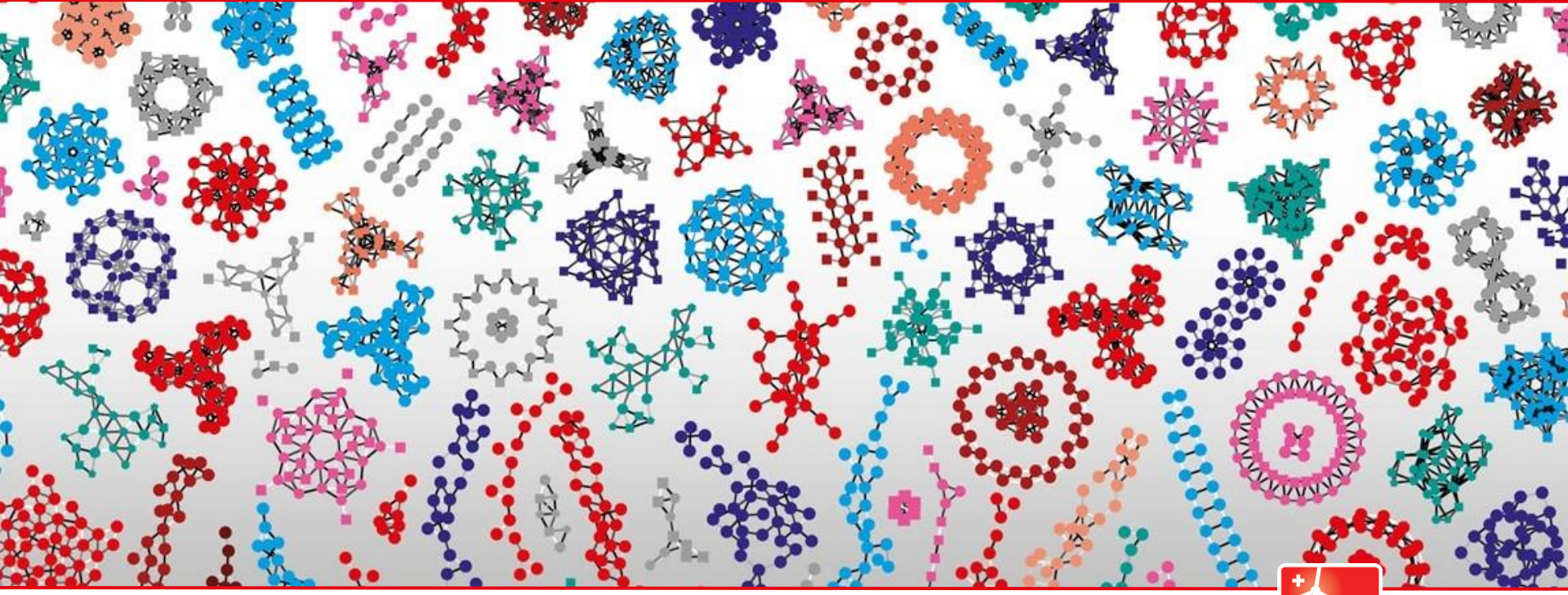
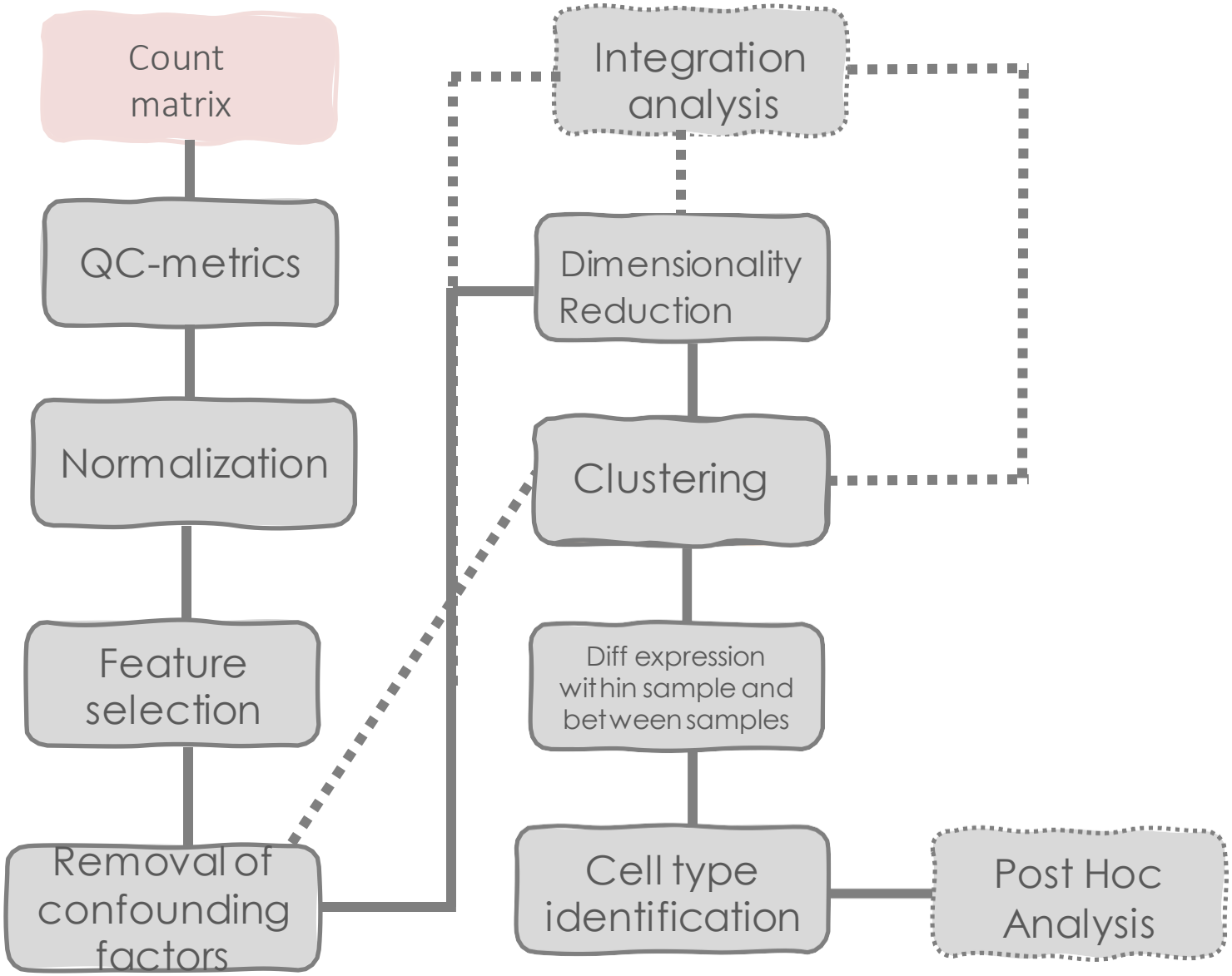


Swiss Institute of Bioinformatics

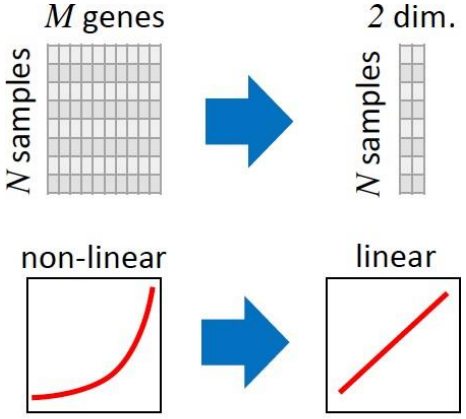
Day 2 : Single cell RNA sequencing: The bioinformatic downstream analysis

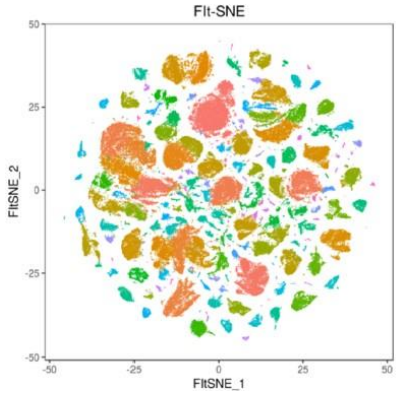
Geert van Geest, Rachel Marcone, Tania Wyss


Count matrix
QC-metrics
Normalization
Feature selection
Removal of confounding factors
Integration analysis
Dimensionality Reduction
Clustering
Diff expression within sample and between samples
Cell type identification
Post Hoc Analysis

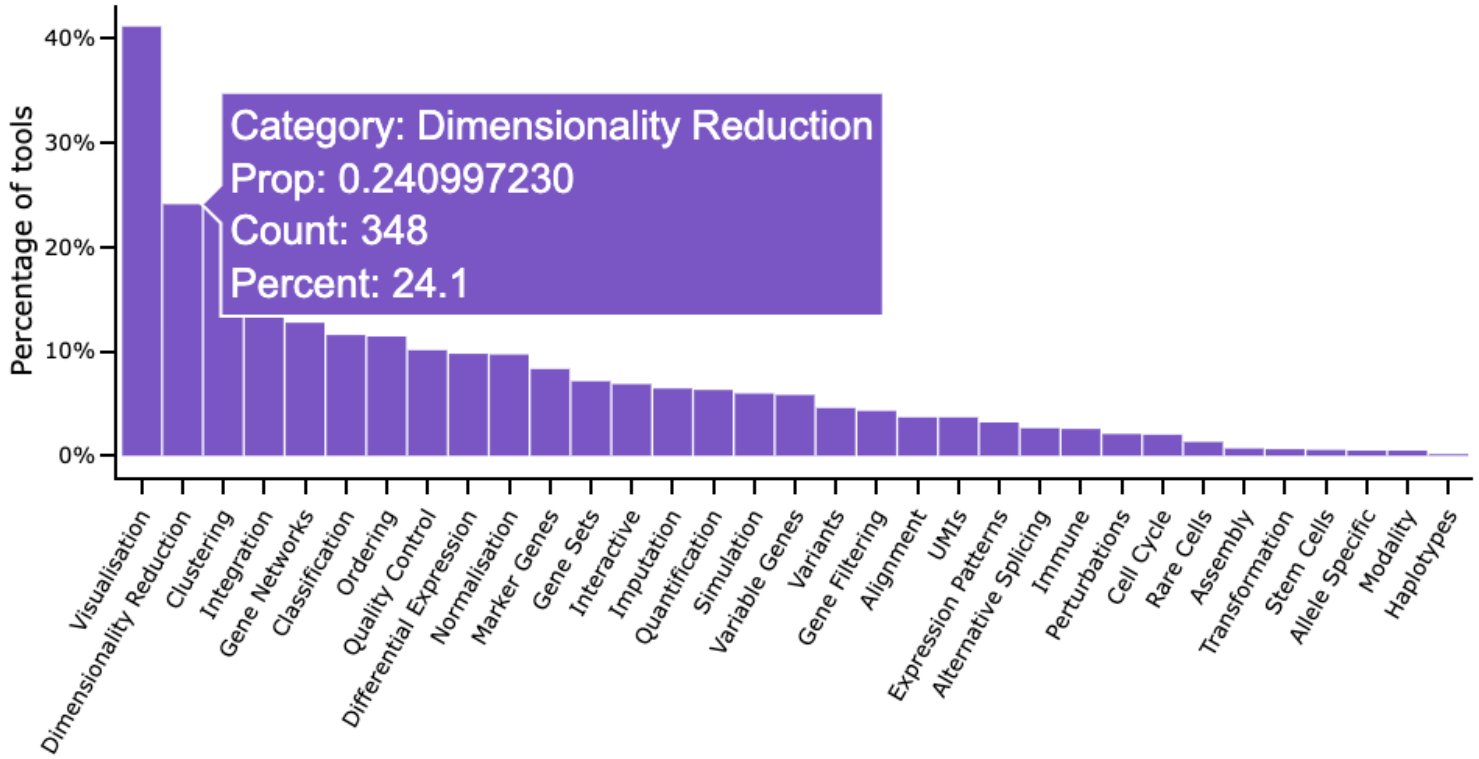
Dimensionality Reduction

- **Simplify complexity**, so it becomes easier to work with.
  - Reduce number of features (genes)
- "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**





Dimentionality reduction: Algorithms



scrna-tools.org

In Seurat

- PCA- Principal Component Analysis
- TSNE- T-distributed stochastic neighborhood embedding
- UMAP- Uniform manifold approach and projection

```
obj <-RunPCA( obj )
obj <-RunTSNE( obj )
obj <-RunUMAP( obj )
```

Vevox

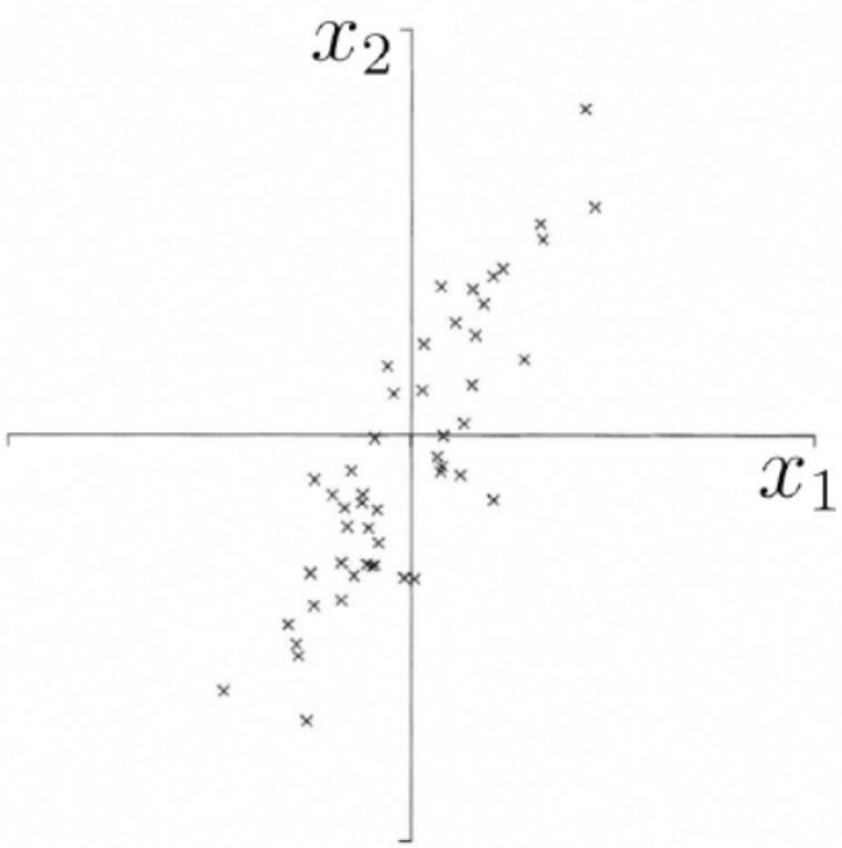
PCA- Principal component analysis

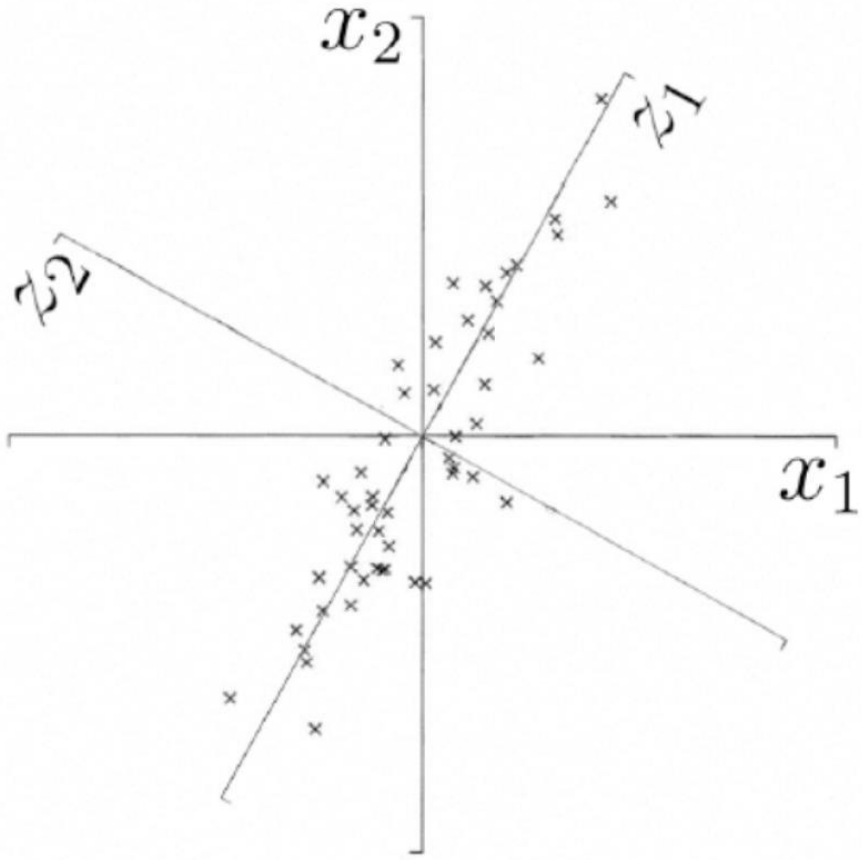
-PCA is based on variance
-PCA is the best angle to see and evaluate the data
-New axis that are linear combination of the original axes

PCA- Principal component analysis

Which and how ?

PCA- Principal component analysis

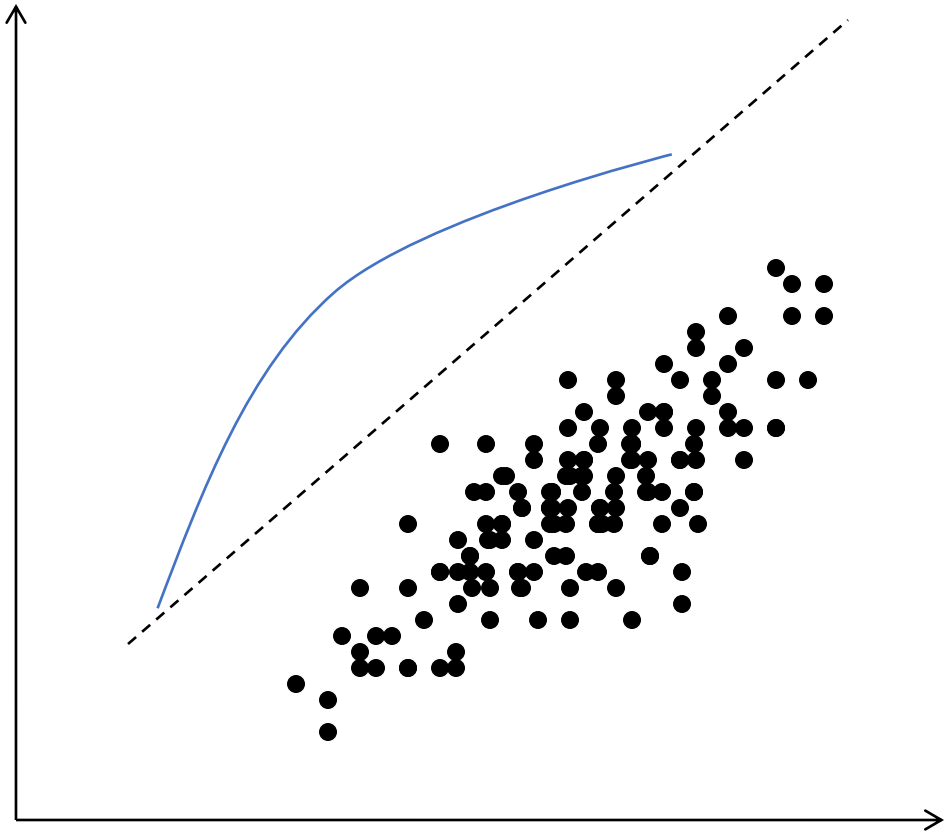




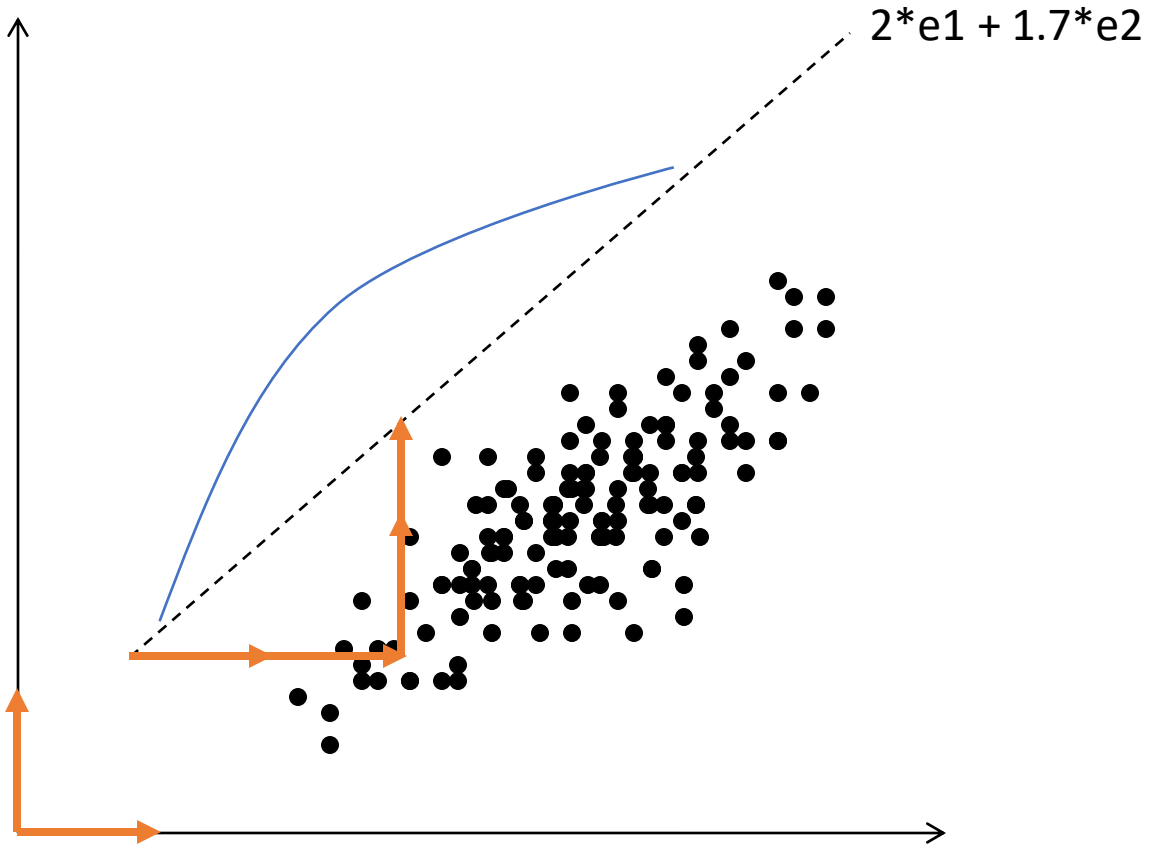
PCA- Principal component analysis

1. Largest variance first

PCA- Principal component analysis



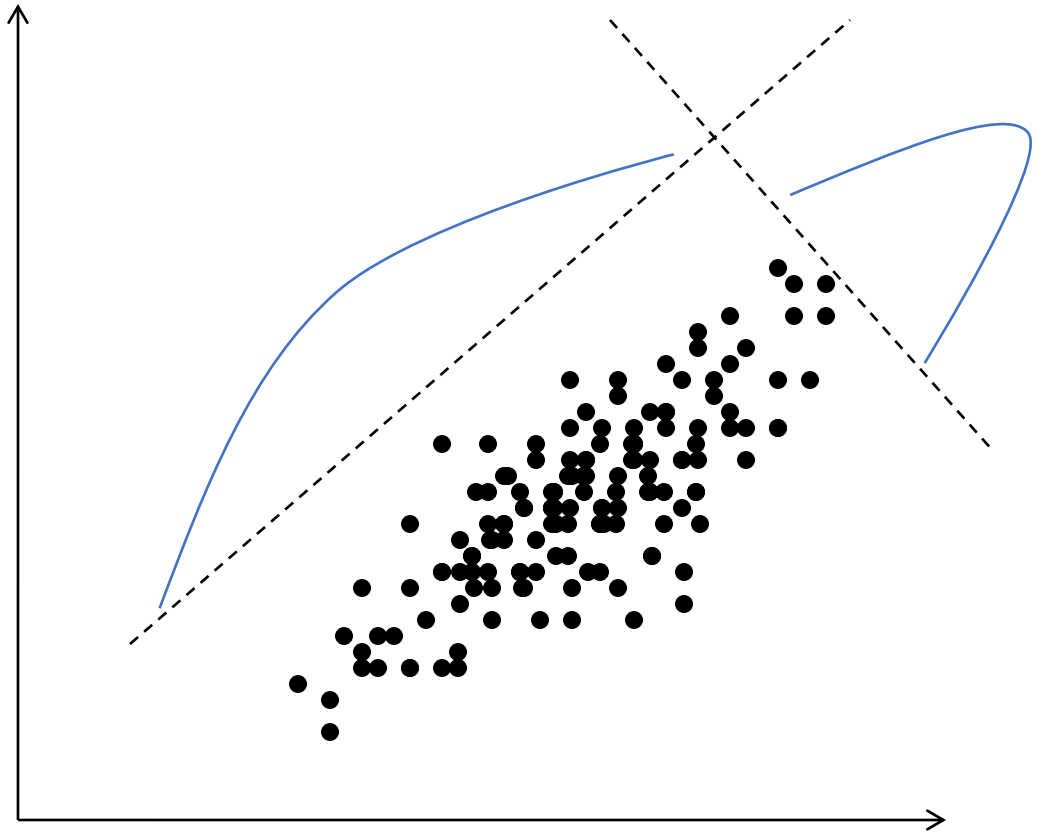
PCA- Principal component analysis



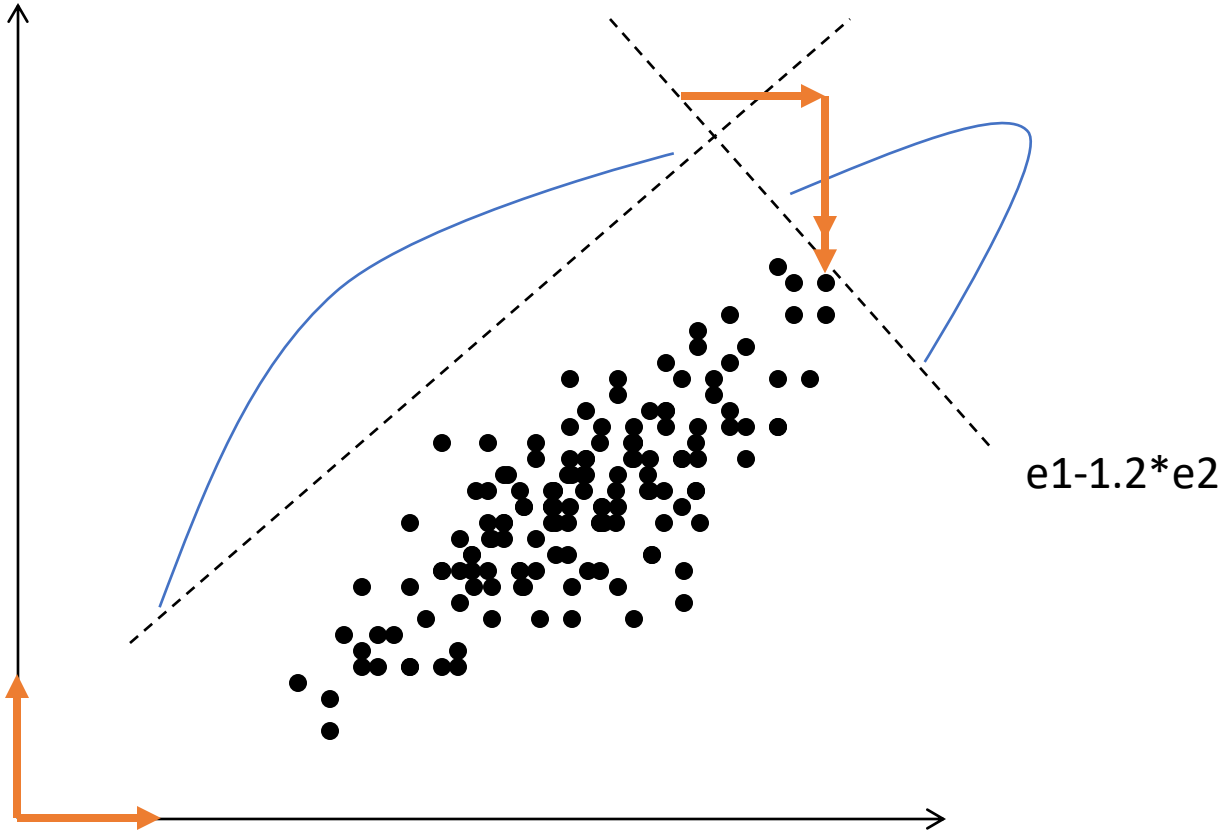
PCA- Principal component analysis

2. Select uncorrelated principal axis (orthogonal)

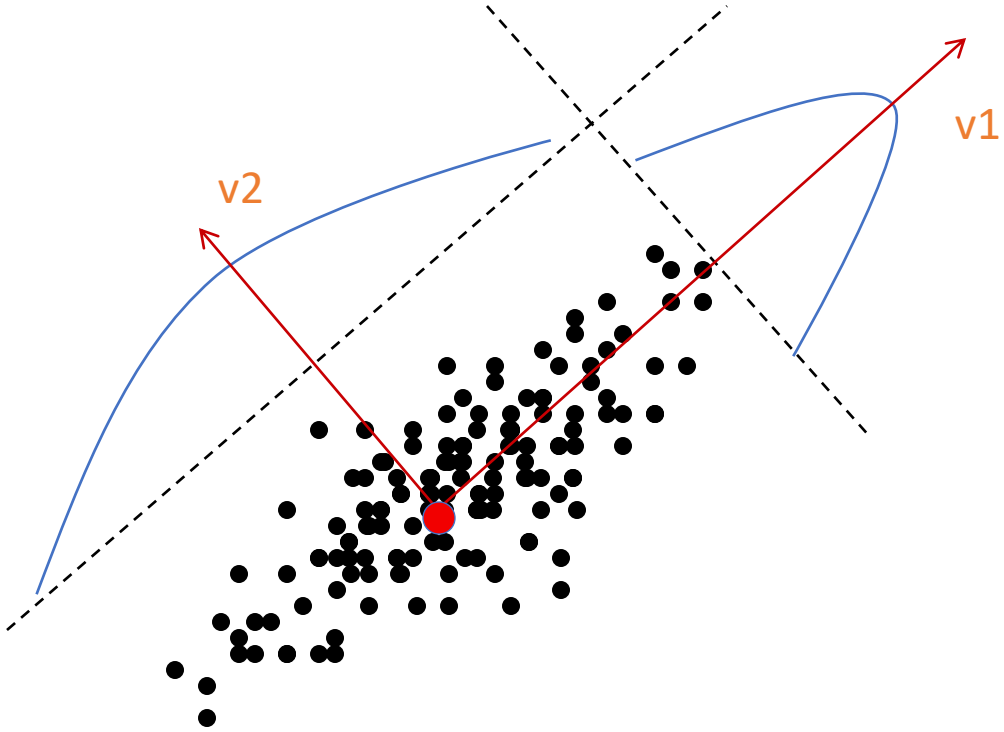
PCA- Principal component analysis



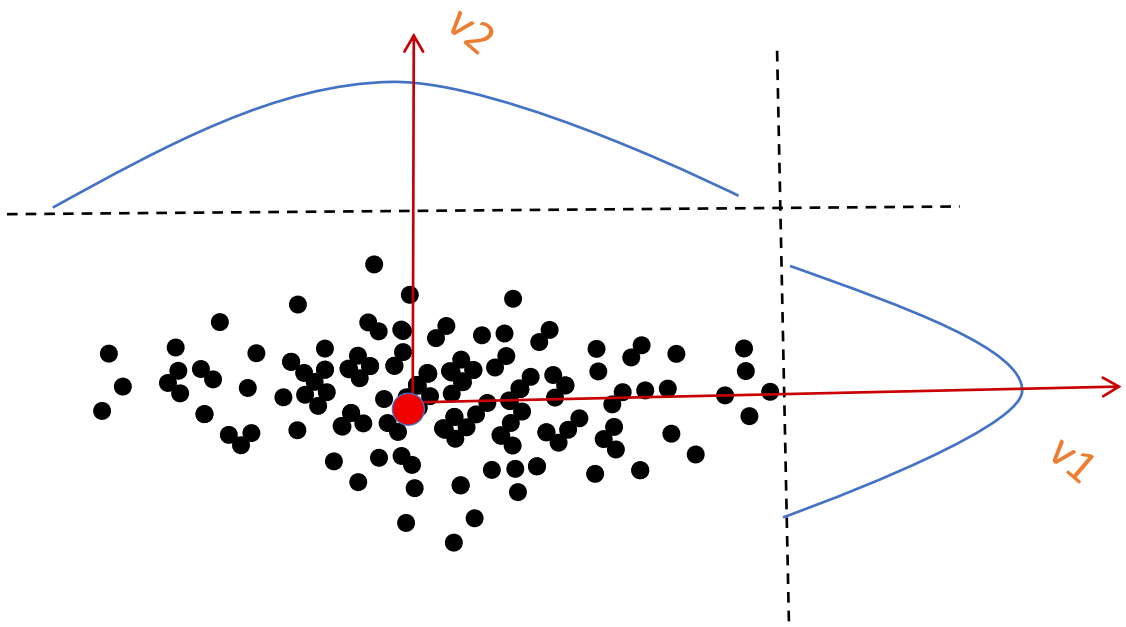
PCA- Principal component analysis



PCA- Principal Component Analysis



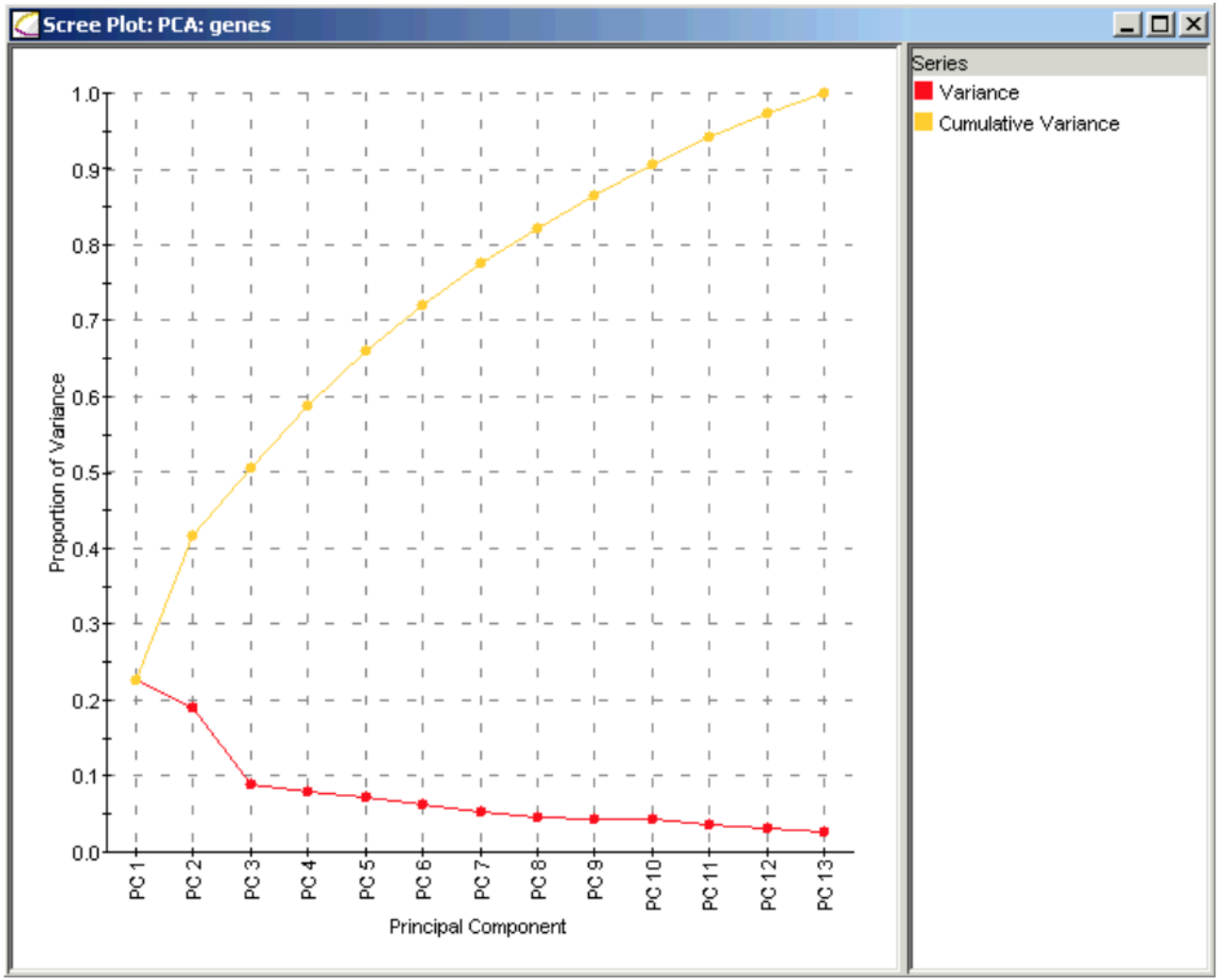
PCA- Principal component analysis



Mathematically

- Calculate the eigenvectors of the **Covariance matrix** are *the directions of the axes where there is the most variance (this is something you can prove mathematically!)*

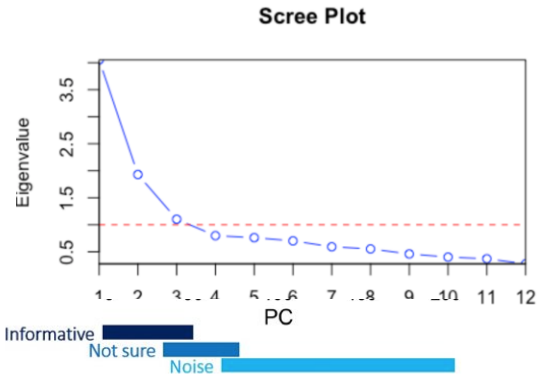
- eigenvalues are the coefficients attached to eigenvectors, which give the *amount of variance carried in each Principal Component*.



Scree Plot for Genetic Data. (Source.)

https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c

In R, Elbow plot



- RunPCA – Computes the PCA with default : 50 pcs.
- Check Elbow plot to see how many pcs are explaining well your data.
- RunPCA will output a message with the genes contributing most to the PC (positif and negatif).
- Uses irlba: Fast Truncated Singular Value Decomposition and Principal Components Analysis for Large Dense and Sparse Matrices (!!Approximation of PCA).
- Usually first PCs only account for few percentages of the total variance.

```
obj <- ScaleData(obj)
obj <-RunPCA(obj)
ElbowPlot(obj,ndims=50)
```

Wikipedia:
https://en.wikipedia.org/wiki/Scree_plot

The PCA axis

- The PC are linear combination of the original axis.
- The estimated parameters of the linear combination is known and therefore we can know positively or negatively how much it goes into one direction or the other one.
- Indeed as the original axis are g1,g2,g3… and the new axis are a1g1 +a2g2…, one takes the ai that are the highest, positively and negatively and therefore knows which genes are mostly representing the axis you see.
- By default, 10 highest positive and negative values are displayed in R with the Seurat package.

- Observation : **Scaling** is important, if one variable is on a different scale than another, it will dominate the PCA procedure as the largest variance might be observed there, and the low dimension plot will really just be visualizing that dimension.

Dimentionality reduction: PCA doesn't fit

- 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
  - Scater library describes correlation between PCs and metadata, take PCs until metadata information is covered

Problems:

- The two first PC in SC-RNAseq often account for only few percent of the total variance
- It performs poorly to separate cells in 0-inflated data types (because of it non-linearity nature)

Vevox

T-SNE

T-SNE = t-distributed stochastic neighborhood embedding

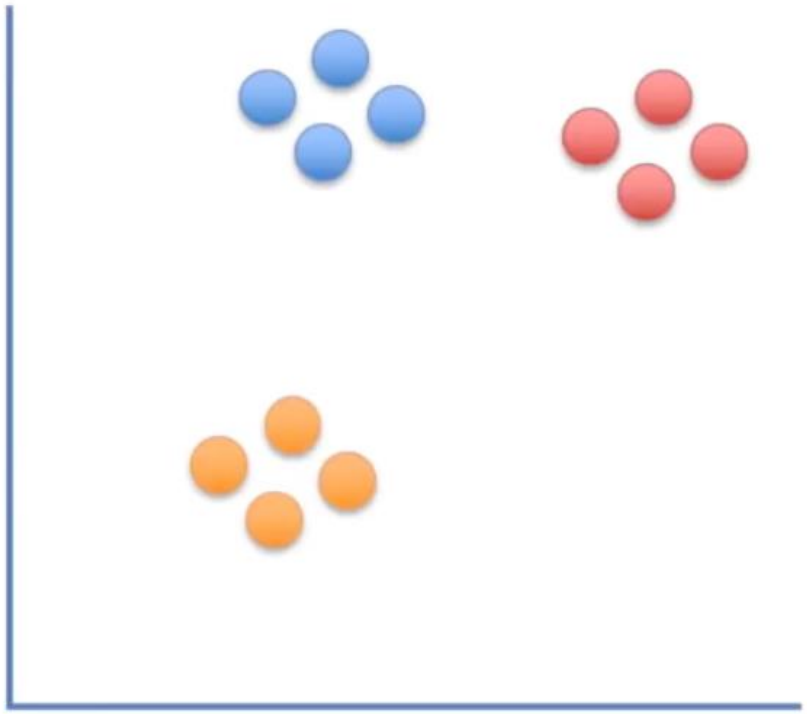
Laurens van der Maaten, Geoffrey Everest Hinton

http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

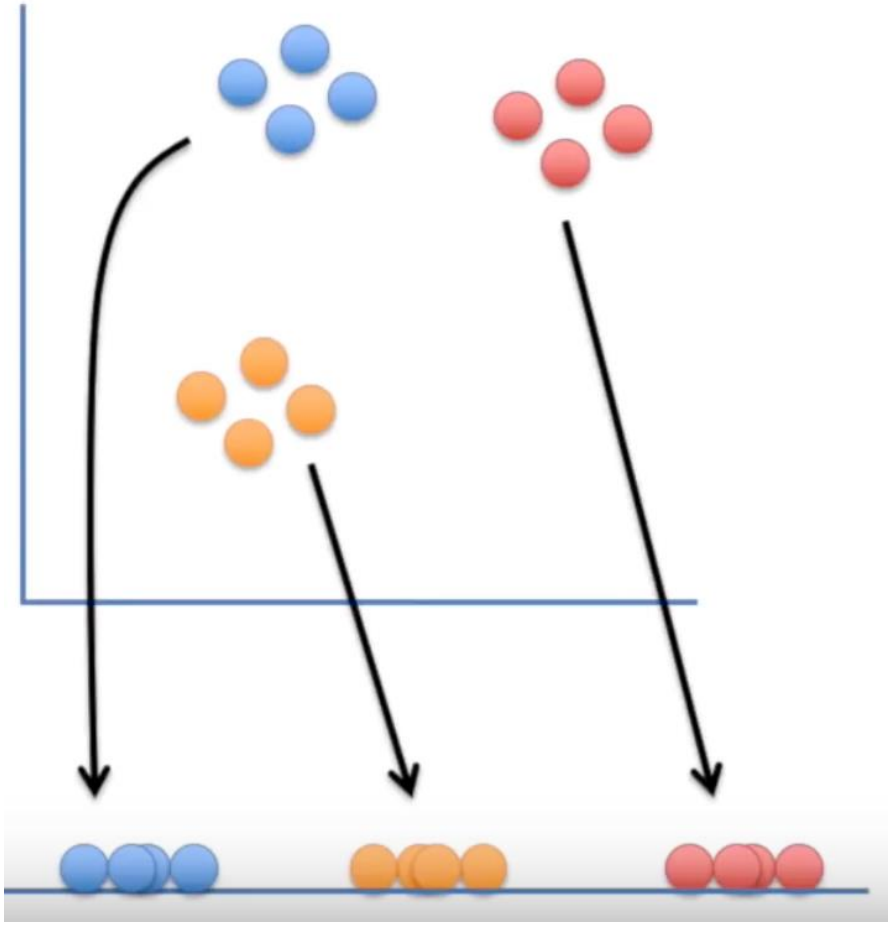
https://www.youtube.com/watch?v=NEaUSP4YerM

Many of the following figures are inspired by this youtube link check out his channel ! (StatQuestion with Josh Starmer)

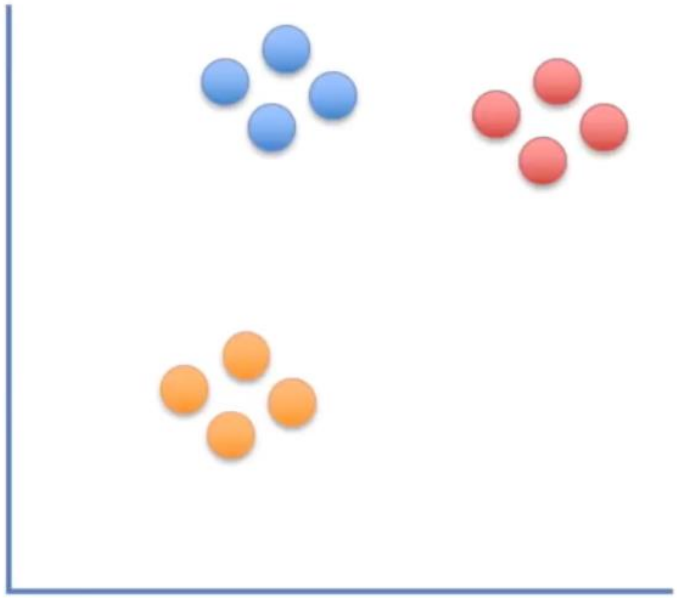
Start with a data set



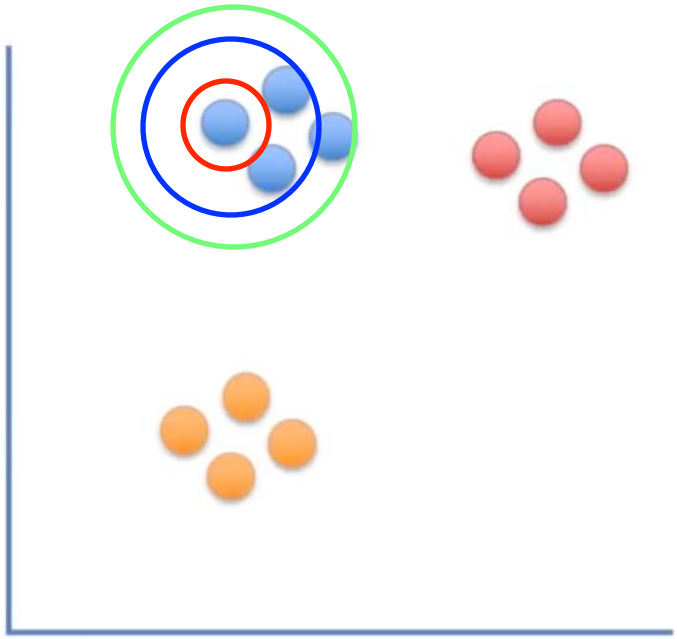
Find a right way to reduce dimension while keeping all the "clusters"



Basic idea (!! set a seed)



Normal distribution around a point B with mean B and variance sigmaB



We calculate

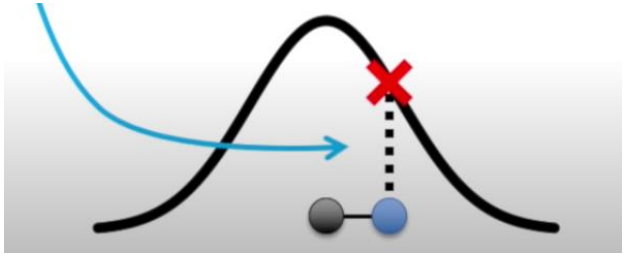
The similarity of datapoint A to datapoint B is the conditional probability, that A would pick B as its neighbor, if neighbors were picked in proportion to their probability density under a Gaussian centered at B with variance σB, written p_A|B.

p_A|A = 0

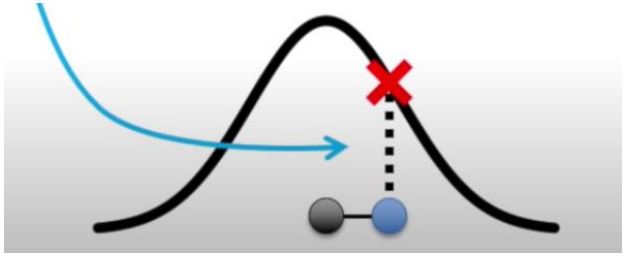
The variance σB of this normal distribution depends on the density around B (the more cells closer to B the lower the variance of this normal distribution will be).

Steps

1. Take a point A.
2. Take another point B
3. Plot that point on a normal distribution distributed around A.
4. Take point B and plot it on that distribution, this will be called the unscaled similarity.

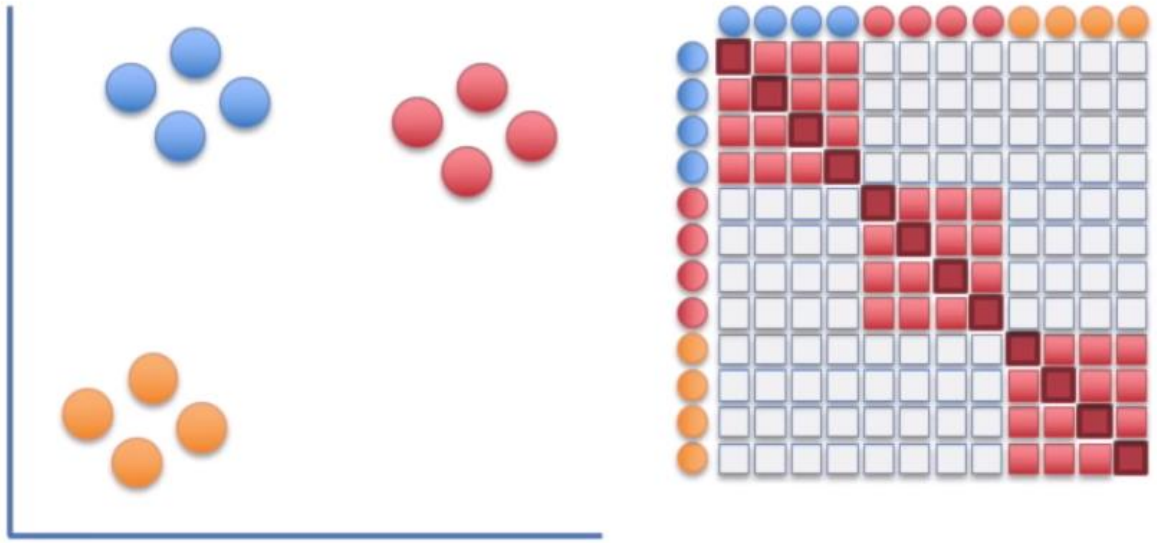


Steps



5. This is done for all the points. Distant points will have a very low similarity, whereas close points a very high similarity.
6.These unscaled similarities are then scaled so that they add up to one.
7. The similarity between A and B might be different than the similarity between B and A, so to correct for that the mean of the two values is taken.

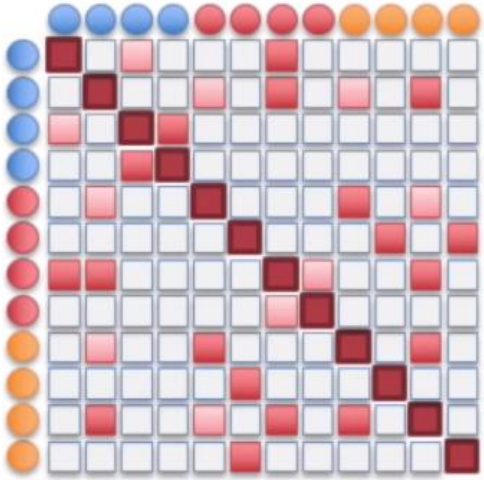
Illustration

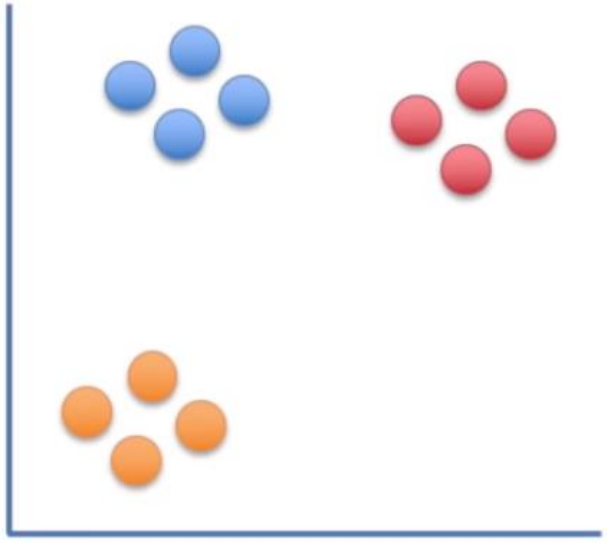


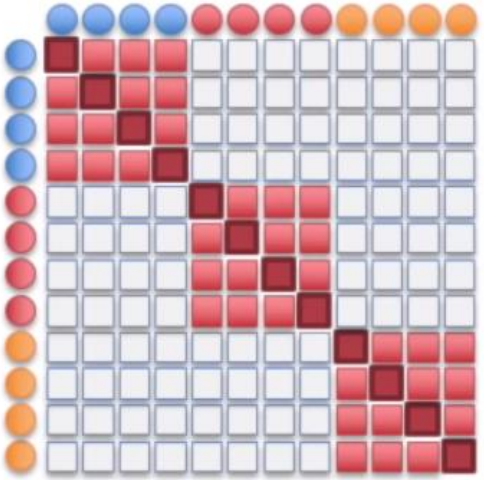
On the projection

Do the same into the randomly projected points.

Using a t-distribution instead of a normal distribution.

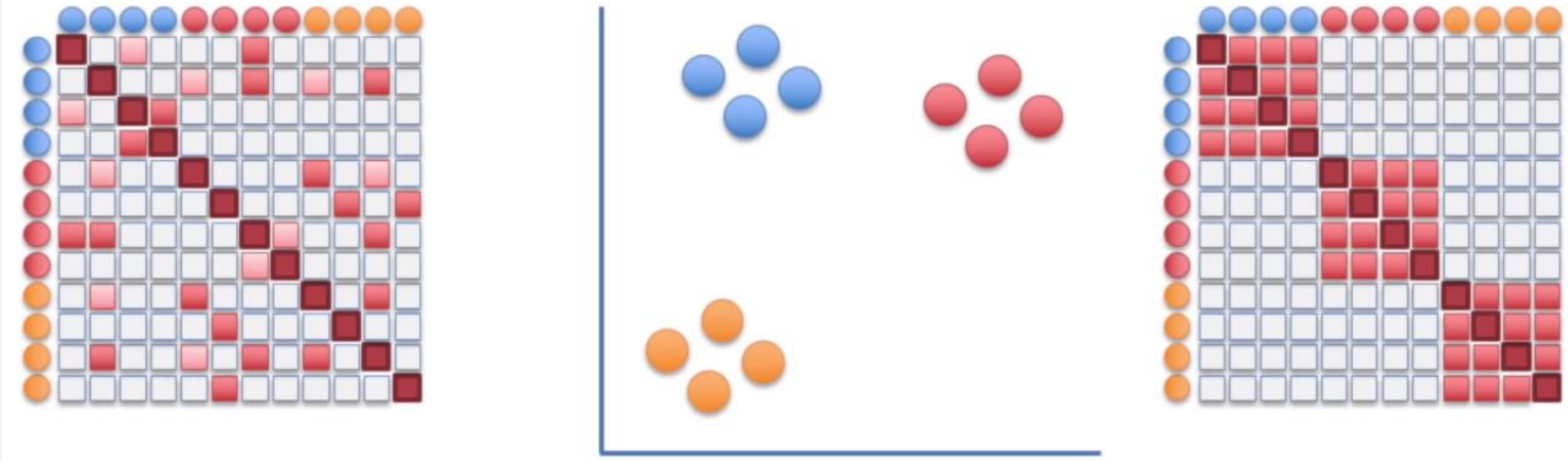






On the projection

Move points little by little and redo calculation until you are « as close as possible » to the original similarity matrix or you reach a certain number of iteration (chosen by the user).



As close as possible

To measure the minimization of the sum of difference of conditional probability t-SNE minimizes the sum of Kullback-Leibler divergence of overall data points using a *gradient descent method*.

In other words : tSNE minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding *low*-dimensional points in the embedding

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}},$$

Parameters for TSNE

perplexity = 30L => linked to parameter σ of all the points

momentum = 0.5, => linked to optimisation

final_momentum = 0.8, => linked to optimisation

A cool webpage

https://distill.pub/2016/misread-tsne/

(used to generate the figures in the next slides)

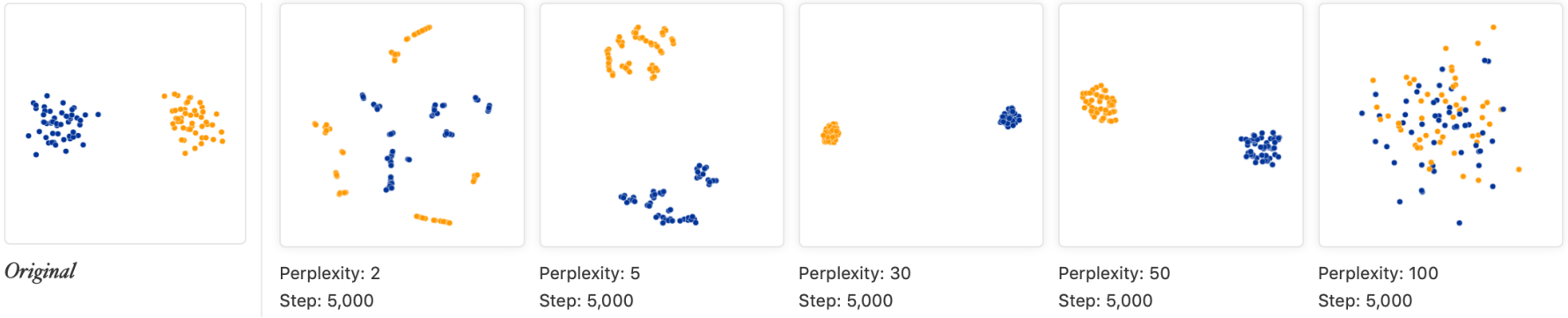
Perplexity-trial and error

The perplexity can be interpreted as a smooth measure of the effective number of neighbors

$$Perp(P_i) = 2^{H(P_i)},$$

where $H(P_i)$ is the Shannon entropy of P_i measured in bits

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}.$$



Original

Perplexity: 2
Step: 5,000

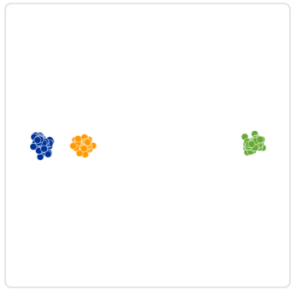
Perplexity: 5
Step: 5,000

Perplexity: 30
Step: 5,000

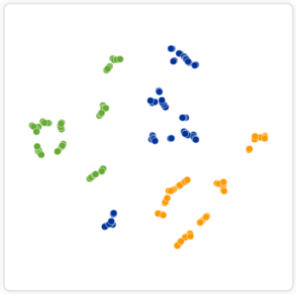
Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

Distances between cluster do not matter

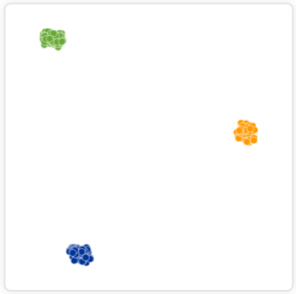


Original

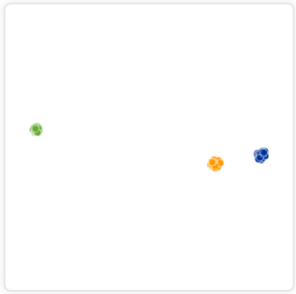


Perplexity: 2
Step: 5,000

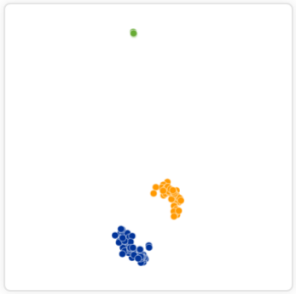
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000

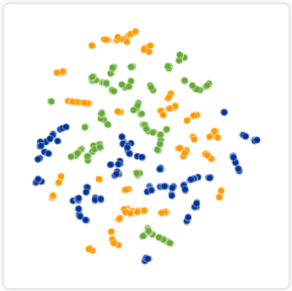


Perplexity: 50
Step: 5,000

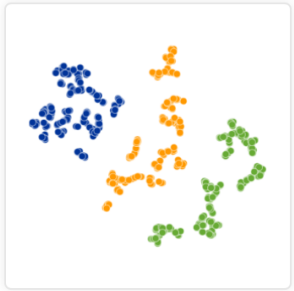


Perplexity: 100
Step: 5,000

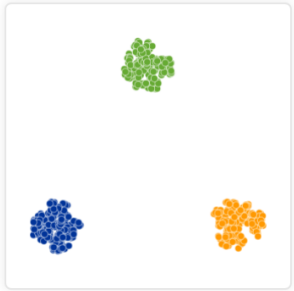
Original



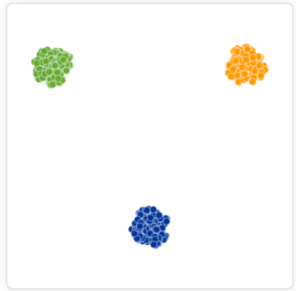
Perplexity: 2
Step: 5,000



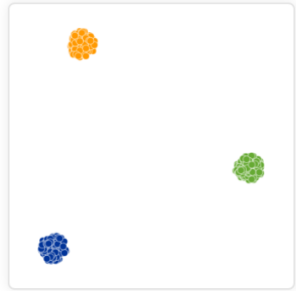
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

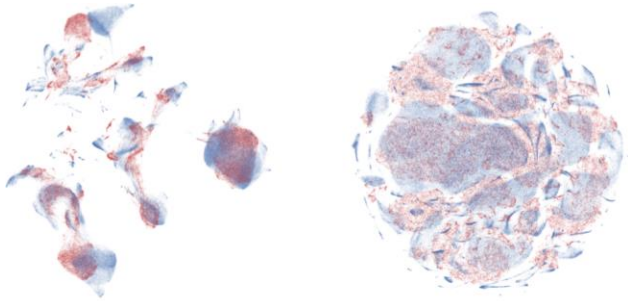


Perplexity: 100
Step: 5,000

Dimension reduction: UMAP

UMAP: **U**niform **M**anifold **A**pproximation and **P**rojection

- It is a NON-LINEAR graph-based method of dimensionality reduction
- UMAP assumes that there is a manifold in the dataset.
- **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**



(a) UMAP (b) t-SNE

UMAP

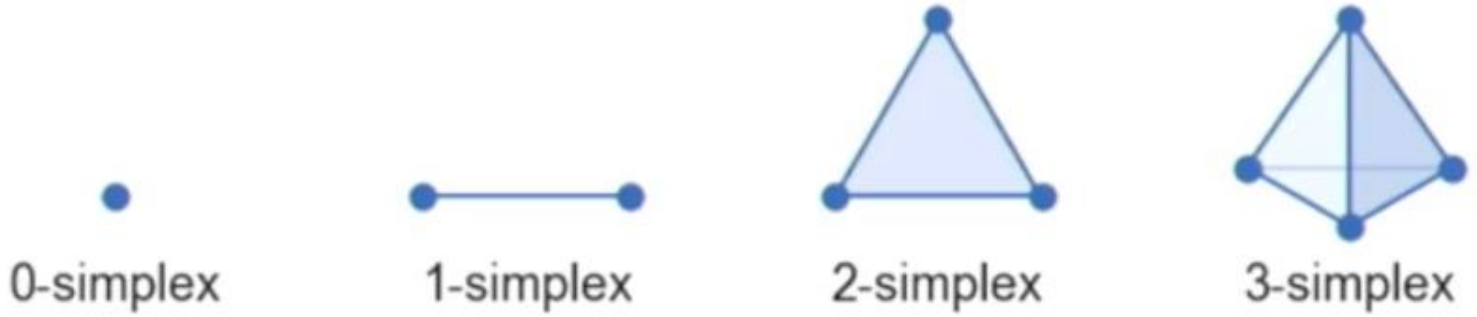
UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

Leland McInnes (Mathematician), John Healy (Computing theorist), James Melville (Computing in R)

https://arxiv.org/abs/1802.03426

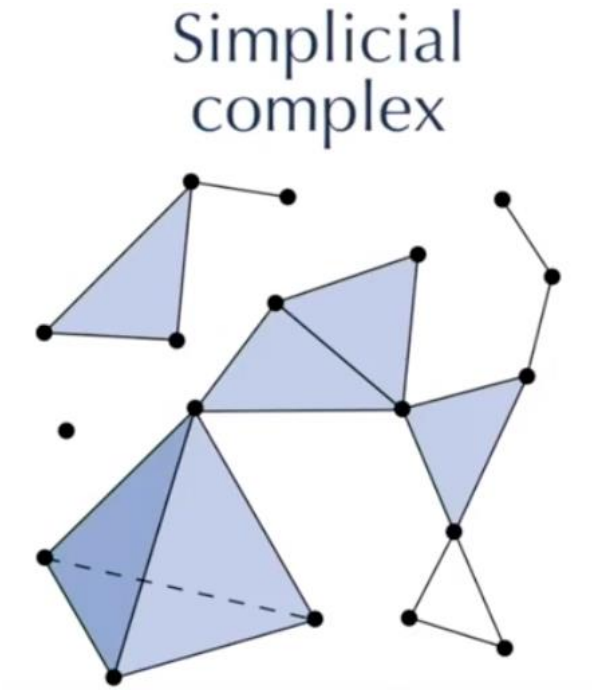
https://www.youtube.com/watch?v=nq6iPZVUxZU

https://umap.scikit-tda.org/parameters.html



From L.McInnes, SciPy 2018

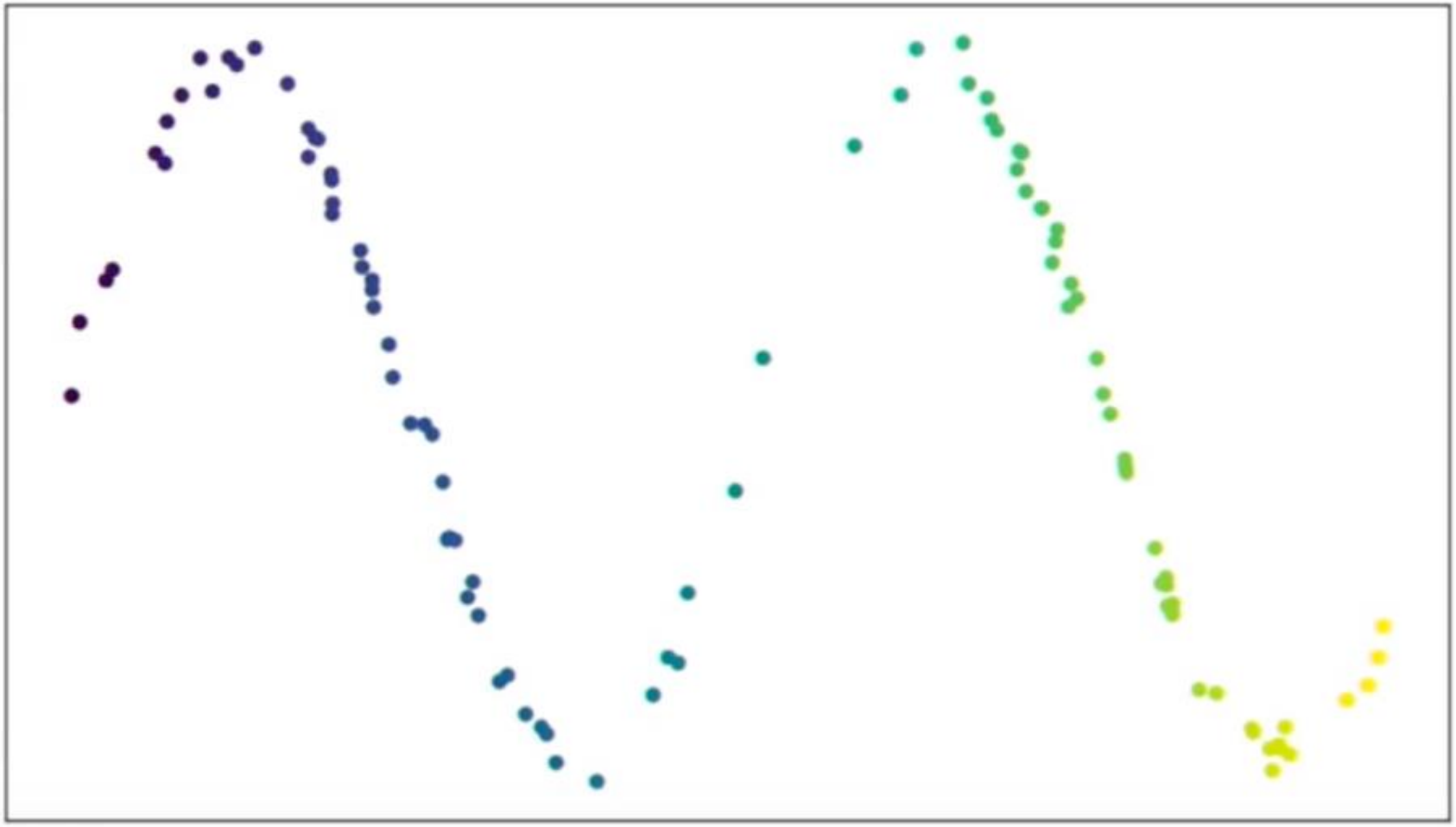
What it enables you to represent



1. Combinatorial
2. Simple to implement
3. Keeps the information of the global structure
4. Nice theorems exist on those (Nerve theorem)

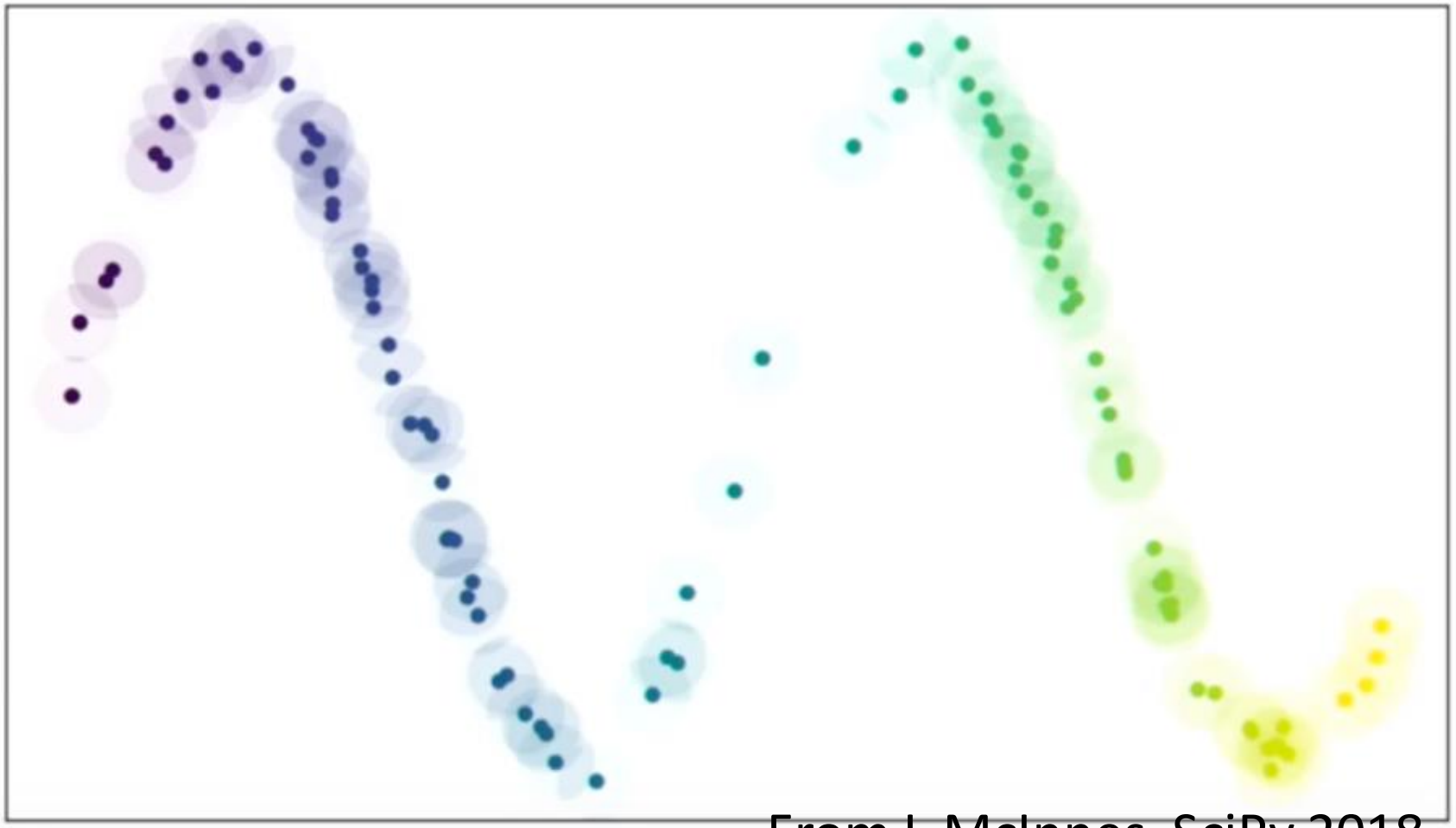
From L.McInnes, SciPy 2018

How do we build a simplicial complex on top of a data set?



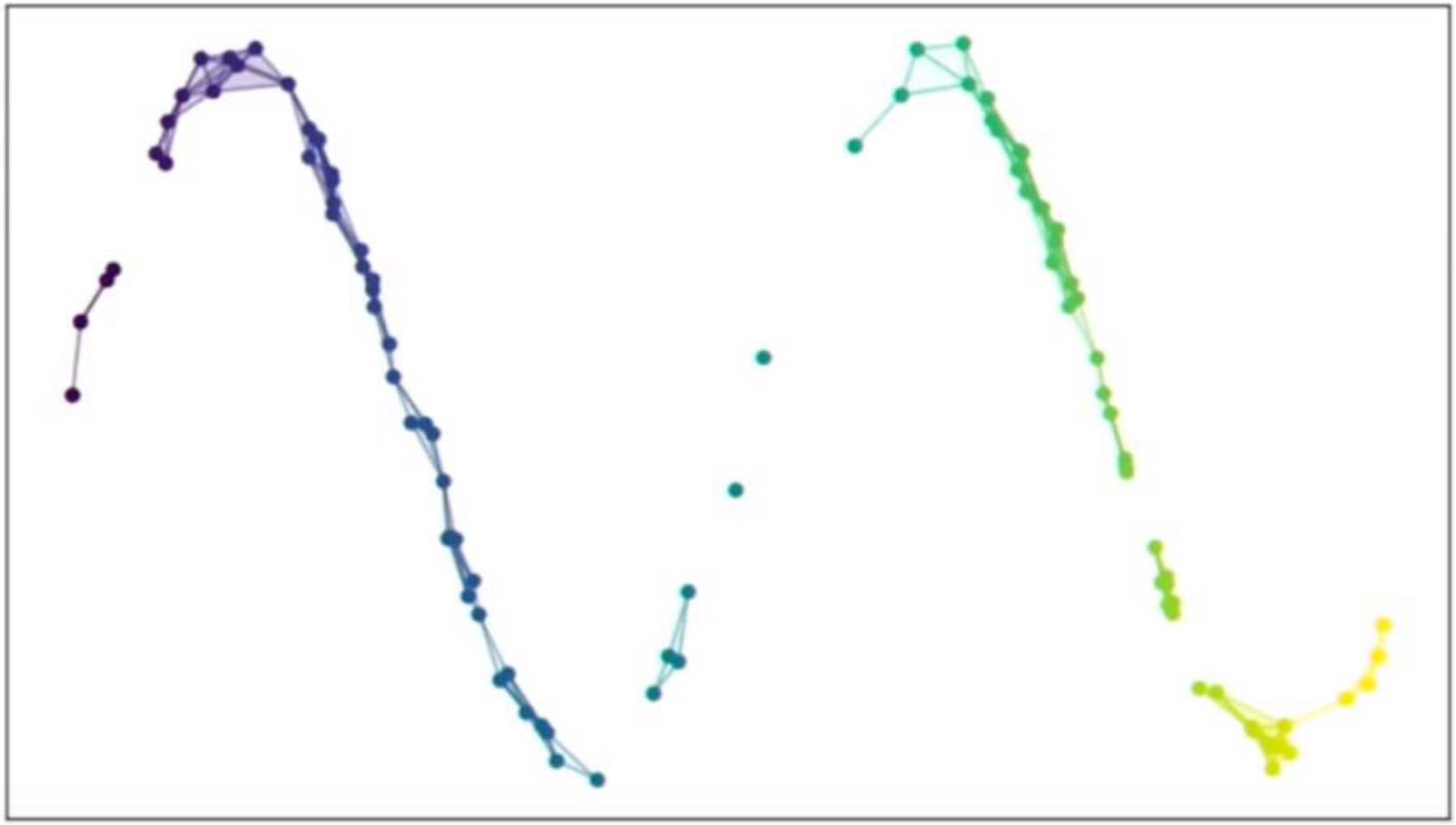
From L.McInnes, SciPy 2018

Step 1: draw unit-balls with a certain metric



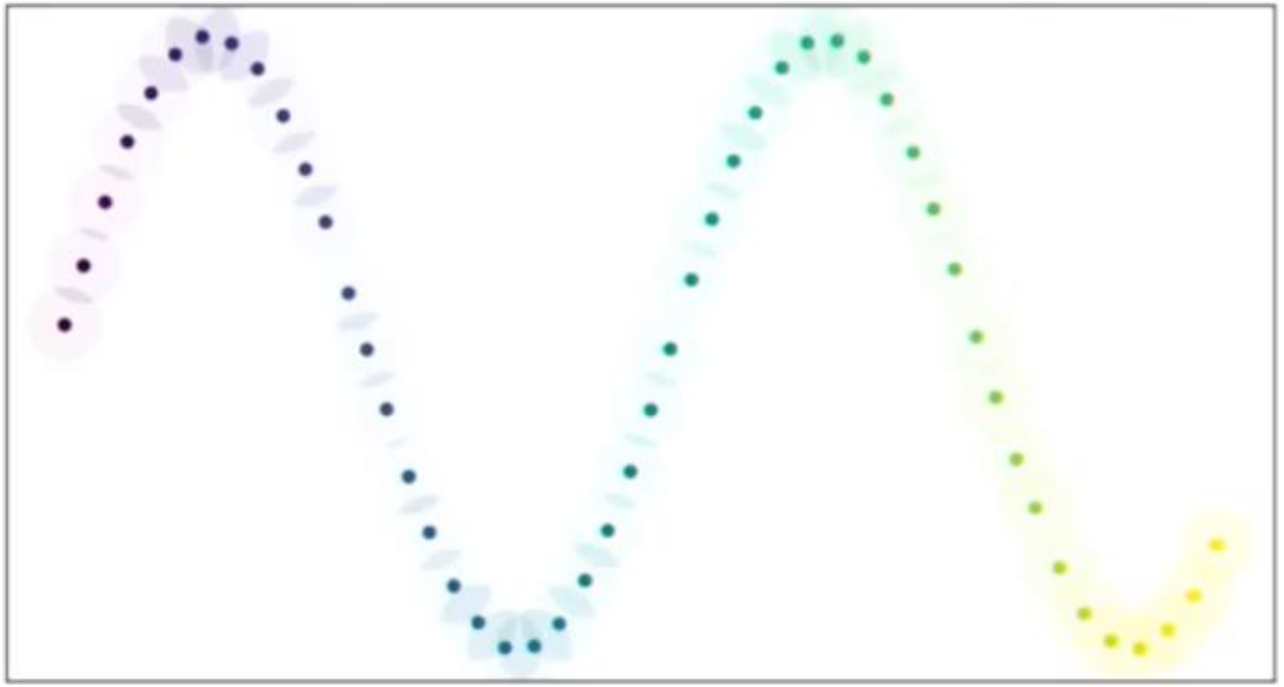
From L.McInnes, SciPy 2018

Step 2: Draw the Nerve of that cover



From L.McInnes, SciPy 2018

The data is not uniformly distributed on the underlying manifold

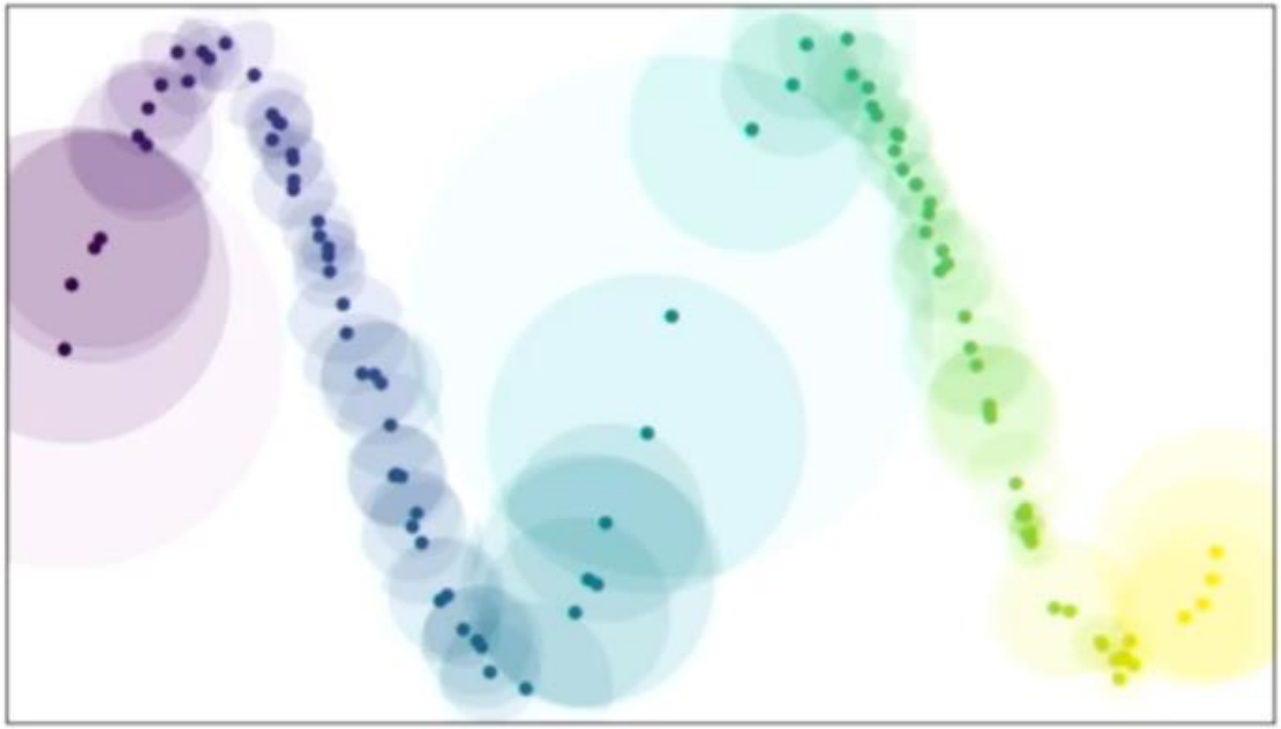


From L.McInnes, SciPy 2018

However… Data is not so nicely distributed

Solution: We vary the notion of metric and effectively the data will be with that metric uniformly distributed on the underlying manifold

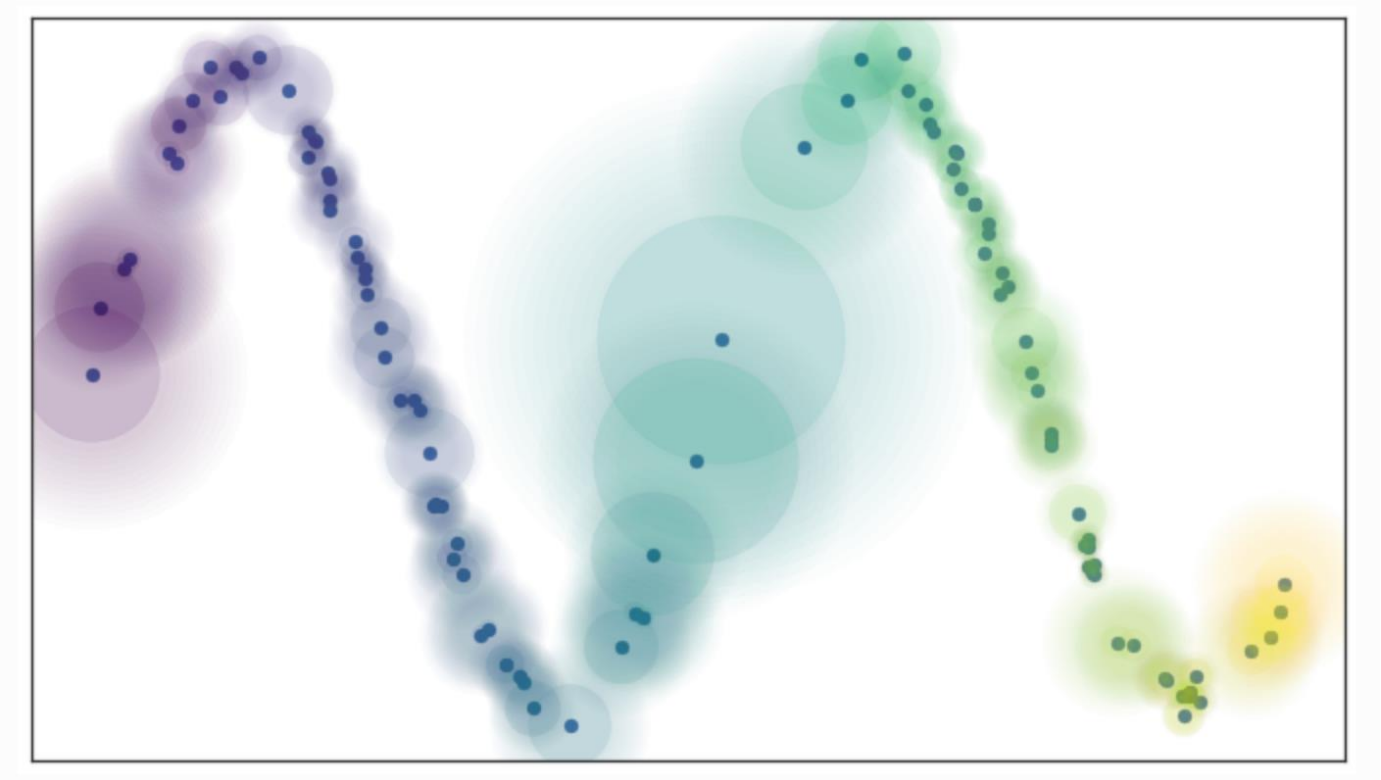
How it looks like on the example



The radius of each ball is equal to one.

From L.McInnes, SciPy 2018

How it looks like on the example

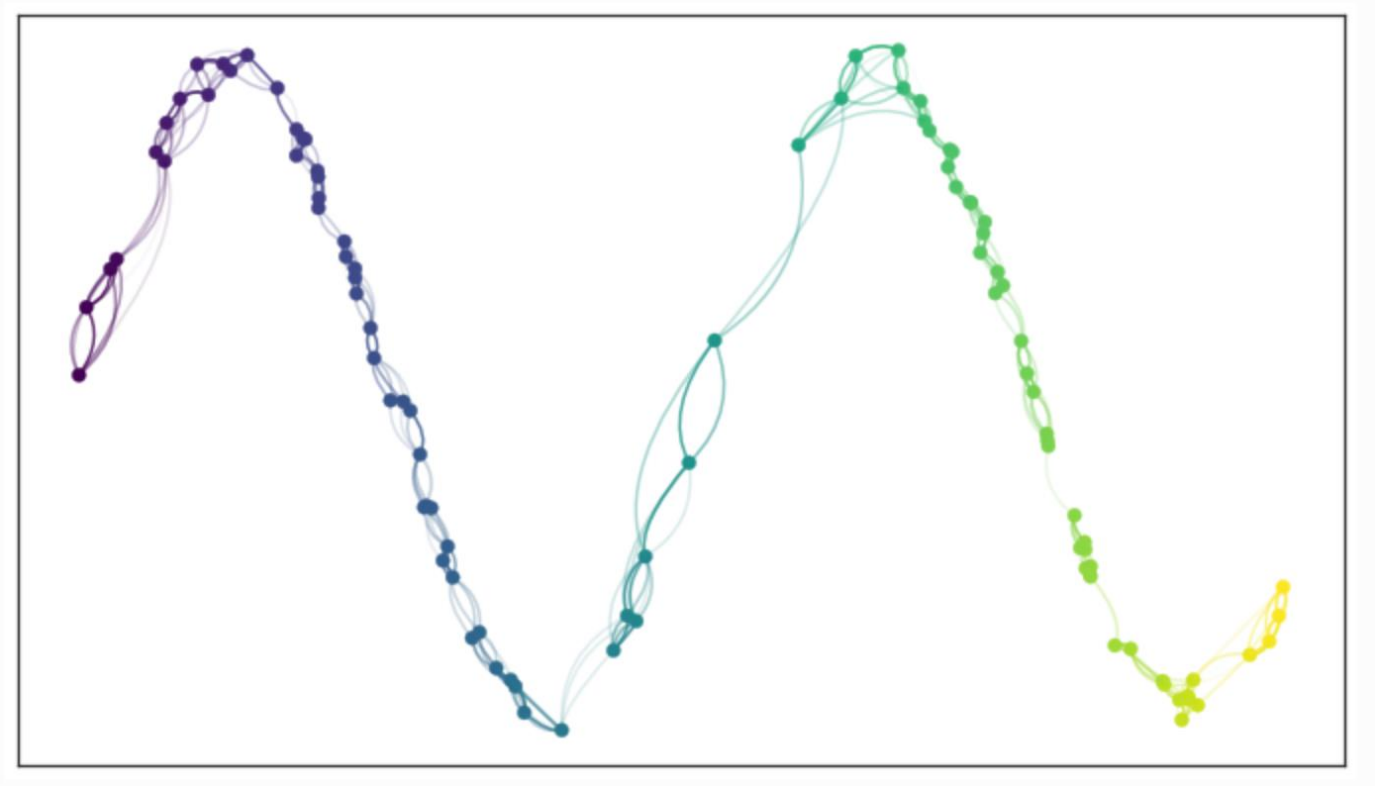


Equivalent to choosing a cover of balls with varying radia. This is what Fuzzy covers try to do.

There are nice theorems again justifying that all of this is valid.

From L.McInnes, SciPy 2018

New directed graph



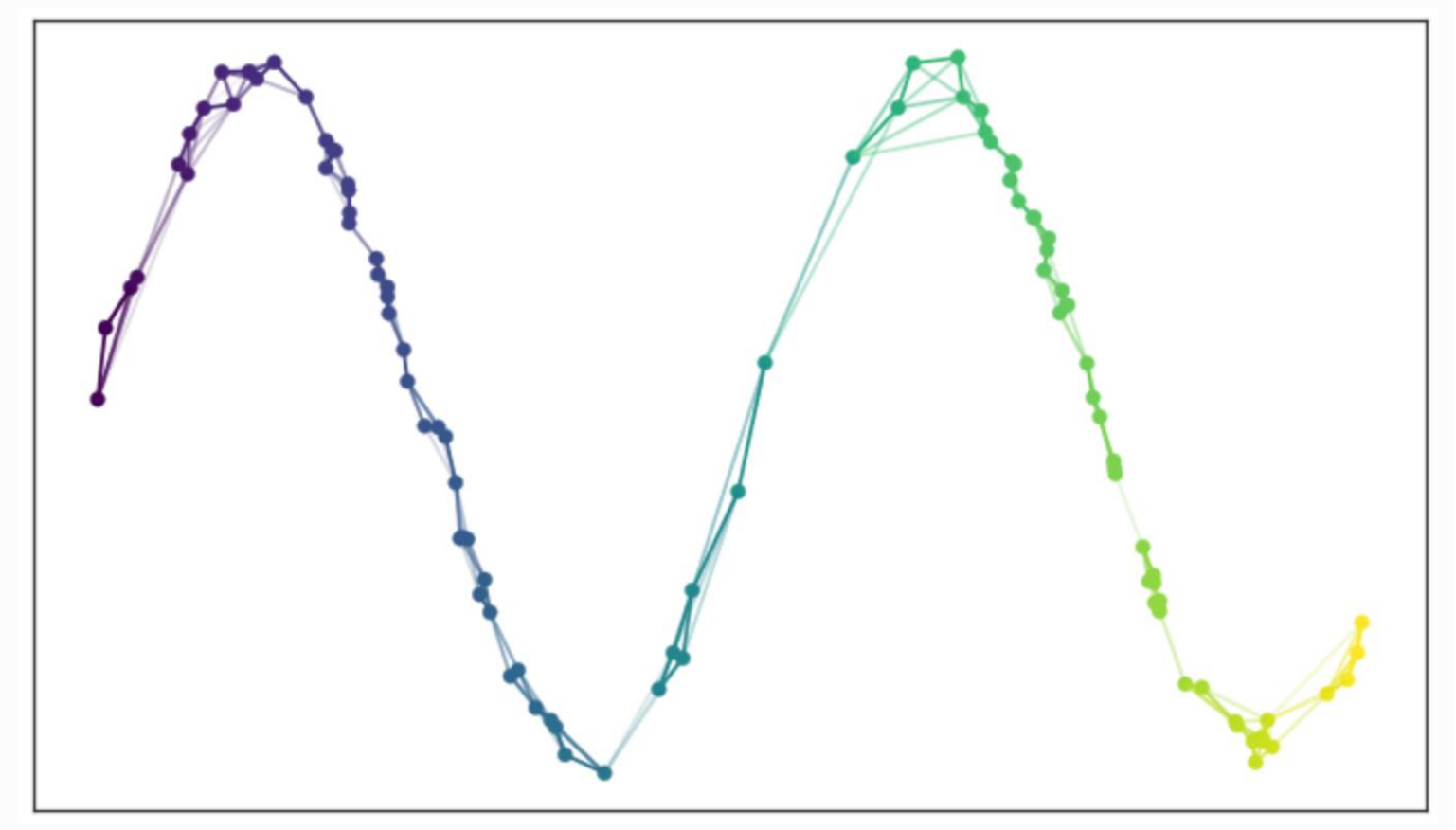
From L.McInnes, SciPy 2018

But we needed a (weighted) simplicial complex…

$$f(a,b) = a+b - a*b$$

Solving the problem…

New simplicial complex



From L.McInnes, SciPy 2018

2nd assumption

The second assumption : the manifold is locally connected.
They use that for mathematics to work but has as an implication that in practice you will not find isolated points in your dataset.

Dimension reduction

Now, UMAP is a dimension reduction method. Let us say you would like to project the data onto IR^2

It will therefore take $Y = \{y1,\ldots,yN\}$ in IR^2

Compute the fuzzy topological considering IR^2 to be the underlying manifold.

Optimizing this dimension reduction

Given fuzzy simplicial set representations : X and Y , a means of comparison is required.

For the purpose of calculations only the 1-skeleton of the fuzzy simplicial sets is considered (the l-skeletons are calculated using the 1-skeleton and can therefore be shown to be negligible)

To compare two fuzzy sets we will make use of fuzzy set ***cross entropy***.

Get the clumps right

$$\sum_{a \in A} \mu(a) \log\left(\frac{\mu(a)}{\nu(a)}\right) + (1-\mu(a)) \log\left(\frac{1-\mu(a)}{1-\nu(a)}\right)$$

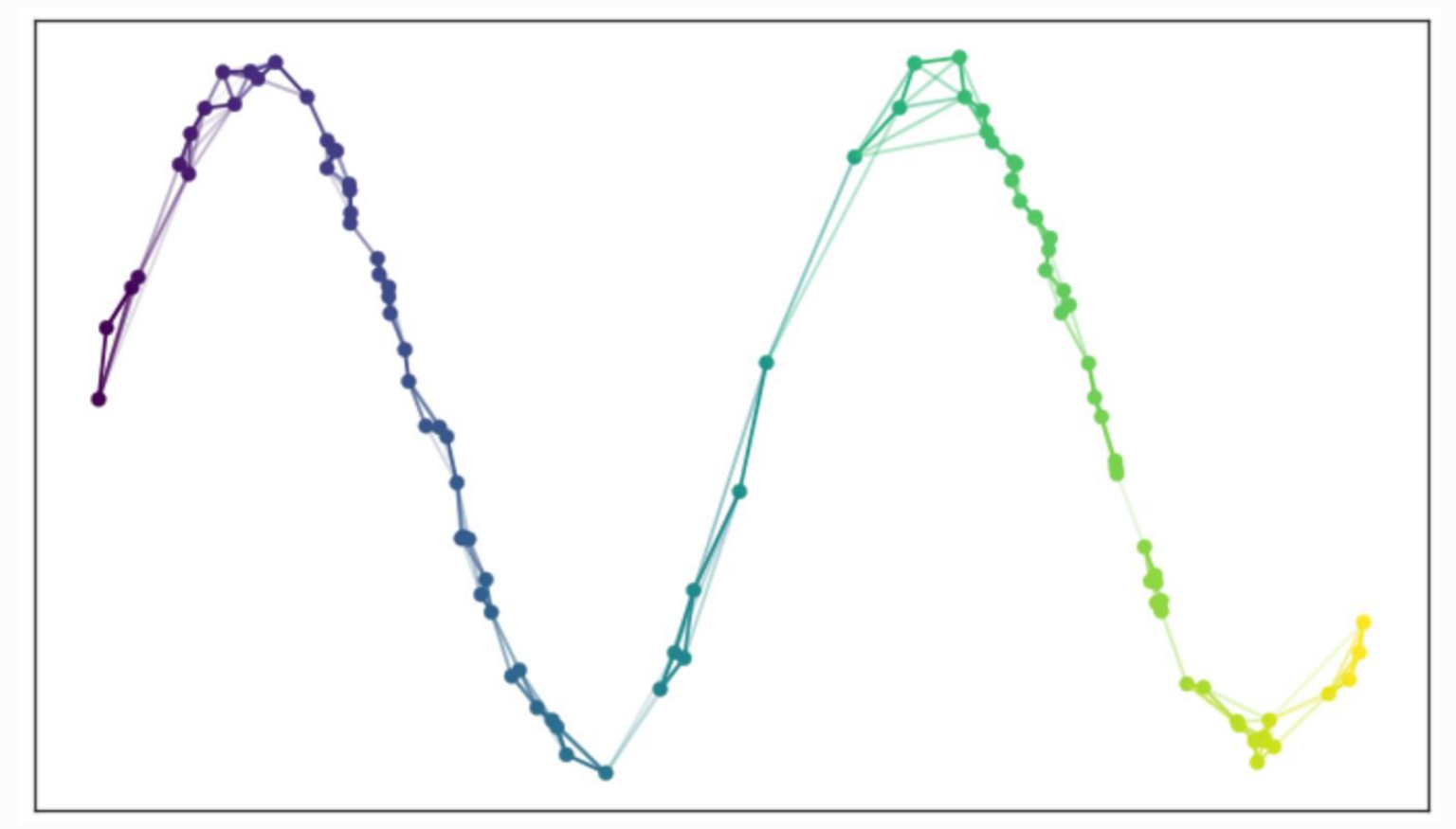
Get the gaps right

Summary

The first phase consists of constructing a fuzzy topological representation (edges and weights).

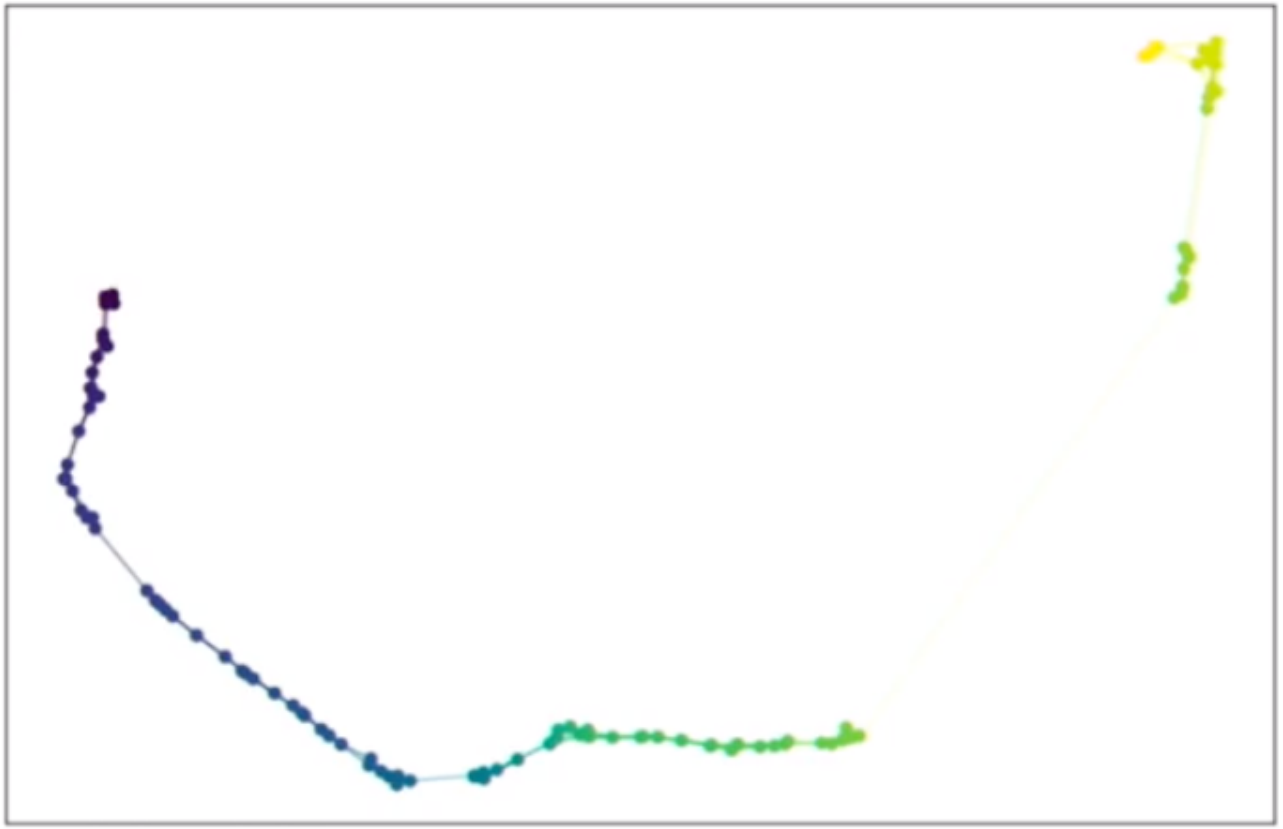
The second phase is optimizing the low dimensional representation to have as close as possible a fuzzy topological representation as measured by cross entropy.

New simplicial complex



From L.McInnes, SciPy 2018

How the UMAP embedding looks



From L.McInnes, SciPy 2018

Input parameters

X: the data

n: the neighborhood parameter: number of neighbors to consider when approximating the local metric

d: the target embedding dimension (2 usually)

min-dist: »beauty» parameter for the local embedding in 2D: the desired separation between close points in the embedding space: this determines how closely points can be packed together in the low dimensional representation

n-epochs: optimization parameter for the local embedding in 2D the number of training *epochs (batches)* to use when optimizing the low dimensional representation.

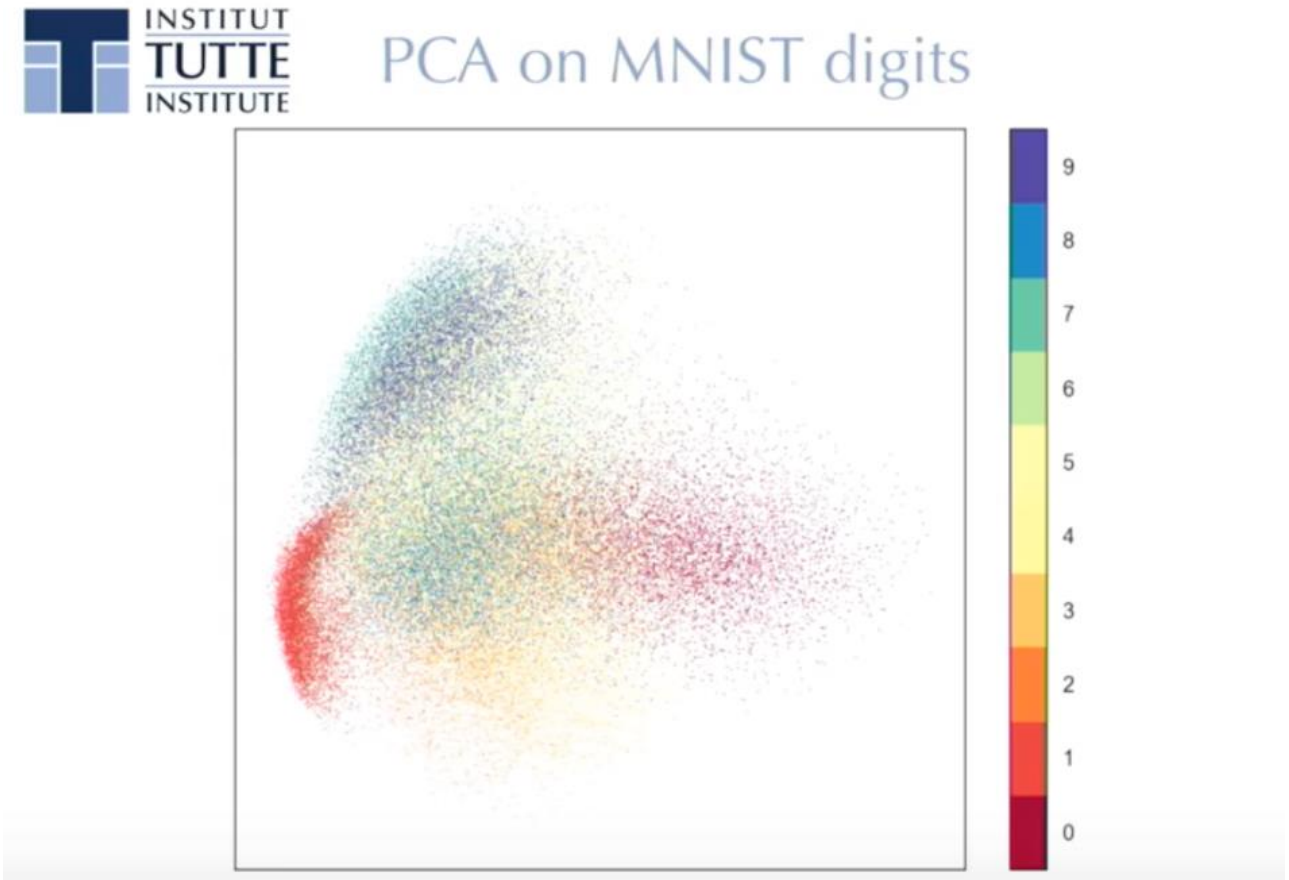
Some parameters in Seurat:

```
n_neighbors = 30L,
min_dist = 0.3,
metric = "correlation",
seed.use = 42,
n_epochs=200
```

Comparing tSNE and UMAP in terms of computation time

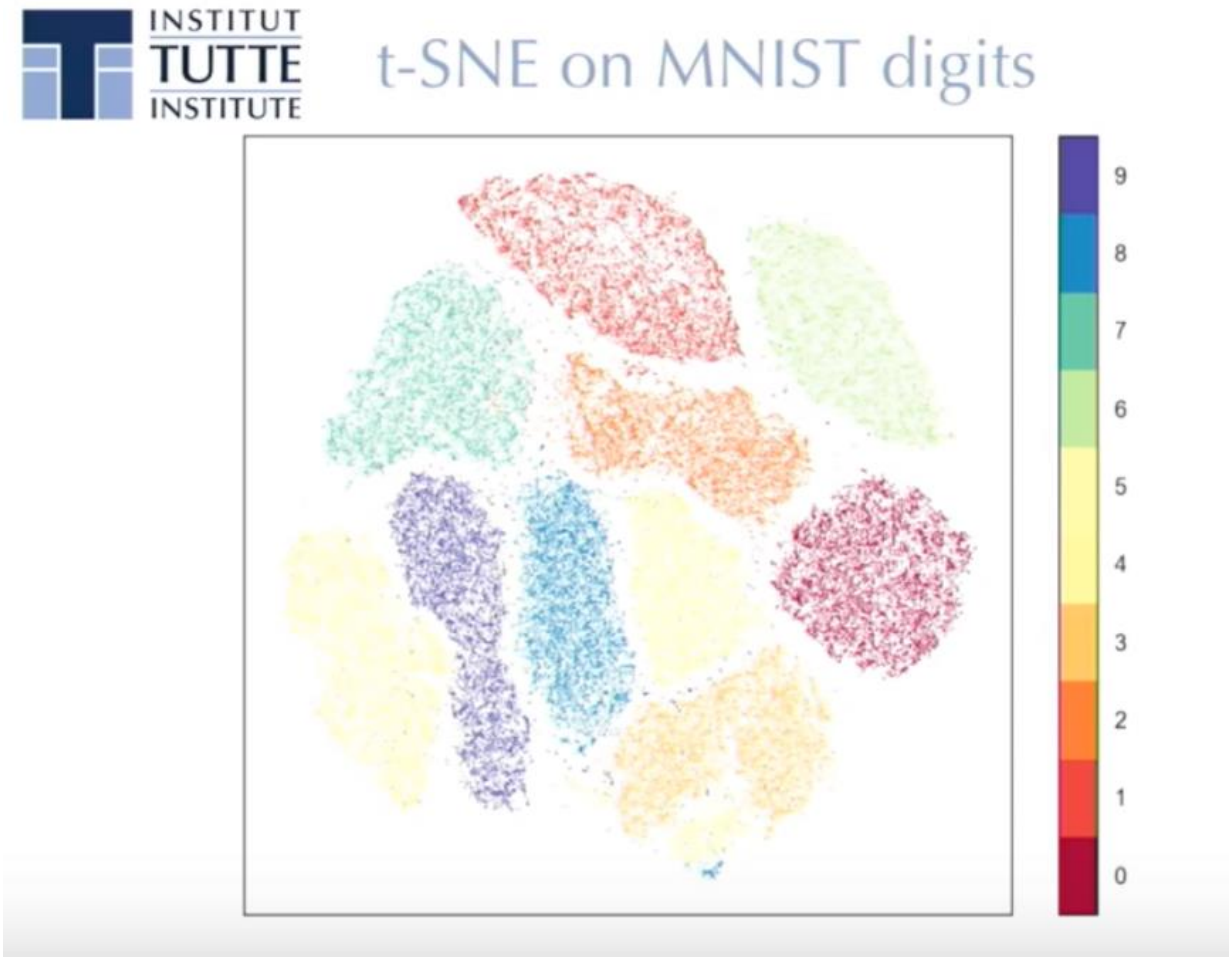
	t-SNE	UMAP
COIL20	20 seconds	7 seconds
MNIST	22 minutes	98 seconds
Fashion MNIST	15 minutes	78 seconds
GoogleNews	4.5 hours	14 minutes

PCA is good, but one can do better!



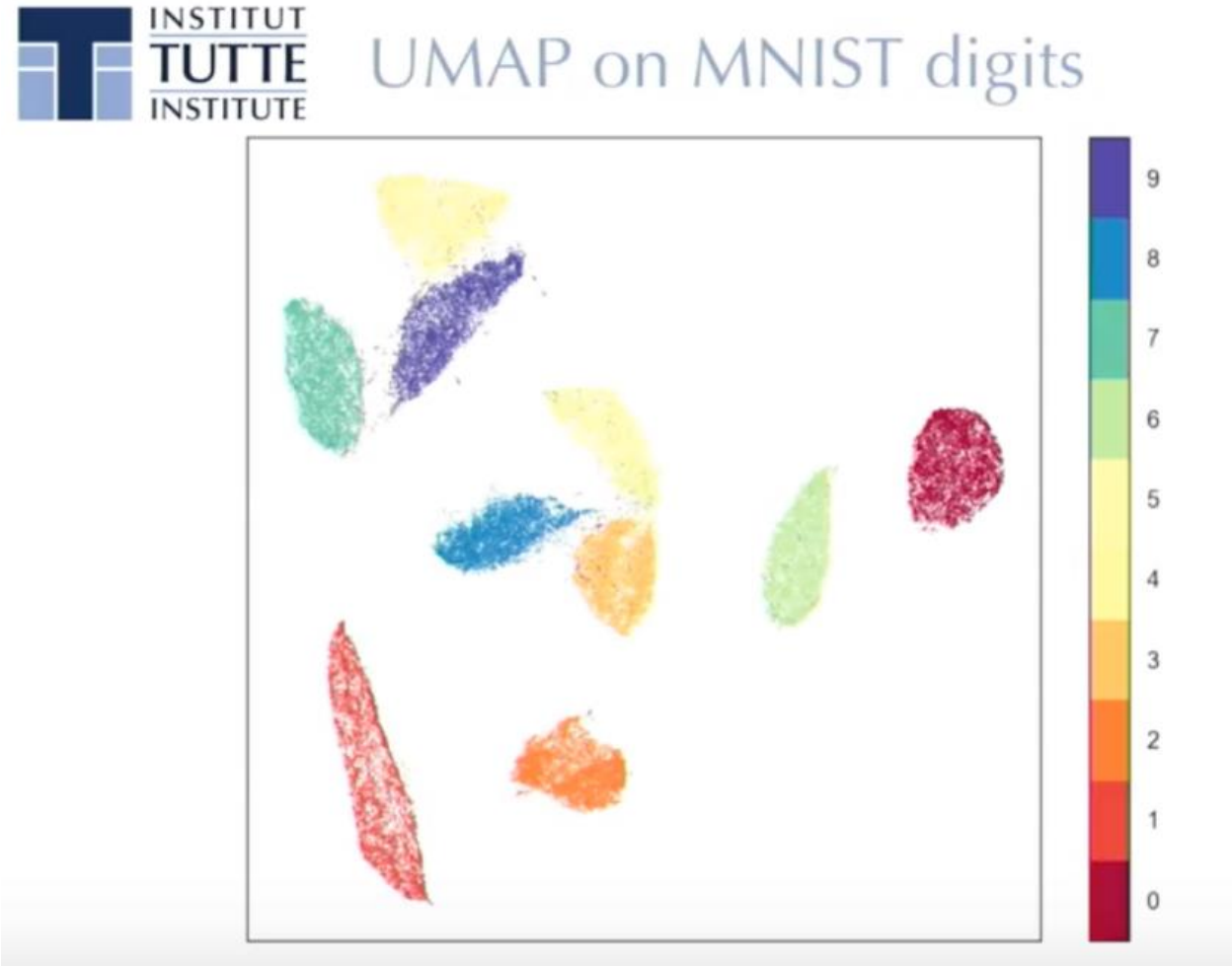
From L.McInnes, SciPy 2018

T-SNE manages to see the local structure



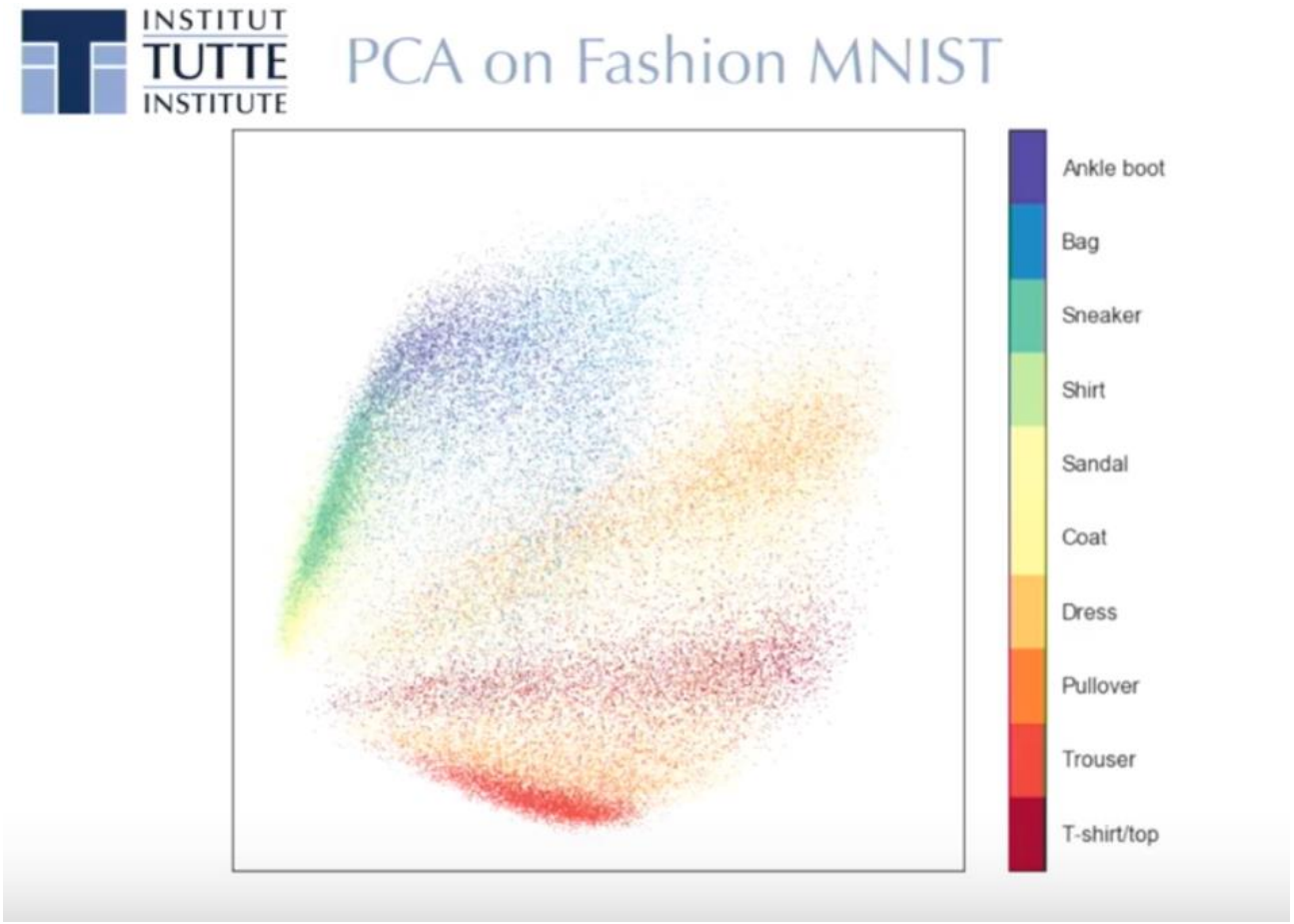
From L.McInnes, SciPy 2018

UMAP



From L.McInnes, SciPy 2018

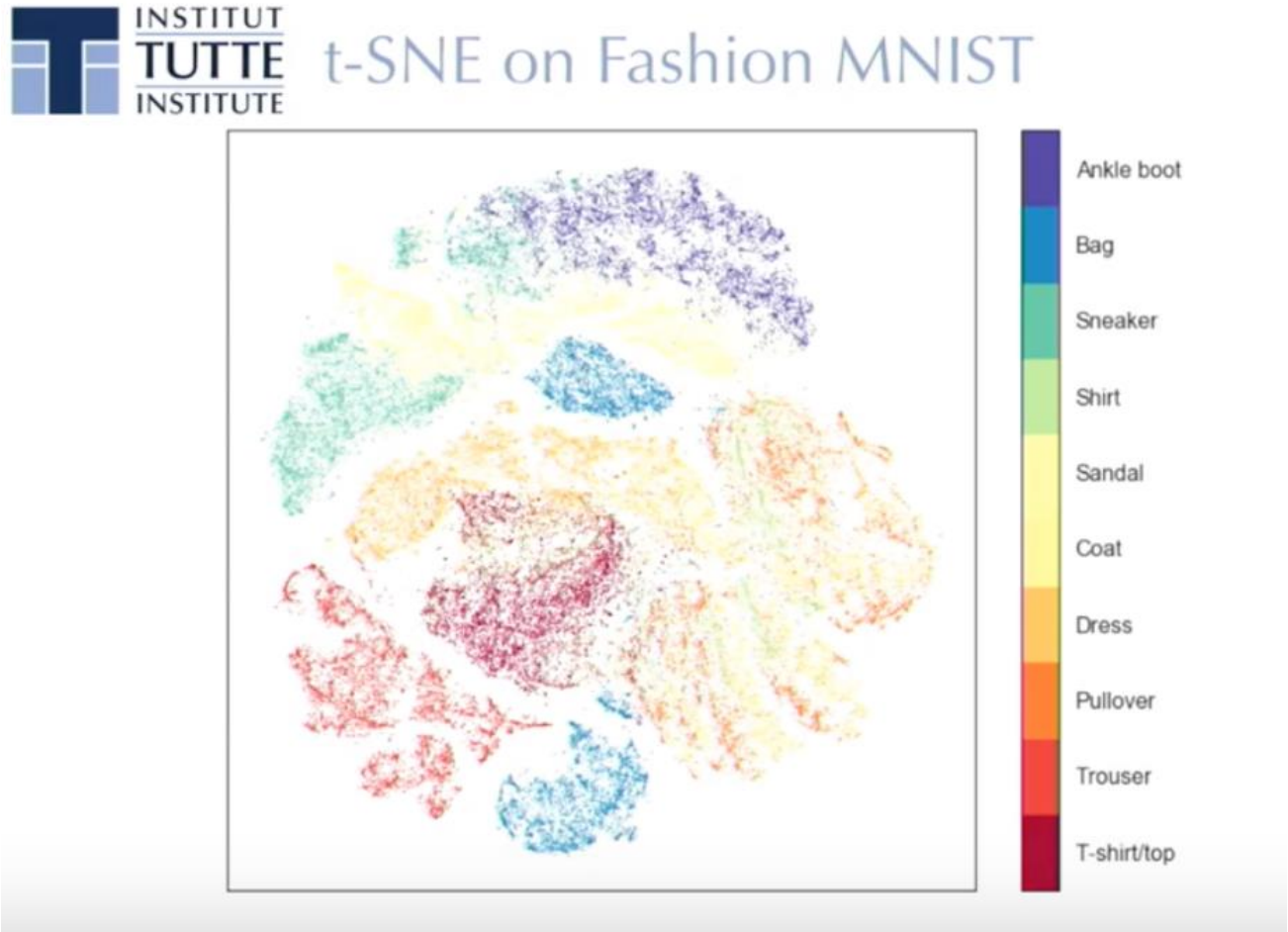
PCA is good, but one can do better!



See the global structure and Interpretable axis

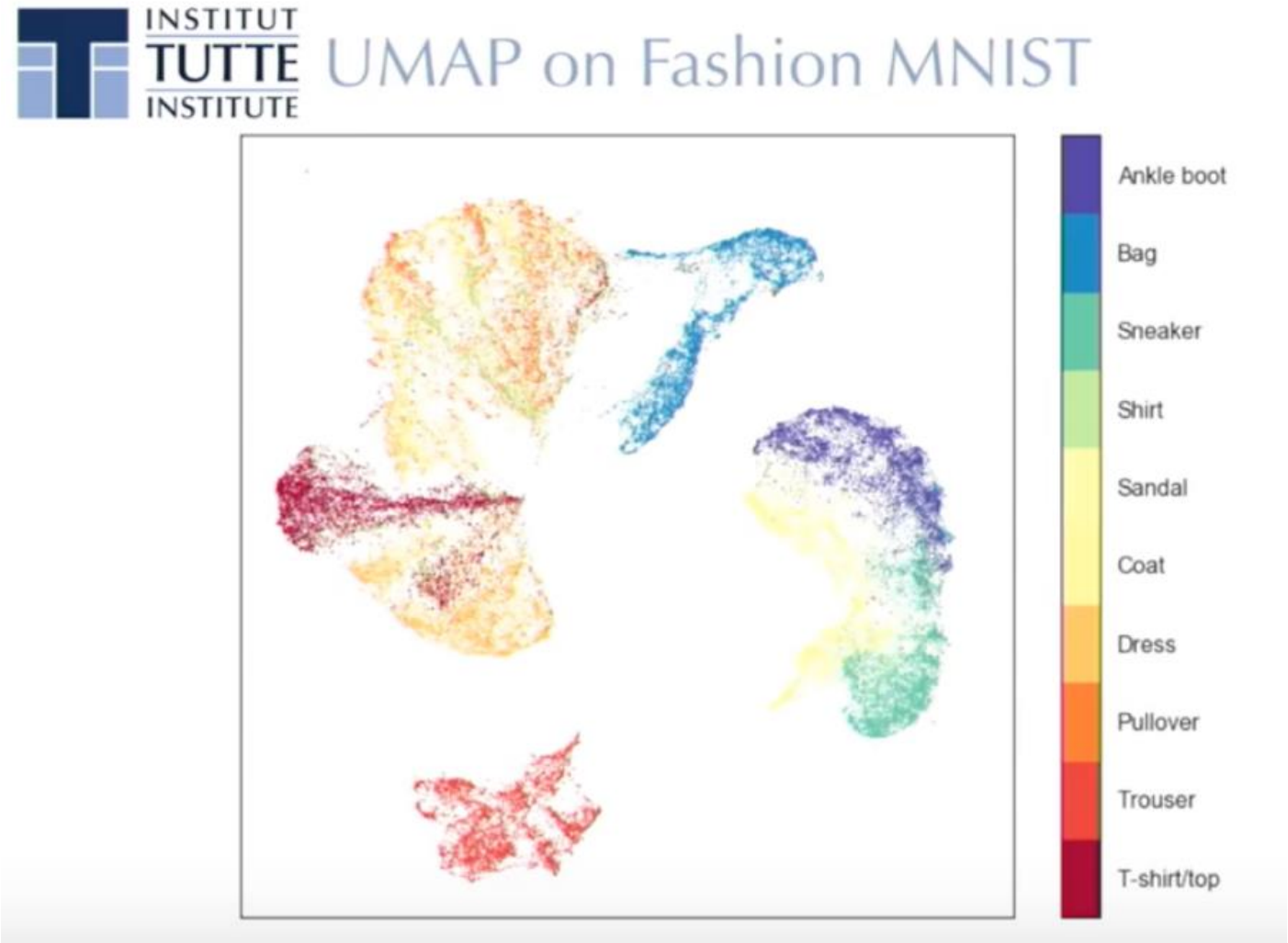
From L.McInnes, SciPy 2018

T-SNE manages to see the local structure



From L.McInnes, SciPy 2018

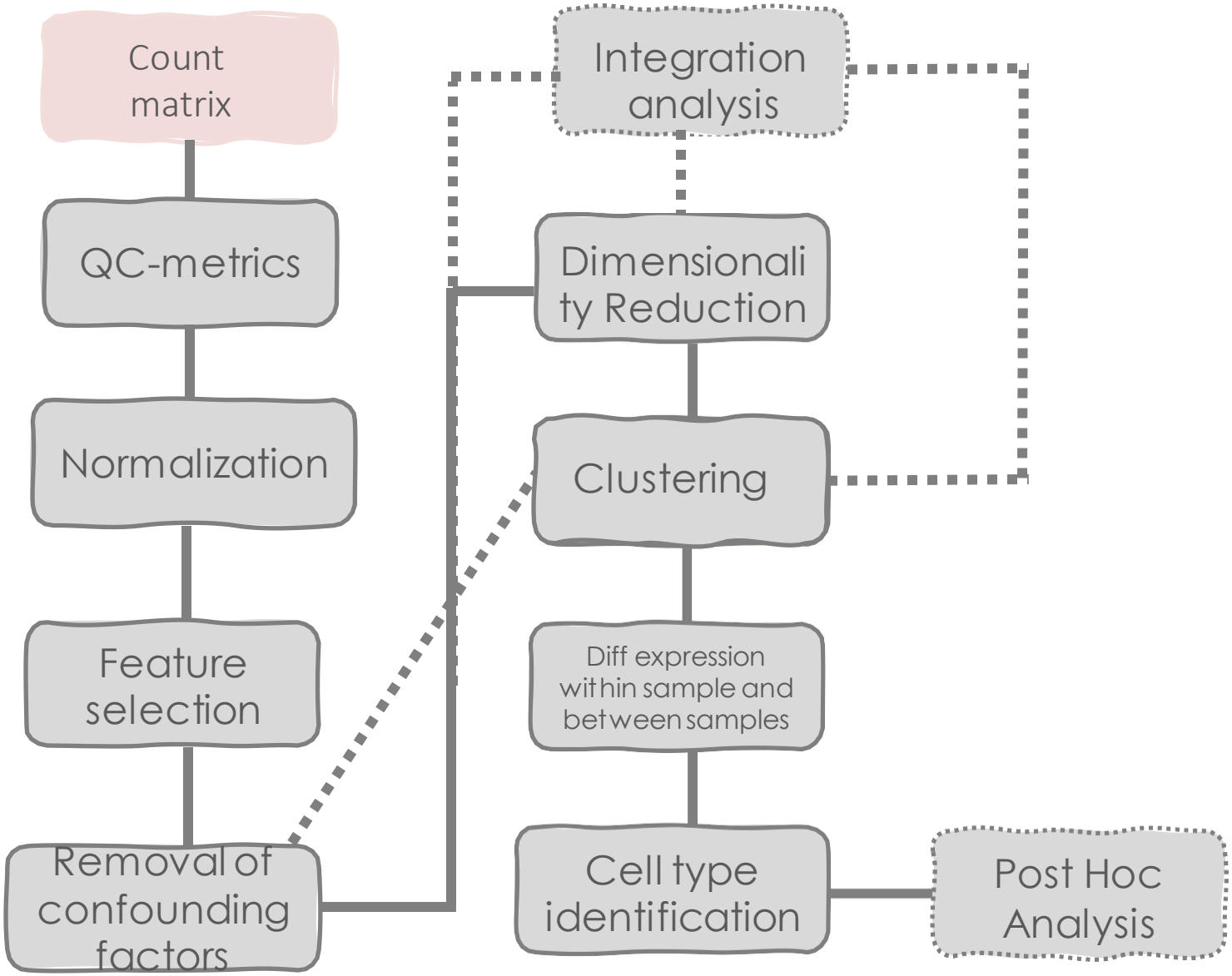
UMAP


INSTITUT
TUTTE
INSTITUTE
UMAP on Fashion MNIST
Ankle boot
Bag
Sneaker
Shirt
Sandal
Coat
Dress
Pullover
Trouser
T-shirt/top

In Seurat

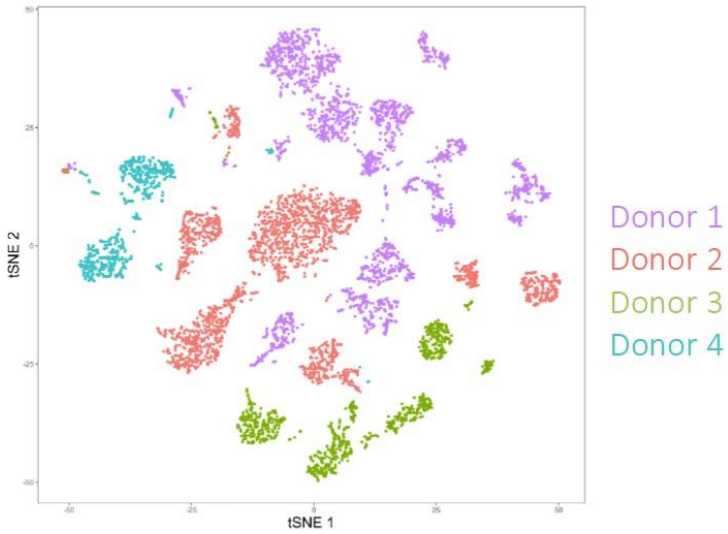
```
obj <-RunPCA( obj )
obj <-RunTSNE( obj )
obj <-RunUMAP( obj )
```

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

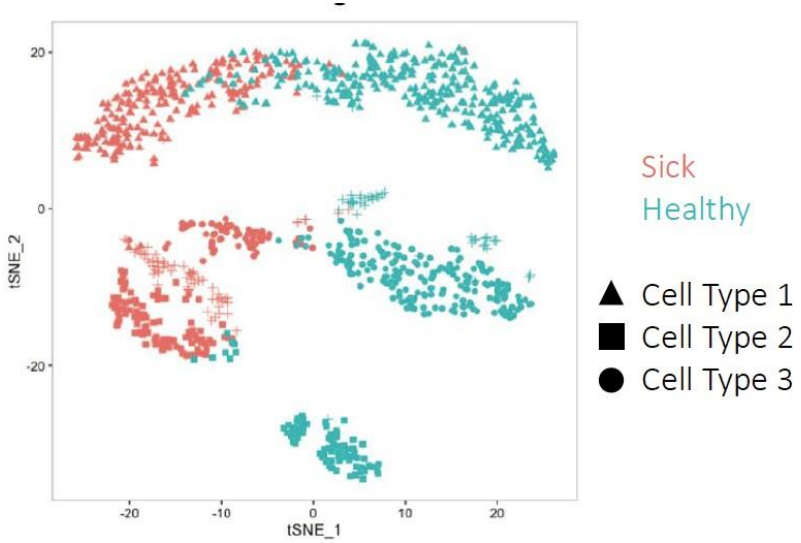

Count matrix
QC-metrics
Normalization
Feature selection
Removal of confounding factors
Integration analysis
Dimensionality Reduction
Clustering
Diff expression within sample and between samples
Cell type identification
Post Hoc Analysis

Integration analysis

- Why do we integrate?



Same tissue from different donors

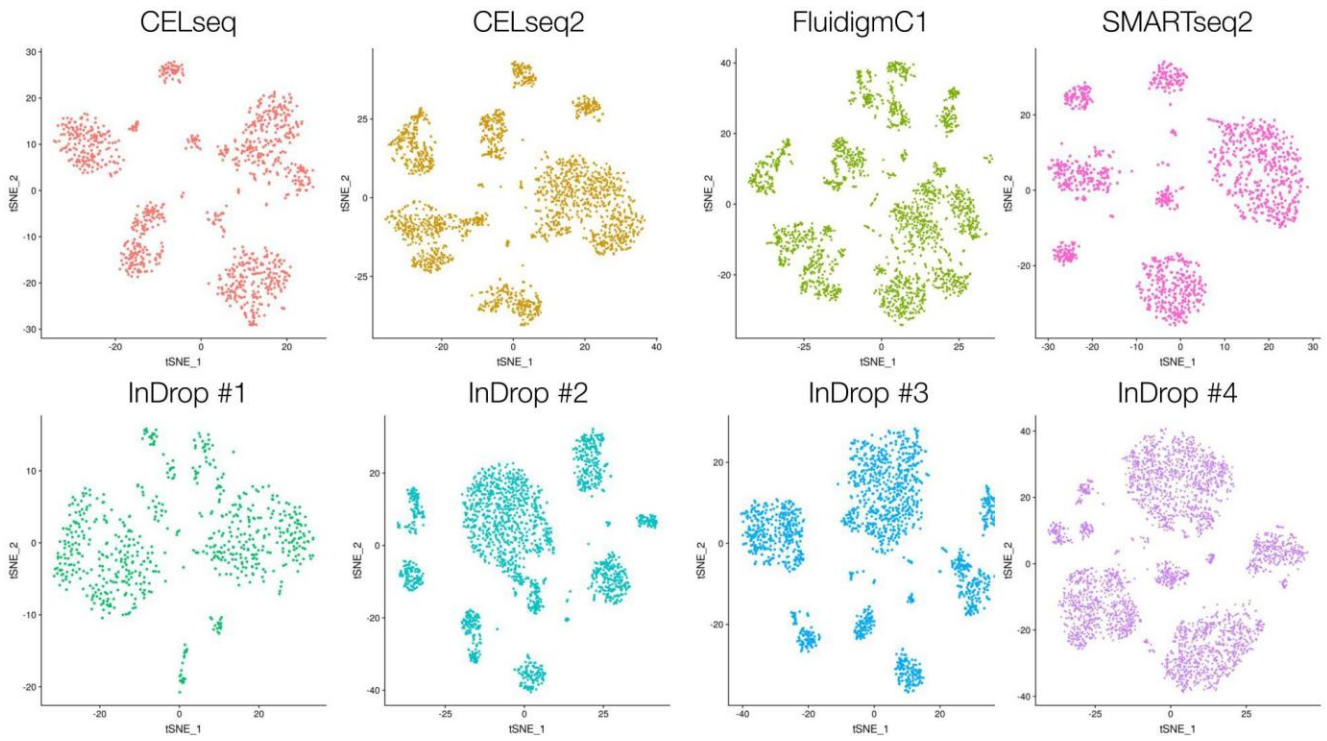


Cross condition comparisons

Courtesy form Ahmed Mahfouz

Integration analysis

- 8 maps from the human pancreas (Seurat tutorial)



Baron et al. 2016, *Cell Syst.*

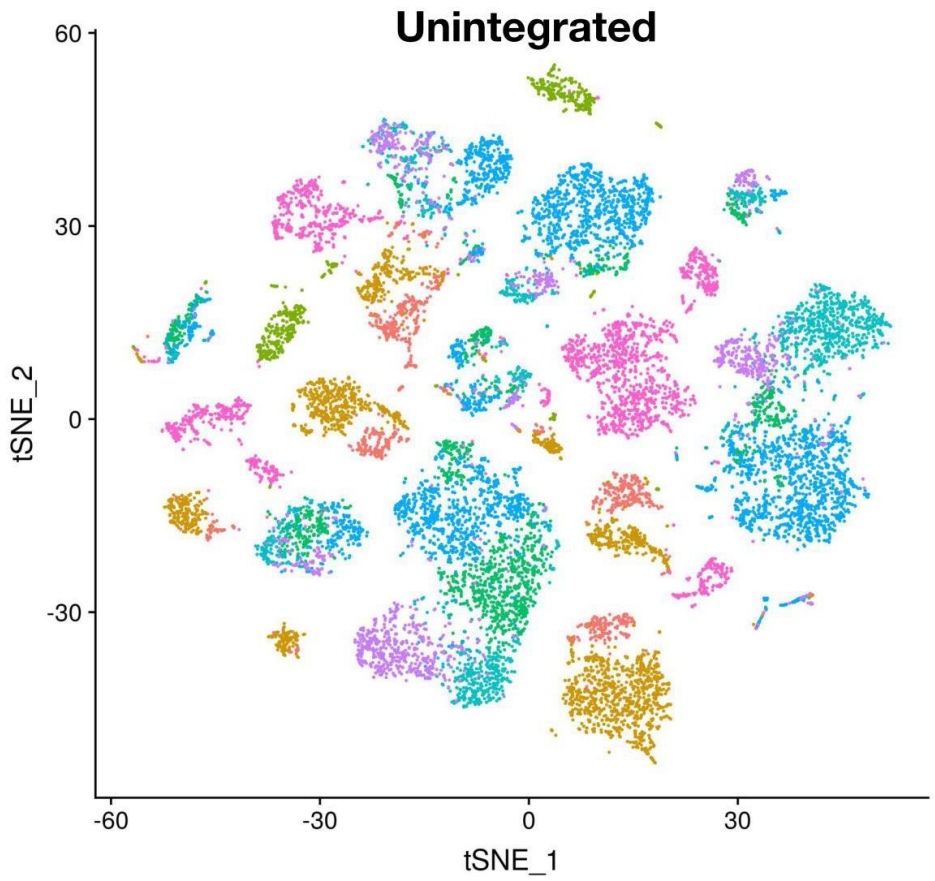
Lawlor et al. 2017, *Genome Res.*

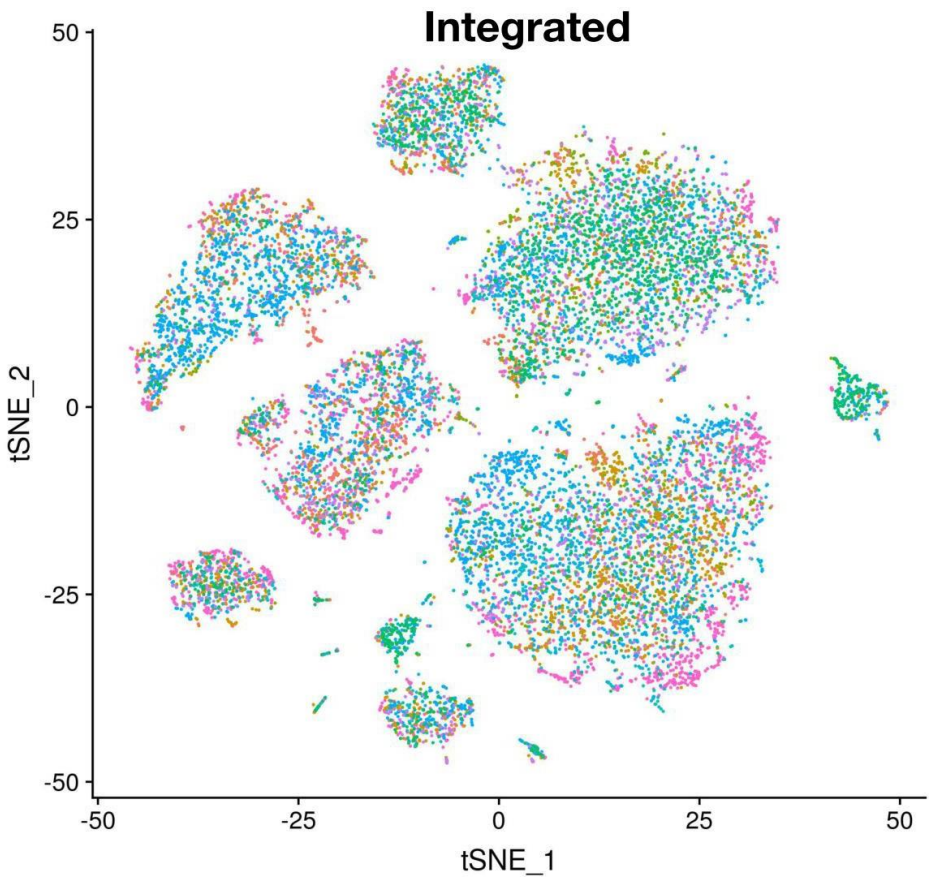
Grun et al. 2016, *Cell Stem Cell*

Muraro et al. 2016, *Cell Syst.*

Integration analysis

- 8 maps from the human pancreas (Seurat tutorial)





Integration analysis: Confounders and batch effect

1. Technical variability
   - Changes in sample quality/processing
   - Library prep or sequencing technology

Technical 'batch effects' confound downstream analysis

2. Biological variability
   - Patient differences
   - Evolution! (cross-species analysis)

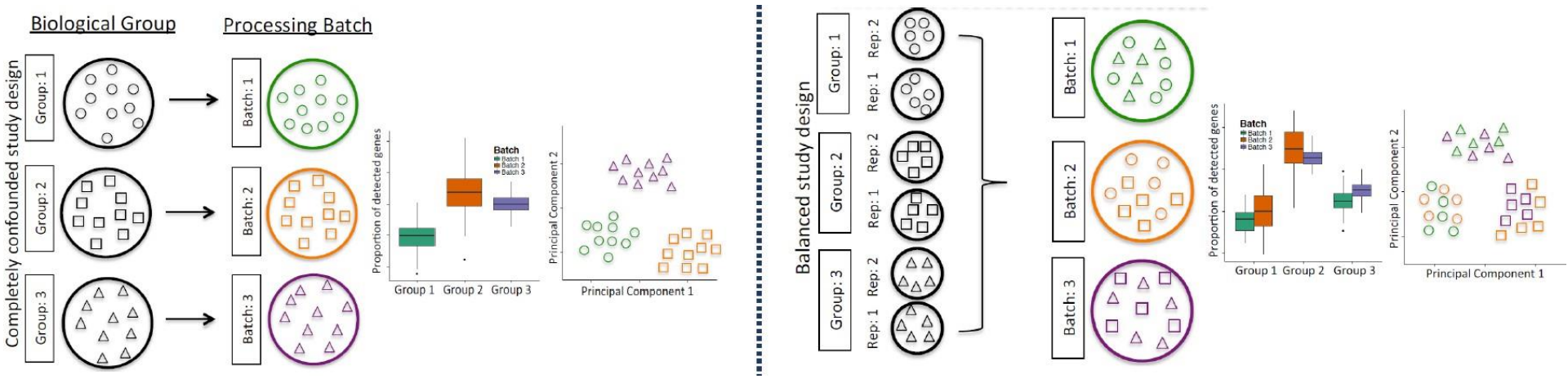
Biological 'batch effects' confound comparisons of scRNA-seq data

Shaham et al. (https://doi.org/10.1093/bioinformatics/btx196)

Integration analysis: Confounders and batch effect

Confounded design

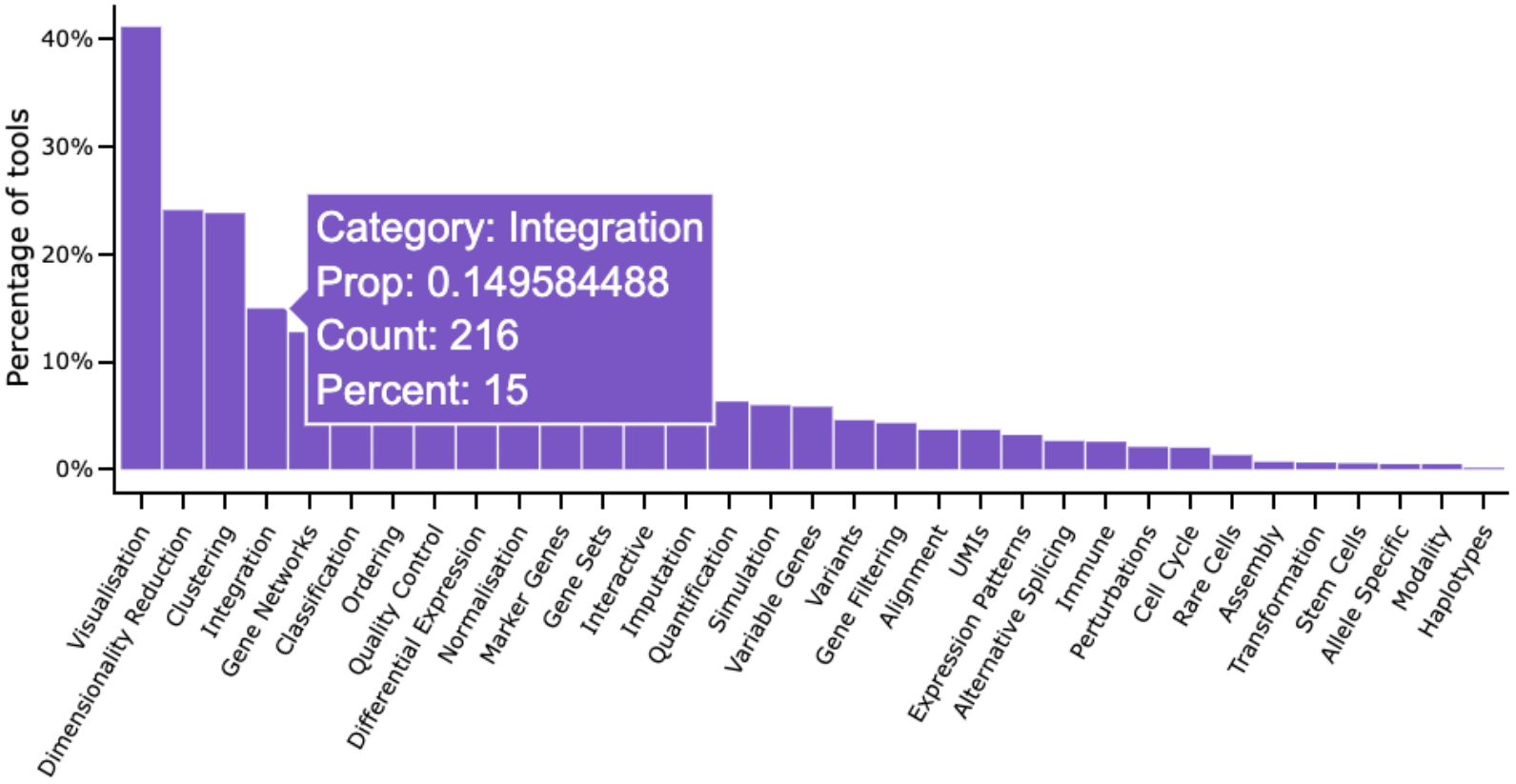
Not confounded design



Good experimental design *does not remove batch effects*, it prevents them from biasing your results.

Hicks et al. (https://doi.org/10.1093/biostatistics/kxx053)

Integration



scrna-tools.org

Integration analysis

- MNNcorrect (https://doi.org/10.1038/nbt.4091)
- CCA +anchors (Seurat v3) (https://doi.org/10.1101/460147)
- CCA +dynamic time warping (Seurat v2) (https://doi.org/10.1038/nbt.4096)
- LIGER (https://doi.org/10.1101/459891)
- Harmony (https://doi.org/10.1101/461954)
- Conos (https://doi.org/10.1101/460246)
- Scanorama (https://doi.org/10.1101/371179)
- scMerge (https://doi.org/10.1073/pnas.1820006116)
- STACAS (https://doi.org/10.1093/bioinformatics/btaa755)

Integration analysis

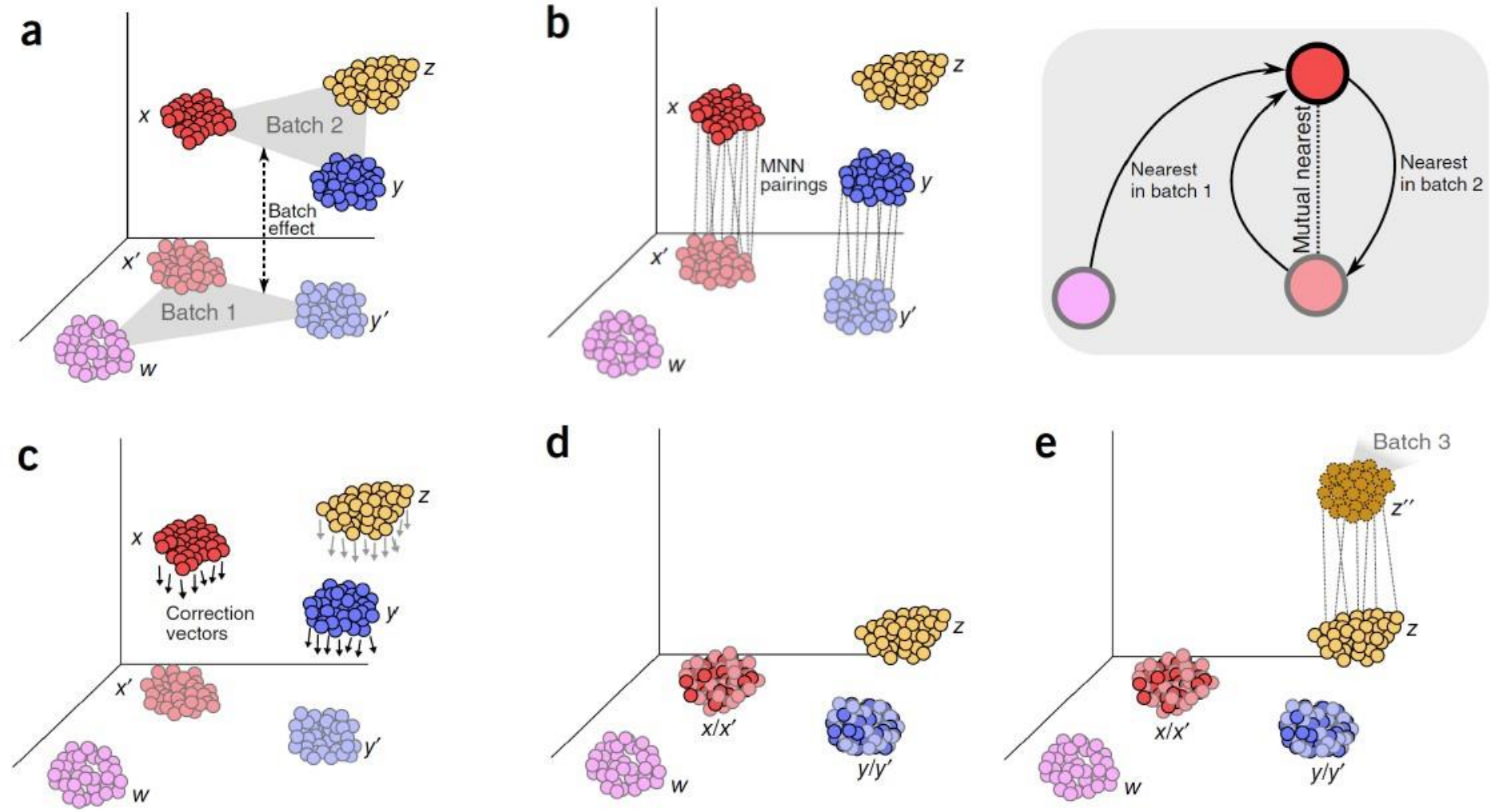
- MNNcorrect (https://doi.org/10.1038/nbt.4091)
- CCA +anchors (Seurat v3) (https://doi.org/10.1101/460147)
- CCA +dynamic time warping (Seurat v2) (https://doi.org/10.1038/nbt.4096)
- LIGER (https://doi.org/10.1101/459891)
- Harmony (https://doi.org/10.1101/461954)
- Conos (https://doi.org/10.1101/460246)
- Scanorama (https://doi.org/10.1101/371179)
- scMerge (https://doi.org/10.1073/pnas.1820006116)
- STACAS (https://doi.org/10.1093/bioinformatics/btaa755)

Integration analysis: Generally speaking

1. Find corresponding cells across datasets (by computing a distance between cells in a certain space)
2. Compute a data adjustment based on correspondences between cells
3. Apply the adjustment

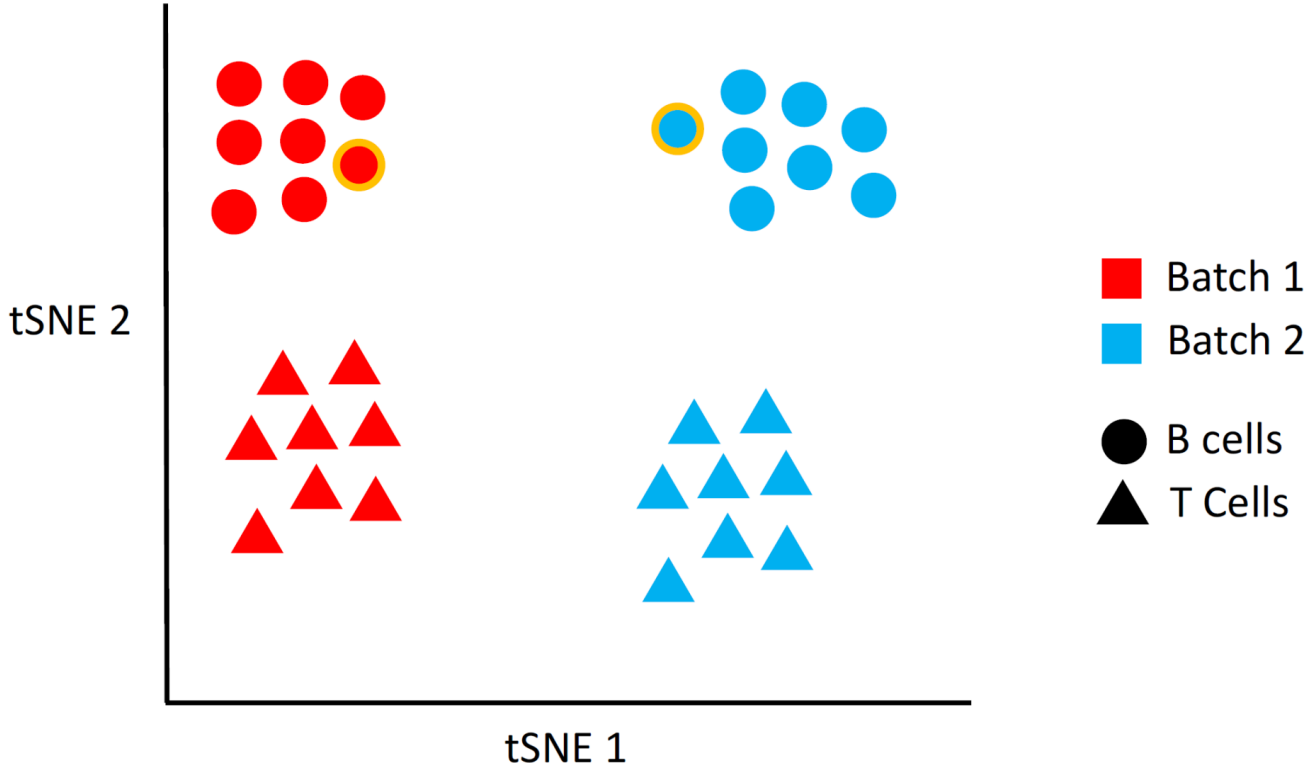
Stuart et al. (https://doi.org/10.1101/460147)

Integration analysis: Mutual Nearest Neighbors (MNN)

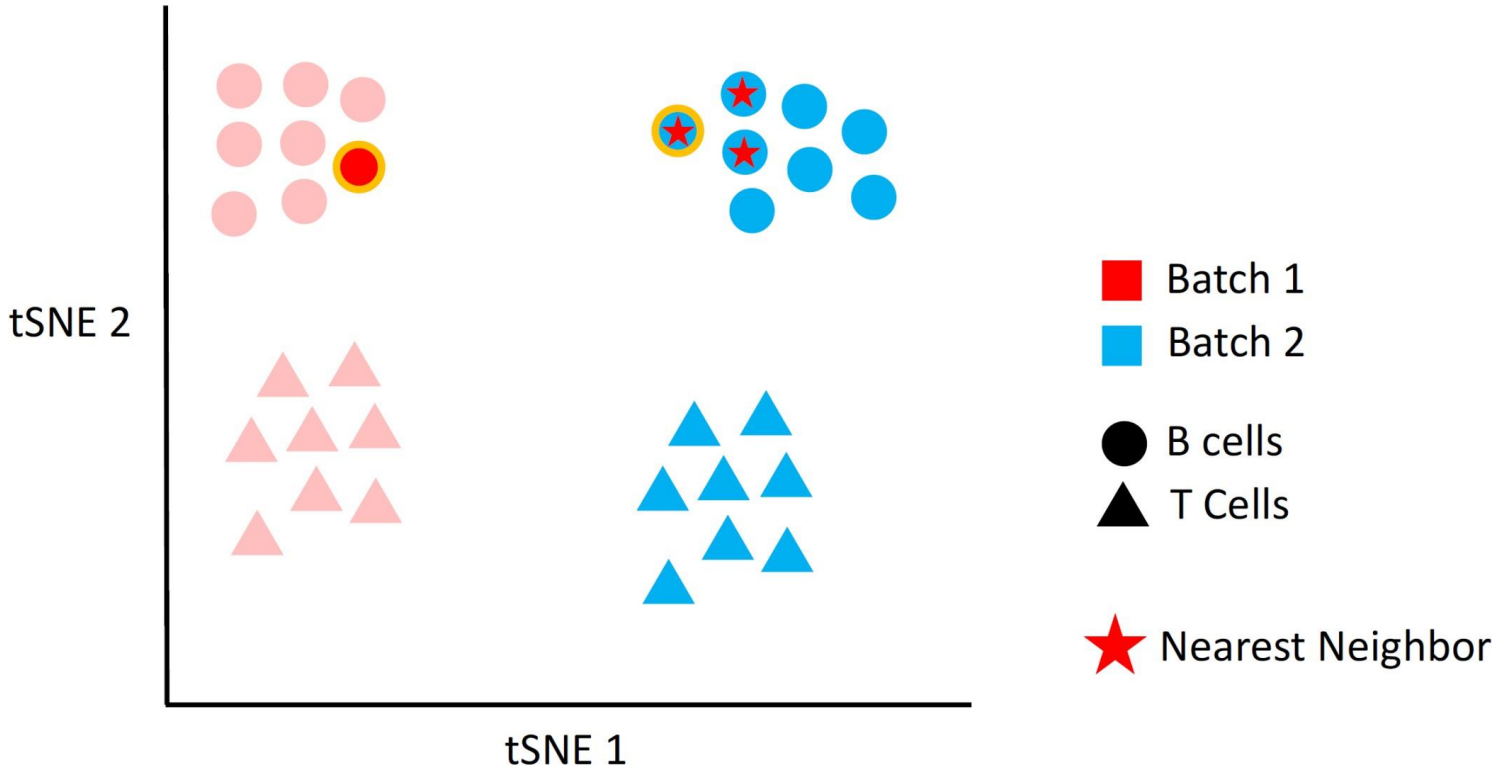


Haghverdi (https://doi.org/10.1038/nbt.4091)

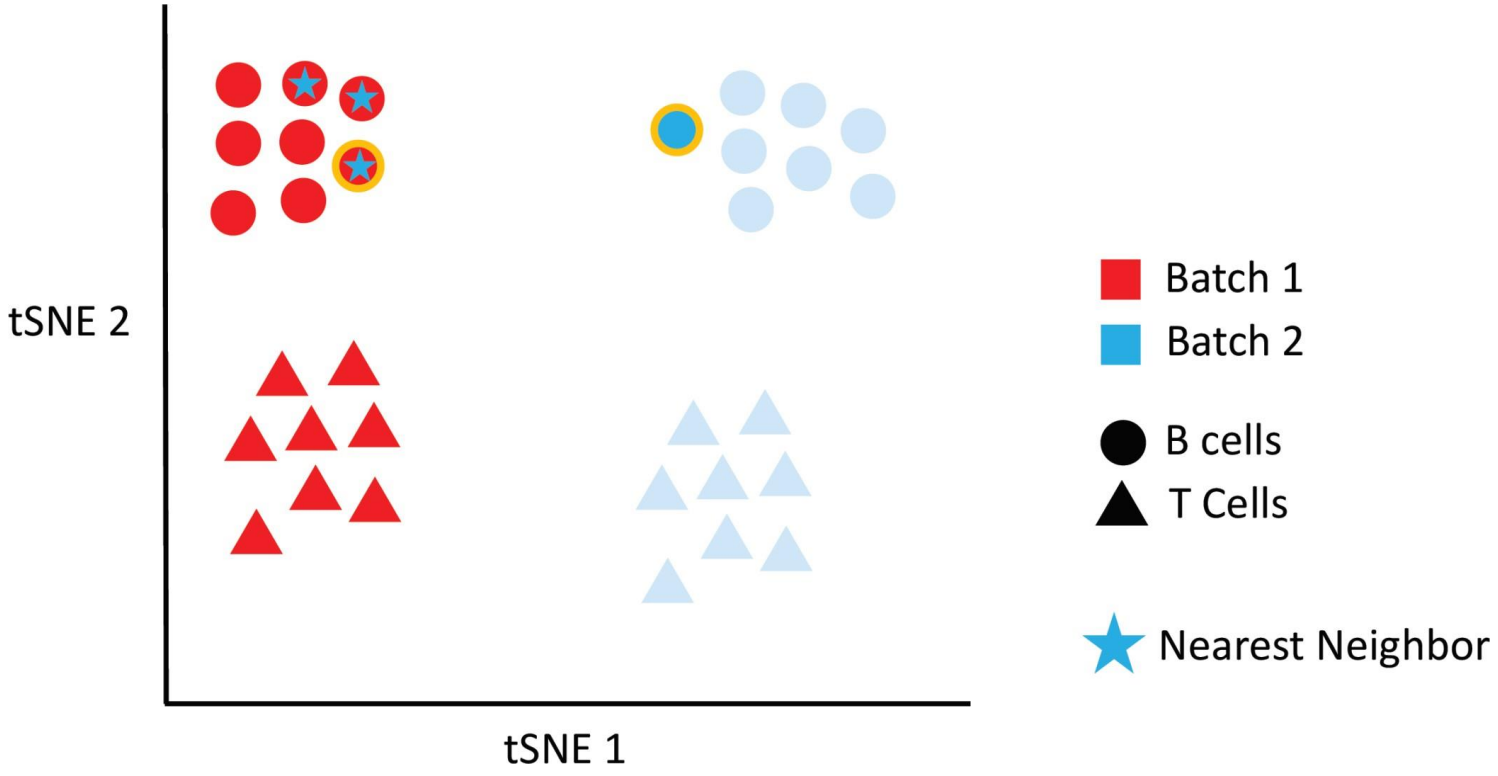
Integration analysis: Mutual Nearest Neighbors (MNN)



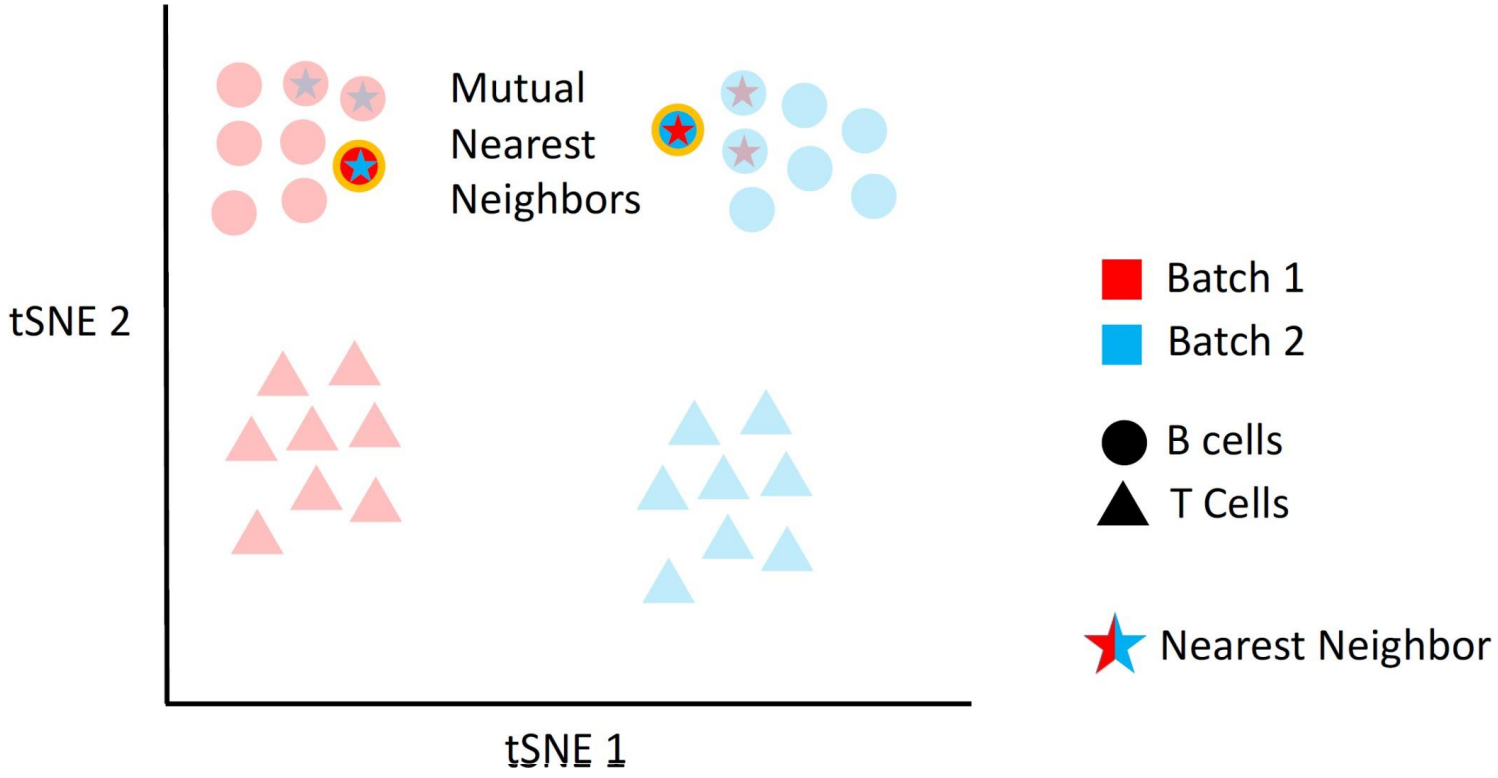
Integration analysis: Mutual Nearest Neighbors (MNN)



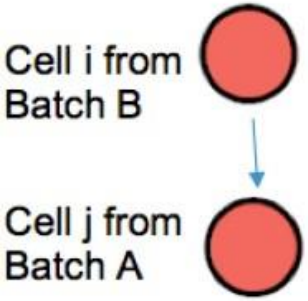
Integration analysis: Mutual Nearest Neighbors (MNN)



Integration analysis: Mutual Nearest Neighbors (MNN)



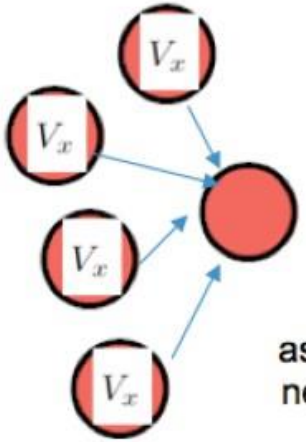
Integration analysis: Mutual Nearest Neighbors (MNN)



1) For each MNN pair, a pair-specific batch-correction vector is computed as the vector difference between the expression profiles of the paired cells.

$$V_x = \begin{pmatrix} gene1_a - gene1_b \\ gene2_a - gene2_b \\ gene3_a - gene3_b \\ \dots \\ geneN_a - geneN_b \end{pmatrix}$$

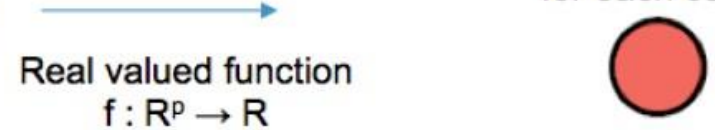
2) A cell-specific batch-correction vector is then calculated as a weighted average of these pair-specific vectors, as computed with a Gaussian kernel.



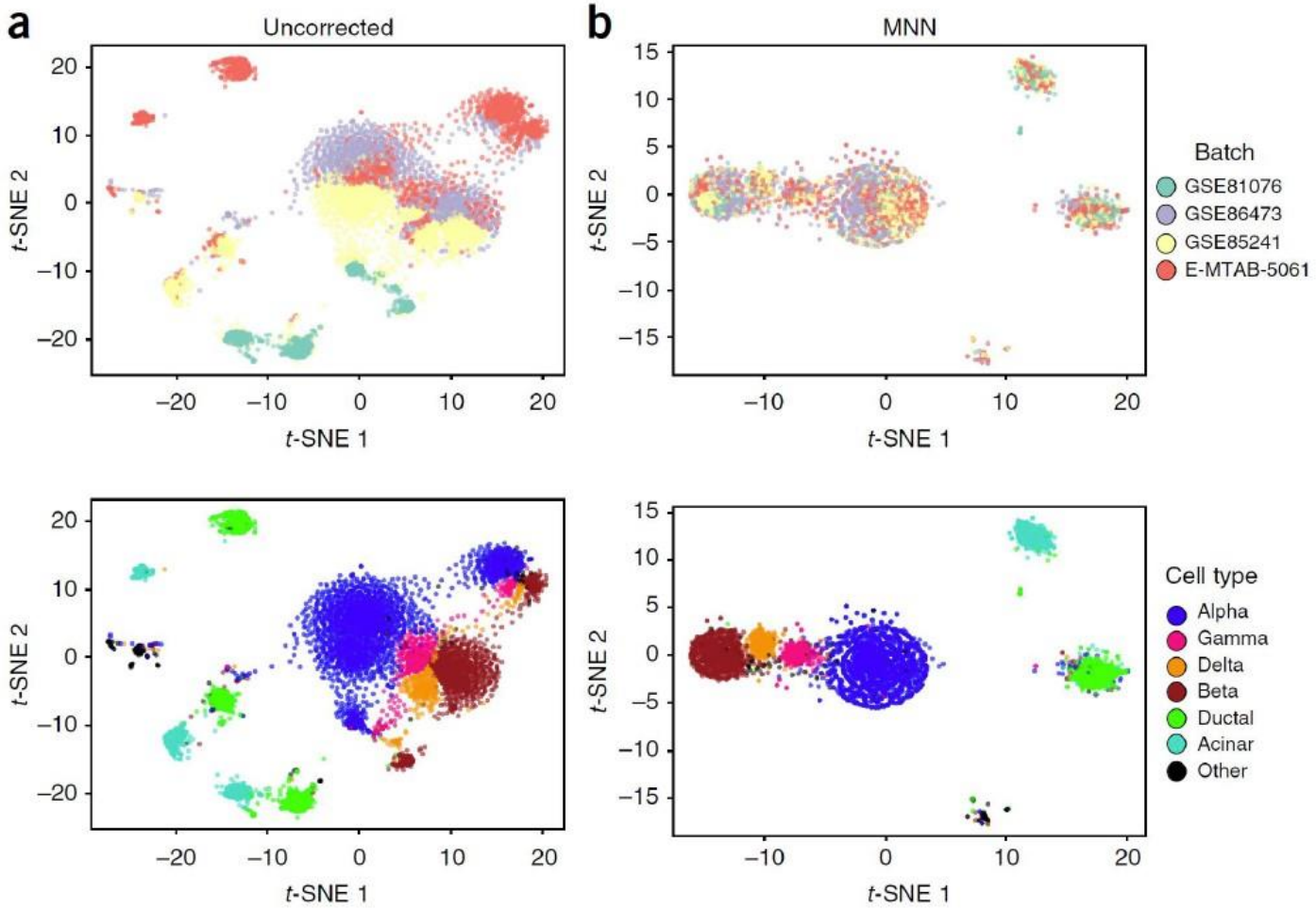
Gaussian Kernel Smoothing

Real valued function
$f : R^p \rightarrow R$
as the weighted average of neighboring observed data

Batch Correction vector for each cell



Integration analysis: Mutual Nearest Neighbors (MNN)



Haghverdi (https://doi.org/10.1038/nbt.4091)

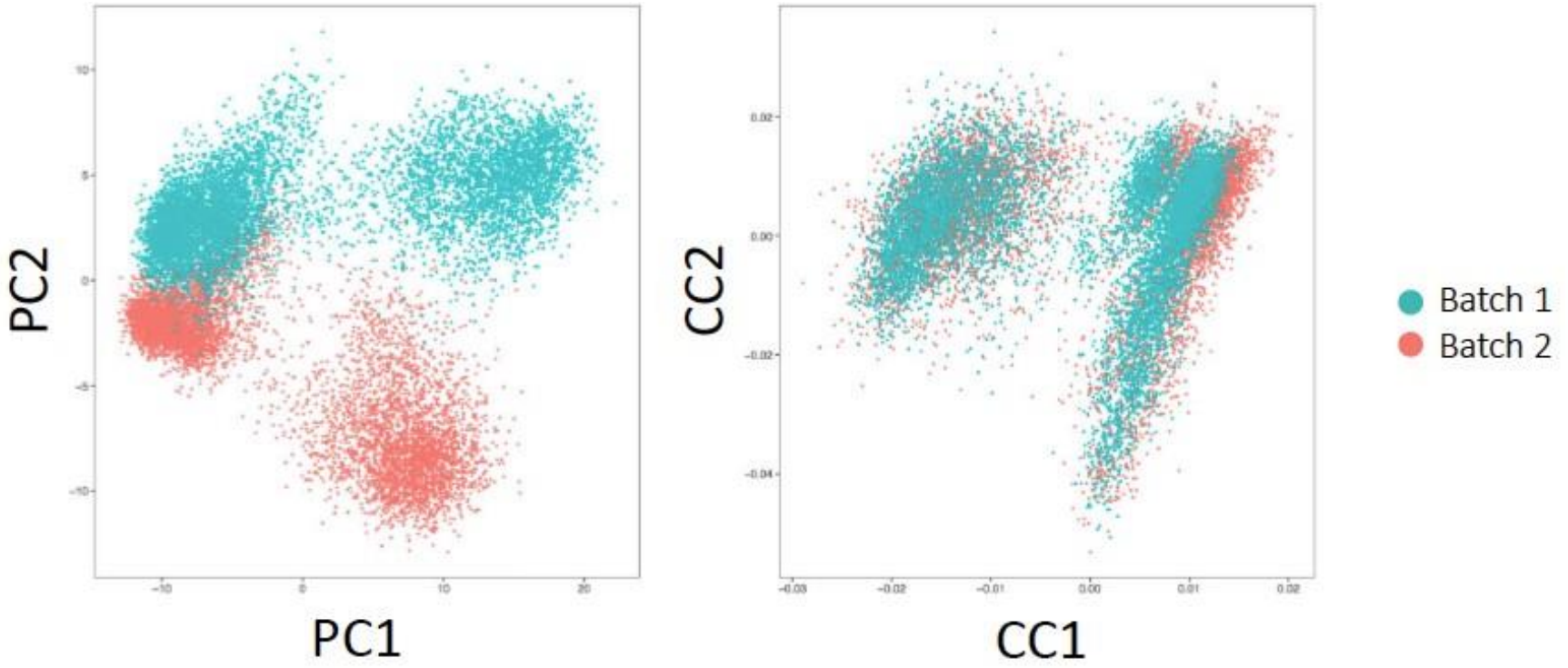
Integration analysis: CCA +anchors (Seurat v3)

1. Find corresponding cells across datasets (anchors)
2. Compute a data adjustment based on correspondences between cells
3. Apply the adjustment

Stuart et al. (https://doi.org/10.1101/460147)

Integration analysis: CCA +anchors (Seurat v3)

1. Find corresponding cells acrossdatasets

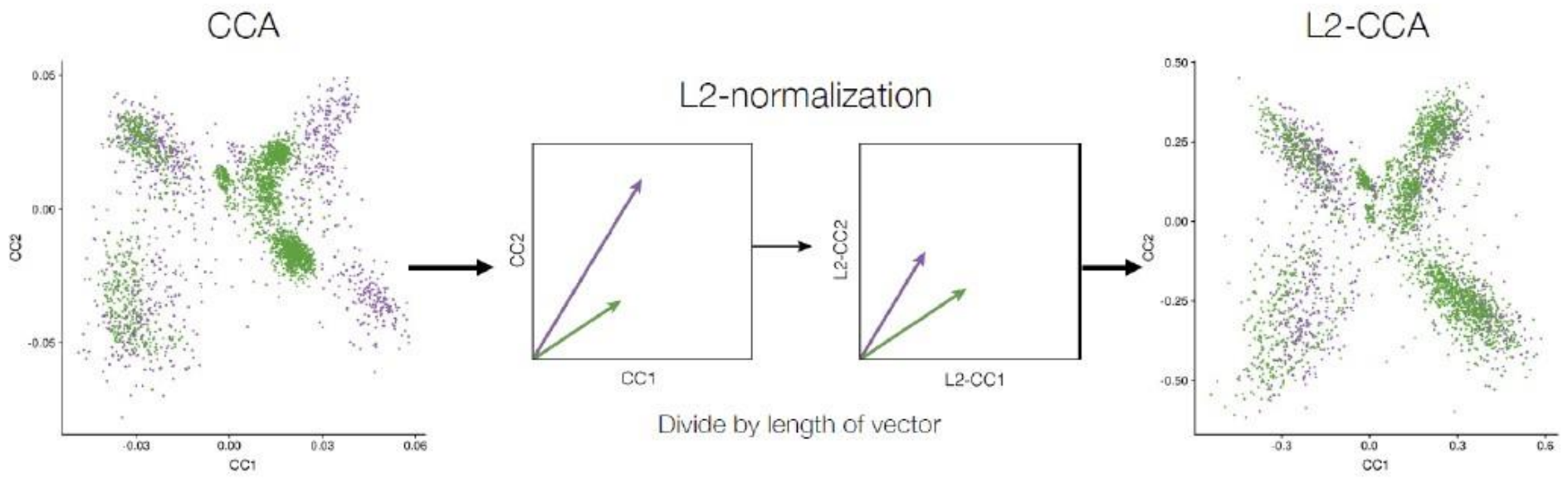


CCA captures correlated sources of variation between two datasets

Stuart et al. (https://doi.org/10.1101/460147)

Integration analysis: CCA +anchors

1. Find corresponding cells acrossdatasets

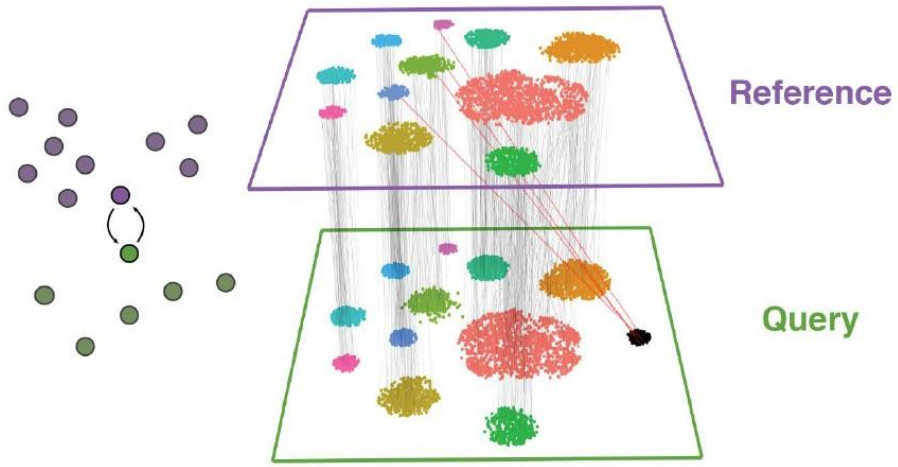


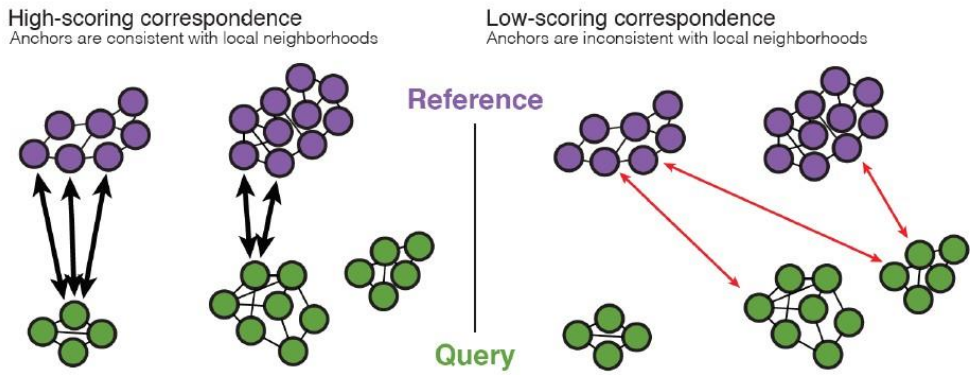
L2-normalization corrects for differences in scale

Integration analysis: CCA +anchors (Seurat v3)

1. Find corresponding cells across datasets

Anchors: Mutual nearest neighbors

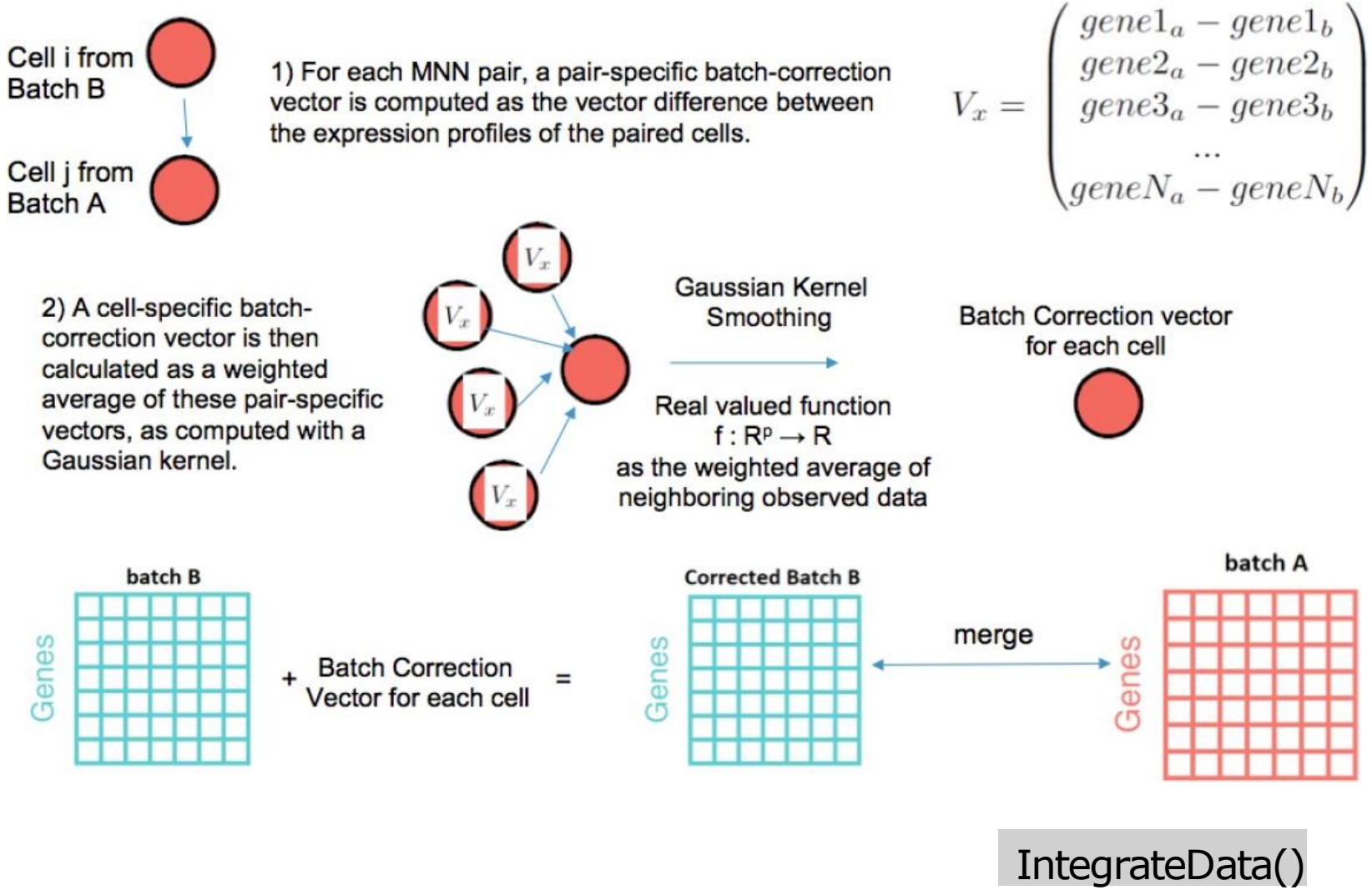




FindIntegrationAnchors()

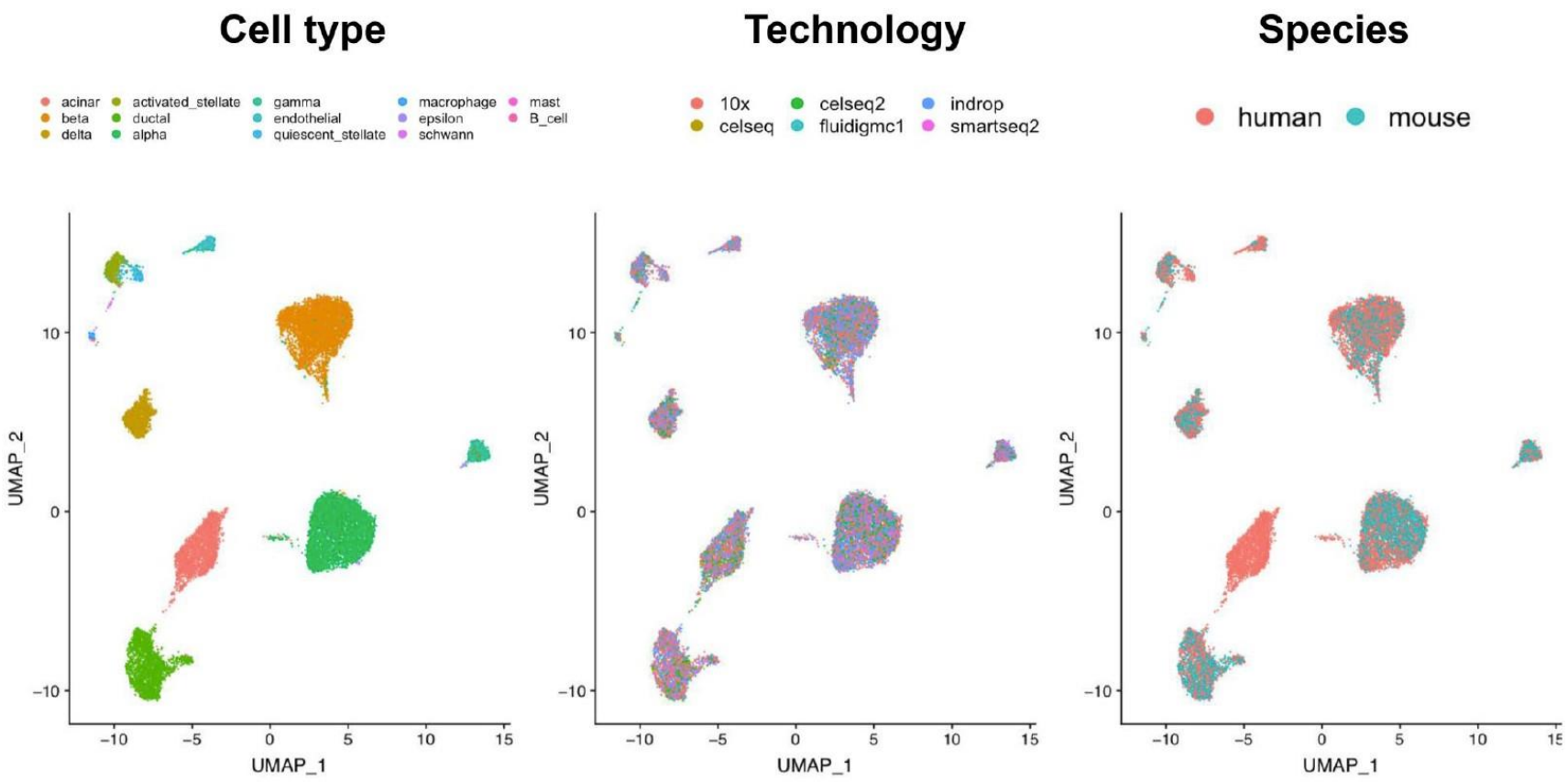
Integration analysis: CCA +anchors

2. Data integration



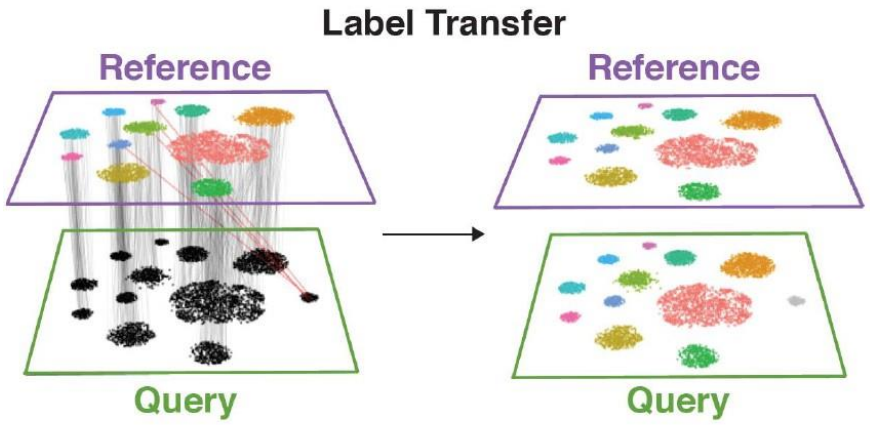
IntegrateData()

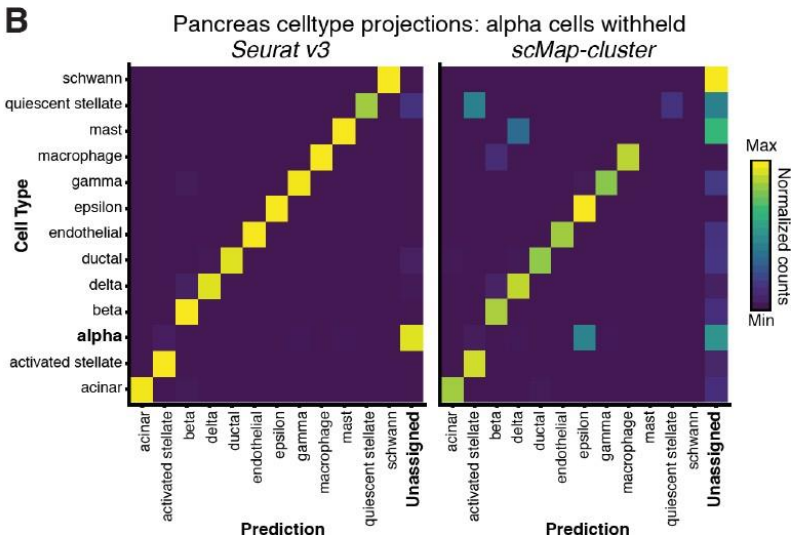
Integration analysis: CCA +anchors



Retinal bipolar datasets: 51K cells, 6 technologies, 2 Species

Label transfer: CCA +anchors





STACAS

- STACAS (https://doi.org/10.1093/bioinformatics/btaa755)
- Sub-Type Anchor Correction for Alignment in Seurat to integrate single-cell RNA-seq data
- Corrected version of Seurat
- Based on labelling of cells-removes "wrong" anchors.