subsequent comparisons among cells will be more influenced by the "signature" (yes/no) rather than expression values
- **Also advised to "scale" the $\log_2$-counts per million of each gene across the cells - shift the expression of each gene so that the mean expression across cells is 0 and its variance 1:** in this way the impact of the actual quantification is further reduced

After normalization: comparing cells

- After normalization we can start to process the data
- Each cell can be seen as **a point in a n-dimensional space,** where n is the number of poly-A genes in the annotation used (about 20-25,000 in human and mouse)
- Each of the n coordinates is defined by the normalized read count of the corresponding gene
- Hence, **points corresponding to two "similar cells" with similar expression profile should be closer in the space**
- Hence, **points corresponding to a subset of cells of "the same type" should be "clustered" together in the space**
- But, also, each "point" (cell) will have most of its n coordinates set to 0 (genes not expressed or not detected)

Multi-dimensional spaces

- The count matrix is usually **sparse:** thousands of genes will have 0 counts in all (or most of) the cells studied
- Values of 0 will make cells look “similar”, but this information is irrelevant: **we want ”similar cells” expressing the same genes, rather than not expressing the same genes**
- Hence, for starters, **remove all genes with counts = 0 across all cells, or counts > 0 in a very limited number of cells**
- Also, the matrix is still very sparse, **keep only the really “informative” genes, that have the highest variance across the cells studied (usually no more than 2,000-3,000)**
- All in all, compare the points (cells) only for those dimensions (genes) that are really relevant, and removes useless similarities (the zeroes) or genes with “random” variations (genes peaking in just a few, e.g. < 10, cells) or genes with uniform expression values (i.e. the housekeeping genes)

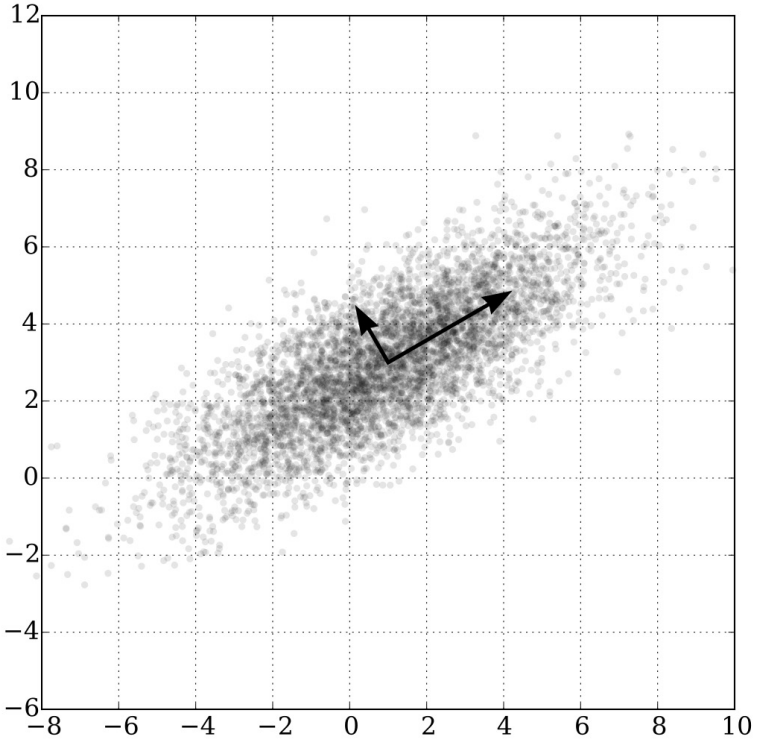
Multi-dimensional spaces

- Even after gene filtering, the number of dimensions remains in the order of thousands (of genes) – too many for downstream analyses
- Before any further processing **dimensionality reduction** is required
- That is, cells are projected into a lower-dimension space, trying to preserve as much as possible their distances in the original full n-dimensional space
- Dimensionality reduction has different goals:
  - Making data **representable for human eye inspection** (i.e. in a two-dimensional space)
  - Making data **treatable for downstream analyses** (i.e. in a lower dimensional space, e.g. 10- or 15-dimensional space)
  - Most important of all, **highlighting the actual sources of variability in the data** and removing irrelevant dimensions with little or no variation among cells (e.g. genes with expression zero in all the cells)

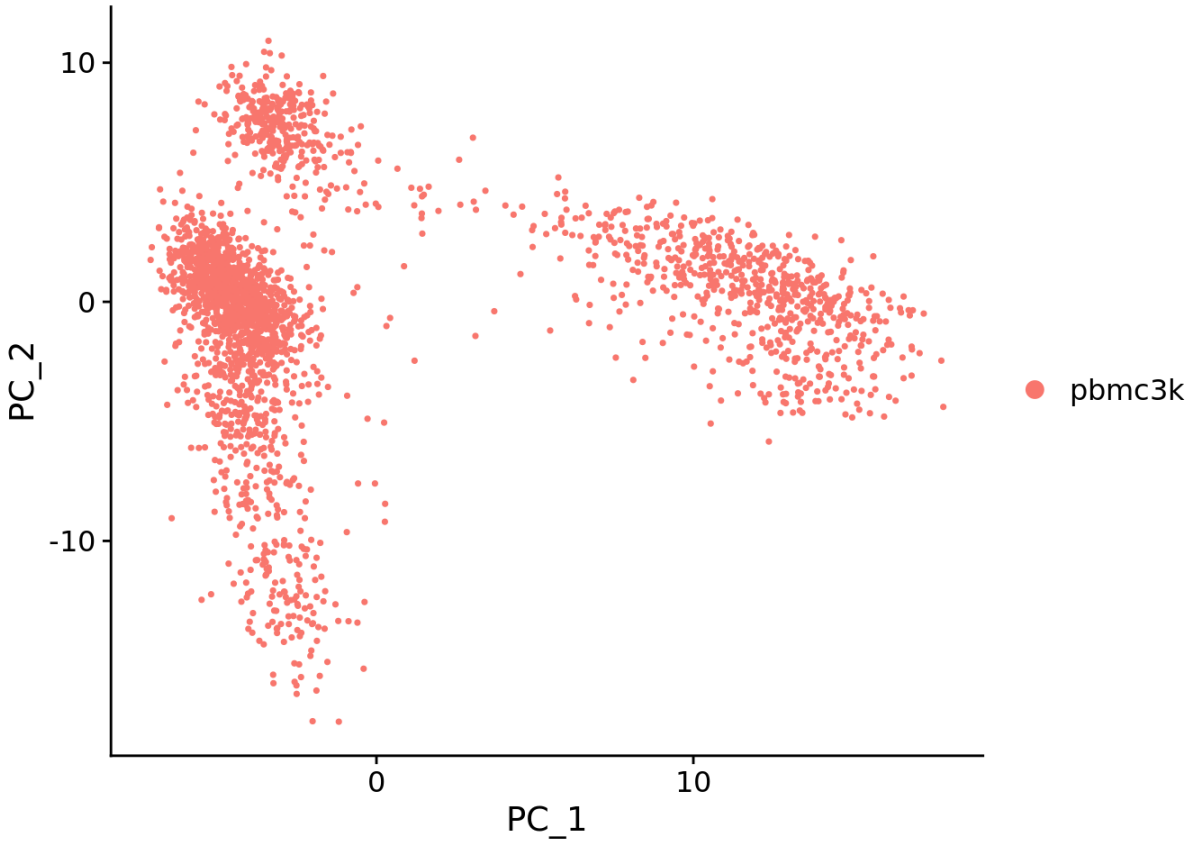
Reduction of dimensionality

- **Principal component analysis (PCA):** widely used also for scRNA-Seq, and indeed usually the "first choice" for all NGS assays for quantification
- Chooses axes in the high-dimensional space that capture the largest amount of variation, and projects points along those axes
- In PCA, **the first axis** (or "principal component", PC) is chosen such that it captures the greatest variance across cells. **The second PC** is chosen such that it is **orthogonal to the first** and captures the greatest remaining amount of variation, and so on.
- The earlier PCs are likely to represent biological structure, as more variation can be captured by considering the correlated behavior of many genes (i.e. genes that "make the difference" tend to show a similar behavior in many cells)
- Random technical or biological noise is expected to affect each gene independently, unlikely to find an axis that can capture random variation across many genes, meaning that noise should mostly be concentrated in the later PCs.
- Only the first and most informative PCs are kept in downstream analyses (there exist simple heuristics for this)

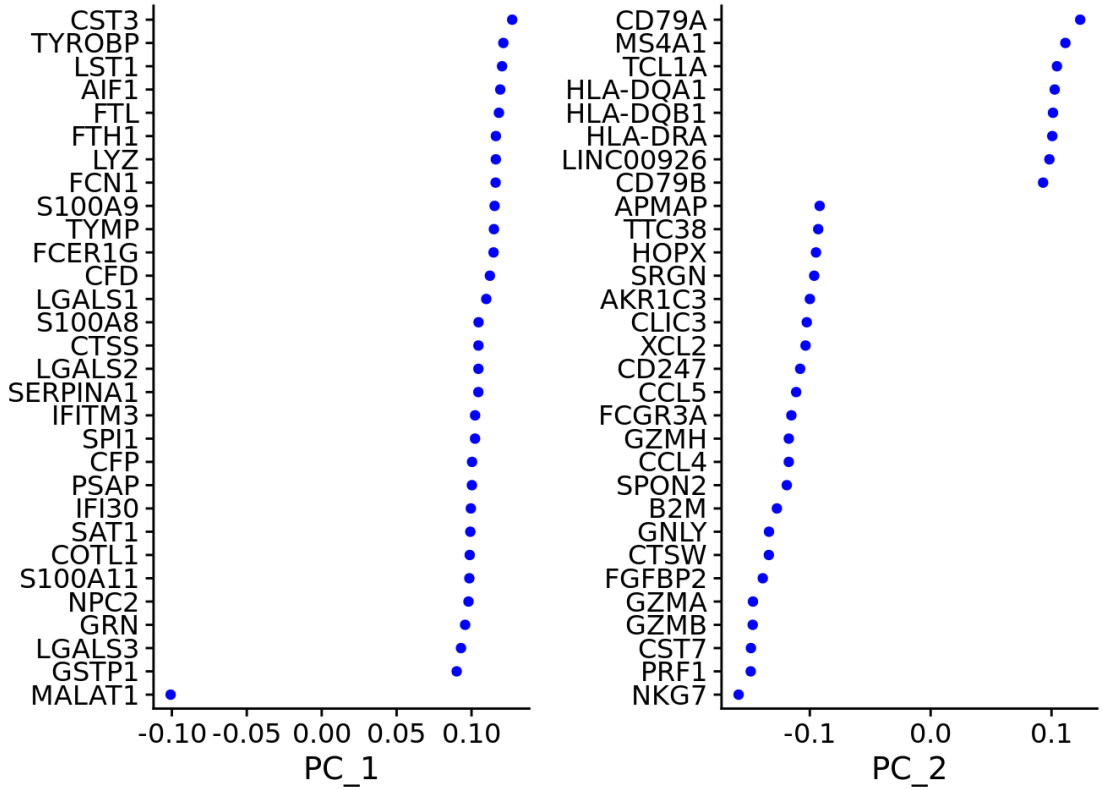
Principal Component Analysis



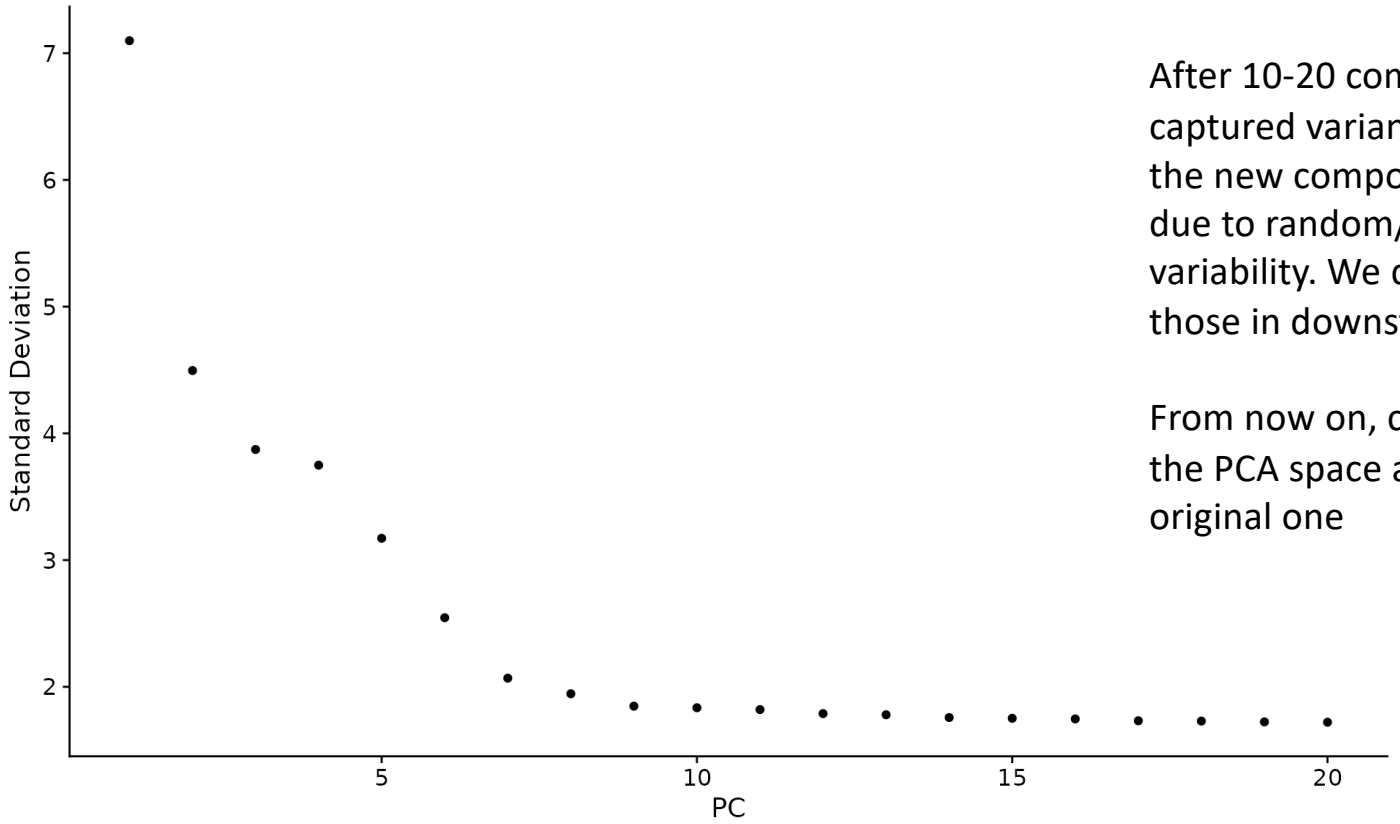
Dimensionality Reduction: PCA (first two)



PCA: most and least variable genes in each PC



PCA: variability across the components



After 10-20 components, the captured variance along the new components is mainly due to random/technical variability. We don't include those in downstream analyses

From now on, cells will be in the PCA space and not in the original one

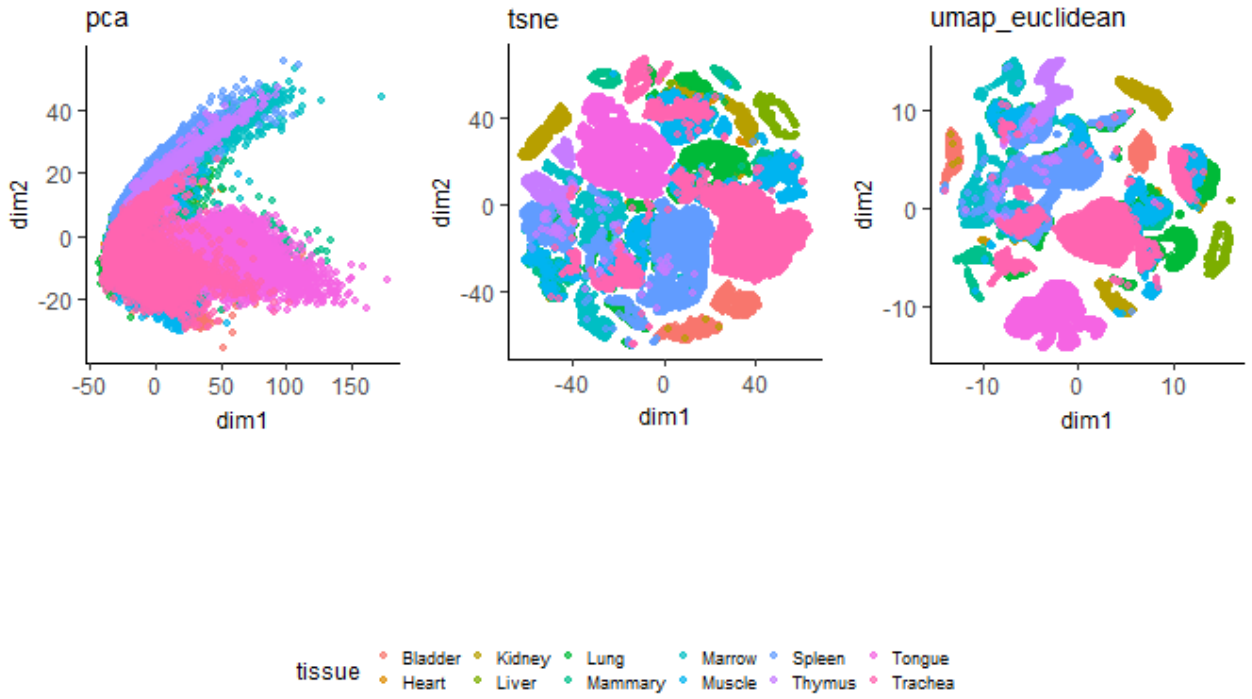
Reduction of dimensionality for visualization

- **t-stochastic neighbor embedding:** for human **visualization** and inspection of data
- Cells are usually represented as points in a **2D space**
- PCA works well by keeping the top *d* PCs, but plotting only the first two PCs often does not provide a meaningful visualization
- Attempts to find a low-dimensional representation of the data that preserves the distances between each point and its neighbors in the high-dimensional space.
- Tends to display “beautiful” clusters in the data
- Not restricted to linear transformations, nor it is guaranteed to accurately represent distances between distant populations.
- It is “**stochastic**”: hence, different runs might have different results

Reduction of dimensionality for visualization

- **Uniform manifold approximation and projection (UMAP)**
- In practice, based on similar principles of t_SNE, virtually replaced it as the "method of choice" for **visualization**
- That is, it also tries to find a low-dimensional **representation** that preserves relationships between neighbors in high-dimensional space.
- Produces more compact (and more beautiful!) visual clusters with more empty space between them. It also attempts to preserve more of the global structure (distance) among clusters
- UMAP is much faster, it is increasingly displacing t-SNE as the method of choice for visualizing large scRNA-seq data sets
- In some cases its result also used as input for the downstream analysis (instead of PCA)

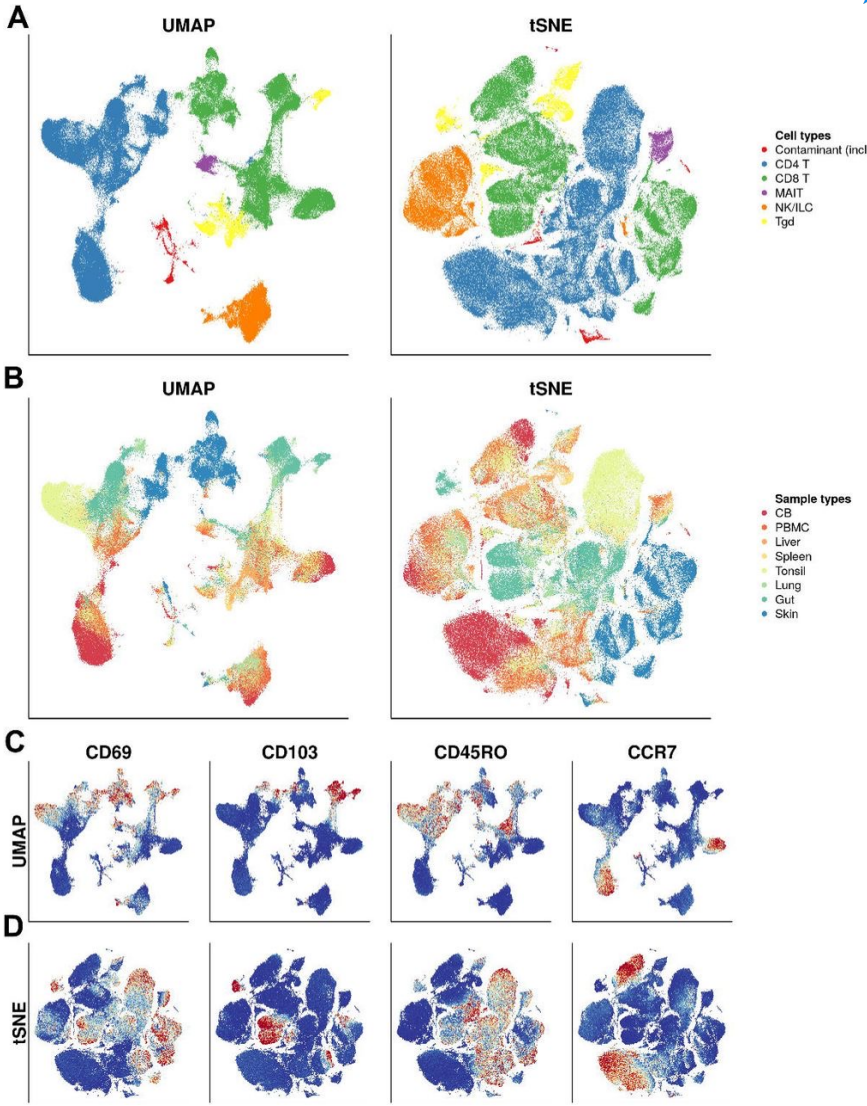
PCA vs t_SNE vs UMAP



Source: «Tabula muris» scRNA-Seq of adult mouse cells

t_SNE
vs UMAP

https://www.biorxiv.org/content/10.1101/298430v1.full

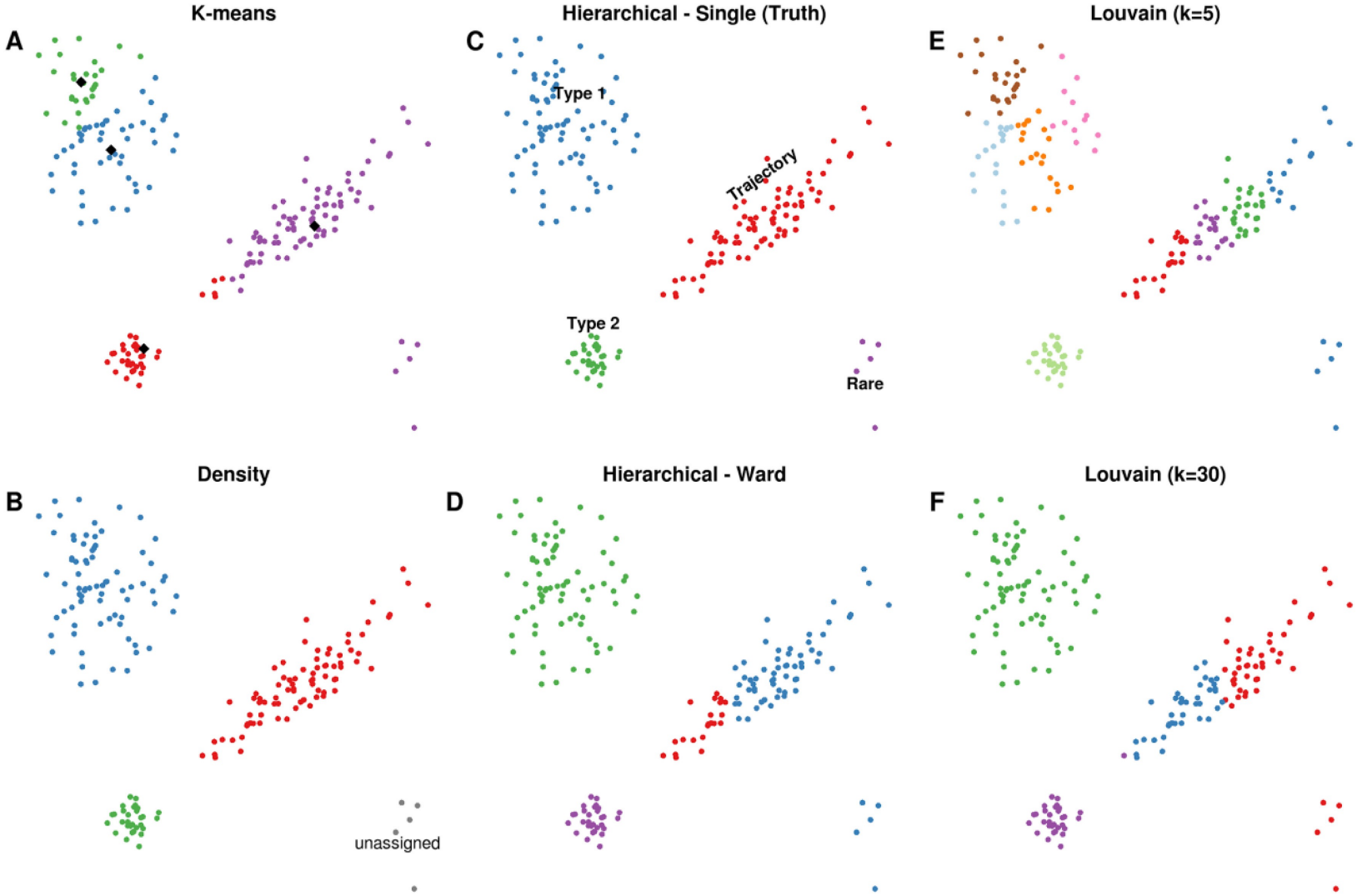


Clustering

- After dimensionality reduction, the next step (and the most important one) for the analysis is to group cells according to their type, or – better – find out the most relevant groups of cells according to similarities in their expression profile
- From a computational point of view, this step is formalized as **clustering**, that is, partition the cells into groups (clusters), hoping to capture the most relevant cell types
- The input for clustering is the initial dataset after dimensionality reduction (e.g. the first n principal components of the previous step)
- The most relevant issue is that usually **we don't know the number of clusters** beforehand
- To be "found", a cluster has to contain at least 10-20 "similar enough" cells (as stated before)

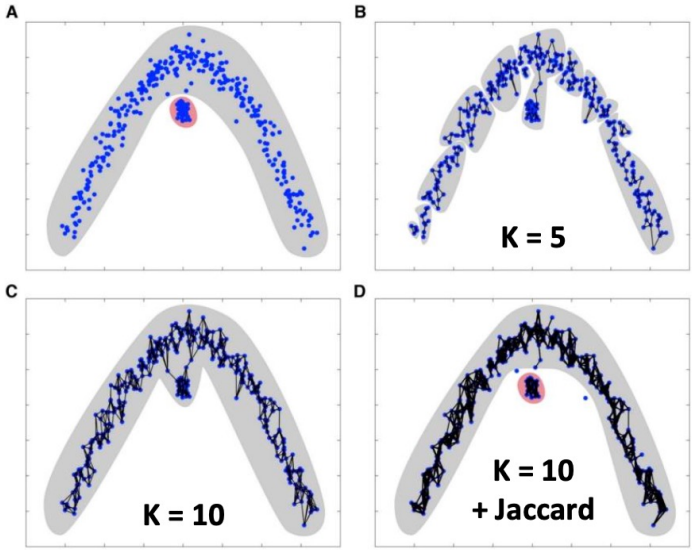
Clustering

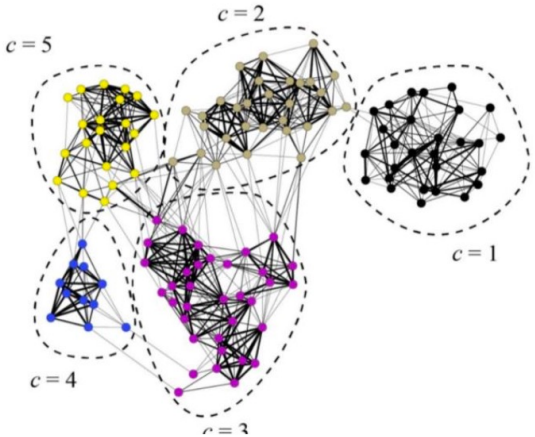
- Also for clustering, the literature in computer science/machine learning is huge
- Different strategies have been (and can be) applied to scRNA-Seq data:
  - **k means** clustering (by trying different values for “k” number of clusters)
  - **hierarchical** clustering
  - **graph-based** clustering
- Once again, the best choice is the one that gives the best results
- For very sparse data (e.g. 10x) **graph based methods** (like the Louvain algorithm) have become the de facto standard
- Changing the strategy can change significantly the results (see next slide)

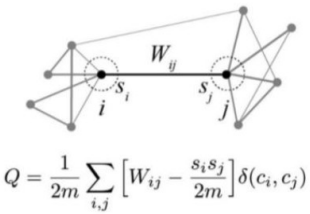

A
K-means
B
Density
unassigned
C
Hierarchical - Single (Truth)
Type 1
Trajectory
Type 2
Rare
D
Hierarchical - Ward
E
Louvain (k=5)
F
Louvain (k=30)

Clustering: kNN

- Construct kNN (k-nearest neighbor) graph based on the Euclidean distance in PCA space
- **Each node is connected to its "k" nearest neighbors**
- Refine the edge weights between any two nodes (at the beginning Euclidean distance) based on the **shared overlap in their local neighborhoods** (Jaccard distance) – that is – how many "friends" that have in common
- **Cluster cells by optimizing modularity (Louvain algorithm)** – a formalization of finding the best "normalized graph cut" in a graph
- You want to **maximize the number of edges connecting nodes in a cluster and minimize the number of edges connecting nodes in different clusters**
- So, at the end **the algorithm will find the optimal number of clusters according to the function used to evaluate the partitioning**

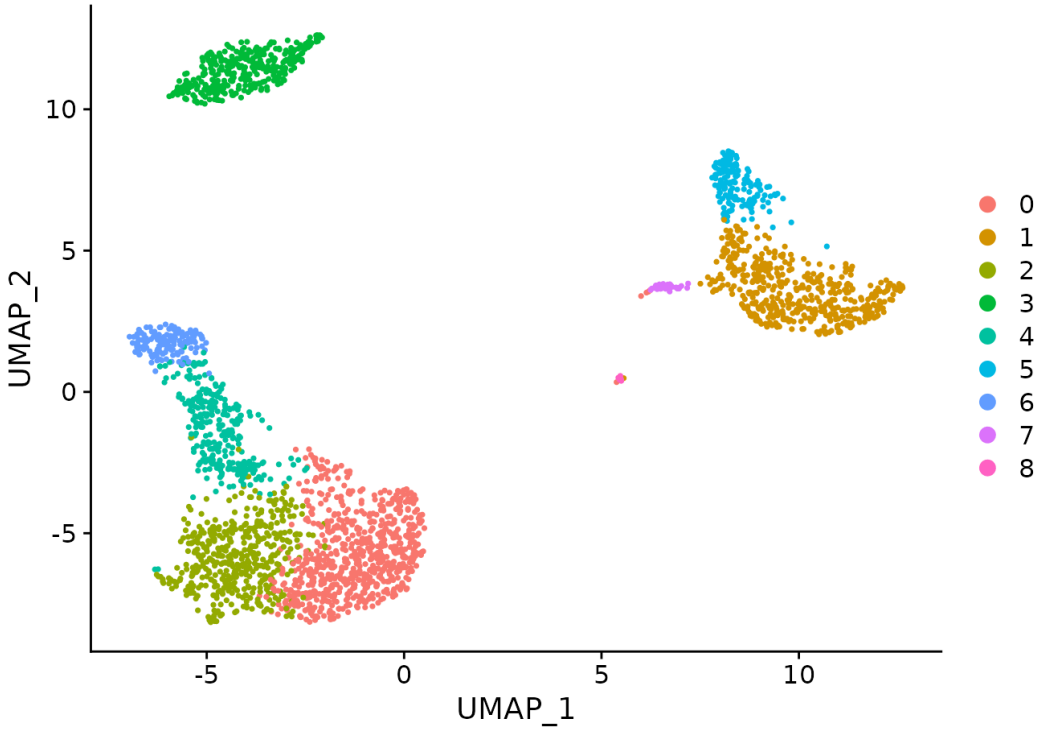






$$Q = \frac{1}{2m}\sum_{i,j}\left[W_{ij} - \frac{s_i s_j}{2m}\right]\delta(c_i, c_j)$$

Clustering example



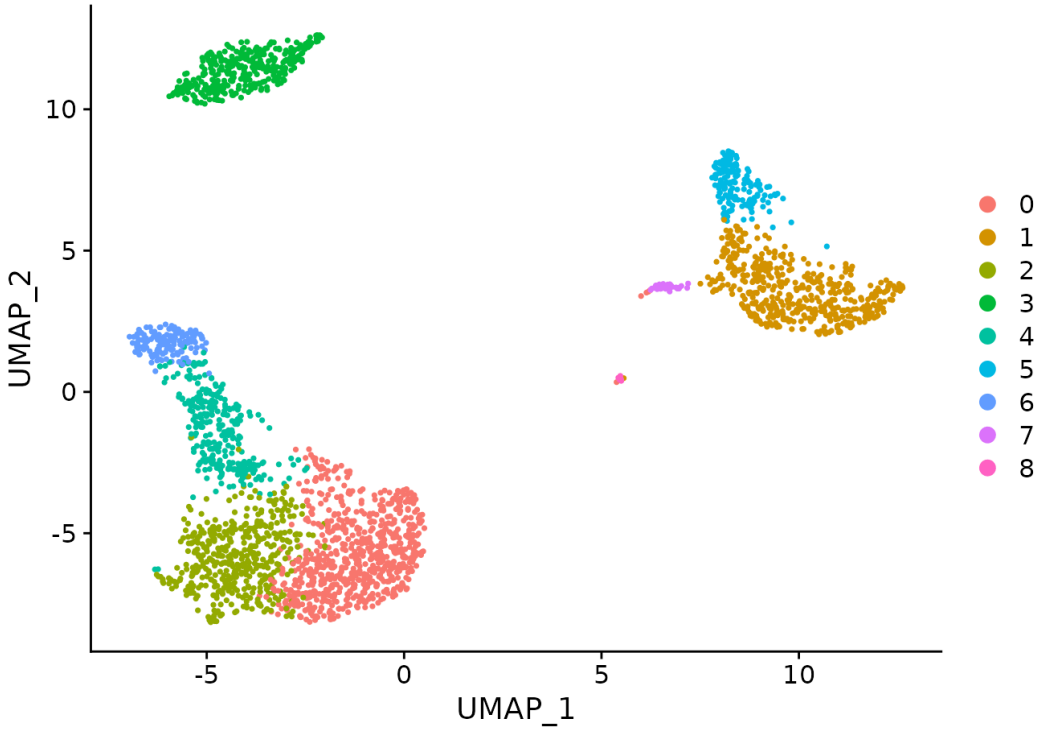
Clustering is performed after PCA (10 components) but the results shown using the UMAP projection
So - this is NOT the space in which cells are clustered

Can we identify which cell type corresponds to each cluster?

Finding differentially expressed genes

- Once we have clusters, the rationale is similar to the one of “bulk” RNA-Seq
- That is, cells of the same cluster are a “condition”: **find differentially expressed genes** in the clusters (e.g. in each cluster with respect to the others)
- **Use the DE genes to infer the cell identity of each cluster**
- In bulk RNA-seq DE genes are the result of a series of pairwise comparisons, with fine tuned modelling of the variability of the genes across the samples
- Here, the comparison we want to make is rather “one cluster against all the other ones”, rather “one cluster against another one”
- Hence, methods for finding DE genes in bulk RNA-Seq conditions and replicates are not so much suitable for this task

Finding DE genes



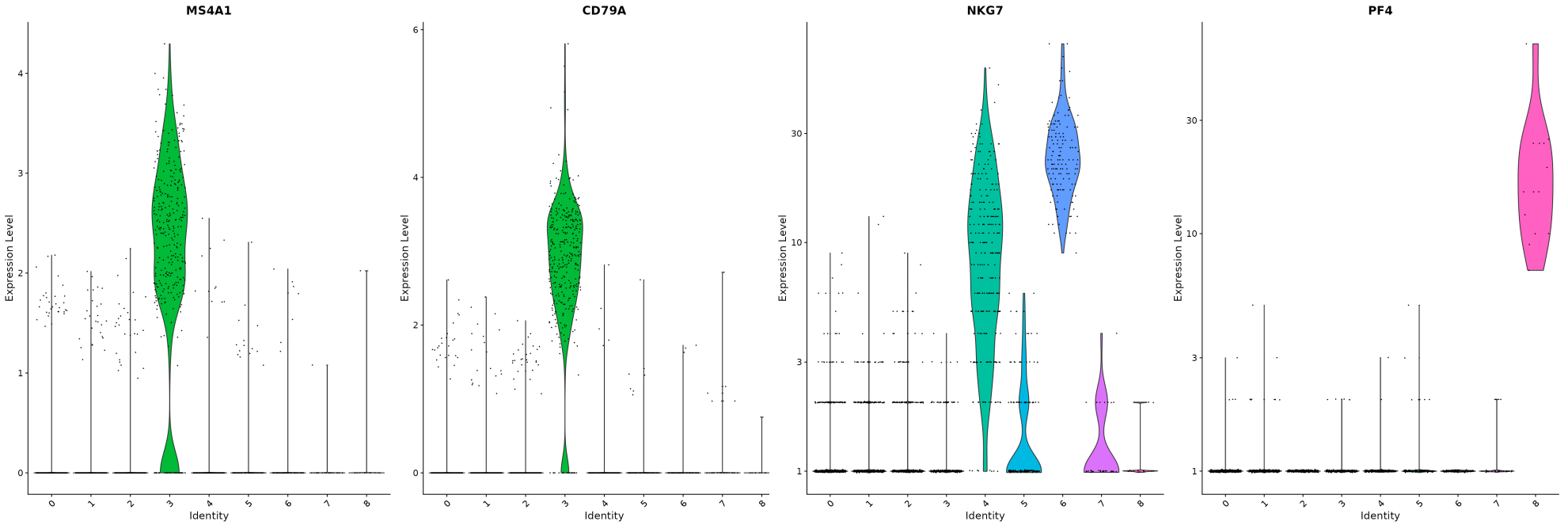
Each cluster is compared to the others taken together
Genes in each cluster are ranked according to significance of their «over-expression» with respect to the others

Finding DE genes

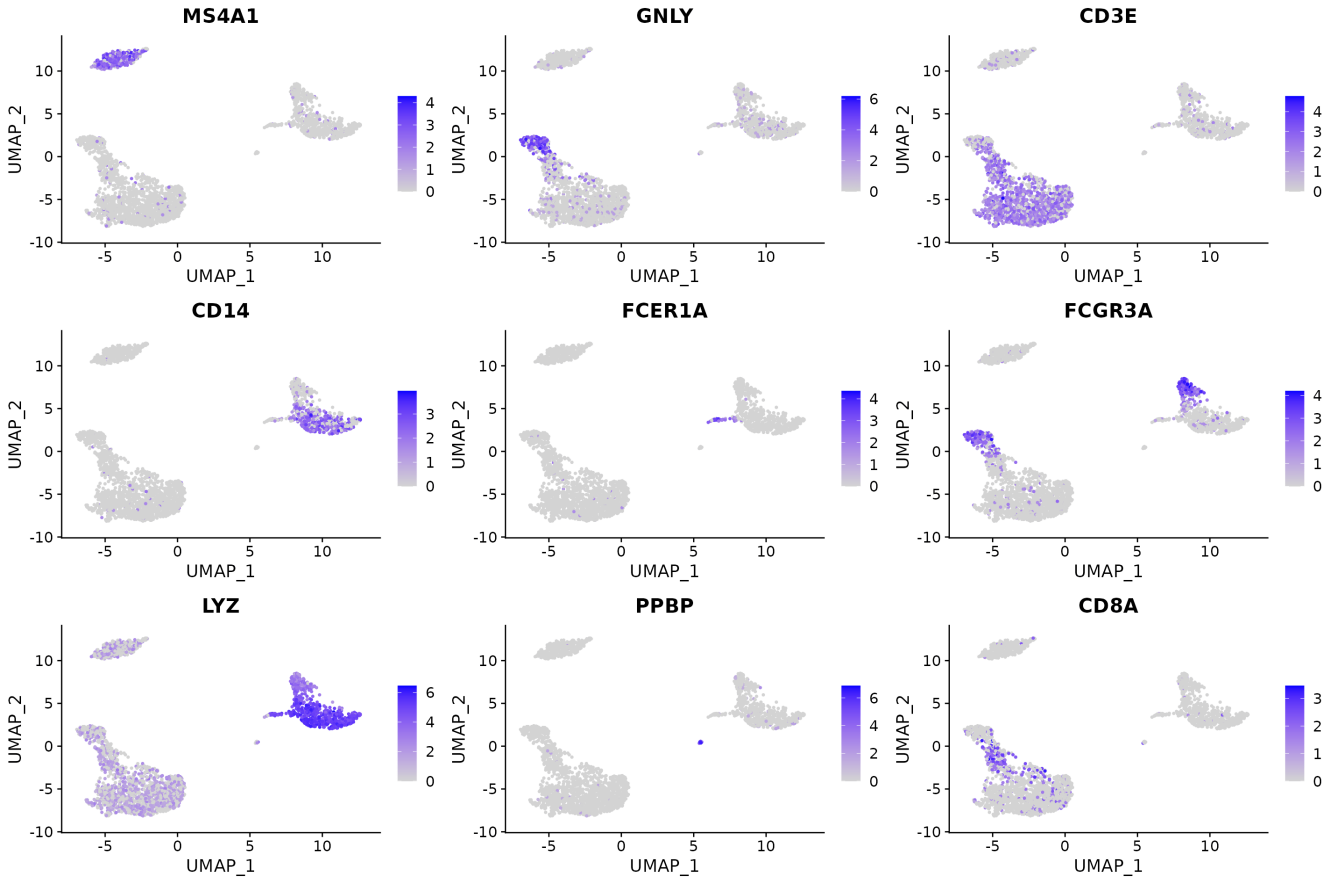
```
## # A tibble: 18 x 7
## # Groups:   cluster [9]
##        p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##        <dbl>      <dbl> <dbl> <dbl>     <dbl> <fct>   <chr>
##  1 1.74e-109       1.07 0.897 0.593 2.39e-105 0       LDHB
##  2 1.17e- 83       1.33 0.435 0.108 1.60e- 79 0       CCR7
##  3 0.              5.57 0.996 0.215 0.        1       S100A9
##  4 0.              5.48 0.975 0.121 0.        1       S100A8
##  5 7.99e- 87       1.28 0.981 0.644 1.10e- 82 2       LTB
##  6 2.61e- 59       1.24 0.424 0.111 3.58e- 55 2       AQP3
##  7 0.              4.31 0.936 0.041 0.        3       CD79A
##  8 9.48e-271       3.59 0.622 0.022 1.30e-266 3       TCL1A
##  9 1.17e-178       2.97 0.957 0.241 1.60e-174 4       CCL5
## 10 4.93e-169       3.01 0.595 0.056 6.76e-165 4       GZMK
## 11 3.51e-184       3.31 0.975 0.134 4.82e-180 5       FCGR3A
## 12 2.03e-125       3.09 1     0.315 2.78e-121 5       LST1
## 13 1.05e-265       4.89 0.986 0.071 1.44e-261 6       GZMB
## 14 6.82e-175       4.92 0.958 0.135 9.36e-171 6       GNLY
## 15 1.48e-220       3.87 0.812 0.011 2.03e-216 7       FCER1A
## 16 1.67e- 21       2.87 1     0.513 2.28e- 17 7       HLA-DPB1
## 17 7.73e-200       7.24 1     0.01  1.06e-195 8       PF4
## 18 3.68e-110       8.58 1     0.024 5.05e-106 8       PPBP
```

Each cluster is compared to the others taken together
Genes in each cluster are ranked according to significance of their «over-expression» with respect to the others
p-values and adjousted p-values are computed usually with non parametric statistical tests

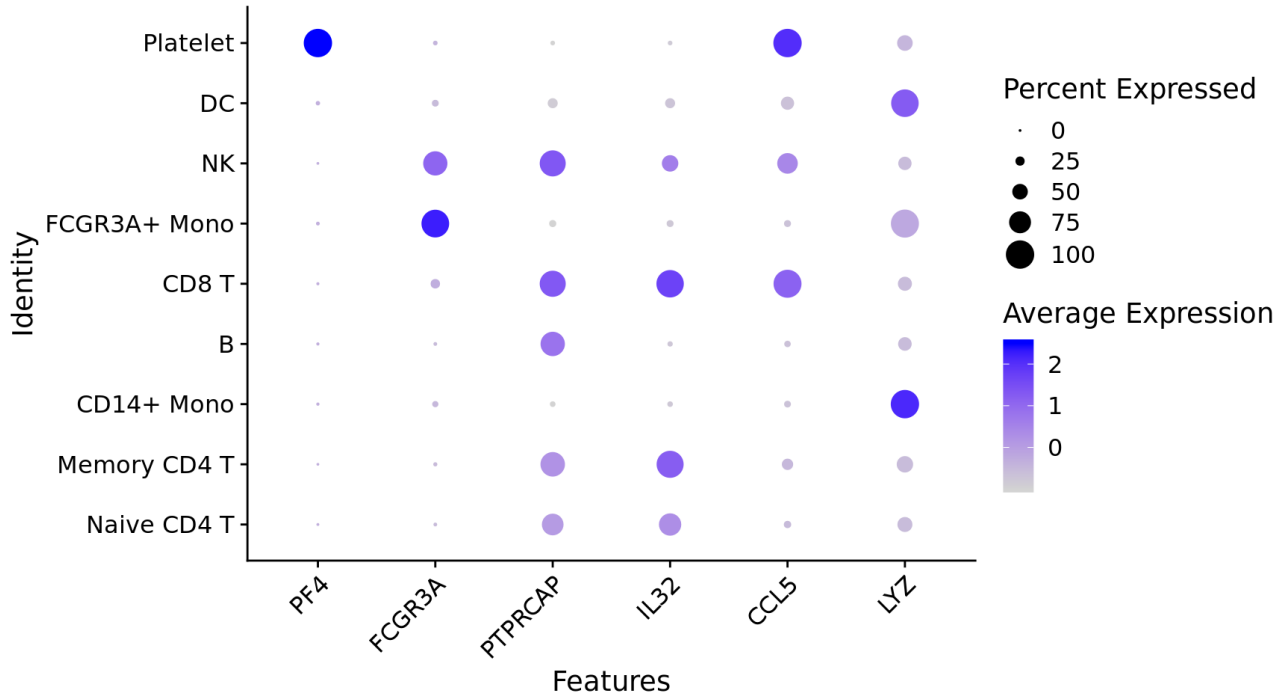
Differentially Expressed Genes in clusters



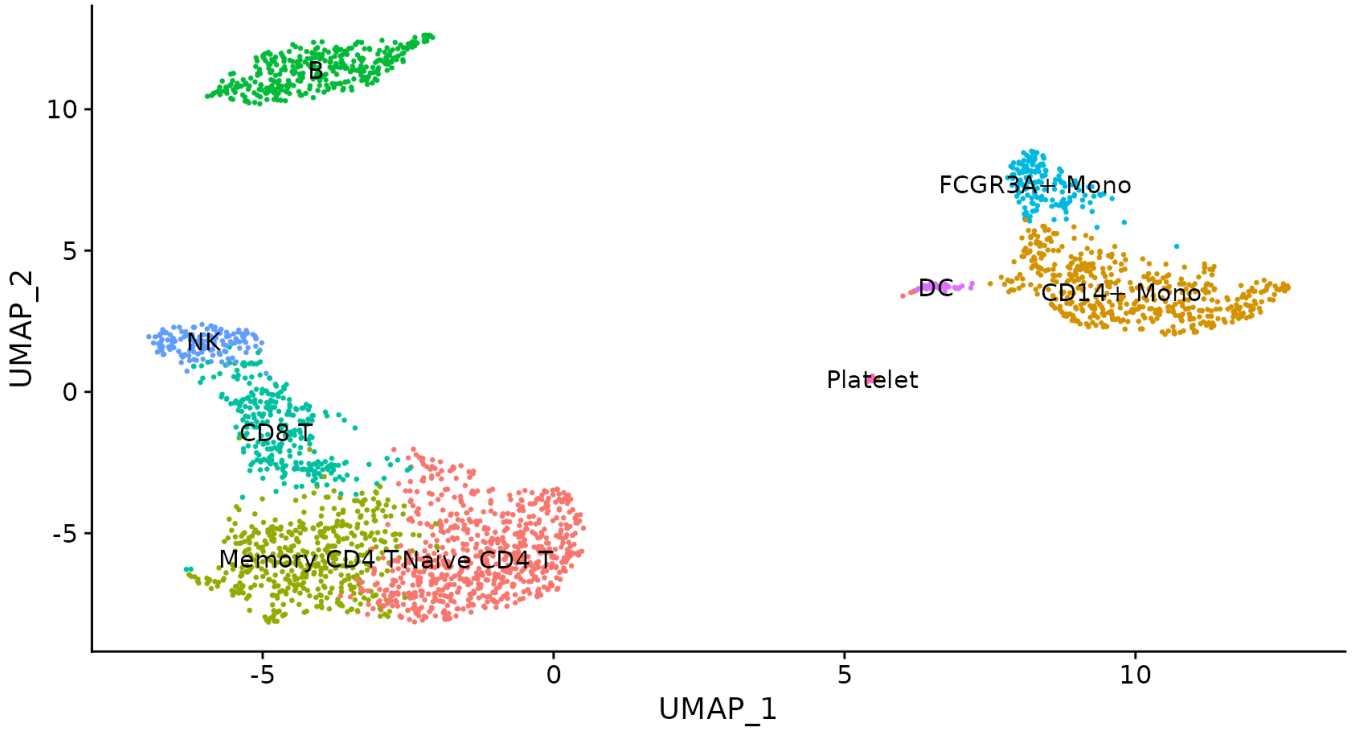
Differentially Expressed Genes



DE/Marker Genes and cell types



From DE gene to cluster annotation



By using the DE genes in each cluster we finally try to «guess» the cell type of each cluster

In this way we can also evaluate the quality of clustering: if «clear» cell types and sub-types do not emerge we can reconsider the clustering parameters and strategy

Summary

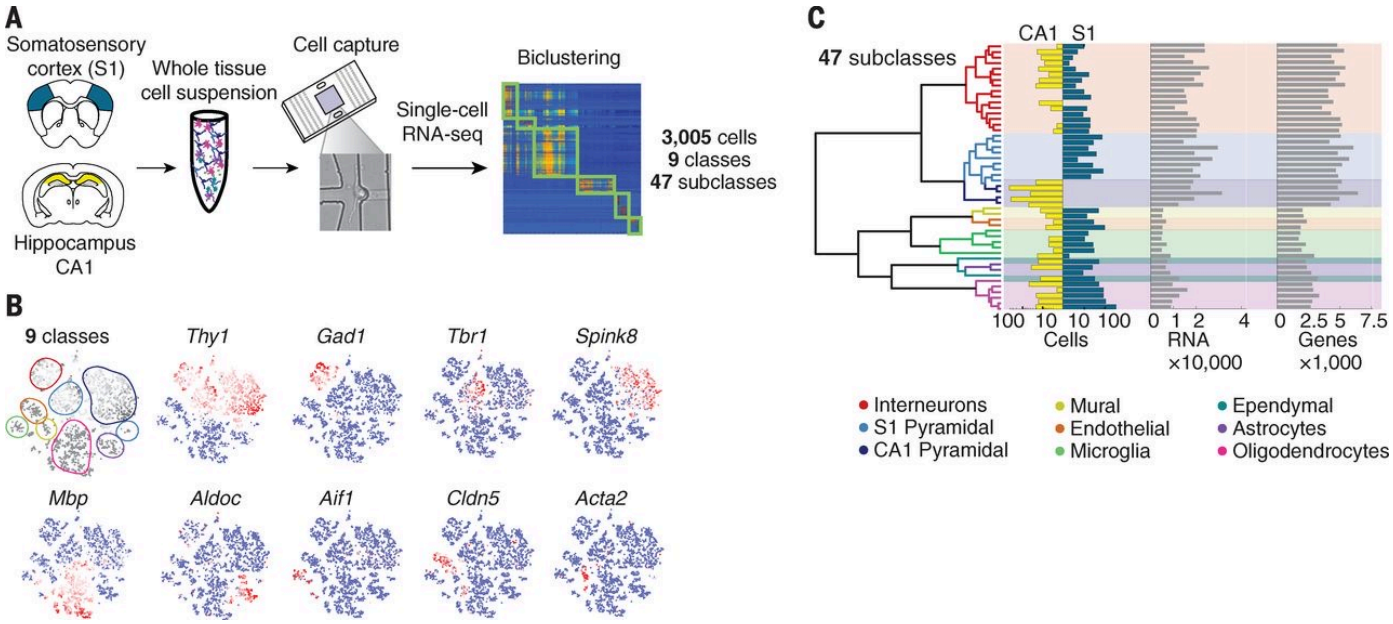


Fig. 1 Molecular census of somatosensory S1 cortex and hippocampus CA1 by unbiased sampling and single-cell RNA-seq.

Amit Zeisel et al. Science 2015;347:1138-1142

Published by AAAS

Summary – scRNA-Seq and 10x

- **Sequencing**:
  - Only poly-A - only one read pair per RNA molecule
  - 1 read "tagging" the 3' of the transcript + 1 read with cell barcode and UMI
  - Few thousands reads per cell
- **Mapping**: usually done with STAR on the genome
- **Counting**: identical reads with identical barcode and UMI count as 1
- **Quality control**: remove empty droplets, doublets, damaged cells
- **Normalization**: transform counts in "log-counts per 10,000"
- **Selection of genes:** keep only genes with the highest variability of expression across the cells (usually 2,000-3,000)
- **Scaling**: transform counts so that each gene as mean 0 and variance 1 across all cells
- **Dimensionality reduction for analysis**: Principal Component Analysis – choose the number of dimensions (usually from 5 to 20)
- **Dimensionality reduction for visualization**: tSNE or UMAP projection in 2D
- **Clustering**: graph based, kNN, Louvain algorithm
- **Finding DE/marker genes**: non parametric Wilcoxon test
- **Assign a cell type to each cluster**: better done by eye/manual inspection but dozens of automated methods are available

Further analysis

- The workflow we have just seen can be the starting point for further analysis/post processing:
  - Deconvolution analysis of bulk RNA-Seq samples
  - Integration (harmonization) of different scRNA-Seq samples
  - Integration with other types of single cell omic data (e.g. scATAC-Seq)
  - Integration with spatial information
  - Trajectory analysis
- The “core” of the data analysis remains anyway the workflow we have just seen, understanding it will make understanding all of the above much easier, and also understanding while in some cases some variations of the methods we have seen are reported to be more advisable (e.g. different approaches to normalization and the statistical test for DE genes)