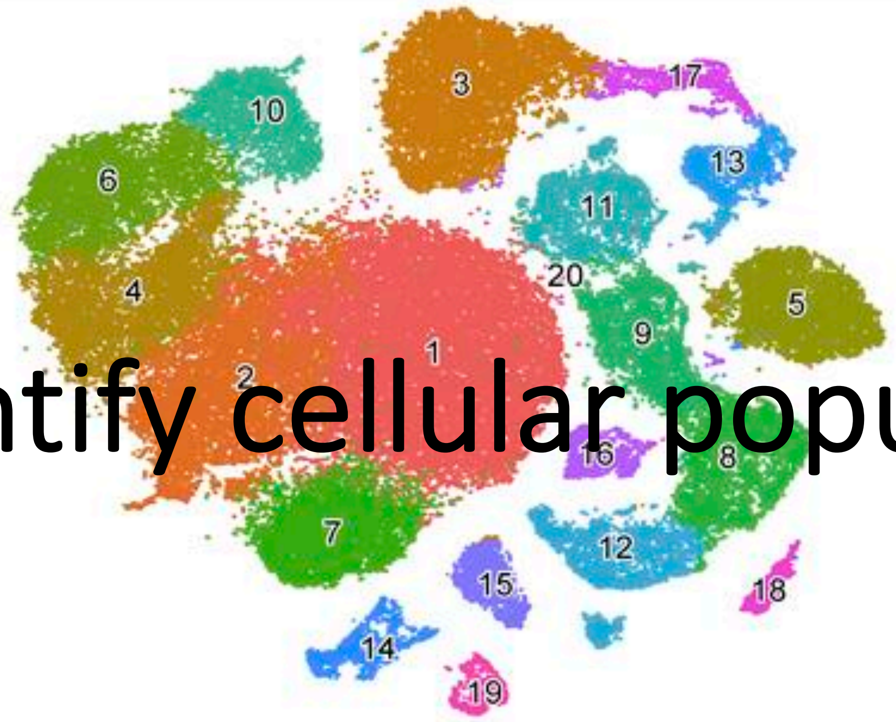
SincellTE 2024 : 5th Workshop in single cell data analyses

Transcriptomics, Spatial and Long-reads



Identify cellular populations

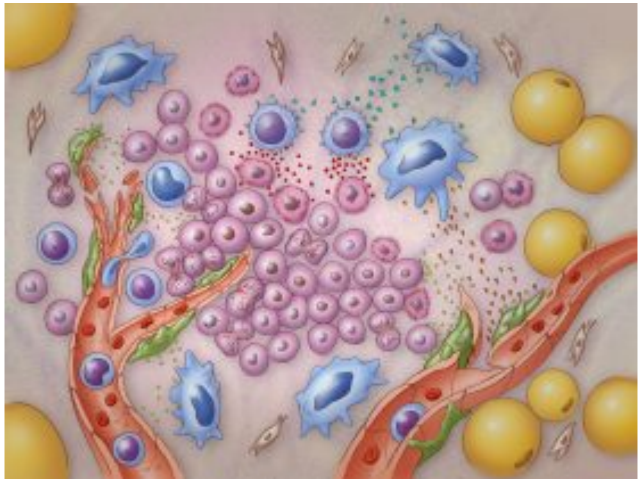
Lorette Noiret

Sorbonne Université, Institut Curie

Morning program

- Identify cellular population (~2h)
  - understand the pipeline
    - statistical highlights : PCA and Graph clustering
  - practical session
- coffee break
- Recap Pratical session
- Integration
- Summary

Identify cellular populations



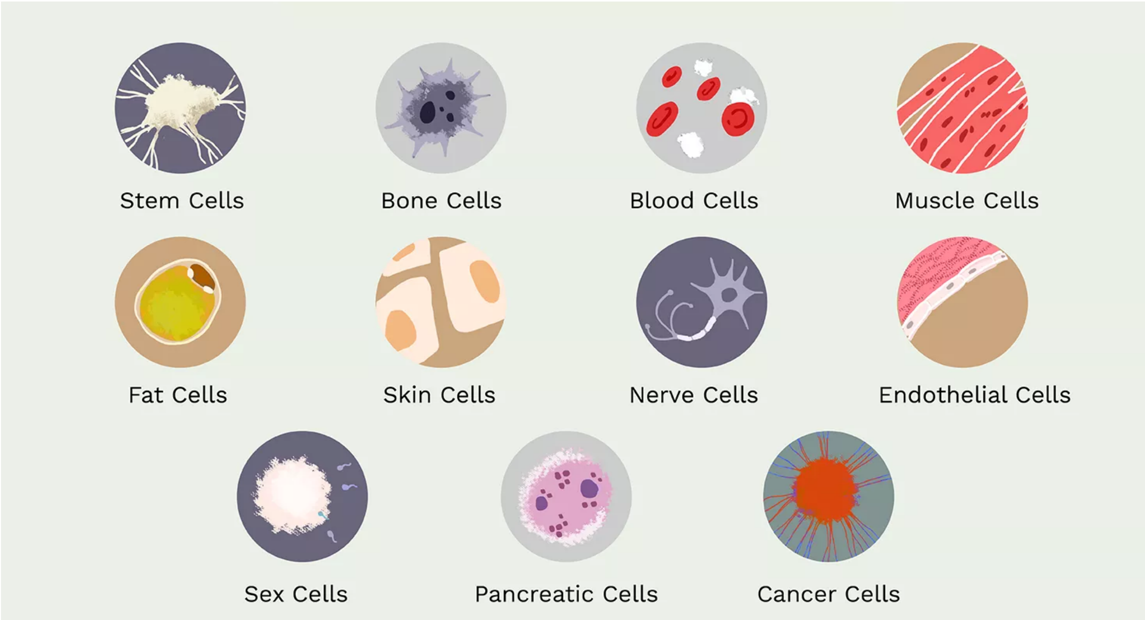
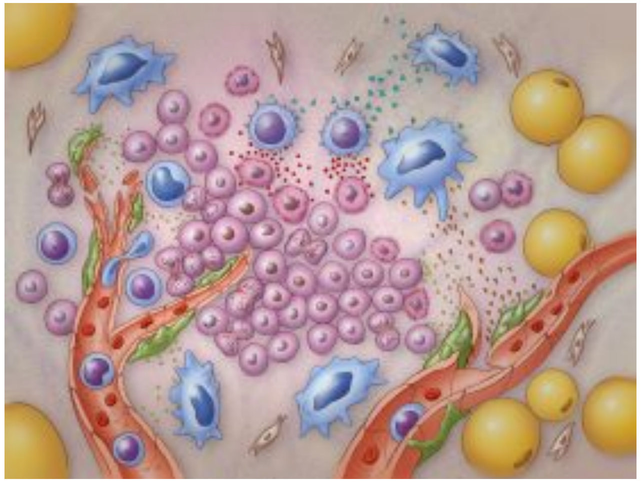


Image :
https://www.mdanderson.org/cancerwise/what-is-the-tumor-microenvironment-3-things-to-know.h00-159460056.html
https://www.thoughtco.com/types-of-cells-in-the-body-373388

Identify cellular populations

Why ?



Differential Expression
Enrichment Analysis (GO, GSEA)
Gene regulatory network (Scenic+)
Ligand-Receptor interactio n (cell2cell)
Pseudo Time analyse …

Answering biological questions

- What define cell identity ?
- Identify new cell types, rare populations
- What are the transcription factors that control cell identity ?
- How cells differentiate to a new cell type ?
- How cells communicate together (ligand/receptor) ?
- What is a cancer cells ?
- How cells are affected by a disease ?
- Biomarker discovery
- …

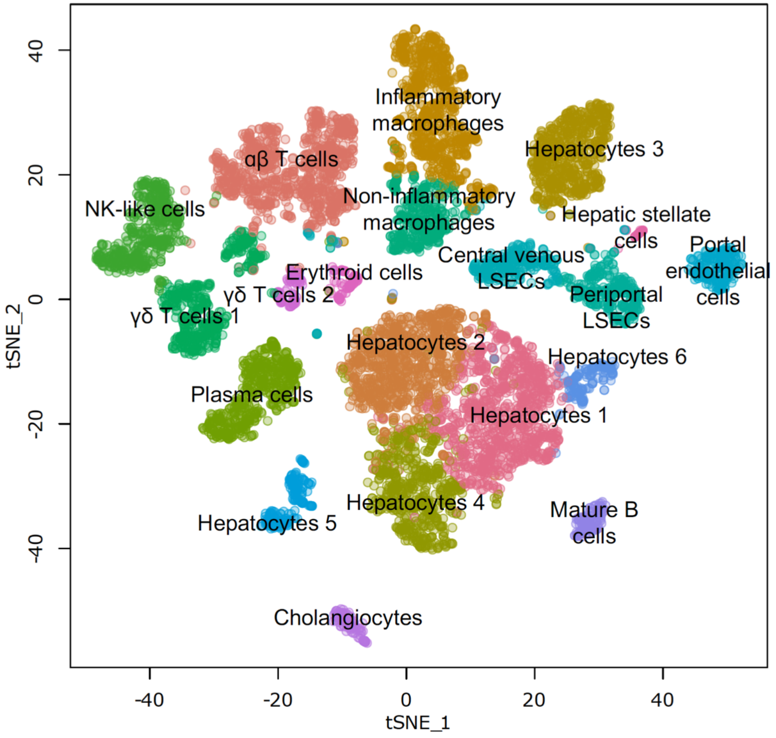
Identify cellular populations

How ?

1. **Regroup similar cells together** (in terms of gene expression profils) : **clustering**

2. **Annote the clusters of cells: give an identity (cell type) to each cell**
   - Manual : list of marker genes
   - Automated : use an existing annotation using a supervised model (annotation transfer)

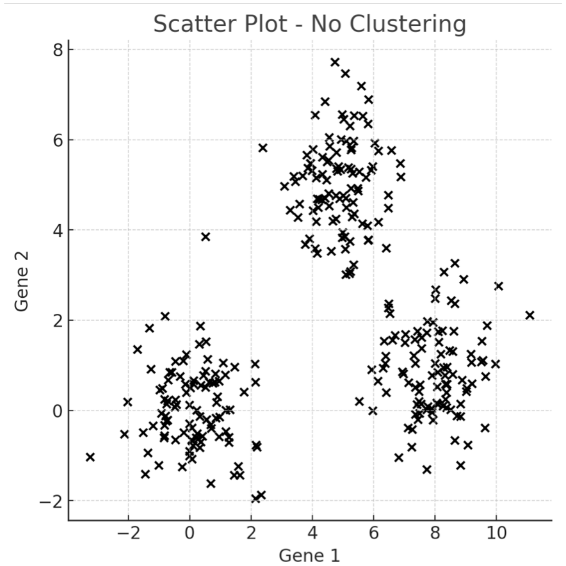
2-D Vizualization of the groups

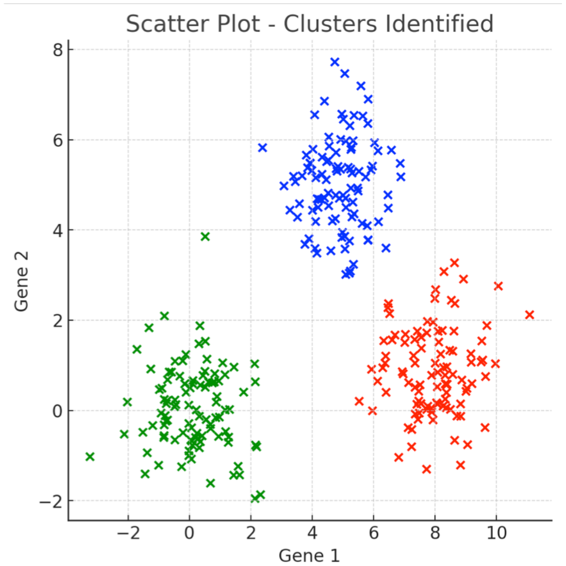


Z. Clark et al. Nature Protocole (2021) Tutorial: guidelines for annotating single-cell transcriptomic maps

Clustering

Clustering : a statistical learning method that groups observations into homogeneous "clusters," which share common characteristics.





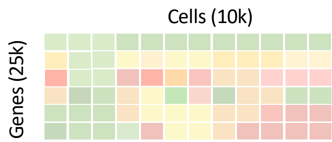
Clustering and scRNAseq : challenges

- **High dimensionality** (~10k-30k genes and ~1k to 100k of cells)
  - **Curse of dimensionality**
  - As the number of dimensions (features or genes) increases, the volume of the space grows exponentially. This means data points become increasingly sparse, making it difficult to identify meaningful clusters or patterns.
  - **Scalability**
- **Sparse**, noisy signal :
  - zeros dominating the data matrix
    - genes not expressed in every cells
    - 10X : ~30% of mRNA transcripts are captured per cell
  - Sparse data can make it harder to discern meaningful signals from the data
- **Interpretability**

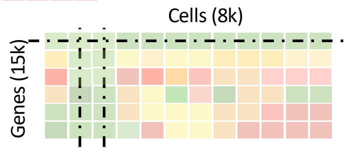
Bioninformatic pipeline

yesterday

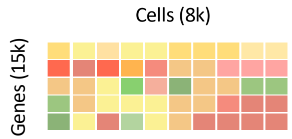
raw data



Quality control and filtering

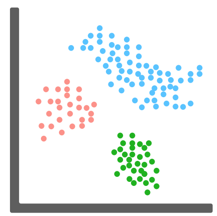


Data normalization



We need to concentrate the genetic signal
to efficiently separate the groups

Clustering



courtesy Nathalie Lehmann

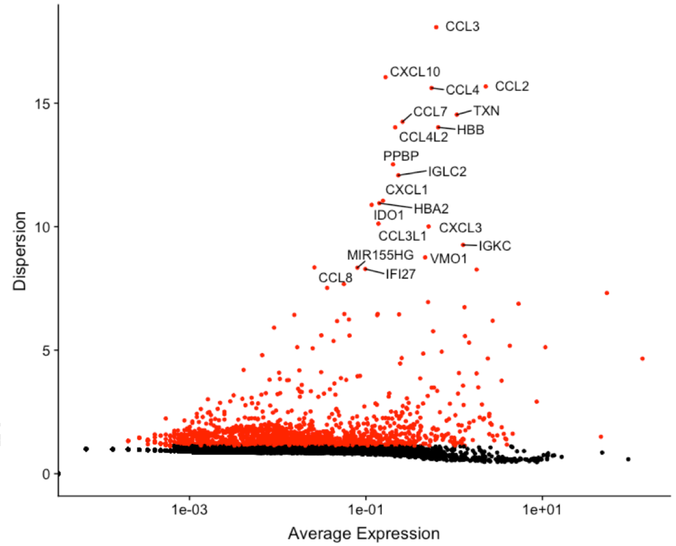
Reduce the dimensionality before clustering

- **Challenges** : high dimensionality, sparse, noisy signal
- **Solutions :** Reduce the dimensionality, the sparsity and the noise in the signal !

1) Work on a subset of genes : **Highly Variables Genes** (HVGs)

- Keep the genes that varies the most
- Remove lowly variables genes: house keeping genes…
- more likely to capture biologically meaningful differences between cells types
- From ~10,000-30,000 genes to **500-3,000 HVGs**

Method « vst »



FindVariableFeatures(object, ...)

Specify the number of genes to keep

Reduce the dimensionality before clustering

- **Challenges : h**igh dimensionality, sparse, noisy signal
- **Solutions :** Reduce the dimensionality, the spasity and the noise in the signal

1) Work on a subset of genes : **Highly Variables Genes** (HVGs)

- From ~10,000-30,000 genes to **500-3,000 HVGs**

2) Perform a **dimension reduction** (e.g. PCA) of this subset

- From 500-3,000 HVGs to **10-50 principal components**

Stastistical highlights
Dimension reduction with PCA

Principal component Analysis

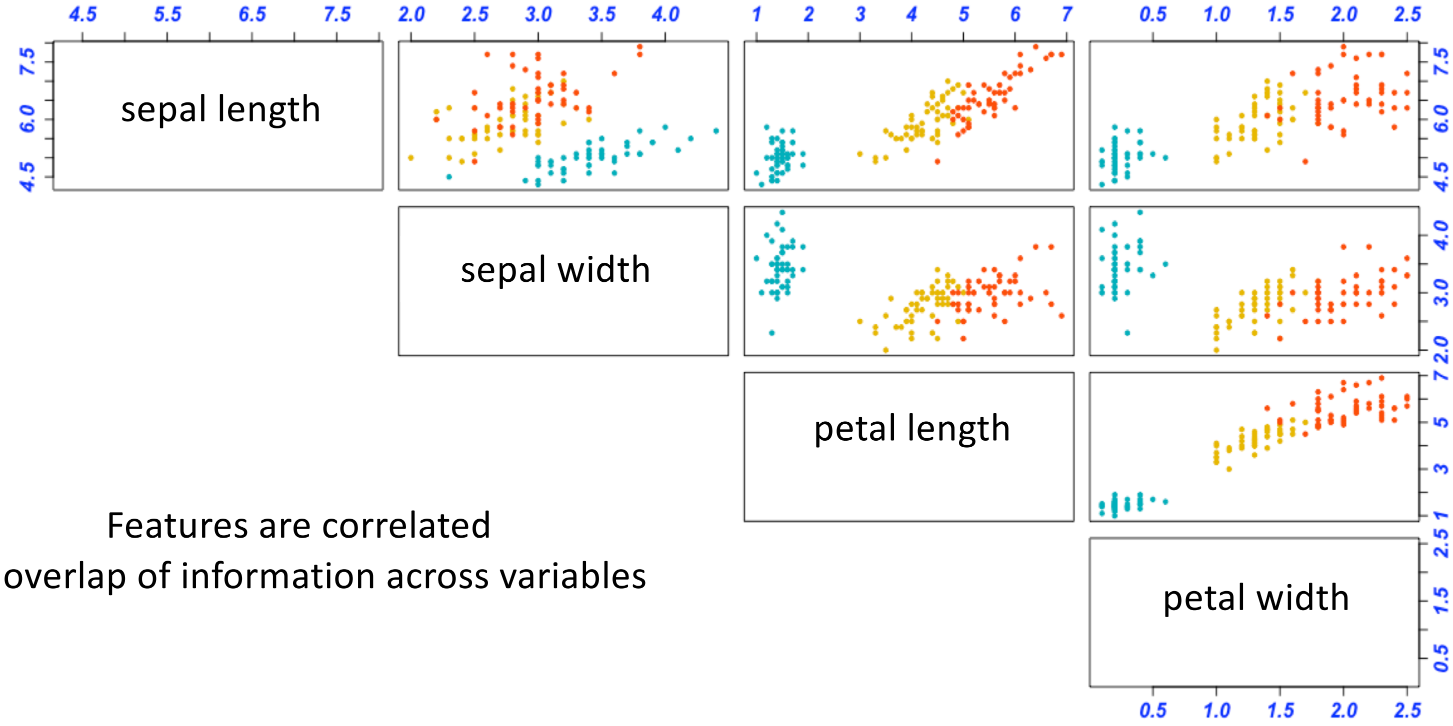
Why ?

- Enables **quick visualization of the main trends** in your data
- **Reduces the number of variables** needed for their representation

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.8	1.9	0.4	setosa
4.8	3.0	1.4	0.3	setosa
5.1	3.8	1.6	0.2	setosa
4.6	3.2	1.4	0.2	setosa
5.3	3.7	1.5	0.2	setosa
5.0	3.3	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor

Données iris : Anderson, 1936; Fisher, 1936)

Principal component Analysis - Principle

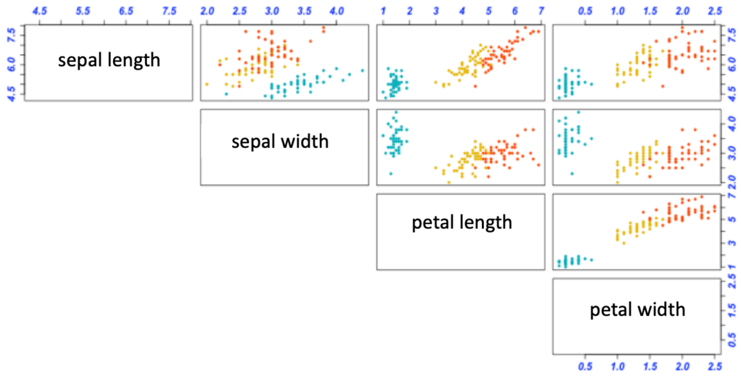


Features are correlated
-> overlap of information across variables

R: pairs()

Photos : Eric Hunt CC BY-SA 4.0 Iris_virginica_2.jpg, CC BY-SA 3.0 Iris_versicolor_4.jpg; : Irissetosa1.jpg (https://upload.wikimedia.org/wikipedia/commons/)

Principal component Analysis - Principle



Correlation matrix

	sepal length	sepal width	petal length	petal width
sepal length	1	-0,11	**0,87**	**0,82**
sepal width	-0,11	1	-0,42	-0,36
petal length	0,87	-0,42	1	**0,96**
petal width	0,82	-0,36	0,96	1

- We don't really need all 4 variables to describe each iris.
- Idea : Create a new variables that summarize the overlaping information
- Make a coordinate change so that the maximum amount of information is summarized in the first few axes."

Principal component Analysis - Principle

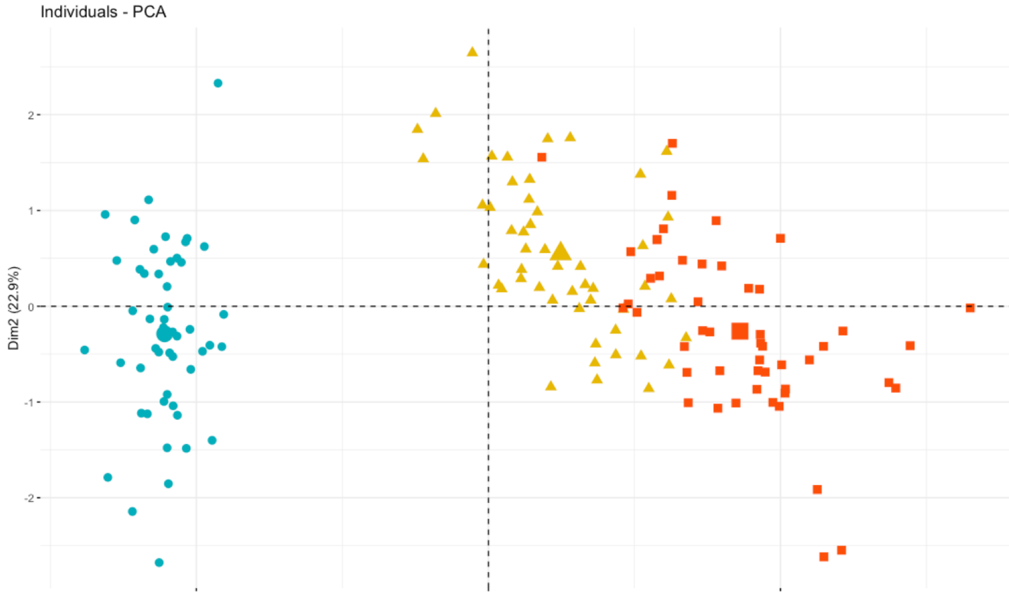
- Inertia = the total variability in the data = sum variances of each feature

PCA create new variables (PC1, PC2…) such that the maximum amount of inertia is summarized in the first few axes.

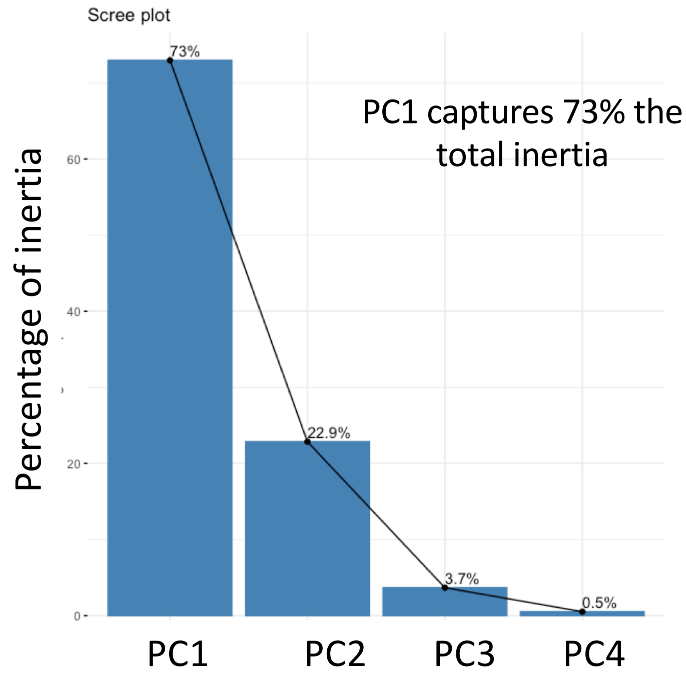
					Inertie
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	
Variance	0.6856935	0.1899794	3.1162779	0.5810063	4.572957
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	
Variance scaled data	1	1	1	1	4
	PC1	PC2	PC3	PC4	
PC variance	2.91	0.92	0.15	0.02	4

PCA – key concepts

Iris represented in thefirst two PC



Scree plot
(Elbow plot)

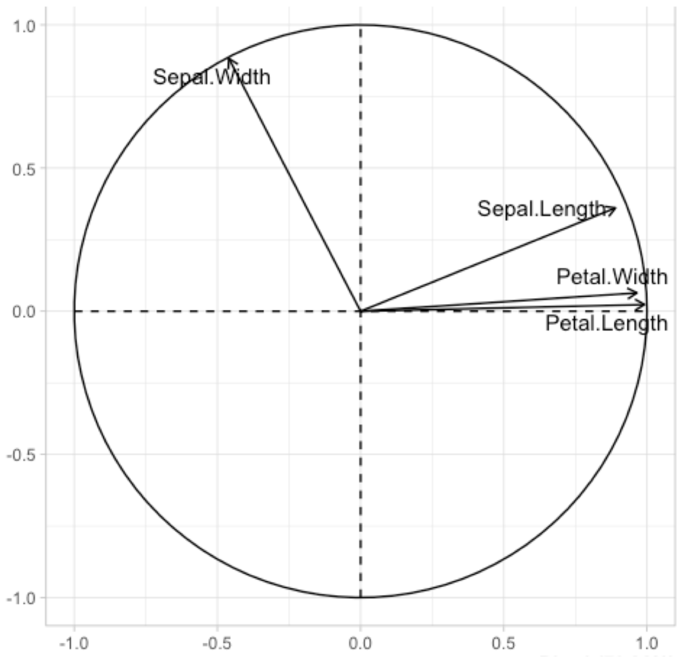


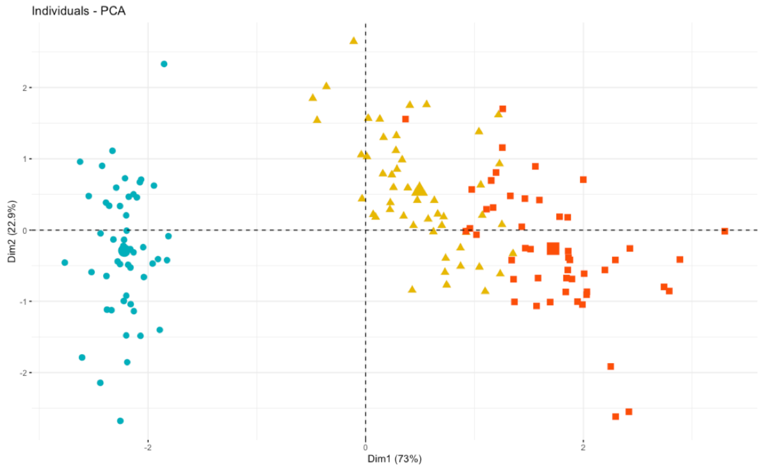
PC1 = -0,901*sepal_length + 1,032*sepal_width -1,341*petal_length -1,313* petal_width

library("FactoMineR")
library("factoextra")

PCA – key concepts

PCA loadings

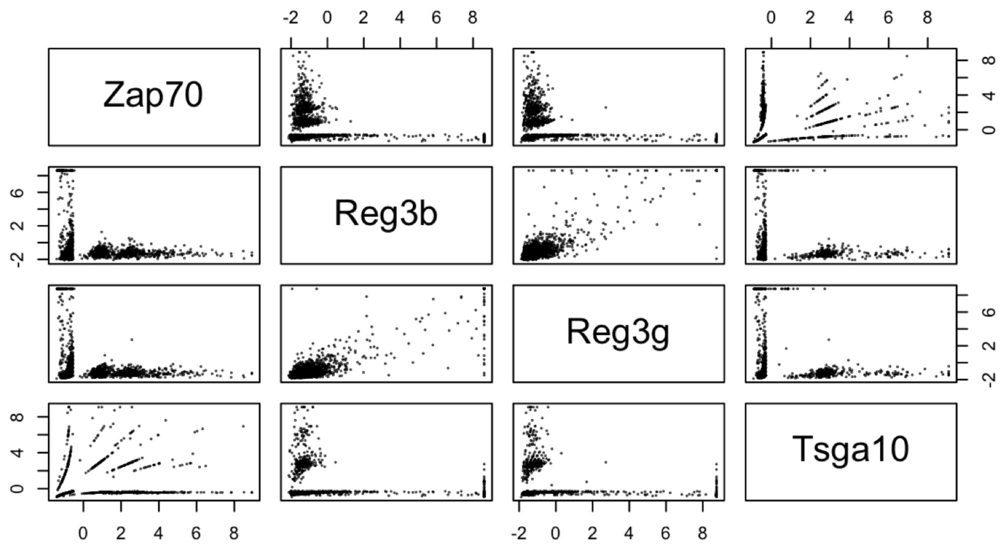




- Axe 1 : Petal.length, Petal.width et Sepal.length
- Axe 2 : Sepal.Width

Loadings: correlation between features and new PC axes

Principal component analysis in scRNAseq



Information overlap

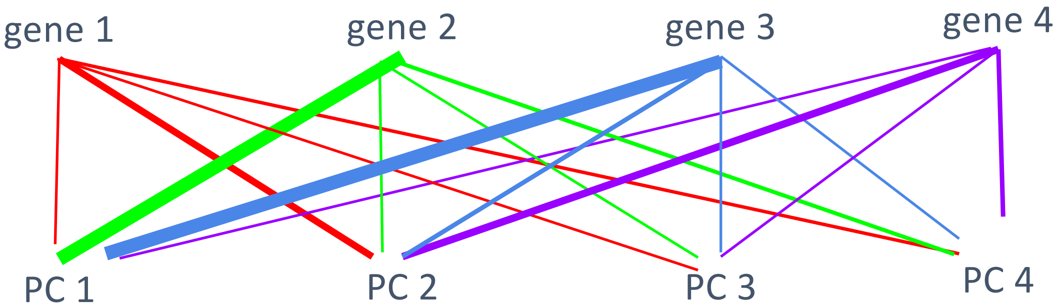
Some genes are co-expressed in some cells

- PCA construct axes that summarize the shared genetic information

Principal component analysis in scRNAseq

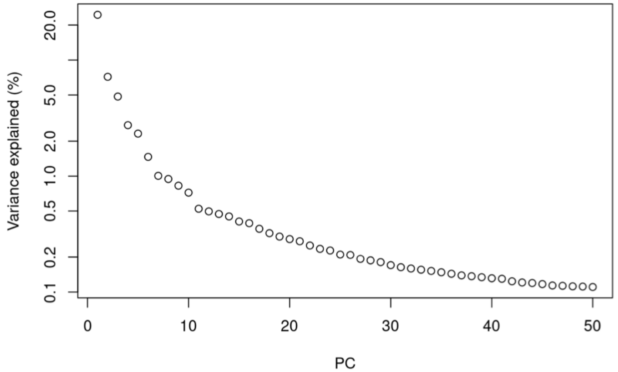
- New coordinate system

RunPCA()



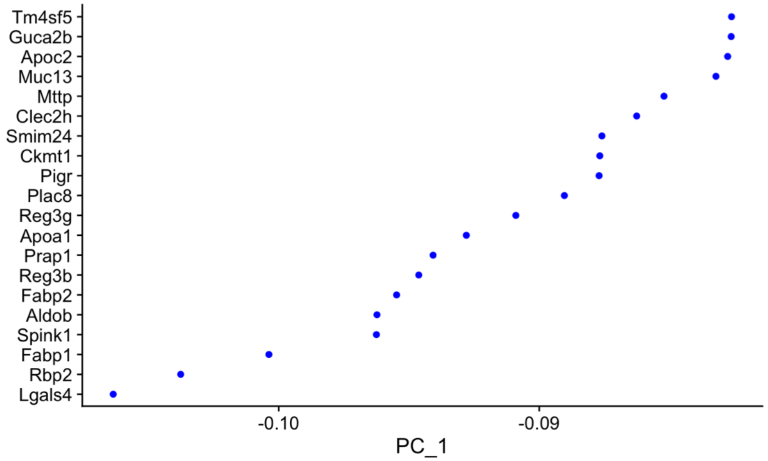
- The first axes of the reference system capture the main trends in the data.
- keep between typically **10-50 (max 100) PC axes**
- Elbow plot : can help you to choose the number of axes to keep for downstream analysis (clustering)

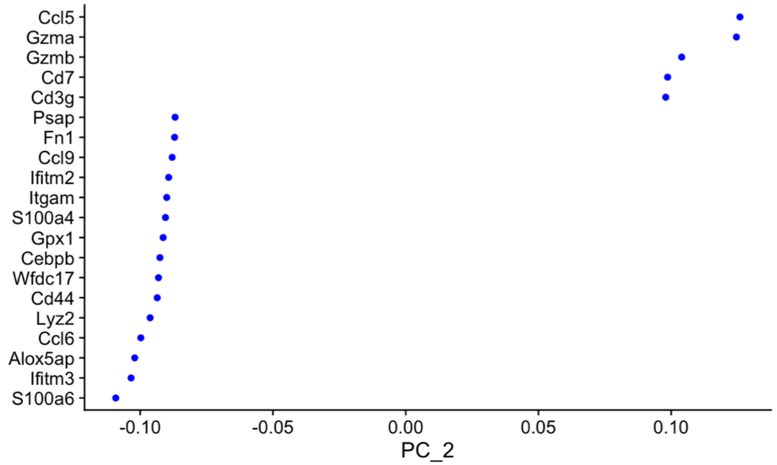
ElbowPlot()



Interpretation PCA axes with loadings (features)

VizDimLoadings()

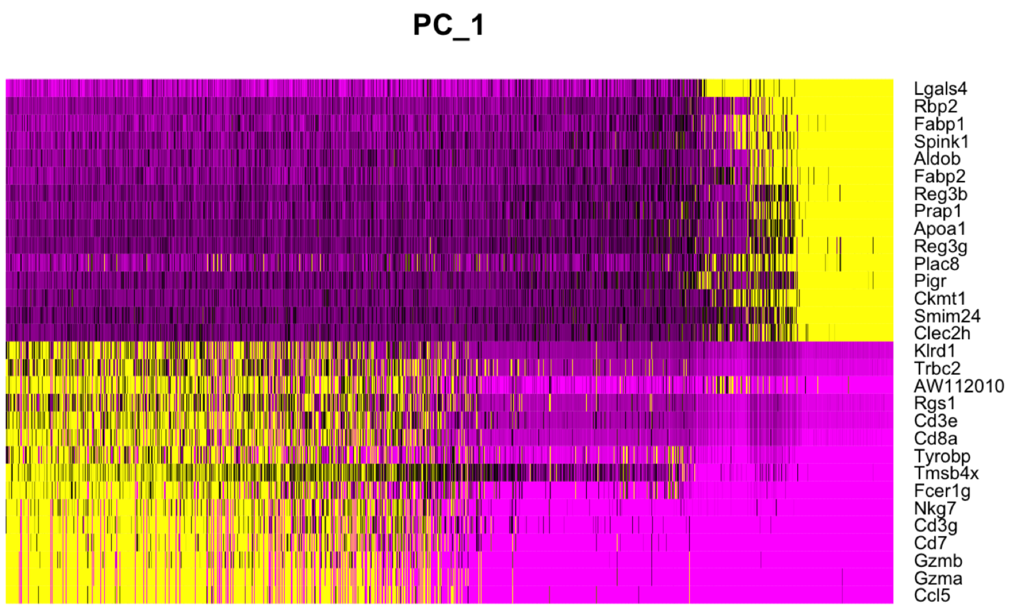




Interpretation of PCA axes (features and cells)

DimHeatmap()

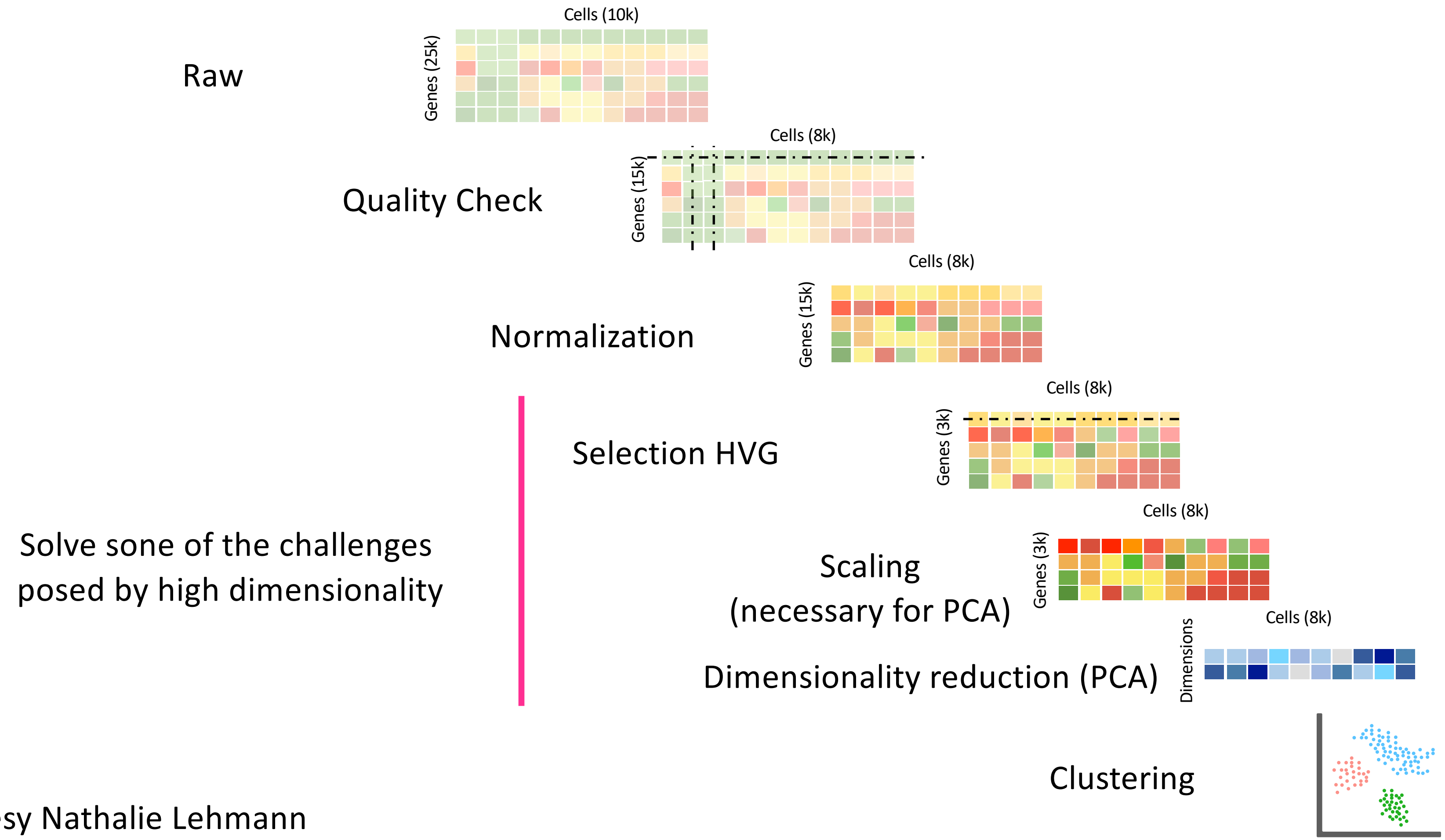
Visualize the genes that are driving the components and allows to get some insight about the heterogeneity of the data..



Genes are ordered according their loadings

Cells re ordered according to their coordinates on PC axe

Bioninformatic pipeline



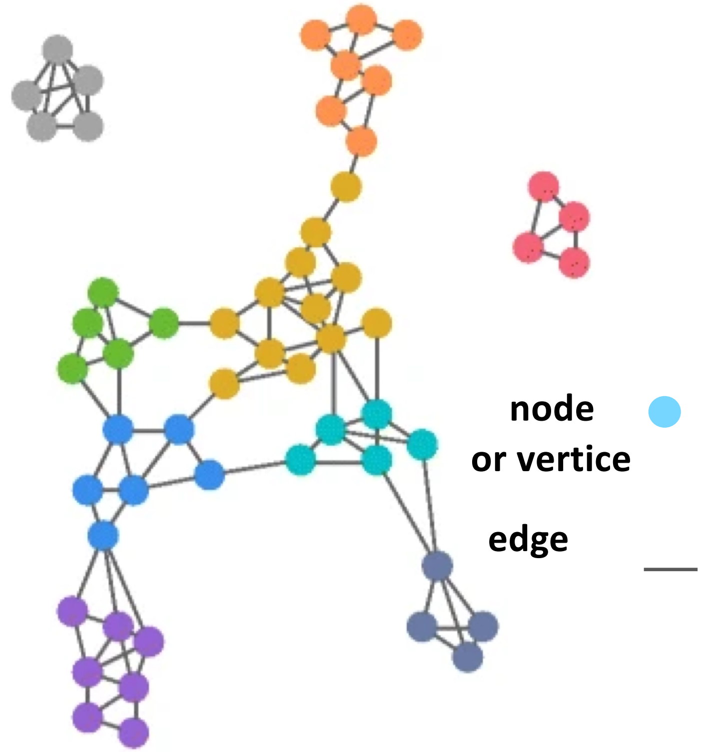
courtesy Nathalie Lehmann

Clustering scRNAseq

Strategy : represent the proximity between cells in the form of a graph, which will then be partitioned

a. Construct a graph from the principal components (PC)
   Nearest Neighbor graph: kNN or **sNN**

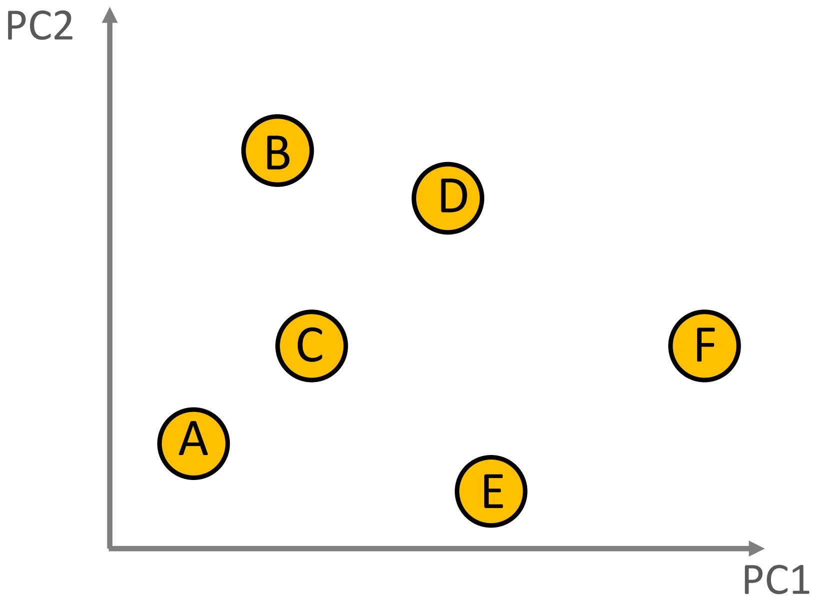
b. Partition the graph using Louvain or **Leiden** algorithm



Levine, J. H. et al.. *Cell* **162**, 184–197 (2015).

k Nearest Neighbors (kNN) graph

- **Calculate the distance between all the pairs of cells**



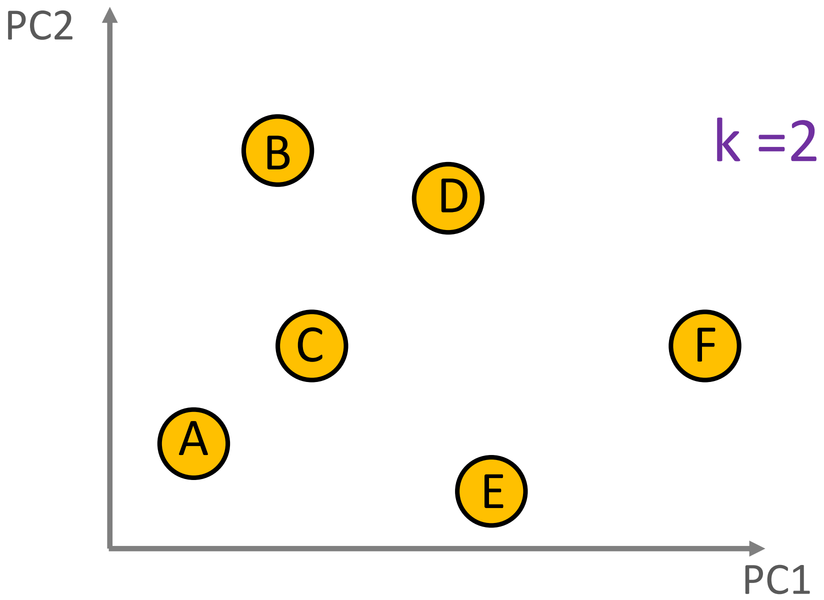
Example : distance between cells calculated using the first PC components

cells	PC1	PC2
A	8	7
B	9	10
C	9,4	8
D	11	9,5
E	11,5	6,5
F	14	8

$$\text{dist(A,B)} = \sqrt{(9-8)^2 + (10-7)^2} \approx 3{,}2$$

k Nearest Neighbors (kNN) graph

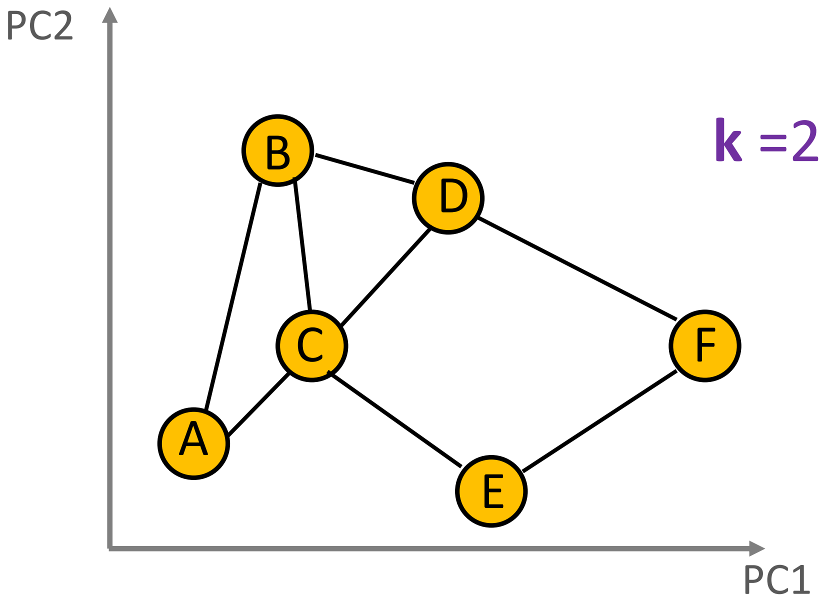
- Calculate the distance between all the pairs of cells
- **For each cell, we define its k nearest neighbors.**
- Each entity (cell) is connected to its k nearest neighbors.



$$
\begin{array}{c|c}
 & \begin{matrix} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & \mathrm{E} & \mathrm{F} \end{matrix} \\
\begin{matrix} \mathrm{A} \\ \mathrm{B} \\ \mathrm{C} \\ \mathrm{D} \\ \mathrm{E} \\ \mathrm{F} \end{matrix} &
\begin{bmatrix}
0 & 3.2 & 1.7 & 3.9 & 3.5 & 6.1 \\
3.2 & 0 & 2 & 2.1 & 4.3 & 5.4 \\
1.7 & 2 & 0 & 2.2 & 2.6 & 4.6 \\
3.9 & 2.1 & 2.2 & 0 & 3 & 3.4 \\
3.5 & 4.3 & 2.6 & 3 & 0 & 2.9 \\
6.1 & 5.4 & 4.6 & 3.4 & 2.9 & 0
\end{bmatrix}
\end{array}
$$

k Nearest Neighbors (kNN) graph

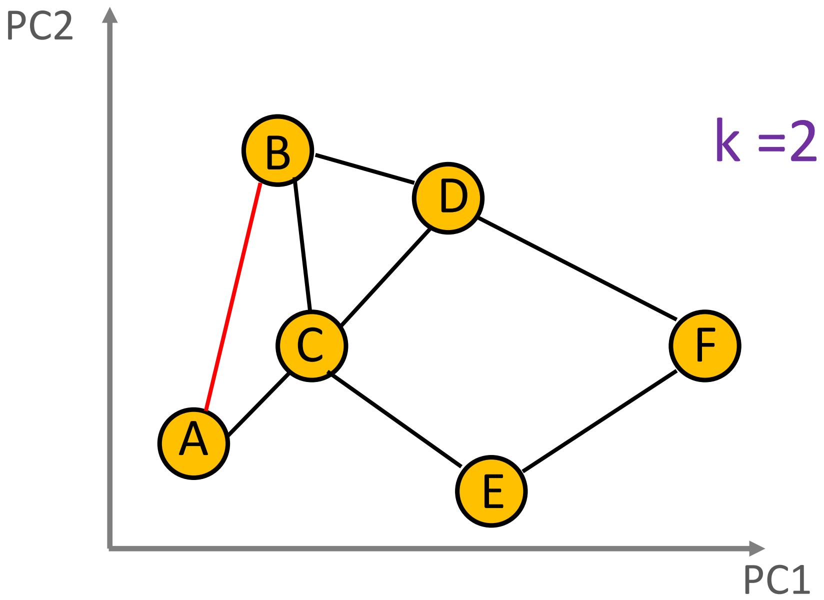
- Calculate the distance between all the pairs of cells
- For each cell, we define its k nearest neighbors.
- **Each entity (cell) is connected to its k nearest neighbors.**



$$
\begin{array}{c c}
 & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} &
\begin{bmatrix}
0 & 3.2 & 1.7 & 0 & 0 & 0 \\
0 & 0 & 2 & 2.1 & 0 & 0 \\
1.7 & 2 & 0 & 0 & 0 & 0 \\
0 & 2.1 & 2.2 & 0 & 0 & 0 \\
0 & 0 & 2.6 & 0 & 0 & 2.9 \\
0 & 0 & 0 & 3.4 & 2.9 & 0
\end{bmatrix}
\end{array}
$$

Shared Nearest Neighbors (sNN) graph

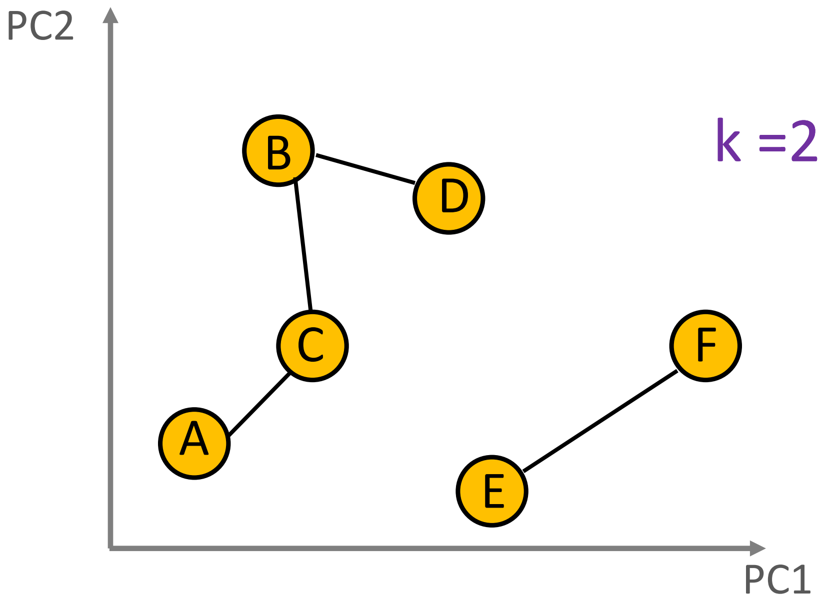
- An **snn** graph is built **from a knn** graph.
- Two cells are connected if they share a neighbor.



$$
\begin{array}{c c}
 & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} &
\begin{bmatrix}
0 & 3.2 & 1.7 & 0 & 0 & 0 \\
0 & 0 & 2 & 2.1 & 0 & 0 \\
1.7 & 2 & 0 & 0 & 0 & 0 \\
0 & 2.1 & 2.2 & 0 & 0 & 0 \\
0 & 0 & 2.6 & 0 & 0 & 2.9 \\
0 & 0 & 0 & 3.4 & 2.9 & 0
\end{bmatrix}
\end{array}
$$

Shared Nearest Neighbors (sNN) graph

- An **snn** graph is built **from a knn** graph.
- Two cells are connected if they share a neighbor.



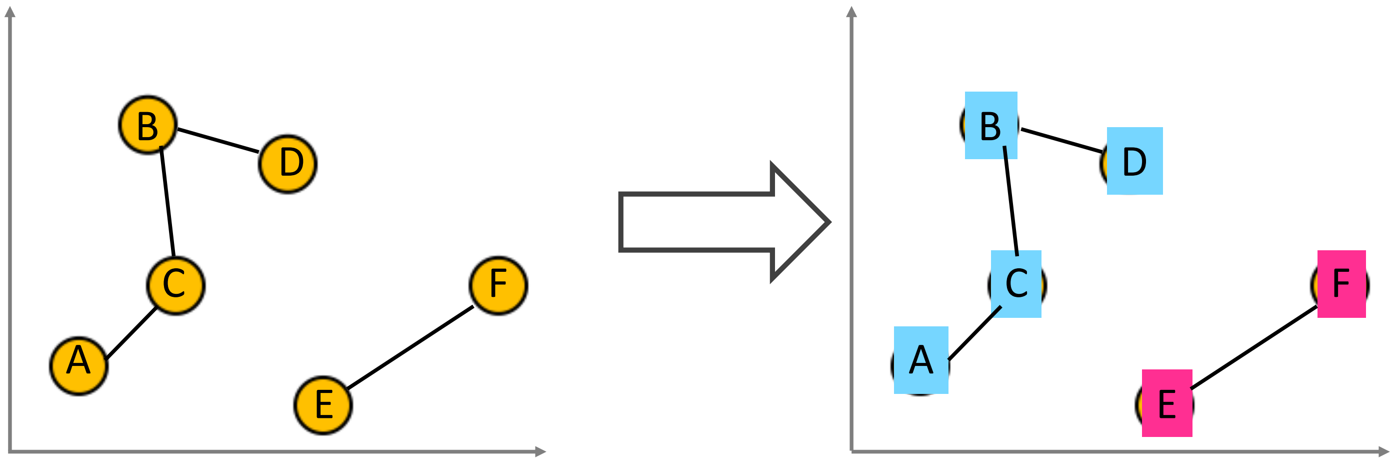
$$
\begin{array}{c c}
 & \begin{array}{cccccc} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & \mathrm{E} & \mathrm{F} \end{array} \\
\begin{array}{c} \mathrm{A} \\ \mathrm{B} \\ \mathrm{C} \\ \mathrm{D} \\ \mathrm{E} \\ \mathrm{F} \end{array} &
\begin{bmatrix}
0 & 3.2 & 1.7 & 0 & 0 & 0 \\
0 & 0 & 2 & 2.1 & 0 & 0 \\
1.7 & 2 & 0 & 0 & 0 & 0 \\
0 & 2.1 & 2.2 & 0 & 0 & 0 \\
0 & 0 & 2.6 & 0 & 0 & 2.9 \\
0 & 0 & 0 & 3.4 & 2.9 & 0
\end{bmatrix}
\end{array}
$$

FindNeighbors(sobj, dims = 1:nPC, k.param = 20)

Communauty (clustering)

We want to group node (cells) together

How many groups shoud you take ?

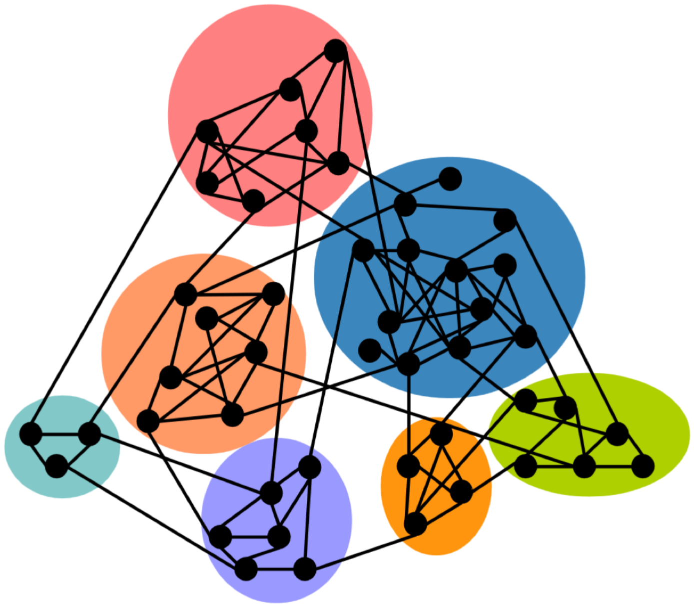


Community: a group of vertices that are strongly connected to each other and weakly connected to the rest of the network..

Community detection (clustering)

Ideal partitioning:

- many connections within communities
- few connections between communities.



Modularity : measure the quality of a partition

- evaluates the density of connections within communities compared to those between communities

Community detection (clustering)

We seek a partitioning that **optimizes modularity**

How

Louvain algorithm
Leiden

Hierarchical approach

The number of groups depends on a parameter called "resolution. »

Test several resolution parameters.

Generate more clusters than the expected number of cell populations.

```
FindClusters(sobj, resolution = c(0.1,0.2,0.5,1))
```

Clustering scRNAseq

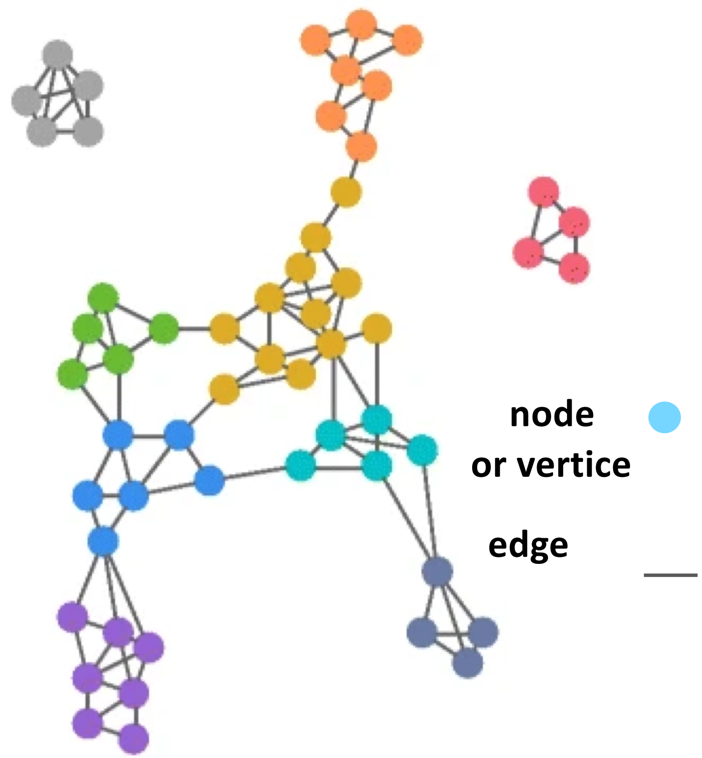
Strategy : represent the proximity between cells in the form of a graph, which will then be partitioned

a. Construct a graph from the principal components (PC

FindNeighbors(sobj, dims = 1:nPC)

b. Partition the graph using Louvain or **Leiden** algorithm

FindClusters(sobj, resolution = c(0.1,0.2,0.5,1))



Levine, J. H. et al.. *Cell* **162**, 184–197 (2015).

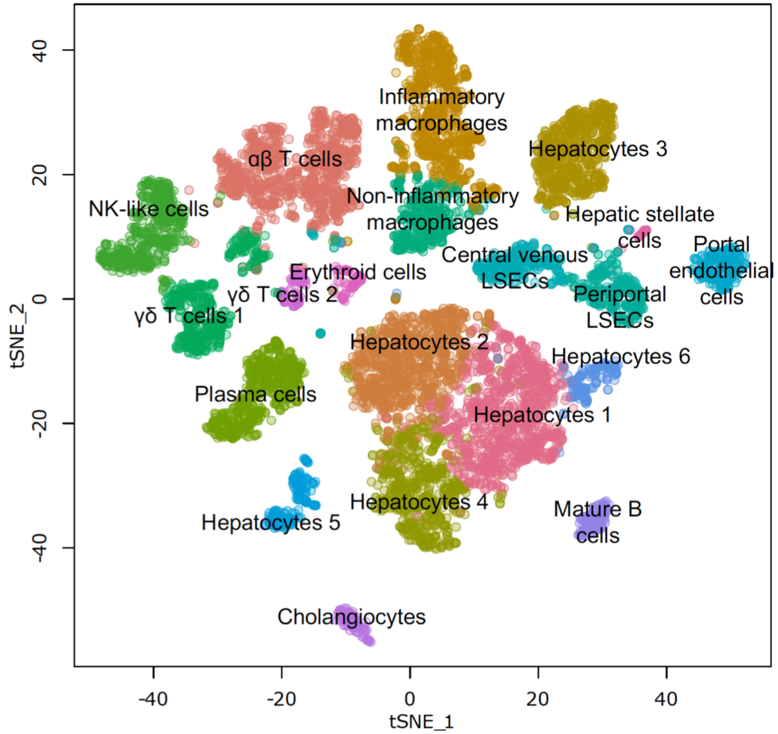
Clustering scRNAseq : key points

Aims : obtain groups of cells

a. Reduce the dimensionality of the data
   - choose the **number of HVG** (500-3000)
   - perform PCA on HVGs (keep 10-50 components)

b. Construct a graph from the principal components
   - choose the **number of PC**
   - (choose the number of neighbors)

c. Partition the graph using Louvain or Leiden algorithm
   - try various **resolution**

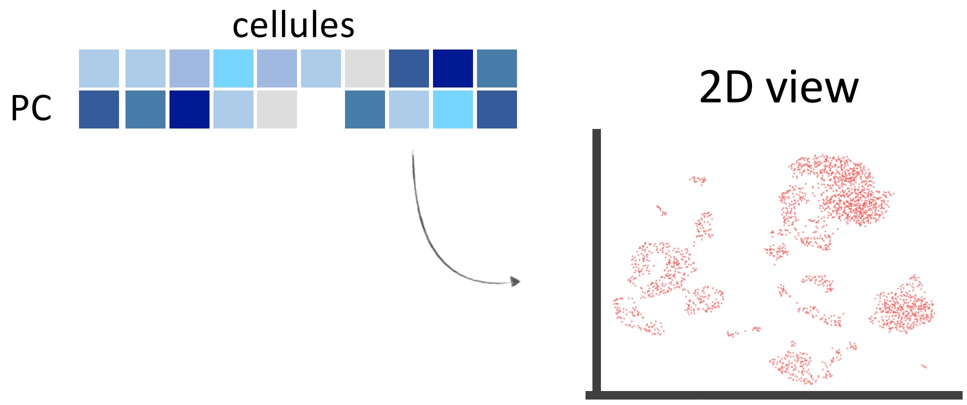


Visualize the cluster on a 2D embedding

Z. Clark et al. Nature Protocole (2021) Tutorial: guidelines for annotating single-cell transcriptomic maps

Visualizing High-Dimensional Data in 2D

- Cells are often represented on a two-dimensional plot to identify interesting characteristics.
- **UMAP** (and **t-SNE**): Non-linear methods
  - very useful to visualize clusters
  - for **visualization only** : distance between clusters cannot be interpreted



Visualizing High-Dimensional Data in 2D

t-SNE (t-distributed stochastic neighbor embedding)

- Focuses on the local structure of the data
- **Perplexity**: the higher this number, the more importance is given to the global structure

UMAP (Uniform Manifold Approximation and Projection)

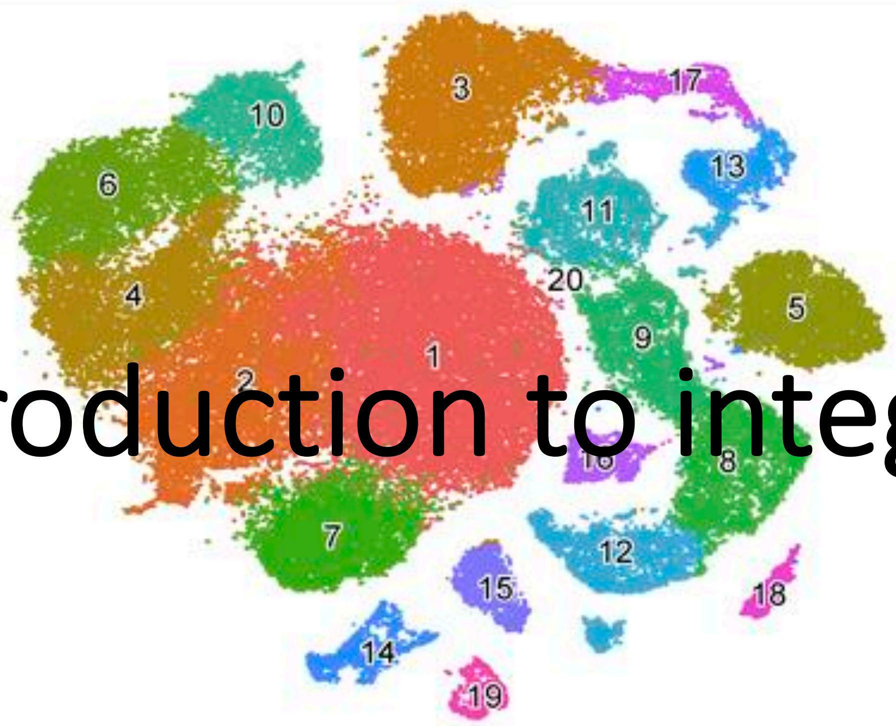
- Better preserves the global structure of the data
- Computation time is more suited to large datasets
- New data can be added to an existing projection
- **n_neighbors**: number of neighbors used during the similarity computation phase between cells (similar to perplexity)
- **min_dis**t: affects the appearance (tightness of points)

Excellent link : https://pair-code.github.io/understanding-umap/

SincellTE 2024 : 5th Workshop in single cell data analyses

Transcriptomics, Spatial and Long-reads

sincellTE



Introduction to integration

Lorette Noiret

Sorbonne Université, Institut Curie

Integration of multiple scRNAseq datasets: challenges

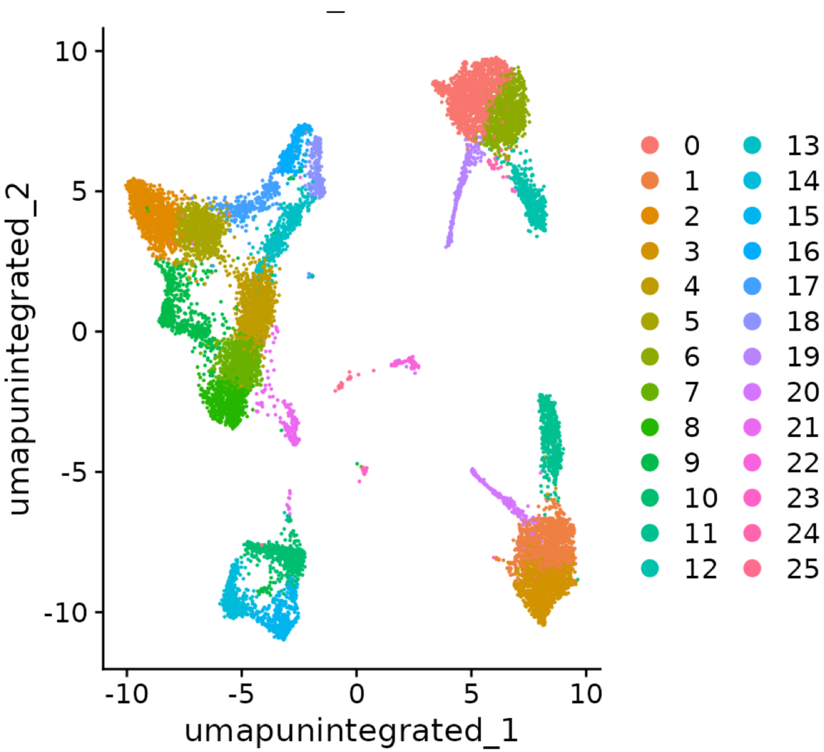
Batch effects : technical or biological variations within and across datasets

- different replicates (patients)
- different conditions (WT/KO, control/disease)
- different experimental protocols (data published datasets from different labs)
- different sequencing plateforms…

Clustering may separate the replicate / condition instead of the cell types

Batch effects

Combine 2 conditions (CTRL/STIM)
and cluster all the cells



Sample identity

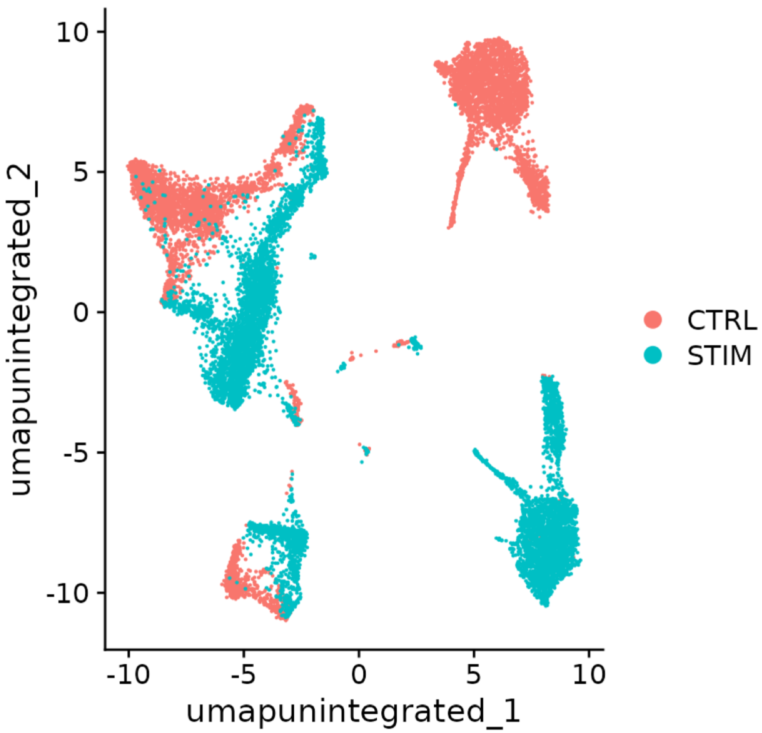


Image : Seurat

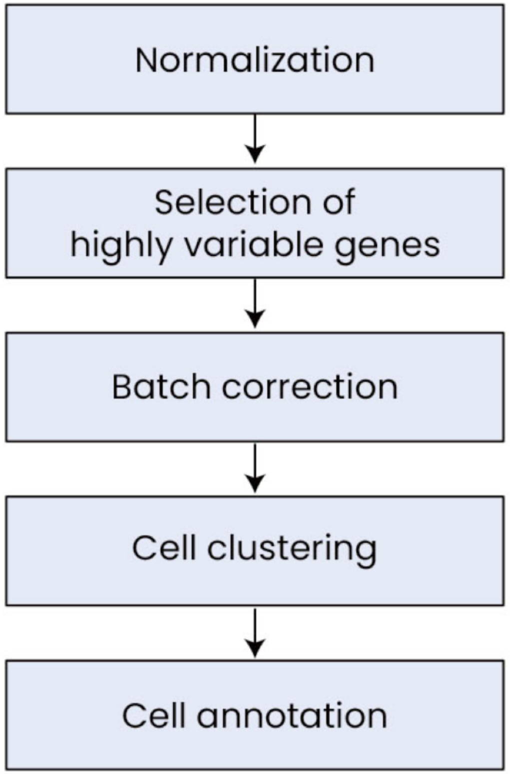
Batch effects : solution 1 – cluster cell separately

- Perform clustering on each sample separately
- Advantages : no issues with batch effect
- Disavantage :
    - more work
    - small number of cell per sample and noisy data : reliability clusters ?

Batch effects : solution 2 : Integration

- **Align cells** across datasets (correct the expression to remove batch effect) before perfoming clustering
- Advantages :
  - remove unanted variation (batch effect)
  - identify shared and unique cell types
- Disavantages :
  - modify the signal : risk of overcorrection
  - identification of rare populations of cells more challenging

Bioinformatic pipeline with integration



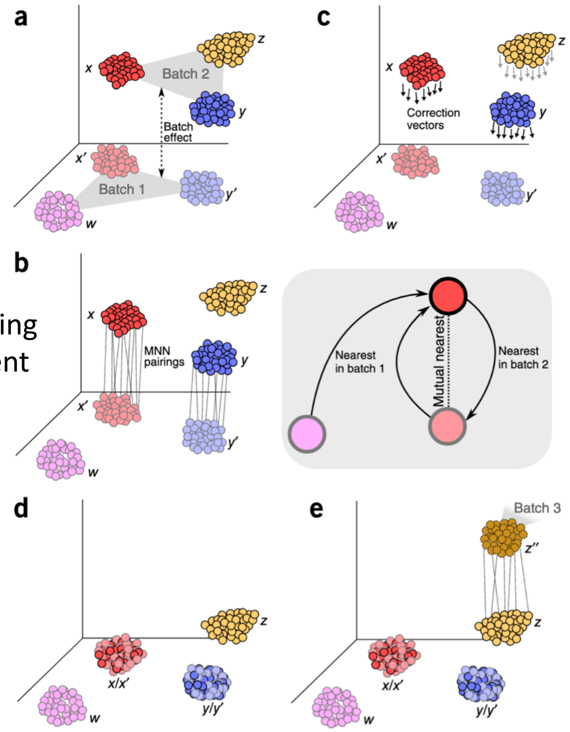
Mol. Cells 2023; 46(2): 106-119

Highly variable genes most frequently selected across the batches

New embedding (transform expression)

Mutual Nearest Neighbors (MNN)

Search corresponding cells across different datasets



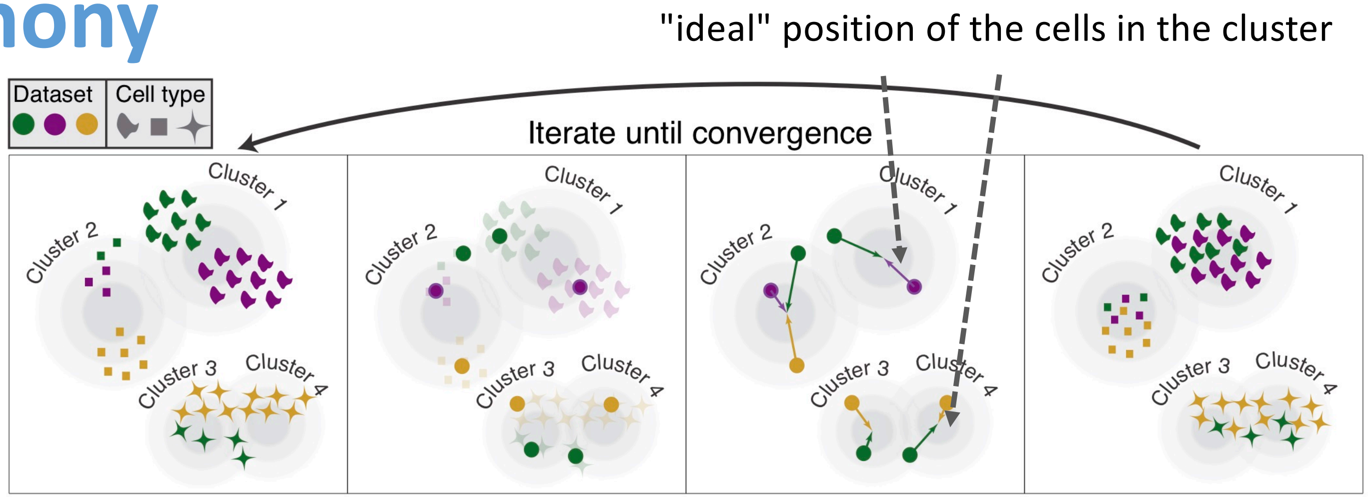
Iidentifies matching cell types by finding **MNN pairs** of cells

- calculate batch-correction vectors between the MNN pairs.
  - Batch 1 is regarded as the reference, and batch 2 is integrated into it by subtraction of correction vectors.

- The integrated data are considered the reference, and the procedure is repeated for integration of any new batch.

Laleh Haghverdi et al. Batch effects in single-cell Rna-sequencing data are corrected by matching mutual nearest neighbors. nature biotechnology 2018

Harmony



A Soft assign cells to clusters, favoring mixed dataset representation

B Get cluster centroids for each dataset

C Get dataset correction factors for each cluster

D Move cells based on soft cluster membership

- scale the data and perform dimension reducion (PCA)
- cluster the cell and evaluate cluster diversity (sample origin)
- within each cluster, adjust/correct the cell embeddings (positions in the PCA space) to reduce batch effects.
- Repeat until the corrections are small

https://portals.broadinstitute.org/harmony/articles/quickstart.html

Detailed Walkthrough of Harmony Algorithm : https://portals.broadinstitute.org/harmony/advanced.html

Clustering scRNAseq : key points

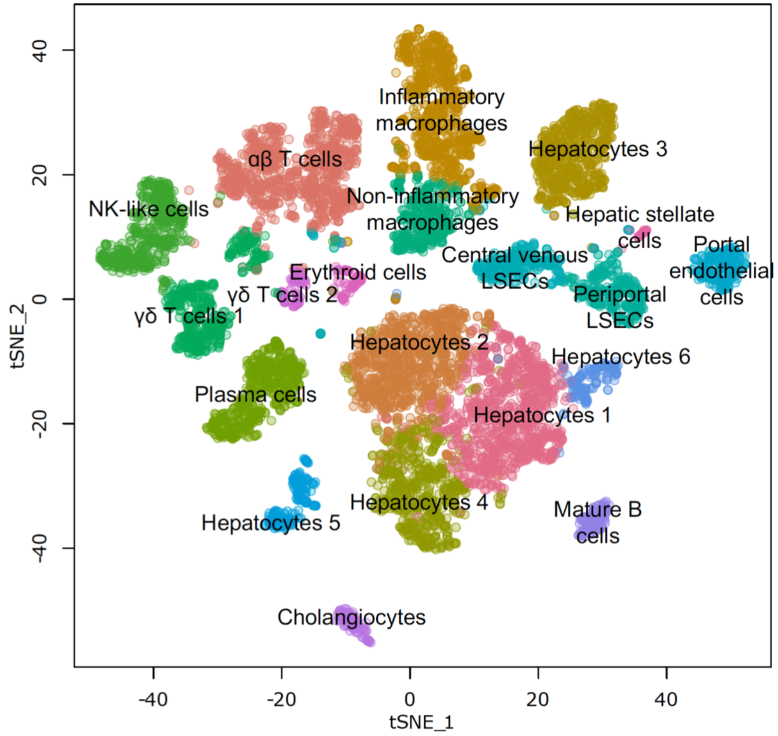
Combine all the samples in one Seurat object

Perform clustering without integration

- select HVG
- perform PCA on HVGs
- perform clustering
- check if clusters are biased toward a sample

Perform clustering with integration

- select HVG (common to each samples)
- integration : correct for batch effects
- perform clustering on integrated data
- check lusters after integration



Z. Clark et al. Nature Protocole (2021) Tutorial: guidelines for annotating single-cell transcriptomic maps