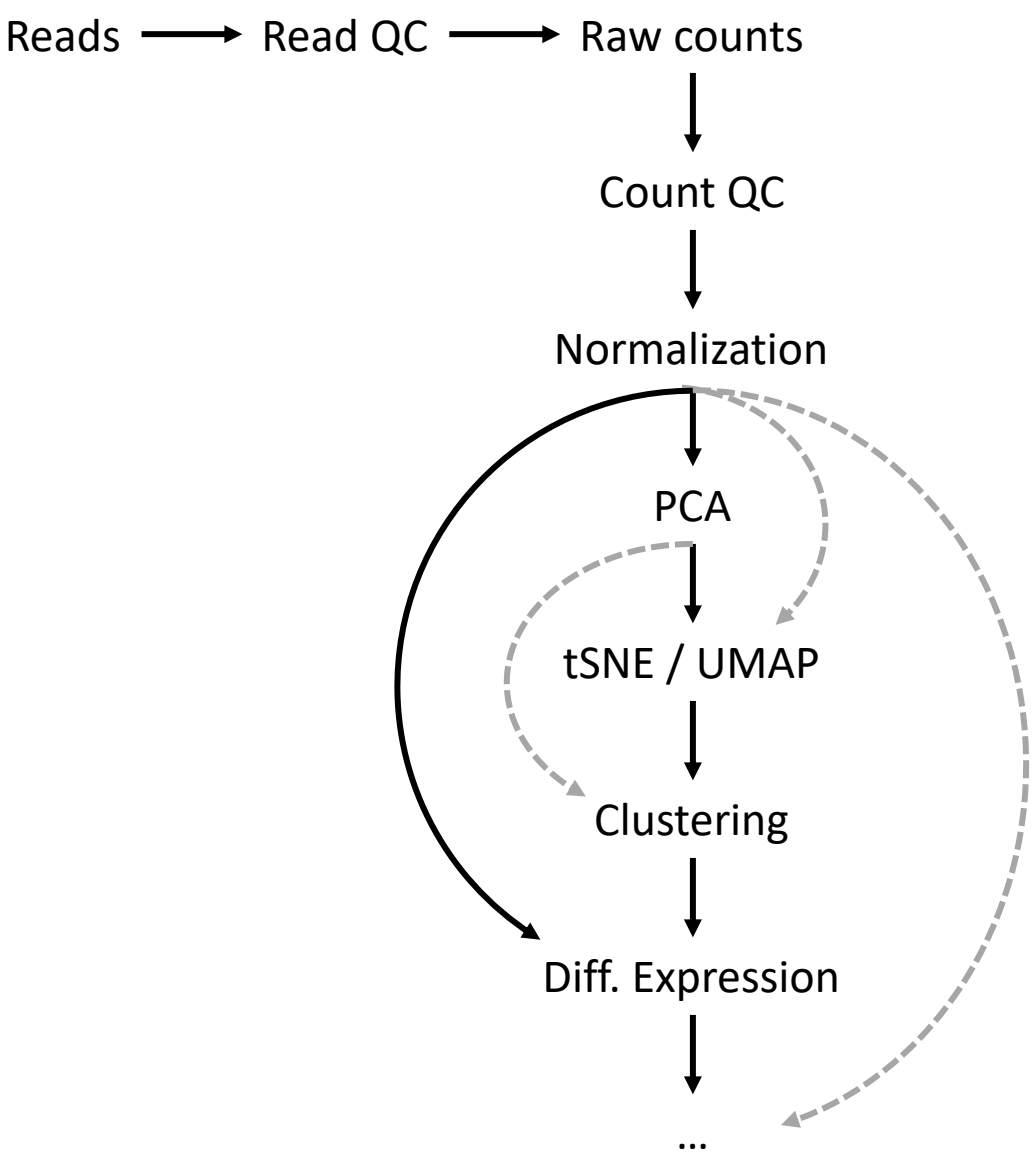
Dimensionality reduction

Paulo Czarnewski, **ELIXIR-Sweden (NBIS)**

European Life Sciences Infrastructure for Biological Information
www.elixir-europe.org

A general single cell analysis workflow



The workflow is dataset-specific:

- Research question
- Batches
- Experimental Conditions
- Sequencing method
- …

Why dimensionality reduction?

- Simplify complexity, so it becomes easier to work with.
  Reduce number of features (genes)
  In some: Transform non-linear relationships to linear

- "Remove" redundancies in the data

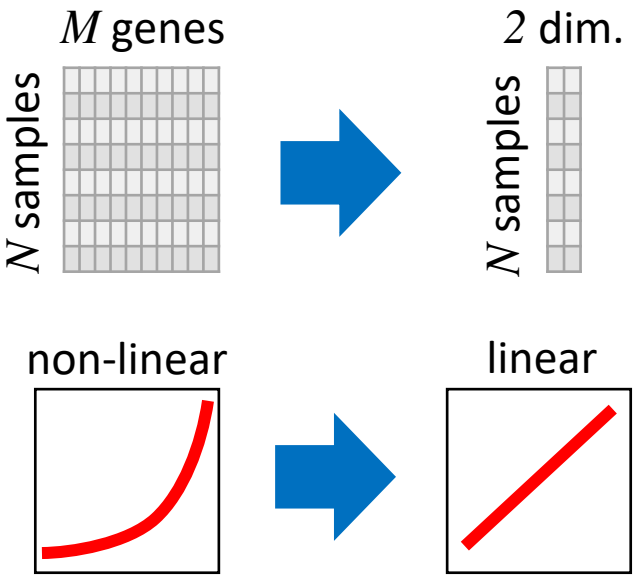
- Identify the most relevant information (find and filter noise)

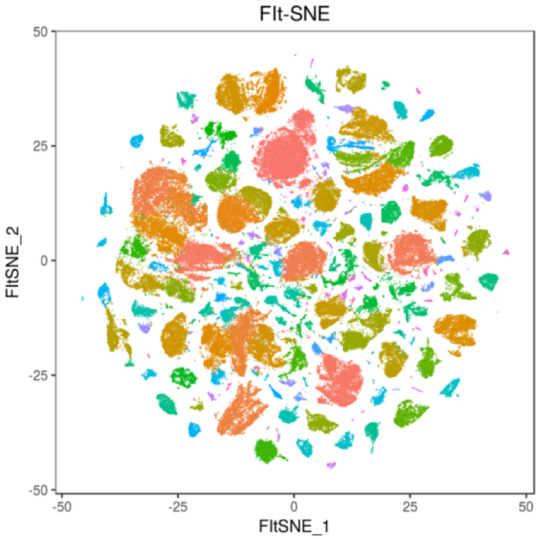
- Reduce computational time for downstream procedures

- Facilitate clustering, since some algorithms struggle with too many dimensions

- Data visualization

… and more …





Some dimensionality reduction algorithms

They can be divided into 3 major groups:

PCA	linear	Matrix Factorization		
ICA	linear	Matrix Factorization		
MDS	non-linear	Matrix Factorization		
Sparce NNMF	non-linear	Matrix Factorization	2010	https://pdfs.semanticscholar.org/664d/40258f12ad28ed0b7d4c272935ad72a150db.pdf
cPCA	non-linear	Matrix Factorization	2018	https://doi.org/10.1038/s41467-018-04608-8
ZIFA	non-linear	Matrix Factorization	2015	https://doi.org/10.1186/s13059-015-0805-z
ZINB-WaVE	non-linear	Matrix Factorization	2018	https://doi.org/10.1038/s41467-017-02554-5

Diffusion maps	non-linear	graph-based	2005	https://doi.org/10.1073/pnas.0500334102
Isomap	non-linear	graph-based	2000	10.1126/science.290.5500.2319
t-SNE	non-linear	graph-based	2008	https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf
- BH t-SNE	non-linear	graph-based	2014	https://lvdmaaten.github.io/publications/papers/JMLR_2014.pdf
- Flt-SNE	non-linear	graph-based	2017	arXiv:1712.09005
LargeVis	non-linear	graph-based	2018	arXiv:1602.00370
UMAP	non-linear	graph-based	2018	arXiv:1802.03426
PHATE	non-linear	graph-based	2017	https://www.biorxiv.org/content/biorxiv/early/2018/06/28/120378.full.pdf

scvis	non-linear	Autoencoder (MF)	2018	https://doi.org/10.1038/s41467-018-04368-5
VASC	non-linear	Autoencoder (MF)	2018	https://doi.org/10.1016/j.gpb.2018.08.003

… and many more

PCA

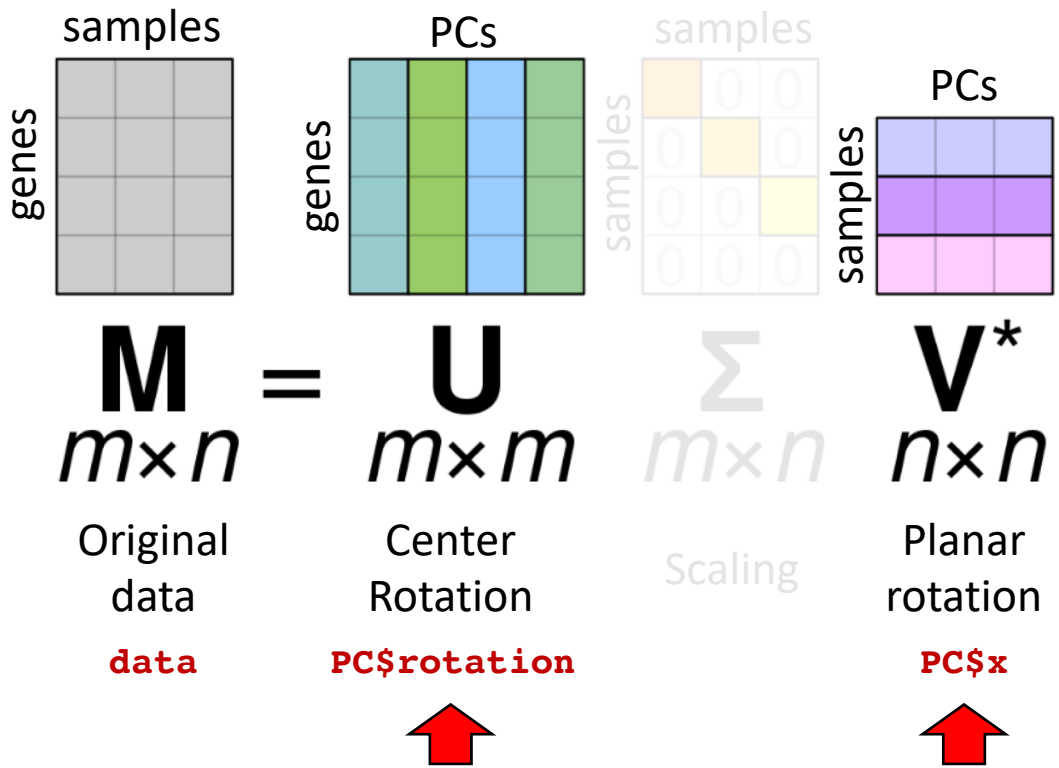
Principal Component Analysis

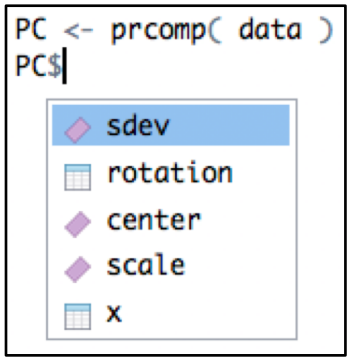
How PCA works

It is a LINEAR algebraic method of dimensionality reduction.

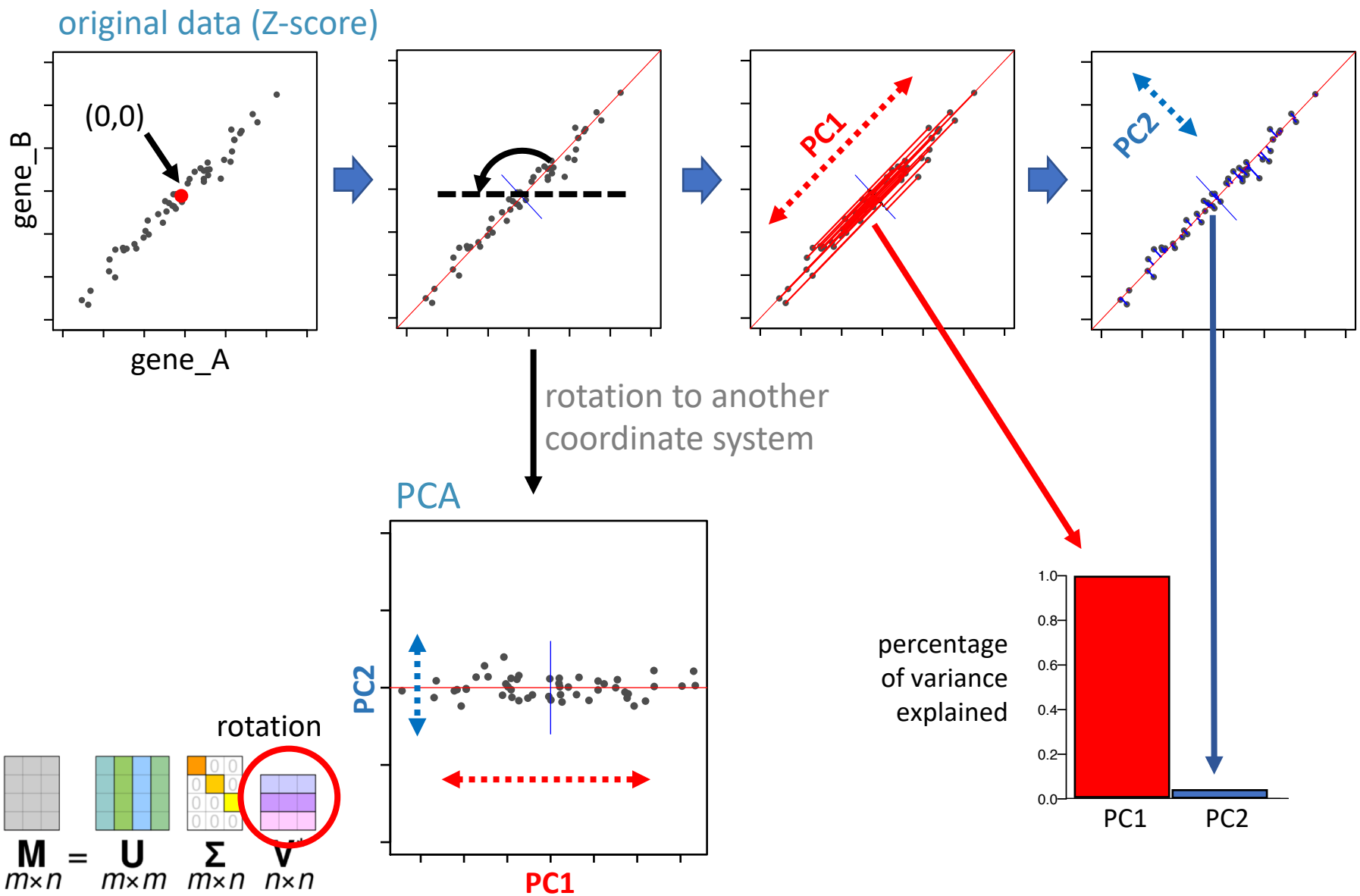
It is a case inside Singular Value Decomposition (SVD) method (data compression)

Any matrix can be decomposed as a multiplication of other matrices (Matrix Factorization).





How PCA works



How PCA works

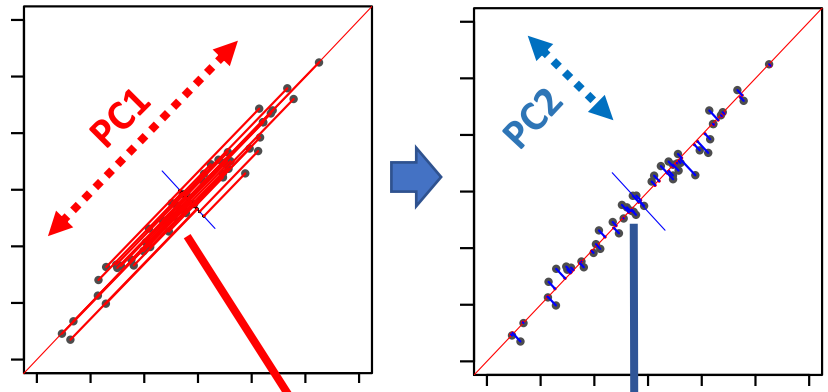
PC1 explains >98% of the variance

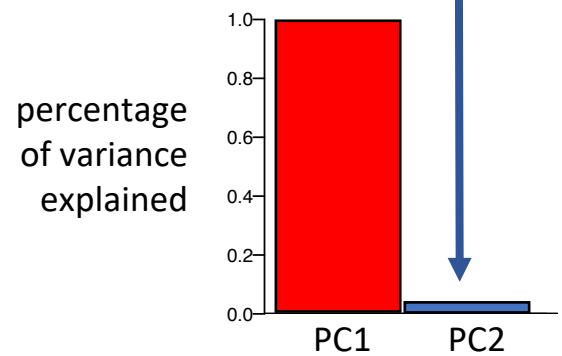
1 PC thus represents 2 genes very well

"Removing" redundancy

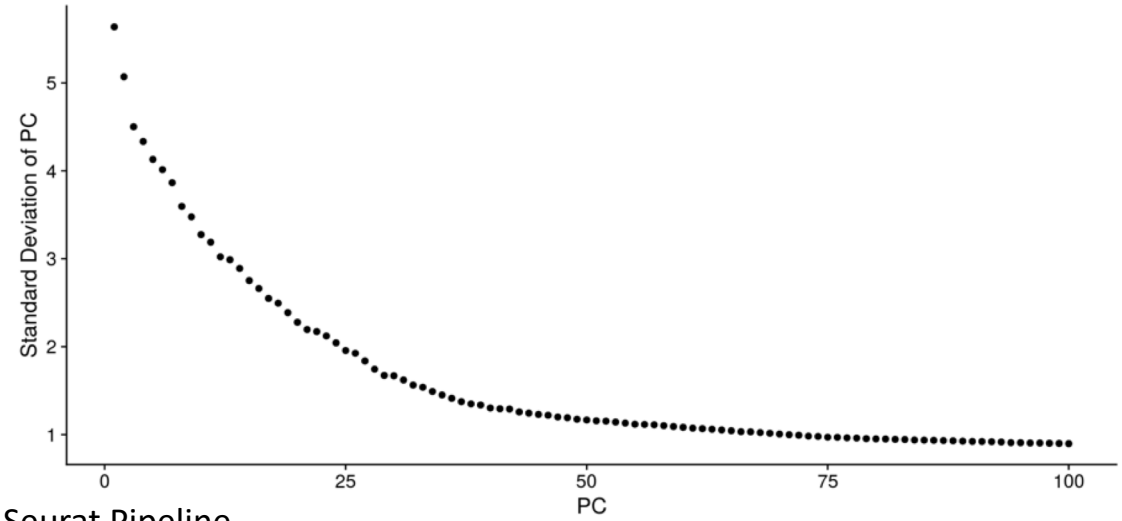
PC2 is nearly insignificant in this example

Could be disregarded



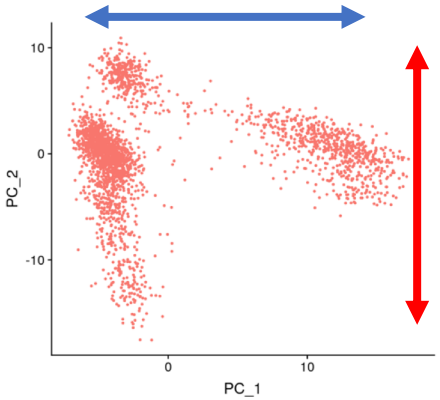


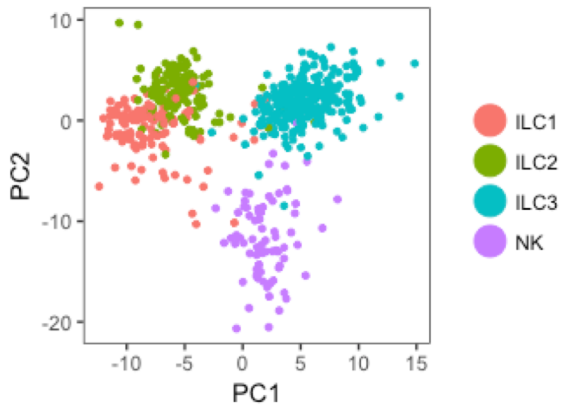
In real life …



Seurat Pipeline

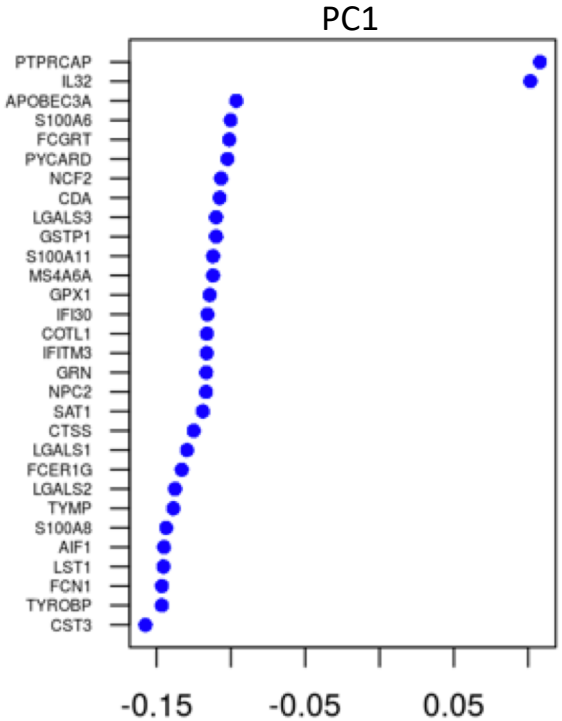
PCA in single cell data

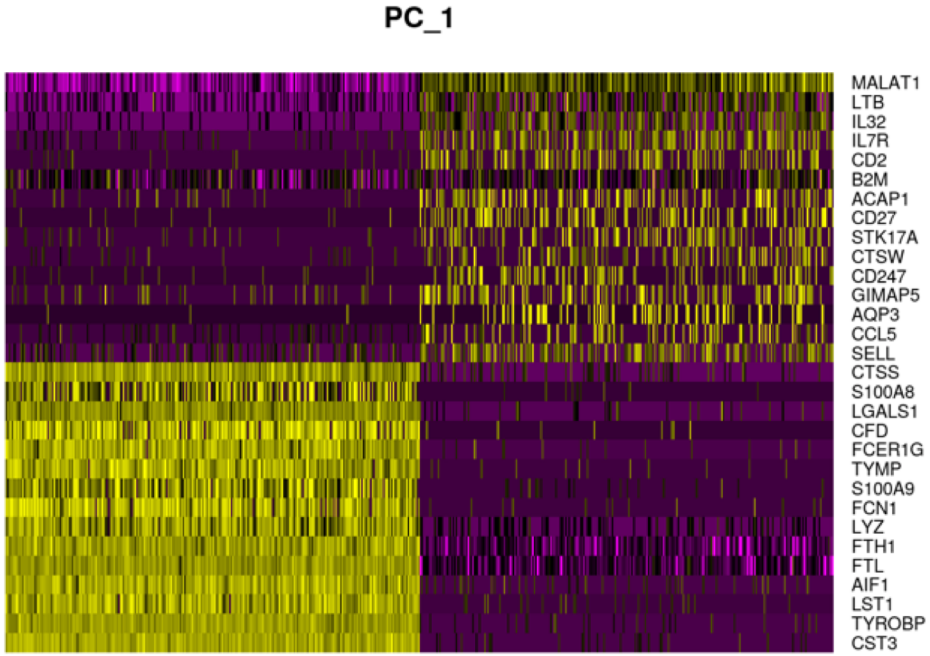




PC1 and PC2 are commonly correlated to sequencing depth and cell heterogeneity/complexity

(but not always …)





Seurat Pipeline / Forkel et all 2015

PCA: Summary

To keep in mind:

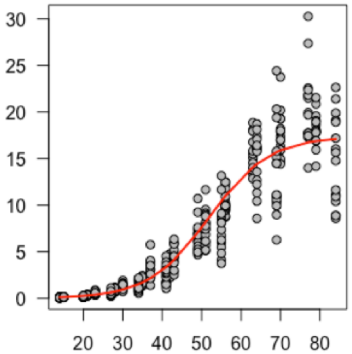
- It is a LINEAR method of dimensionality reduction
- It is an interpretable dimensionality reduction
- Data is usually SCALED prior to PCA (Z-score | see ScaleData in the Seurat)
- The TOP principal components contain higher variance from the data
- Can be used as FILTERING, by selecting only the top significant PCs
  - PCs that explain at least 1% of variance
  - Jackstraw of significant p-values
  - The first 5-10 PCs

Problems:

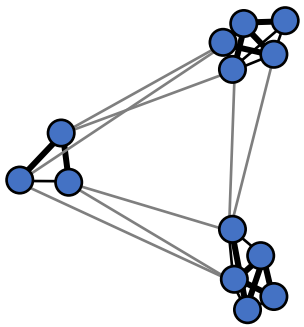
- It performs poorly to separate cells in 0-inflated data types (because of it non-linearity nature)
- Cell sizes and sequencing depth are usually captured in the top principal components

A very brief intro to graphs

Graphs



This is a PLOT



This is GRAPH
(a.k.a. network)

- Each dot is a cell (or a gene)
- Each line represents a connection between 2 cells
- Each connection can be weighted as a proximity between cells
  - Correlation (high and positive)
  - Euclidean distance (low)
  - etc.

Graph-based dimensionality reduction algorithms can be divided into 2 main steps:

1. Construct a weighted graph based on the top k connections
   (a.k.a. k-nearest neighbors, KNN)

2. The low dimensional layout of the graph is computed and optimized

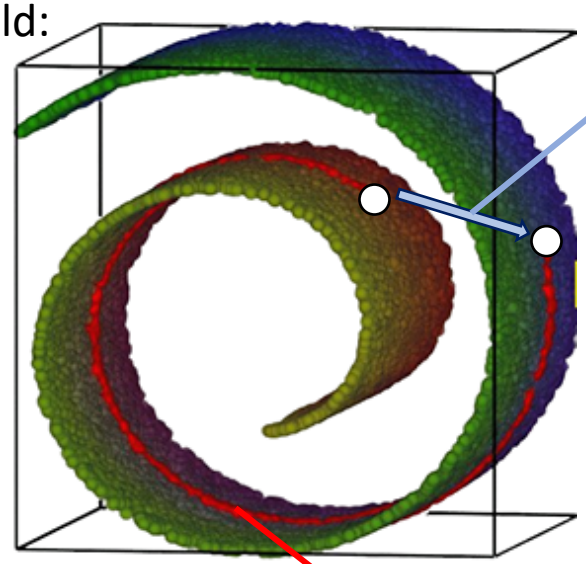
tSNE

t-Stochastic Neighborhood Embedding

How t-SNE works

It is a graph-based NON-LINEAR dimensionality reduction

Manifold:



distance in PCA space
(euclidean distance)

distance in t-SNE space

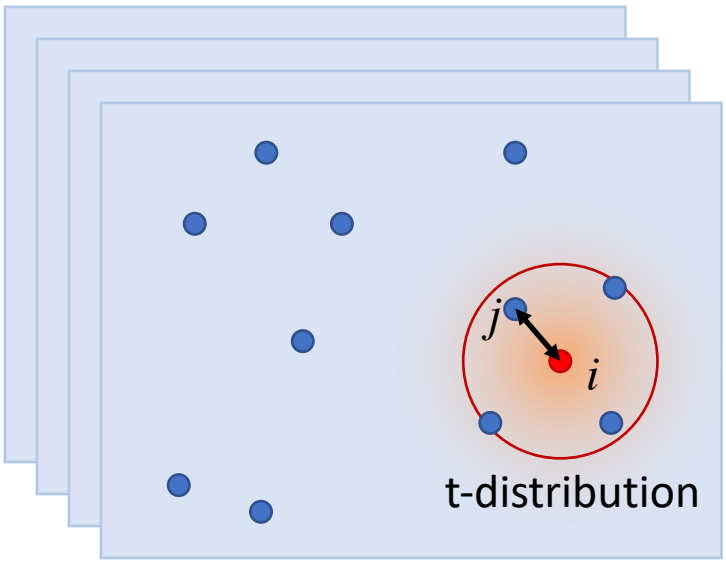
In other words, t-SNE calculates the distances based on the distance to the neighbor cell

Src: http://web-ext.u-aizu.ac.jp/~shigeo/home.html

Maaten et al (2008) *Journal of Machine Learning Research*

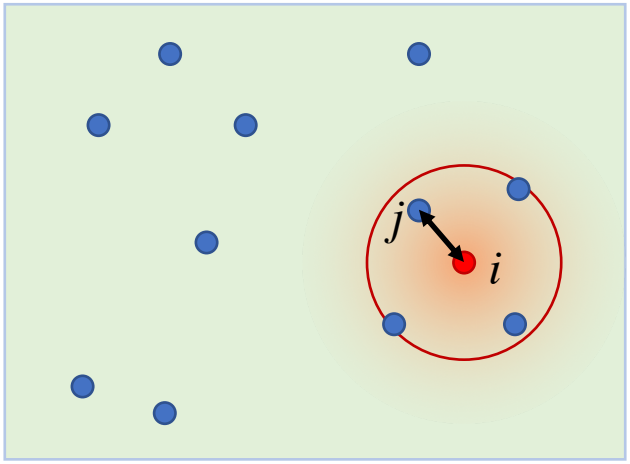
How t-SNE works

High dimension



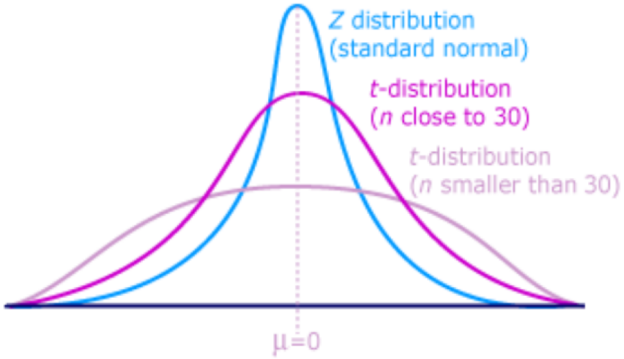
$\leftrightarrow = p_{j|i}$

Low dimension

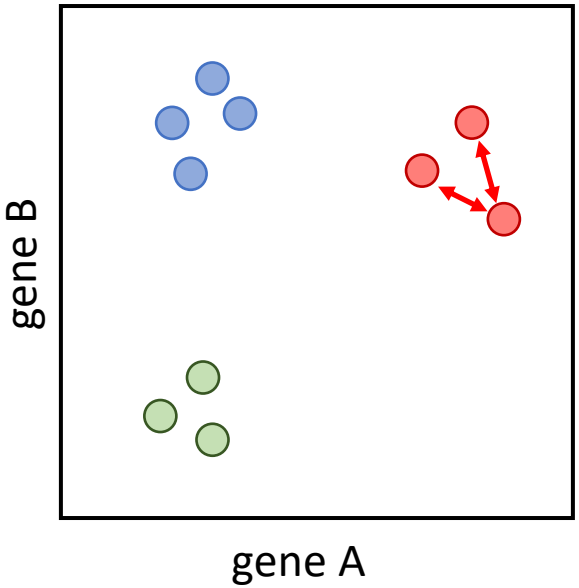


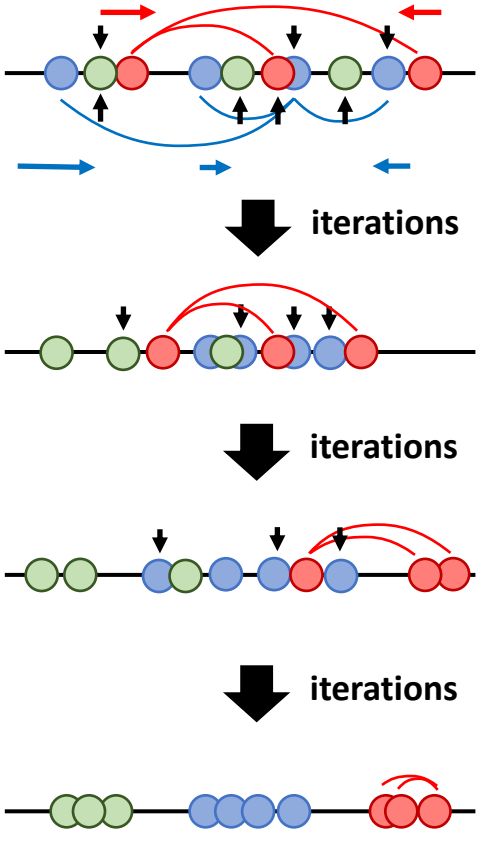
$\leftrightarrow = q_{j|i}$

$p_{j|i}$ and $q_{j|i}$ measure the <u>conditional probability</u> that a point i would pick point j as it's nearest neighbor, in high (p) and low (q) dimensional space respectively.



How t-SNE works





The same concept applies to embedding into 2 dimensions

t-SNE hyper-parameters

- Barnes-Hut's tSNE implementation - $O(n \log n)$

  `Rtsne & Seurat & viSNE (MATLAB)`

Maaten (2014) *Journal of Machine Learning Research*

The definition of the t-SNE and the chances of converging correctly depends on the hyper-parameters ("tuning" parameters).

t-SNE has over 10 hyper-parameters that can be optimized for your specific data.

The most common hyper-parameters are:

- Perplexity
- Number of iterations
- Learning rate
- Theta (for BH t-SNE)

Check this link: https://distill.pub/2016/misread-tsne/

Important notes about t-SNE

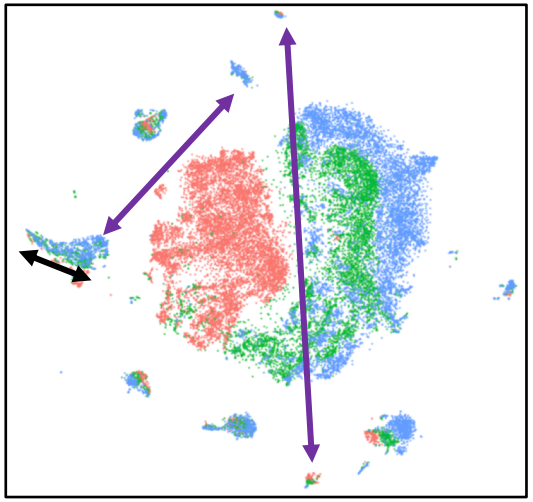
- Unlike PCA, it is a stochastic algorithm, so it will never produce the same output (unless you use a `seed()` to lock the random estimators).

- The cost function never reaches the minima, and it is not an indicator how good the graph looks.

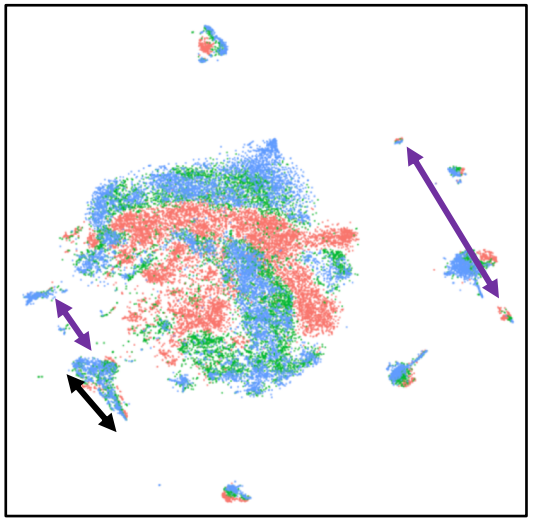
- The cost function in t-SNE minimizes the distance between similar points (and ignore the distant ones – local embedding)
  The distances within a group are slightly meaningful, but not between groups!

- To add more samples, you need to re-run the algorithm from start.

Converged successfully



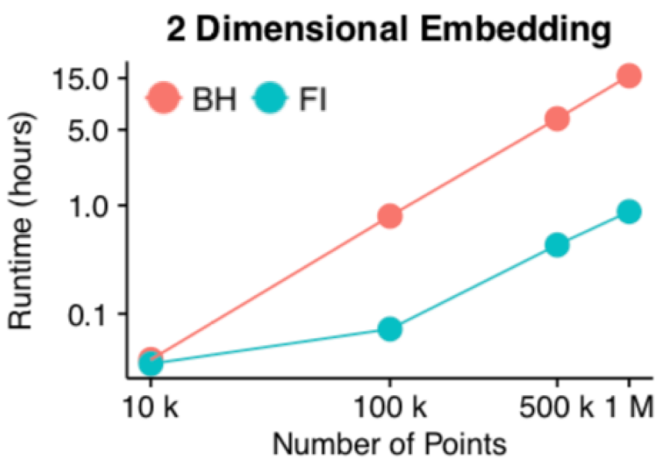
Failed to converge



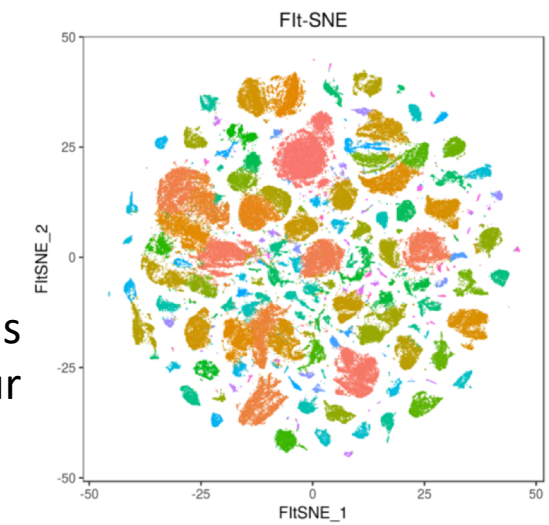
Efficient t-SNE implementation

- Fast Fourier Transform-accelerated Interpolation-based t-SNE - $O(n)$

Linderman et al (2017) *BioRxiv*



250k cells
1 hour



Linderman et al 2017 / Seurat Pipeline

t-SNE: summary

To keep in mind:

- It is a NON-LINEAR method of dimensionality reduction
- It is the current GOLD-STANDARD method in single cell data (including scRNA-seq)
- Can be run from the top PCs (e.g.: PC1 to PC10)

Problems:

- It does not learn an explicit function to map new points
- It's cost function is not convex – This means that the optimal t-SNE cannot be computed
- Too many hyper-parameters to be defined empirically (dataset-specific)
- It does not preserve a global data structure (only local)

UMAP

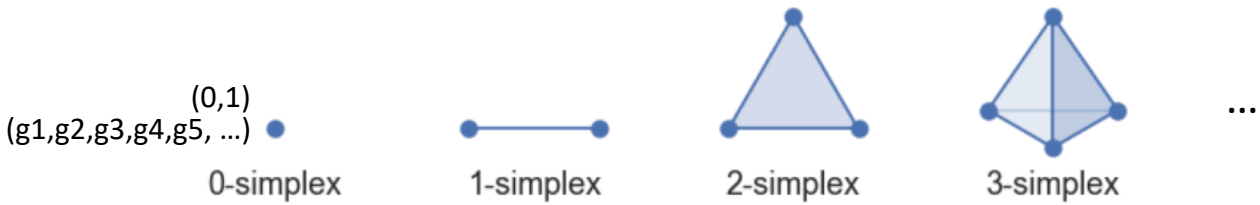
Uniform Manifold Approximation and Projection

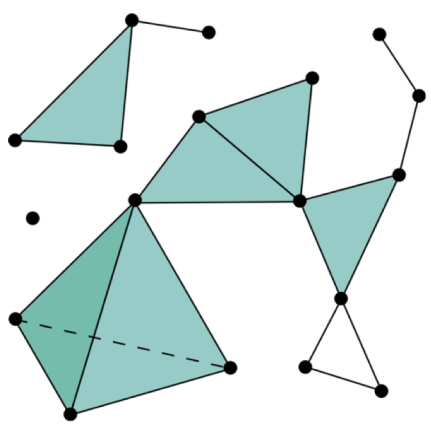
How UMAP works

It is based on **<u>topological structures</u>** in multidimensional space (simplices)

Points are connected with a line (edge) if the distance between them is below a threshold:

- Any distance metric can be used (euclidean)



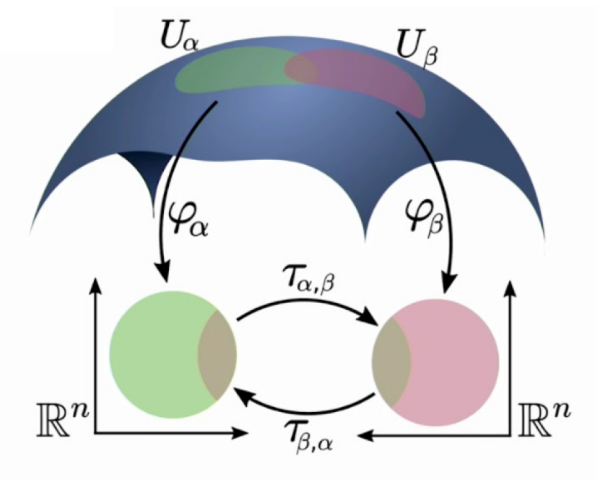


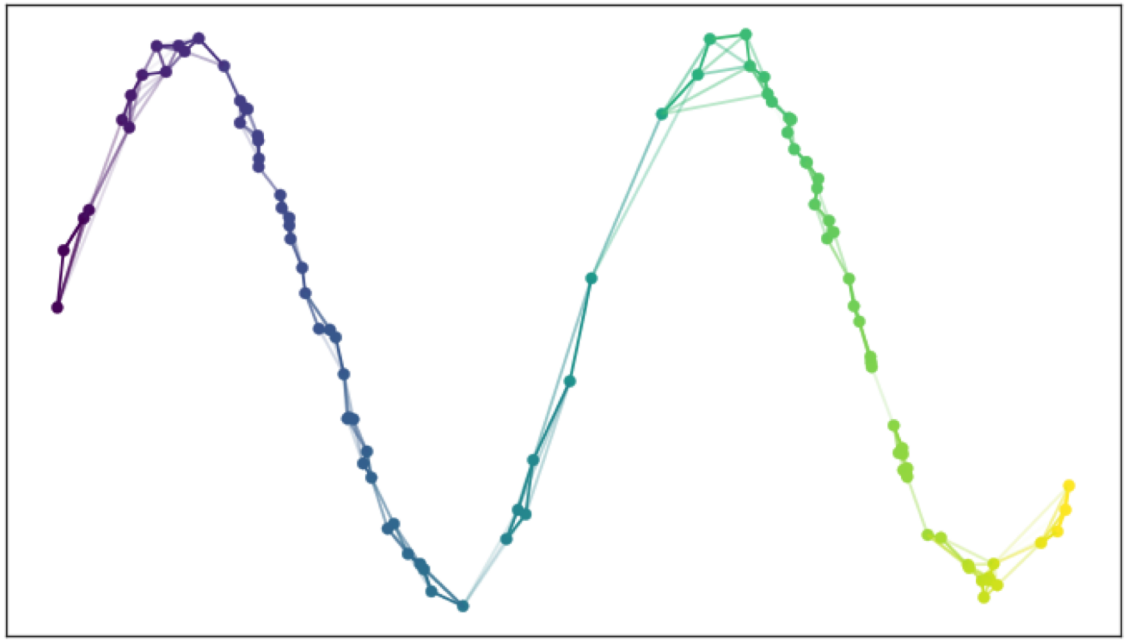
This way, by constructing the simplicial complexes beforehand allows UMAP to calculate the relative point distances in the lower dimension

(instead of randomly assigning as in tSNE)

McInnes et al (2018) *BioRxiv*

How UMAP works





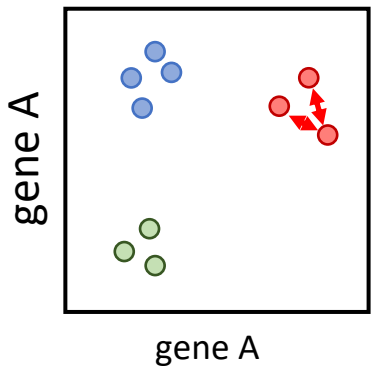
The distance in the manifold are the same, but not in the REAL space.

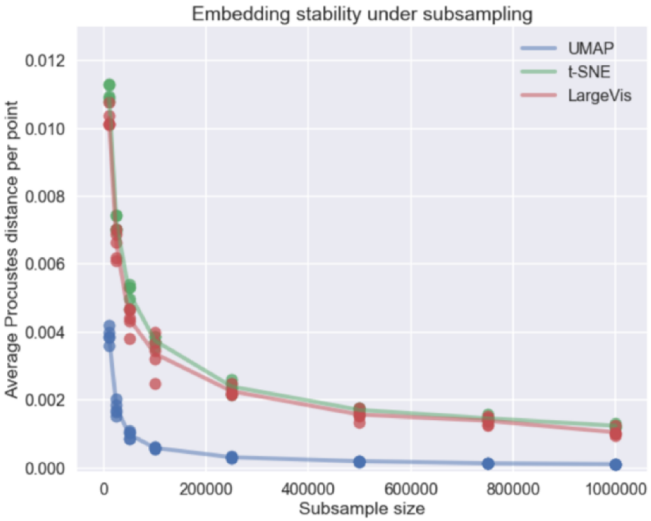
The distance is now “variable” in the REAL space for each point (t-SNE was fixed)

McInnes et al (2018) *BioRxiv*

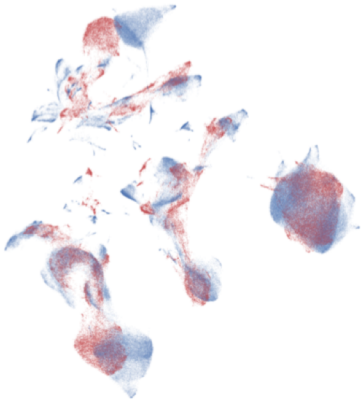
https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

UMAP

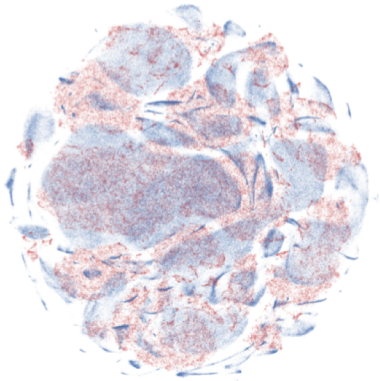




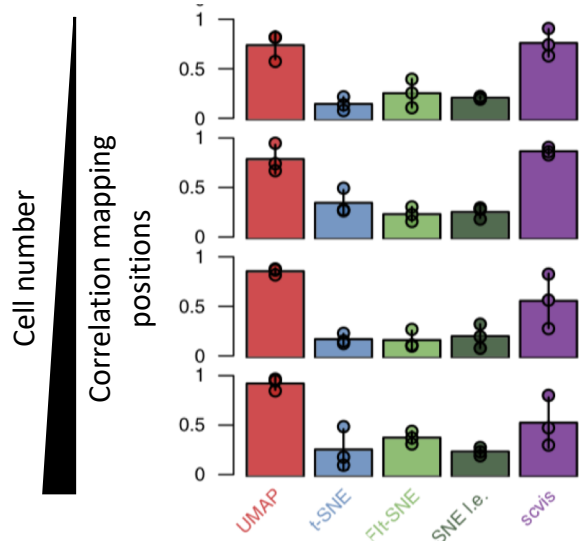
Since UMAP learns the global data structure and is less dependent on random initiators (like t-SNE), it can recreate low dimensional embedding regardless of the dataset size.



(a) UMAP



(b) t-SNE



McInnes et al (2018) *BioRxiv*

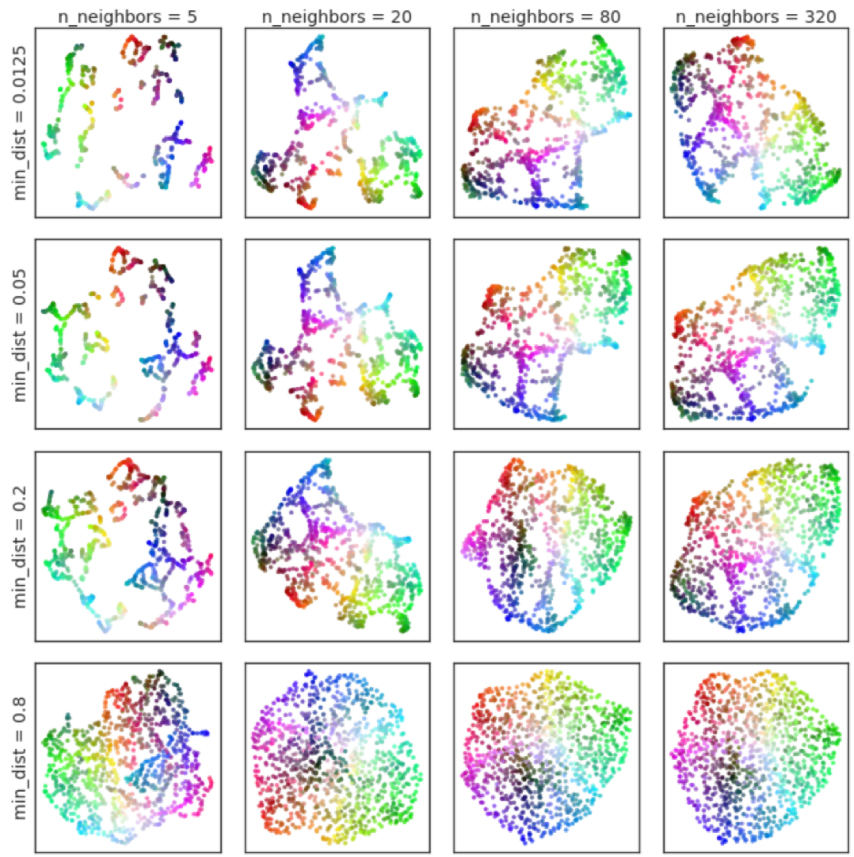
Becht & McInnes et al (2019) *Nat Biot*

UMAP hyper-parameters

UMAP assumes that there is a manifold in the dataset, it could also tend to cluster noise.

As for t-SNE, checking the parameters is also important.

Embedding of random noise

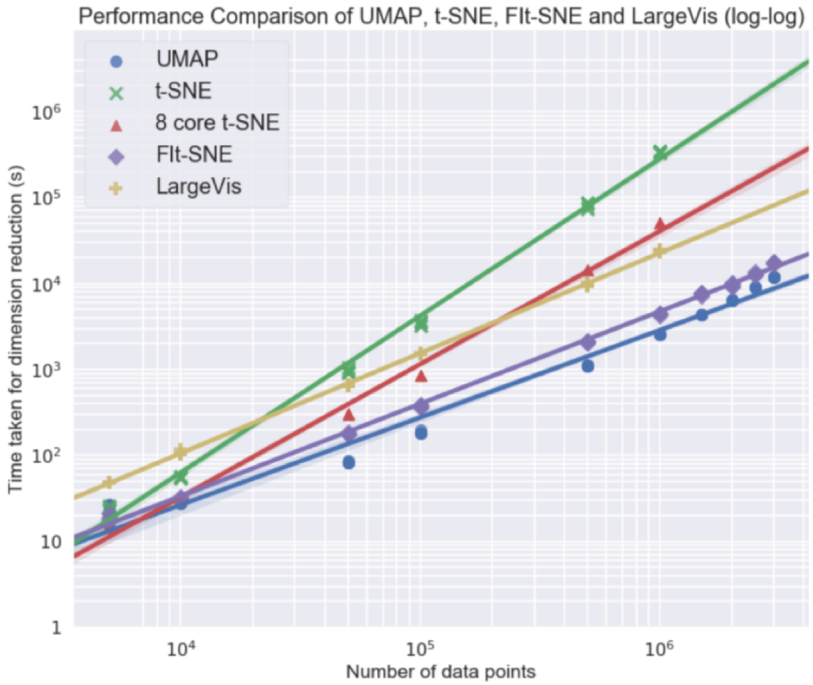


McInnes et al (2018) *BioRxiv*

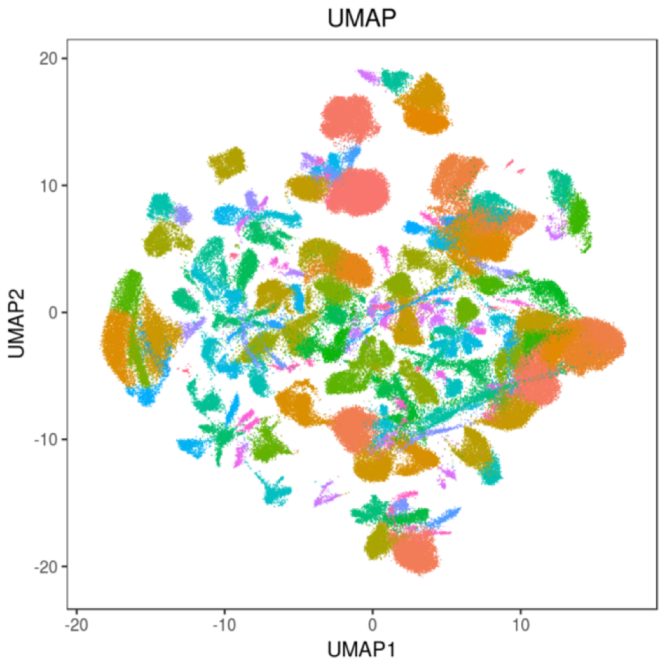
https://umap-learn.readthedocs.io/en/latest/parameters.html

UMAP hyper-parameters

UMAP's mathematical improvements allows much faster computations compared to current state-of-the-art methods.



250k cells
7 min



McInnes et al (2018) *BioRxiv*

Becht & McInnes et al (2019) *Nat Biot*

UMAP: Summary

To keep in mind:

- It is a NON-LINEAR graph-based method of dimensionality reduction
- Very efficient - $O(n)$
- Can be run from the top PCs (e.g.: PC1 to PC10)
- Can use any distance metrics!
- Can integrate between different data types (text, numbers, classes)
- It is no longer completely stochastic as t-SNE
- Defines both LOCAL and GLOBAL distances
- Can be applied to new data points

McInnes et al (2018) *BioRxiv*

Becht & McInnes et al (2019) *Nat Biot*

Wrap-up

Single cell workflows

	Seurat v3	Scater	Pagoda v2	Monocle v3
➡	PCA	PCA	PCA	PCA
	ICA	-	-	ICA
	-	MDS	-	-
➡	tSNE (BH, Flt)	tSNE (BH)	tSNE (BH)	tSNE (BH)
➡	UMAP	UMAP	-	UMAP
	-	-	LargeVis	-
	Diff. Maps	Diff. Maps	Isomap	-
	-	-	-	DDRTree
	PHATE	-	-	-
	-	-	-	SimplePPT

Paper comparing lots of dimensionality reduction techniques:
https://www.biorxiv.org/content/biorxiv/early/2018/06/28/120378.full.pdf

Thank you!

Paulo Czarnewski, **ELIXIR-Sweden (NBIS)**

European Life Sciences Infrastructure for Biological Information
www.elixir-europe.org

Exercise time !!!

Exercises: free exploration

Follow the initial steps of Data Integration Lab

```
pancreas.data <- readRDS(file="session-integration_files/pancreas_expression_matrix.rds")

metadata <- readRDS(file="session-integration_files/pancreas_metadata.rds")

pancreas <- CreateSeuratObject(pancreas.data, meta.data = metadata)

pancreas <- NormalizeData(pancreas, verbose = FALSE)

pancreas <- FindVariableFeatures(pancreas, selection.method = "vst", nfeatures = 2000, verbose = FALSE)

pancreas <- ScaleData(pancreas, verbose = FALSE)

pancreas <- RunPCA(pancreas, npcs = 30, verbose = FALSE)

pancreas <- RunTSNE(pancreas, reduction = "pca", dims = 1:30)

pancreas <- RunUMAP(pancreas, reduction = "pca", dims = 1:30)
```

Exercises: free exploration

PCA

```
obj <- RunPCA( obj )

obj @ reductions $ pca @ cell.embeddings
                       @ feature.loadings
                       @ stdev
```

```
PCAPlot(obj)
ElbowPlot(obj)

JackStraw(obj)
JackStrawPlot(obj)
PCASigGenes(obj)
```

tSNE

```
obj <- RunTSNE( obj )
obj @ reductions $ tsne @ cell.embeddings

reduction="PCA"   dims=1:20      perplexity=30
max_iter=1000    num_threads=0   theta=0.2
```

```
TSNEPlot(obj)
```

UMAP

```
obj <- RunUMAP( obj )
obj @ reductions $ umap @ cell.embeddings

reduction="PCA"   dims=1:20      n.neighbors=50
n.epochs=200     min.dist=0.0001
```

```
UMAPPlot(obj)
```

Exercises: free exploration

Follow the initial steps of Data Integration Lab

1. PCA: Which genes separate PC1? Which genes separate PC2?
2. PCA: Using the Elbow method, how many PCs contain over 1% standard deviation?

3. tSNE: What happens if you use only 5 PCs as input for your tSNE?
4. tSNE: What happens if you increase perplexity to 50 ? And with 5?
5. tSNE: What happens if you decrease theta? Try between 0.1 - 0.2.
6. tSNE: What happens if decrease the number of iteration to 400?

7. UMAP: What happens if you set 100 neighbors?
8. UMAP: can reduce your data to 5 dimensions (instead of only 2)?
9. UMAP: what happens if you reduce the minimum distance to 0.001?