Understanding t-SNE & UMAP

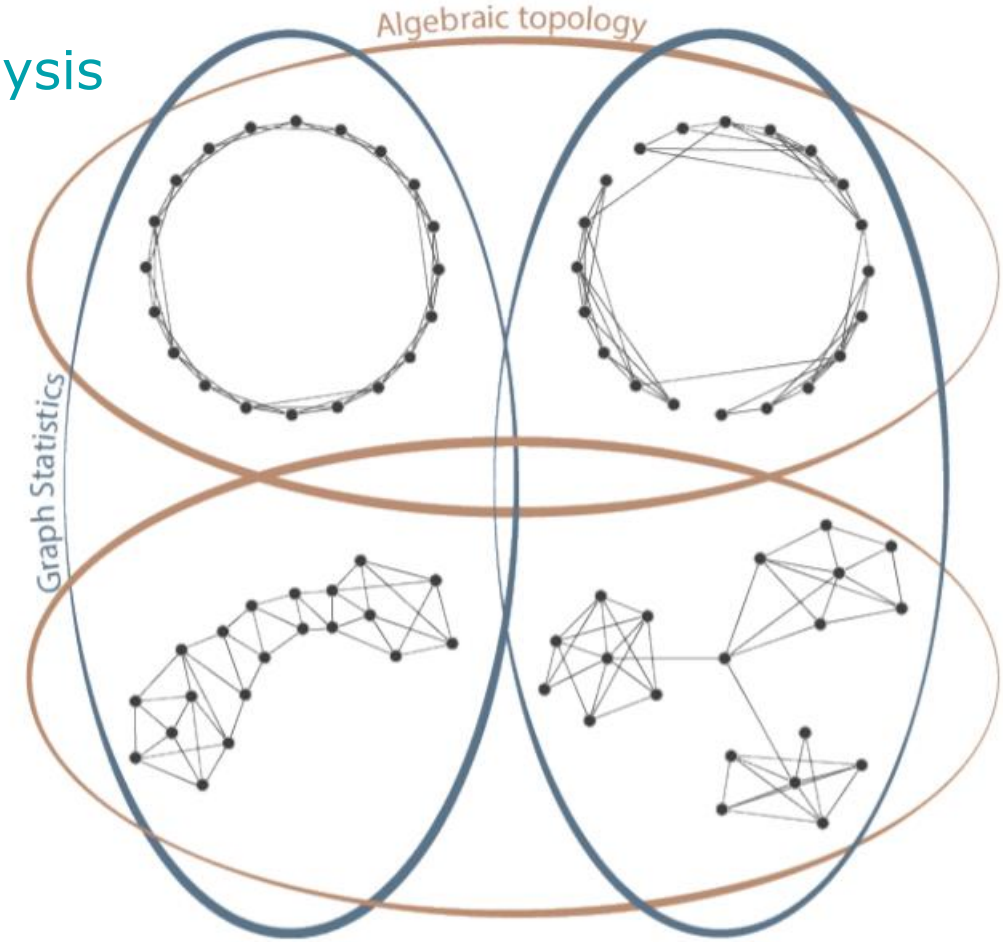
Comparing clustering techniques for single-cell RNA-seq

Razib Obaid, PhD

Yale *Harvey Cushing/John Hay Whitney Medical Library*

Points to discuss

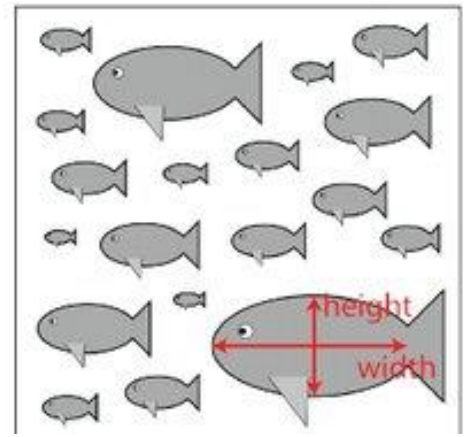
- Principal Component Analysis
- Graph-based clustering
- K-means clustering
- t-SNE
- UMAP

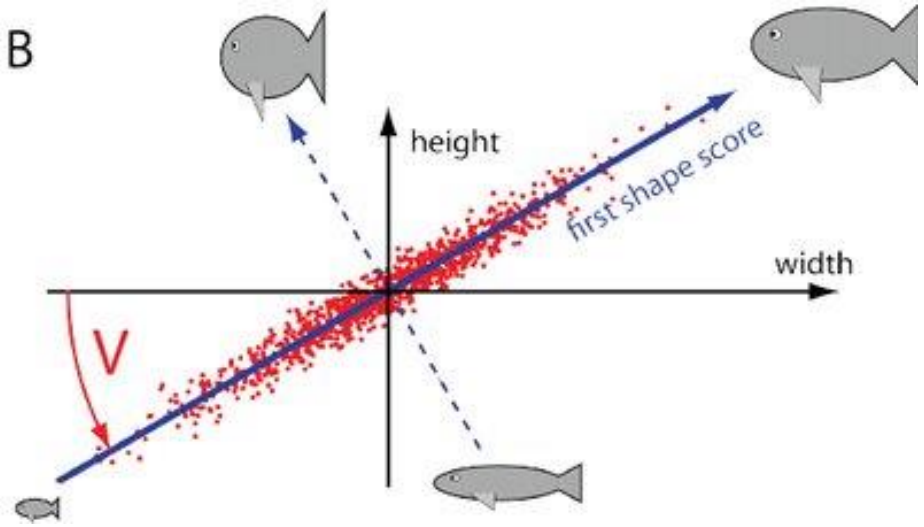


Principal component analysis

High dimensional correlated data → Low dimensional uncorrelated data

(Keeping the variance of the data same for both) UNSUPERVISED





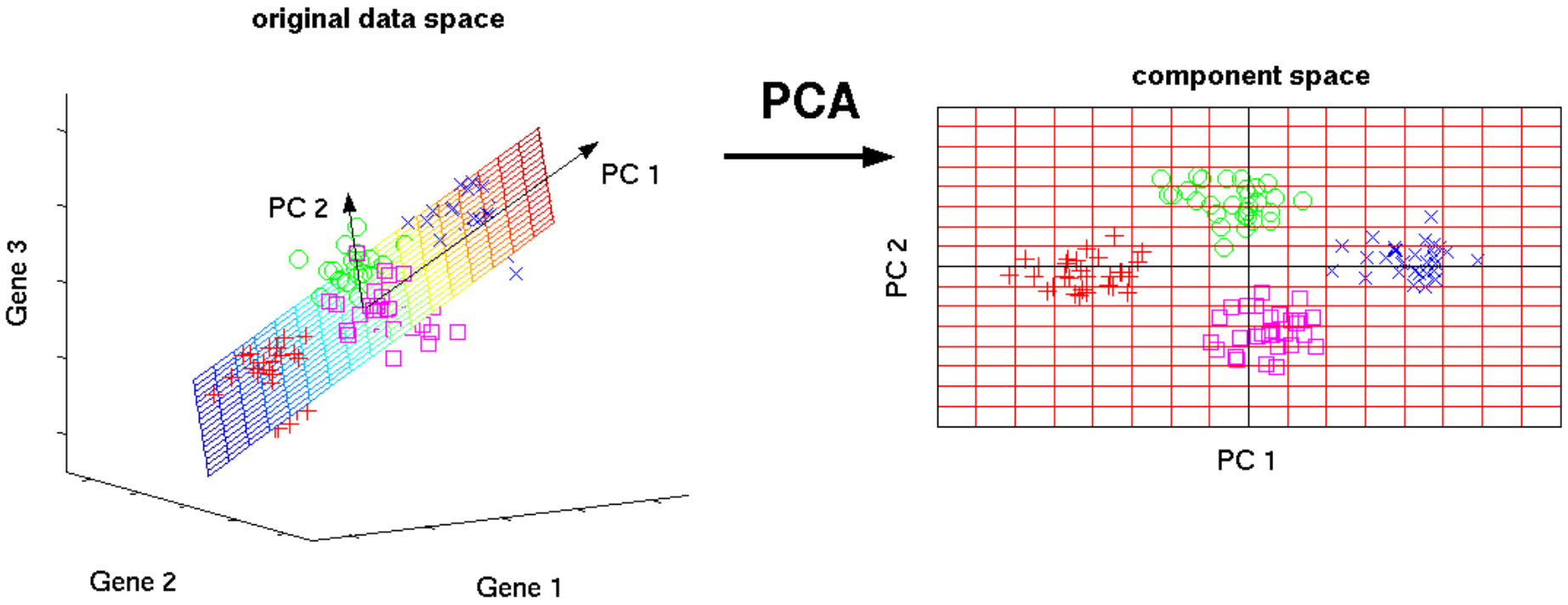
Change of coordinates with maximal feature-feature covariance

Werner, et al., PloS one 9, 11 (2014)

Principal component analysis

High dimensional correlated data → Low dimensional uncorrelated data

(Keeping the variance of the data same for both) UNSUPERVISED



Principal component analysis

High dimensional correlated data → Low dimensional uncorrelated data

(Keeping the variance of the data same for both) UNSUPERVISED

Keys steps of the algorithm:

Take high D data, normalize and center it

Compute covariance matrix

Compute eigenvectors and eigenvalues

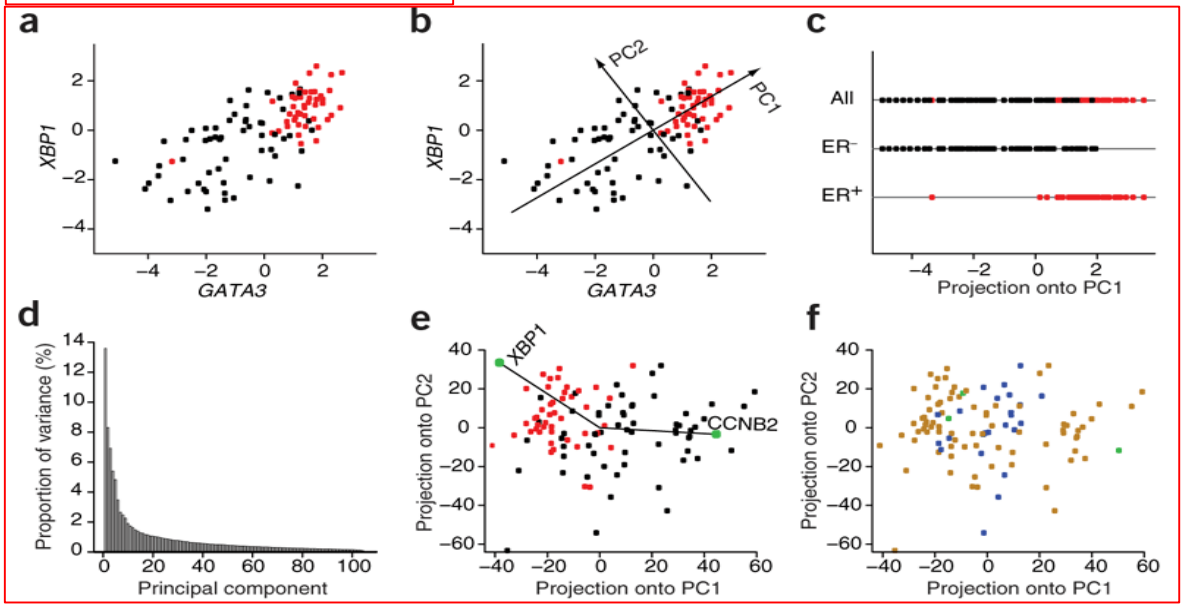
Pick number of eigenvectors ← **This is where the user/analyst comes in!**

Project data points to those eigenvectors

Principal component analysis

Pick number of eigenvectors

This is where the user/analyst comes in!



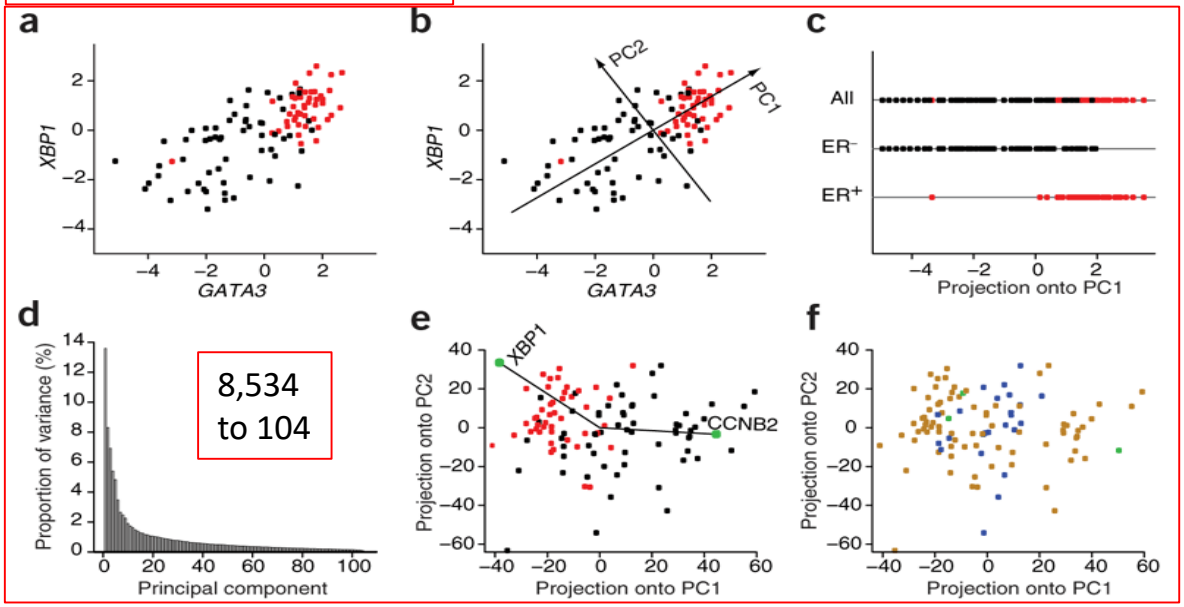
Choose the smallest number of eigenvectors or the principal components which explains the largest amount of datasets

Ringner, Nature Biotechnology, 26, 303-304 (2008)

Principal component analysis

Pick number of eigenvectors

This is where the user/analyst comes in!



Choose the smallest number of eigenvectors or the principal components which explains the largest amount of datasets

Ringner, Nature Biotechnology, 26, 303-304 (2008)

What will PCA tell a biologist about their cell sample?

- Which components have the highest correlation in the dataset?
- How many components are the key player in the effect observed during the experiment?
- Which components should be used as a reference for classification or clustering?

What will happen if you don’t do PCA?

- Will give bad clustering if correlated datapoints are used. (lose)
- Will take a much longer computational time for any clustering (or bad clustering). (lose - lose)
- Wrong conclusion of the results. (lose – lose – lose)

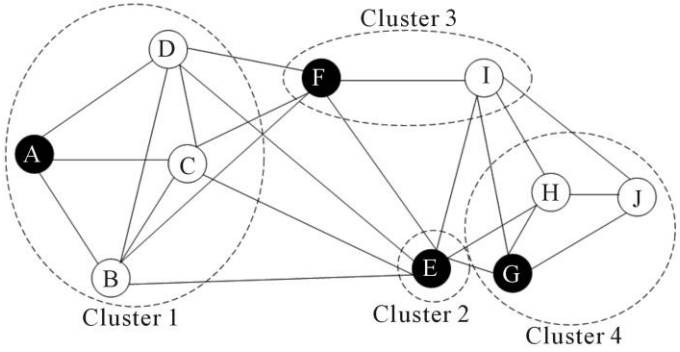
Clustering (WHY?)

- Identify groups of similar cells or samples into meaningful structures.
- Summarize the attributes of large datasets
- No 'a priori' assumptions or hypothesis
- Uses PCA as its input

- Optimization problem
  - **Minimize** within cluster distance
  - **Maximize** between cluster distance

Graph based clustering

- Create cluster by computing neighbor graph based on connectivity
- In scRNA-seq, nodes are cells and edges are cell-cell pairwise distance.
- Commonly used algorithm
  - Louvain algorithm – locate area with most density of neighbor graph



Graph based clustering (Louvain algorithm)

The algorithm does the clustering in two levels:

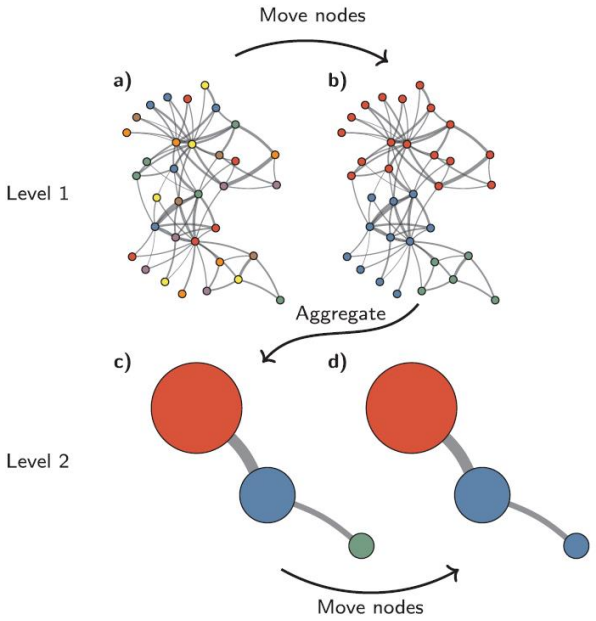
1) Moving of the nodes locally
2) Aggregating them based on density of edges, known as ‘Community’ .

The algorithm tries to maximize a function called ‘Modularity’ defined by:

Actual # of edges in a community – Expected # of edges

$$H = \frac{1}{2m}\sum_{c}(e_c - \gamma K_c^2/2m)$$

Traag et al., Scientific Reports 9, 5233 (2019)



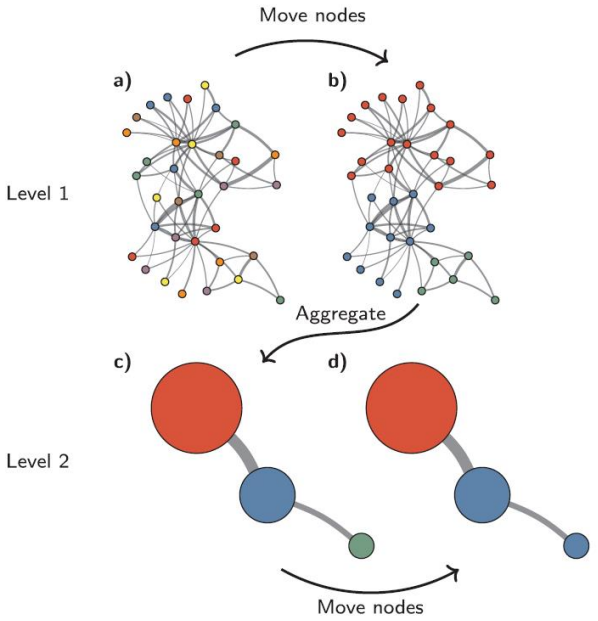
Graph based clustering (Louvain algorithm)

$$H = \frac{1}{2m}\sum_{c}(e_c - \gamma K_c^2/2m)$$

Actual # of edges – Expected # of edges

Expected number of edges is determined by:

- **Number of degrees of nodes (how many near neighbors)**
- **Total number of edges**
- **'Resolution' – a fudge factor that determines how many communities form. Increasing this will produce a larger number of smaller, more well defined clusters.**



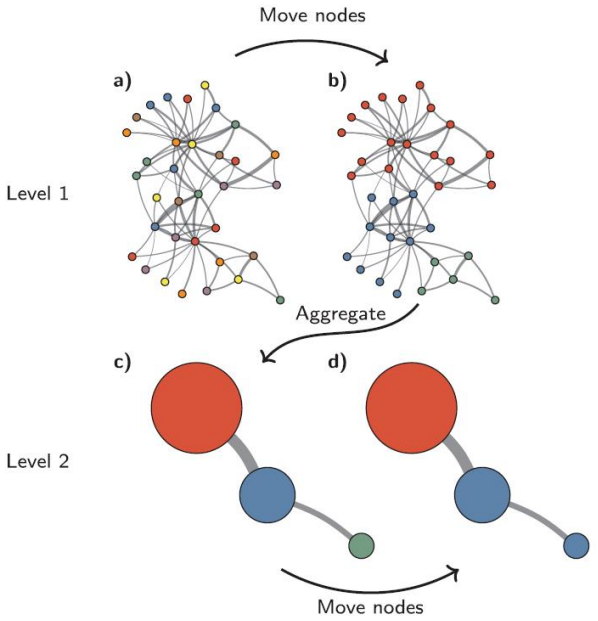
Graph based clustering (Louvain algorithm)

$$H = \frac{1}{2m}\sum_{c}(e_c - \gamma K_c^2/2m)$$

Some points of be aware of:

1) **<u>Resolution:</u> Problem of detecting small communities in a large network.**

2) **<u>Degeneracy:</u> Finding the global maximum and difficult to determine if the global maximum is truly more scientifically important than local maxima.**



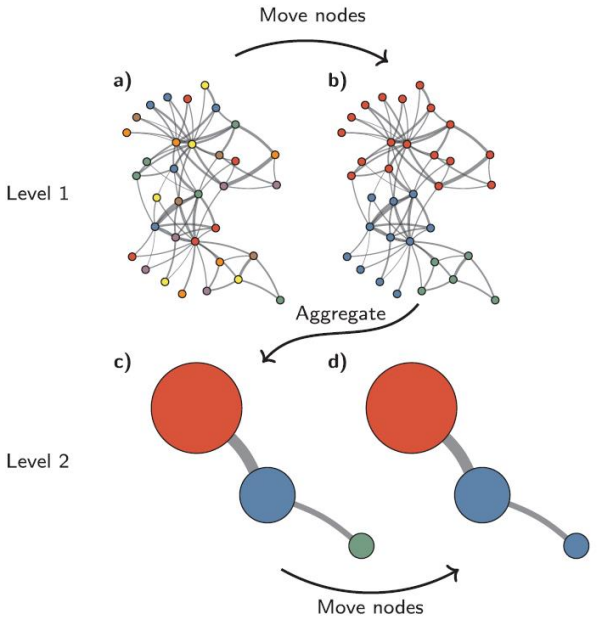
Graph based clustering (Louvain algorithm)

$$H = \frac{1}{2m}\sum_{c}(e_c - \gamma K_c^2/2m)$$

Some points of be aware of:

1) **<u>Resolution:</u> Problem of detecting small communities in a large network.**

2) **<u>Degeneracy:</u> Finding the global maximum and difficult to determine if the global maximum is truly more scientifically important than local maxima.**



<u>Solution 1:</u> Louvain with refinement

Iterative application of Louvain

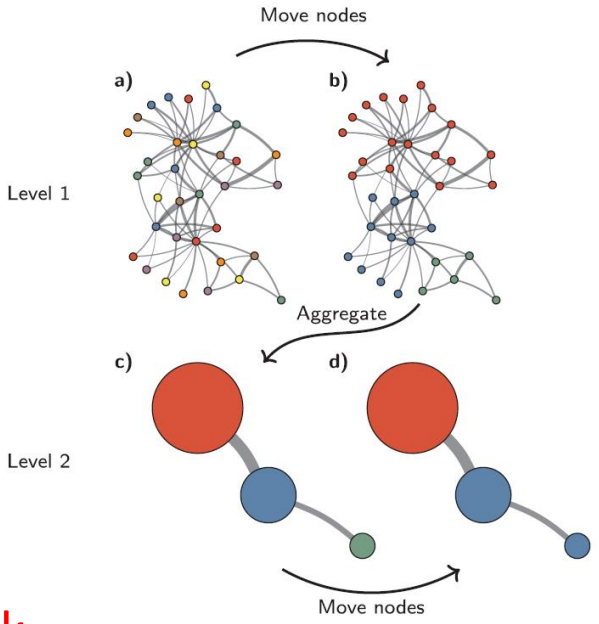
Graph based clustering (Louvain algorithm)

$$H = \frac{1}{2m}\sum_{c}(e_c - \gamma K_c^2/2m)$$

Some points of be aware of:

1) **<u>Resolution:</u> Problem of detecting small communities in a large network.**

2) **<u>Degeneracy:</u> Finding the global maximum and difficult to determine if the global maximum is truly more scientifically important than local maxima.**



<u>Solution 2:</u> Smart Local Moving (SLM)
Communities divided into local network.
Then apply displacement.

K-means clustering

```
Begin
    Assign each item a class in 1 to K (randomly)
    For 1 to max-iteration {
        For each class 1 to K {
            Calculate centroid (one of the "K means")
            Calculate distance from centroid to each item
        }
        Assign each item the class of the nearest centroid
        Exit if no items are re-assigned (convergence)
    }
End
```

K-means clustering

```
Begin
    Assign each item a class in 1 to K (randomly)
    For 1 to max-iteration {
        For each class 1 to K {
            Calculate centroid (one of the "K means")
            Calculate distance from centroid to each item
        }
        Assign each item the class of the nearest centroid
        Exit if no items are re-assigned (convergence)
    }
End
```

Distance method used is mostly Euclidean: $d(x_i, y_i) = \sqrt{\sum_i (x_i - y_i)^2}$

But there are others Spearman, Pearson….

Depends on the dataset. But when in doubt start with Euclidean!

Distance Measure	Equation	Time complexity	Advantages	Disadvantages	Applications
Euclidean Distance	$d_{euc} = \left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{\frac{1}{2}}$	O(n)	Very common, easy to compute and works well with datasets with compact or isolated clusters [27,31].	Sensitive to outliers [27,31].	*K*-means algorithm, Fuzzy *c*-means algorithm [38].
Average Distance	$d_{ave} = \left(\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2\right)^{\frac{1}{2}}$	O(n)	Better than Euclidean distance [35] at handling outliers.	Variables contribute independently to the measure of distance. Redundant values could dominate the similarity between data points [37].	*K*-means algorithm
Weighted Euclidean	$d_{we} = \left(\sum_{i=1}^{n}w_i(x_i - y_i)^2\right)^{\frac{1}{2}}$	O(n)	The weight matrix allows to increase the effect of more important data points than less important one [37].	Same as Average Distance.	Fuzzy *c*-means algorithm [38]
Chord	$d_{chord} = \left(2 - 2\frac{\sum_{i=1}^{n}x_i y_i}{\|x\|_2\|y\|_2}\right)^{\frac{1}{2}}$	O(3n)	Can work with un-normalized data [27].	It is not invariant to linear transformation [33].	Ecological resemblance detection [35].
Mahalanobis	$d_{mah} = \sqrt{(x-y)S^{-1}(x-y)^T}$	O(3n)	Mahalanobis is a data-driven measure that can ease the distance distortion caused by a linear combination of attributes [35].	It can be expensive in terms of computation [33]	Hyperellipsoidal clustering algorithm [30].
Cosine Measure	$\text{Cosine}(x,y) = \frac{\sum_{i=1}^{n}x_i y_i}{\|x\|_2\|y\|_2}$	O(3n)	Independent of vector length and invariant to rotation [33].	It is not invariant to linear transformation [33].	Mostly used in document similarity applications [28,33].
Manhattan	$d_{man} = \sum_{i=1}^{n}(x_i - y_i)$	O(n)	Is common and like other Minkowski-driven distances it works well with datasets with compact or isolated clusters [27].	Sensitive to the outliers. [27,31]	*K*-means algorithm
Mean Character Difference	$d_{MCD} = \frac{1}{n}\sum_{i=1}^{n}\|x_i - y_i\|$	O(n)	*Results in accurate outcomes using the K-medoids algorithm.	*Low accuracy for high-dimensional datasets using K-means.	Partitioning and hierarchical clustering algorithms.
Index of Association	$d_{IOA} = \frac{1}{n}\sum_{i=1}^{n}\left\|\frac{x_i}{\sum_{i=1}^{n}x_i} - \frac{y_i}{\sum_{i=1}^{n}y_i}\right\|$	O(3n)	-	*Low accuracy using K-means and K-medoids algorithms.	Partitioning and hierarchical clustering algorithms.
Canberra Metric	$d_{canb} = \sum_{i=1}^{n}\frac{\|x_i - y_i\|}{(x_i + y_i)}$	O(n)	*Results in accurate outcomes for high-dimensional datasets using the K-medoids algorithm.	-	Partitioning and hierarchical clustering algorithms.
Czekanowski Coefficient	$d_{czekan} = 1 - \frac{2\sum_{i=1}^{n}min(x_i, y_i)}{\sum_{i=1}^{n}(x_i + y_i)}$	O(2n)	*Results in accurate outcomes for medium-dimensional datasets using the K-means algorithm.	-	Partitioning and hierarchical clustering algorithms.
Coefficient of Divergence	$d_{canb} = \left(\frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - y_i}{x_i + y_i}\right)^2\right)^{\frac{1}{2}}$	O(n)	*Results in accurate outcomes using the K-means algorithm.	-	Partitioning and hierarchical clustering algorithms.
Pearson coefficient	$Pearson(x,y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}}$	O(2n)	*Results in accurate outcomes using the hierarchical single-link algorithm for high dimensional datasets.	-	Partitioning and hierarchical clustering algorithms.

*Points marked by asterisk are compiled based on this article's experimental results.

doi:10.1371/journal.pone.0144059.t001

Distance measure

t-SNE (What is it?)

- A popular method for exploring high-dimensional data is t-SNE (t-Distributed Stochastic Neighbor Embedding)

No linear distance calculation.

Minimizes the divergence between two distributions:

distribution that measures pairwise similarities of the input objects

distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding

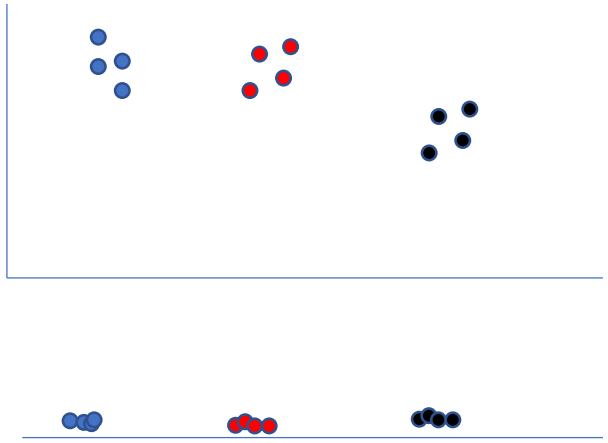
van der Maaten, Journal of Machine Learning Research, 9, 2579–2605 (2008)

t-SNE (What is it?)

Minimizes the divergence between two distributions:

distribution that measures pairwise similarities of the input objects

distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding



Density at a high dimension

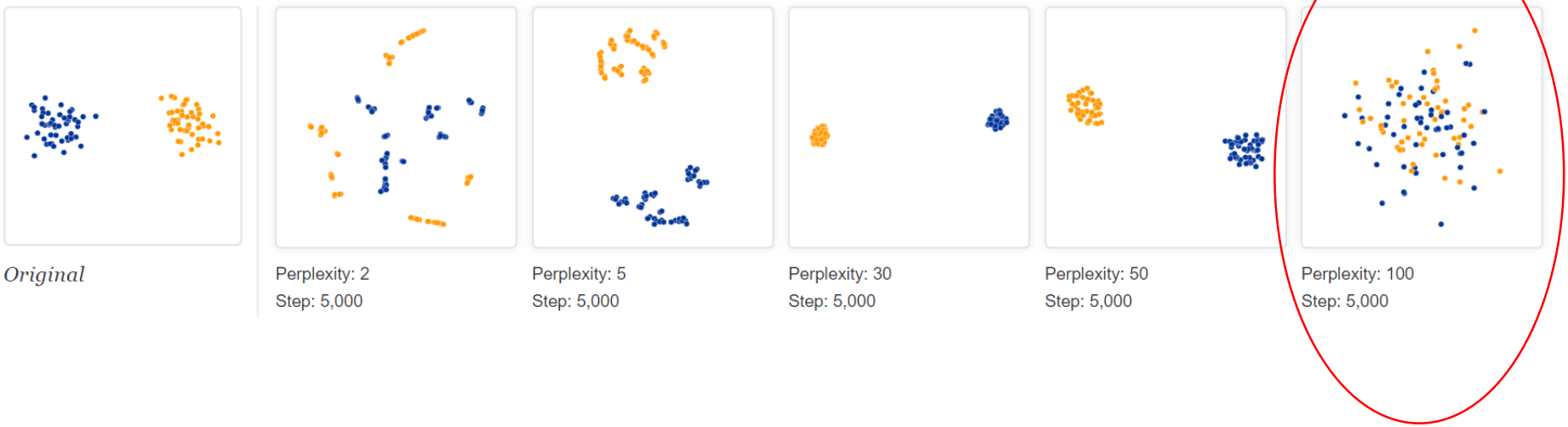
Density at a low dimension

t-SNE (What is it?)

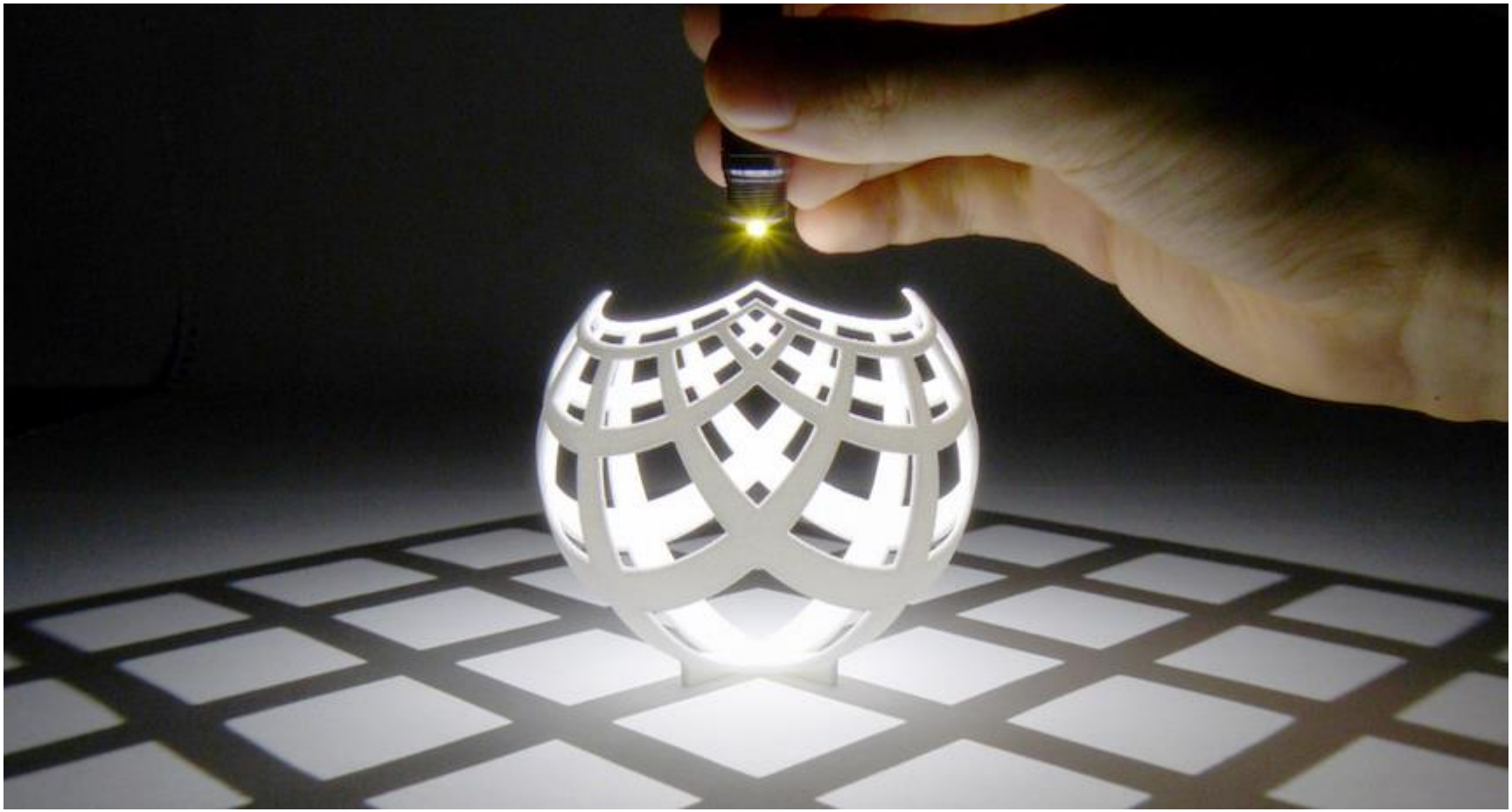
Perplexity: The measure of number of nearest neighbor a point has

This determines how the clustering would look. Typically, lies between 5 and 50.

Too high!



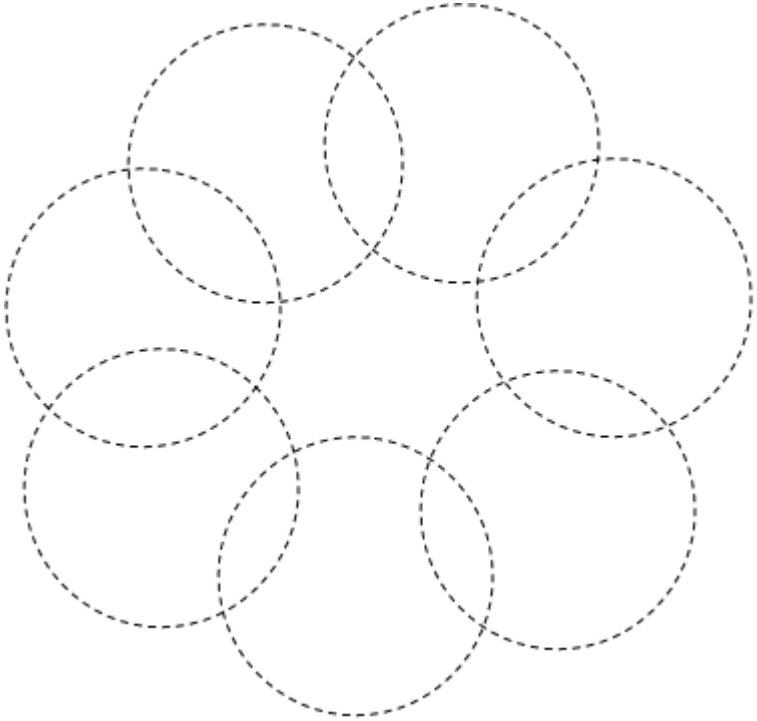
Projection in view!



UMAP (What is it?)

UMAP is a dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis.

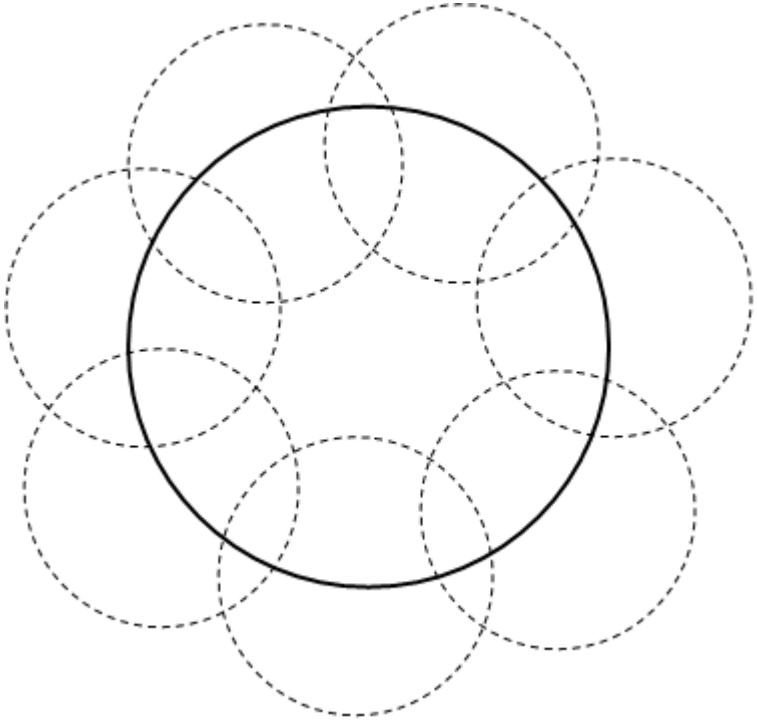
Let’s try to draw some something!



UMAP (What is it?)

UMAP is a dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis.

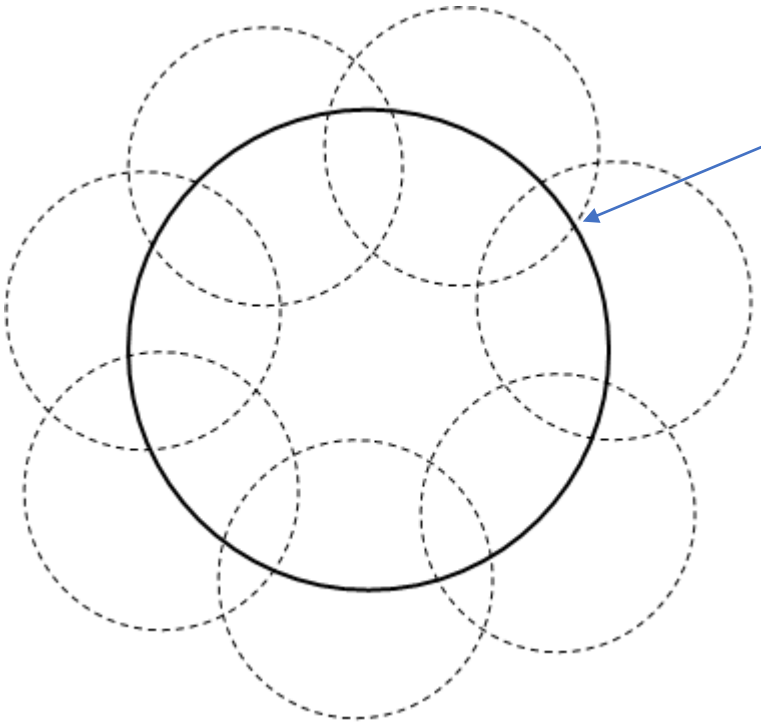
Let's try to draw some something!



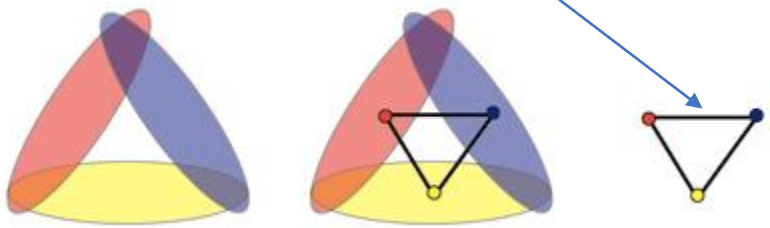
UMAP (What is it?)

UMAP is a dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis.

Let's try to draw some something!

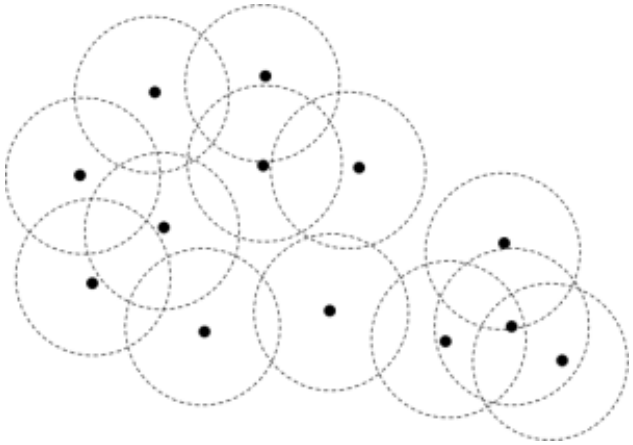


Nerve of the set!



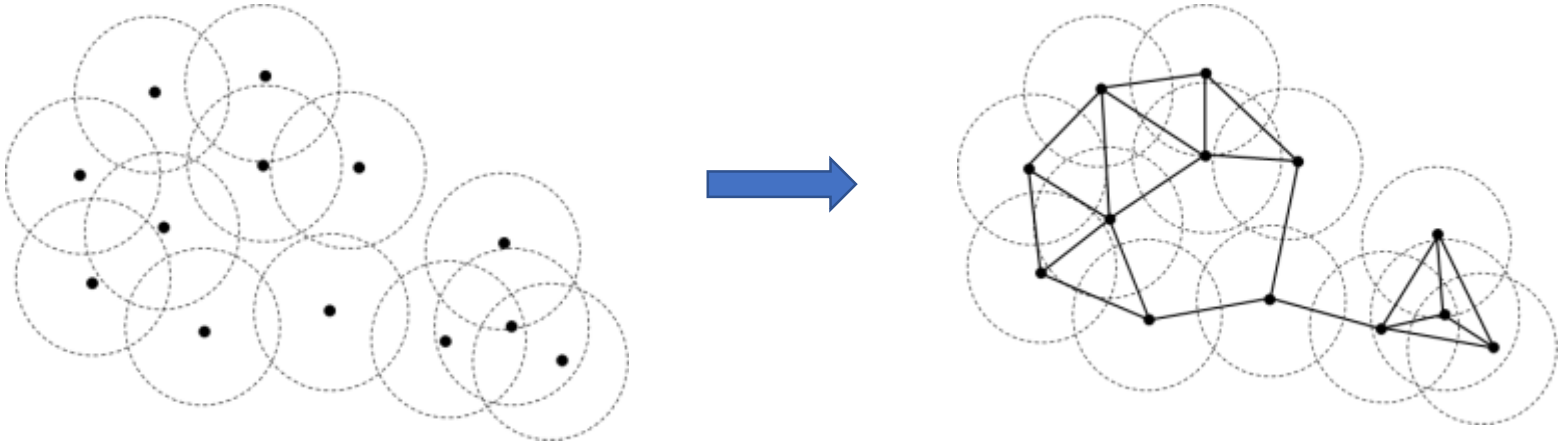
UMAP (What is it?)

UMAP is a dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis.



UMAP (What is it?)

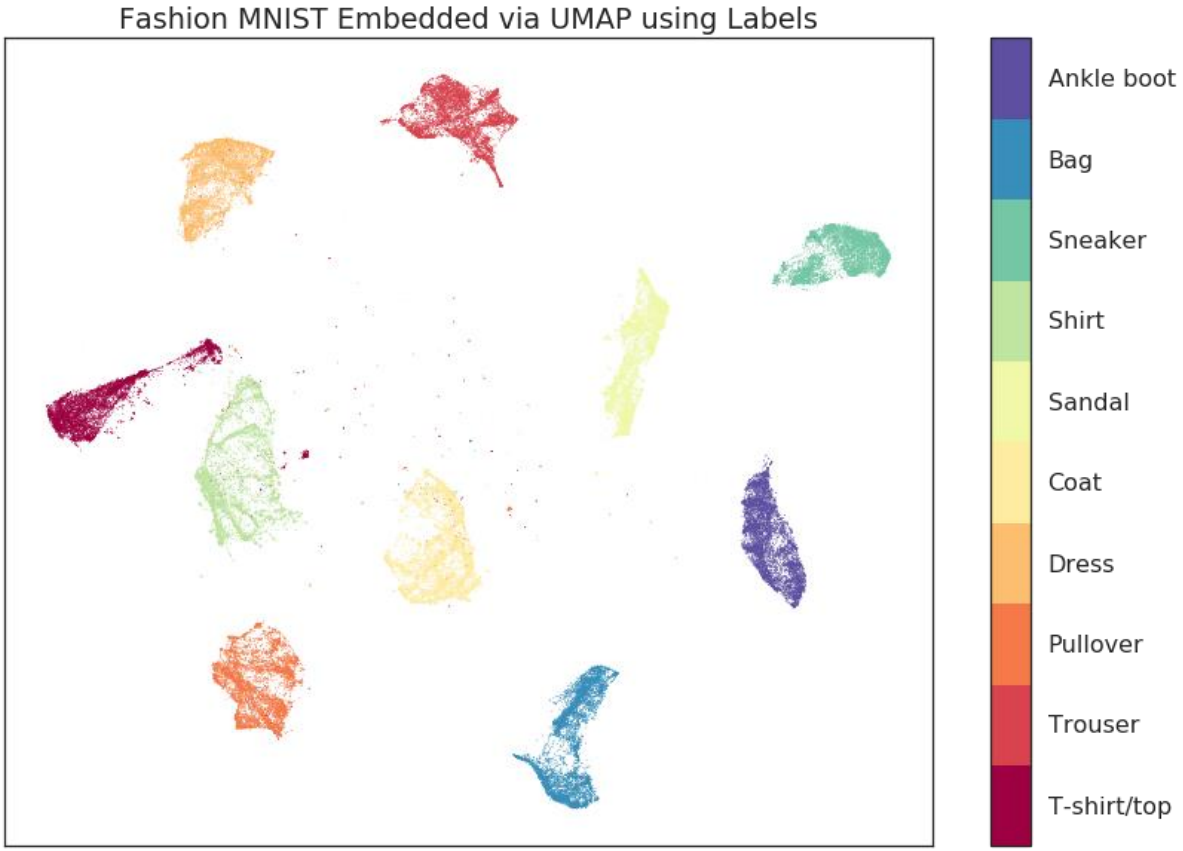
UMAP is a dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis.



Key variables for UMAP: number of neighbors, distance metric, visualization dimension

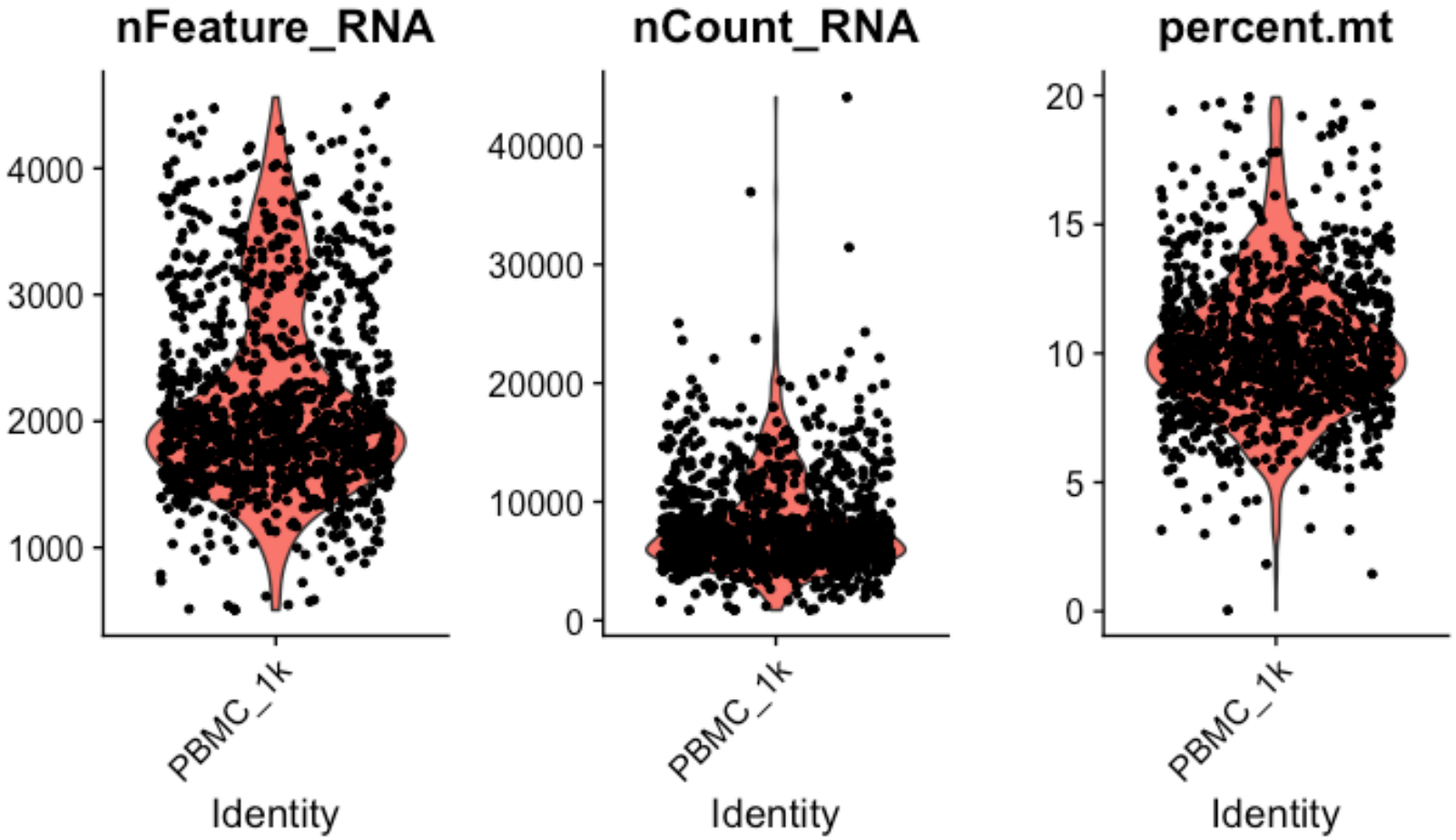
UMAP (What is it?)

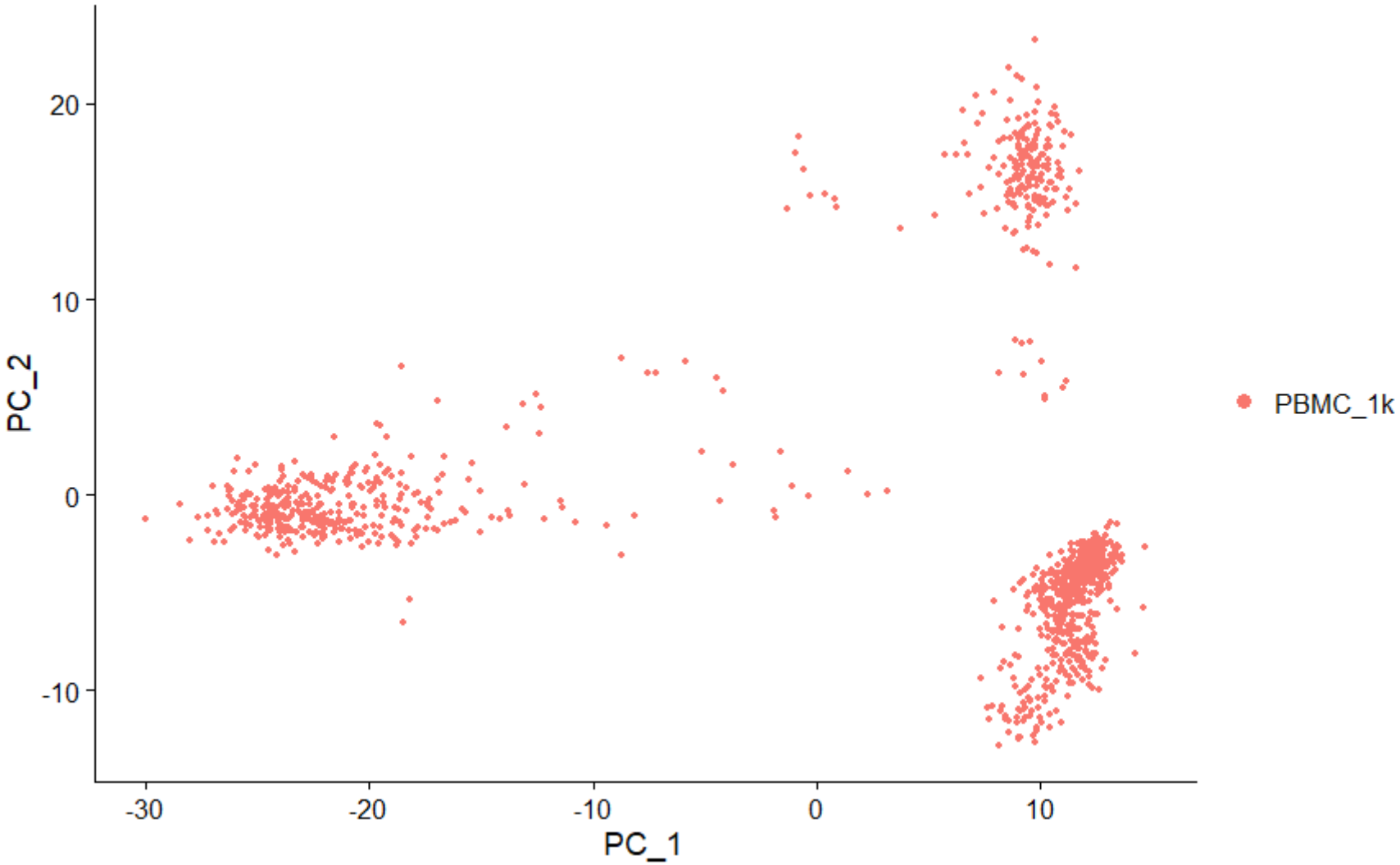
UMAP is a dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis.

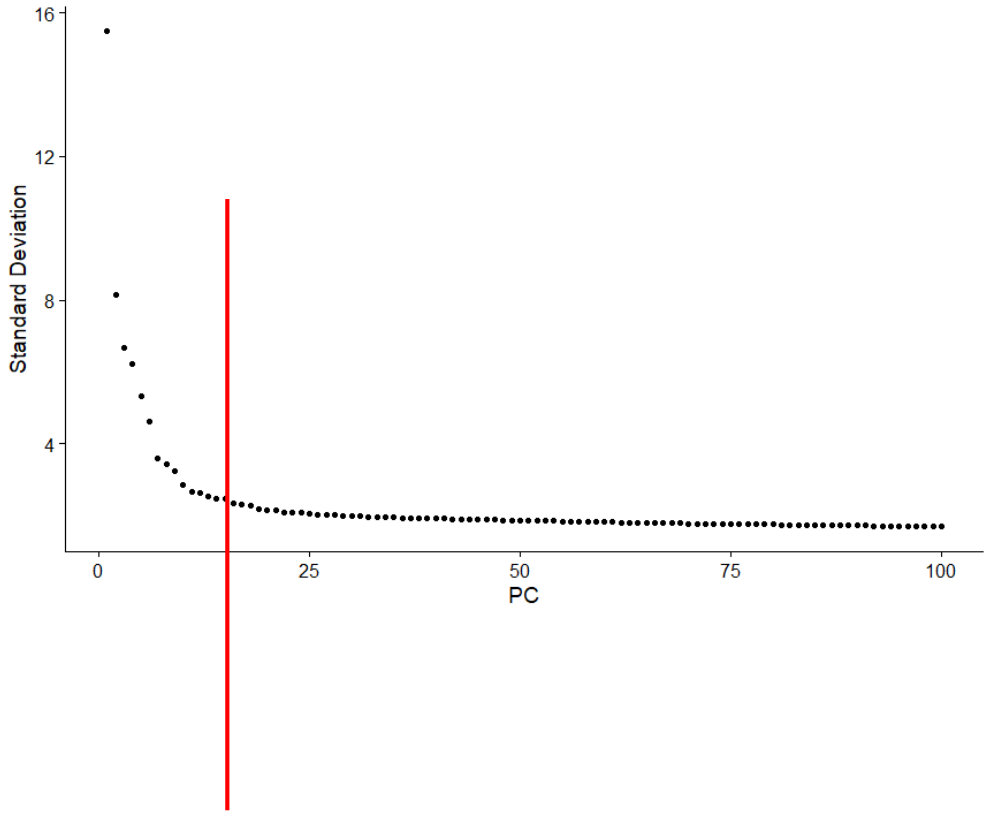


Local and Global structure

Sample datasets

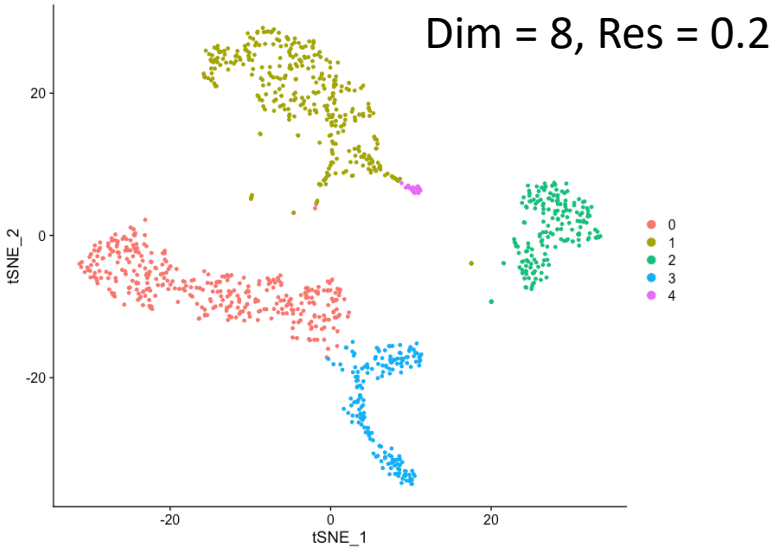


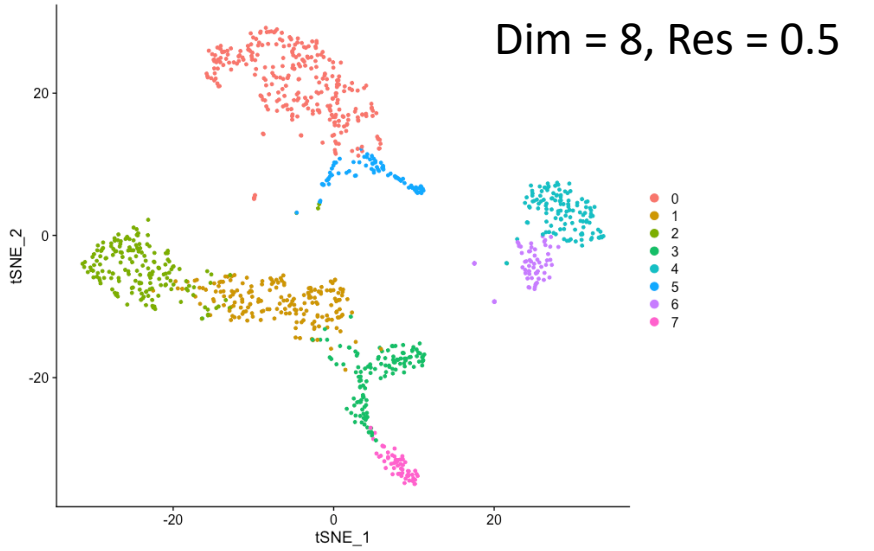

20
10
0
-10
PC_2
-30
-20
-10
0
10
PC_1
PBMC_1k

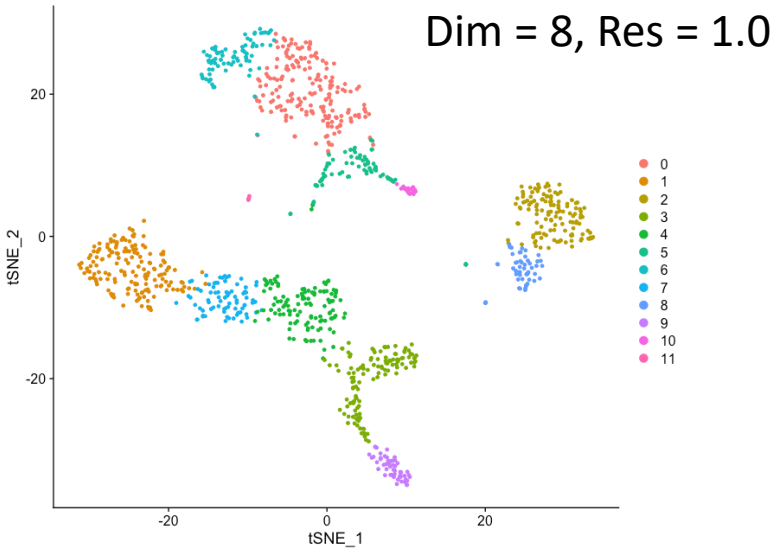


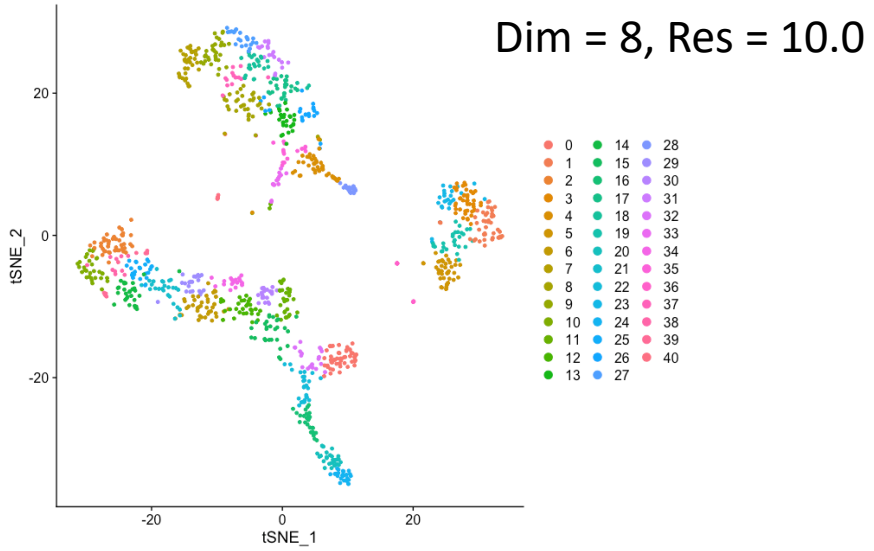
Cutoff for Principal components (measure of dimensions to consider)

t- SNE


Dim = 8, Res = 0.2
tSNE_2
tSNE_1
20
0
-20
-20
0
20
0
1
2
3
4

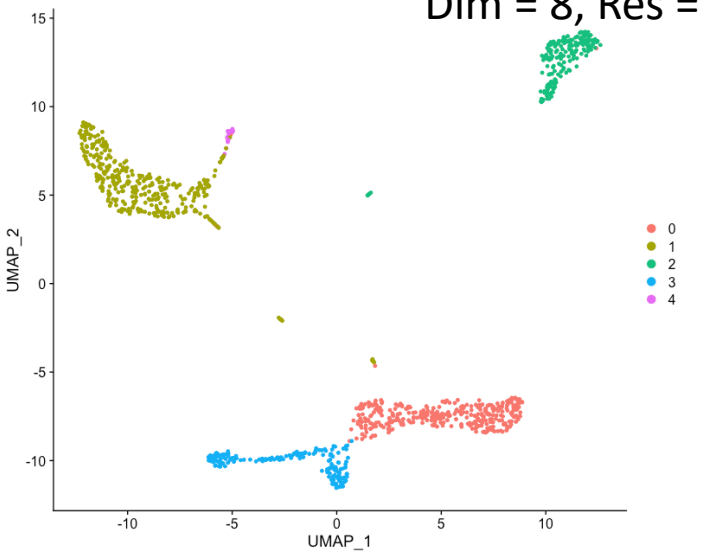

Dim = 8, Res = 0.5
tSNE_2
tSNE_1
20
0
-20
-20
0
20
0
1
2
3
4
5
6
7


Dim = 8, Res = 1.0
tSNE_2
tSNE_1
20
0
-20
-20
0
20
0
1
2
3
4
5
6
7
8
9
10
11

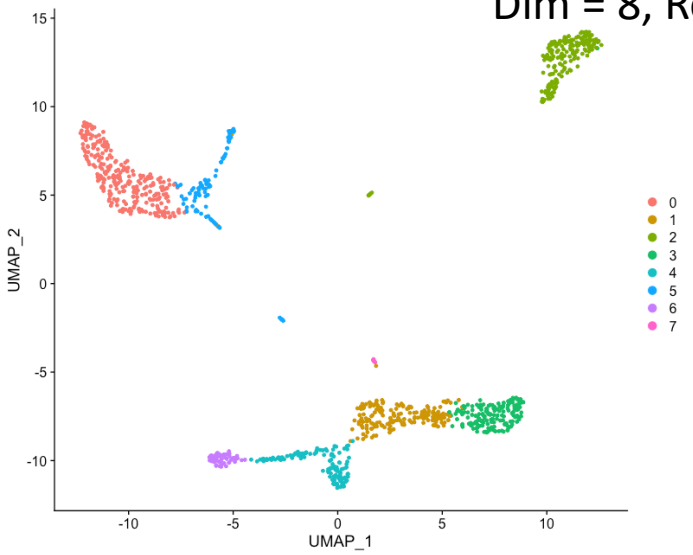

Dim = 8, Res = 10.0
tSNE_2
tSNE_1
20
0
-20
-20
0
20
0 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40

UMAP

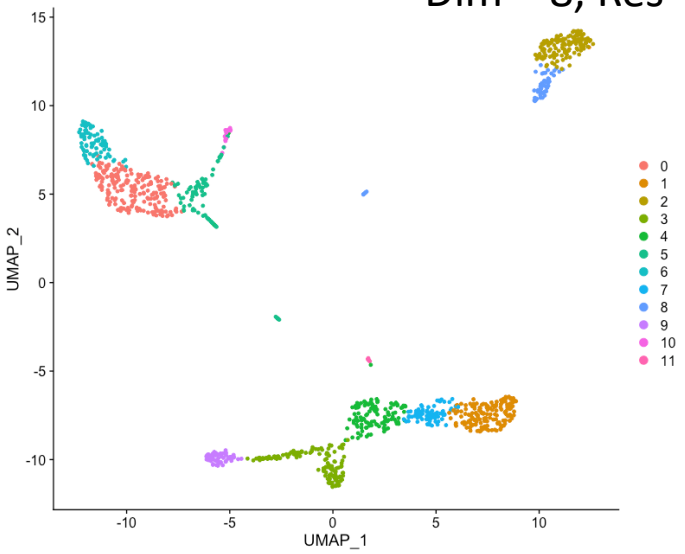
Dim = 8, Res = 0.2



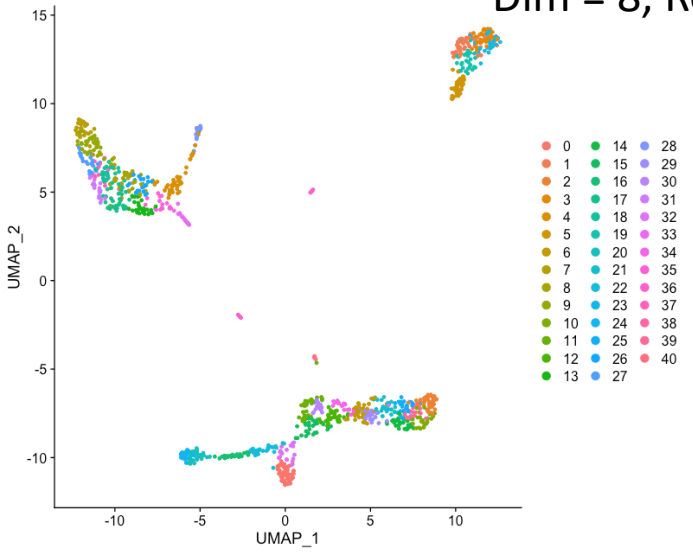
Dim = 8, Res = 0.5



Dim = 8, Res = 1.0

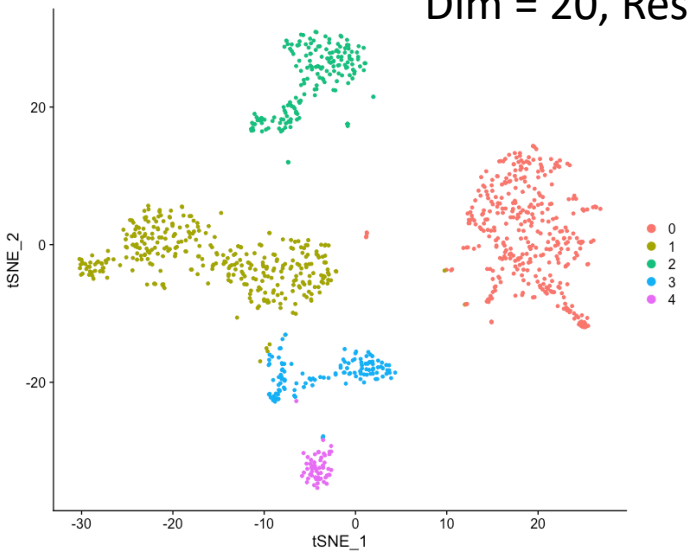


Dim = 8, Res = 10.0

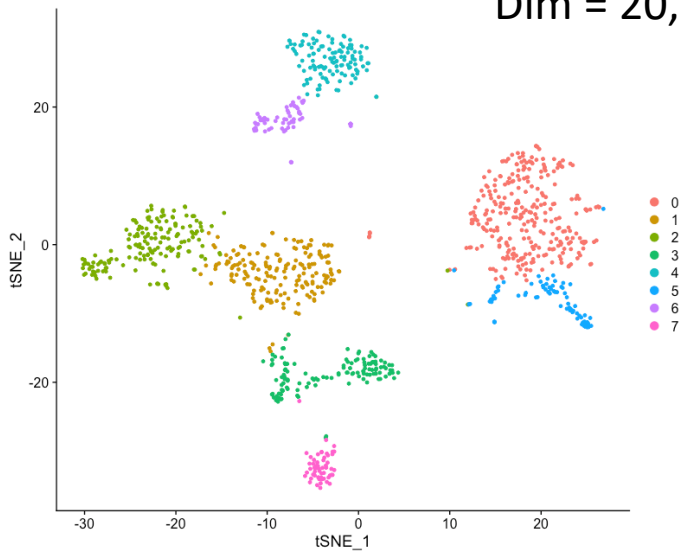


t-SNE

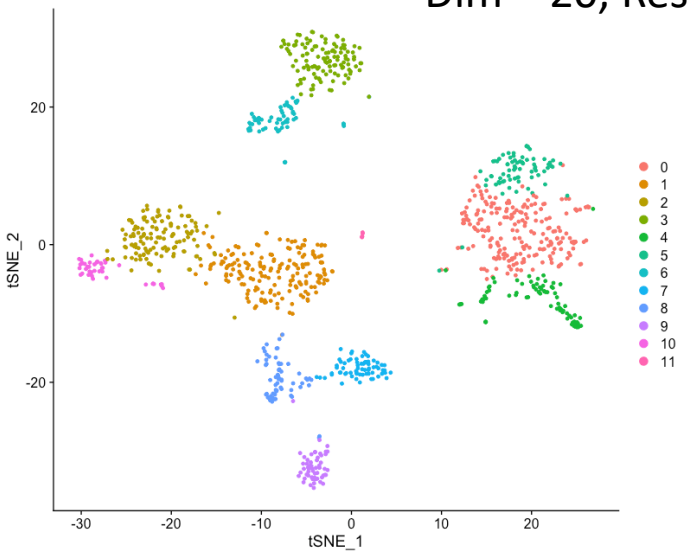
Dim = 20, Res = 0.2



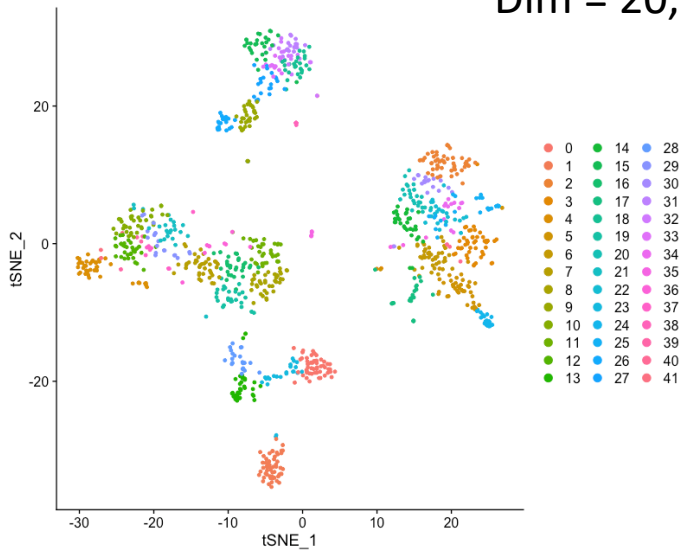
Dim = 20, Res = 0.5



Dim = 20, Res = 1.0

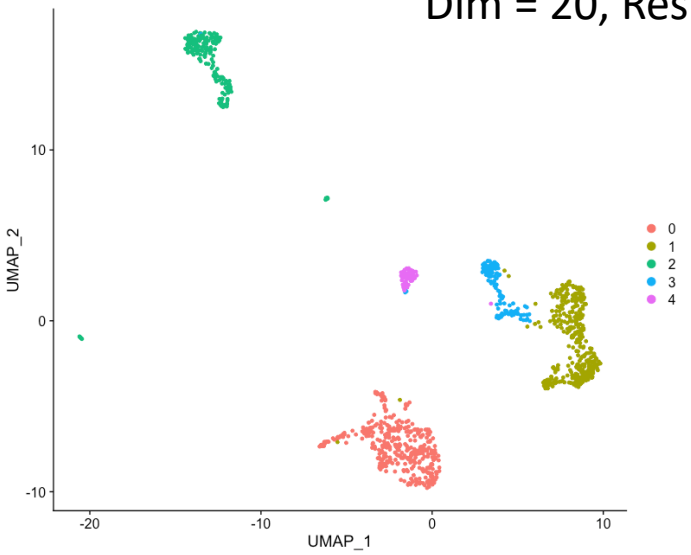


Dim = 20, Res = 10.0

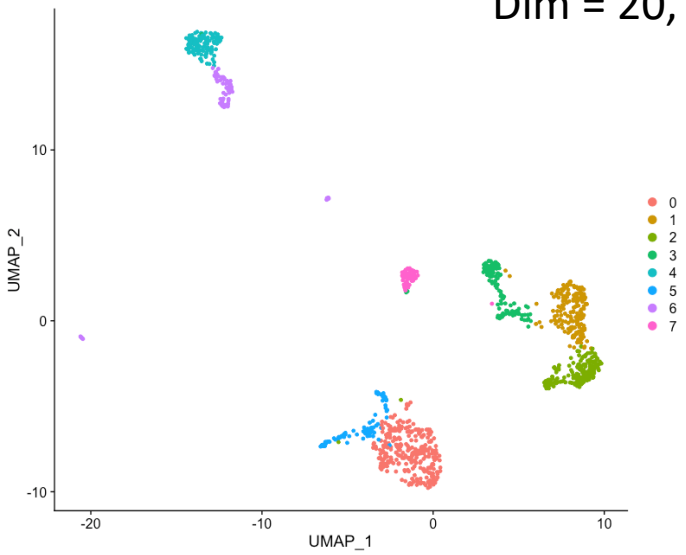


UMAP

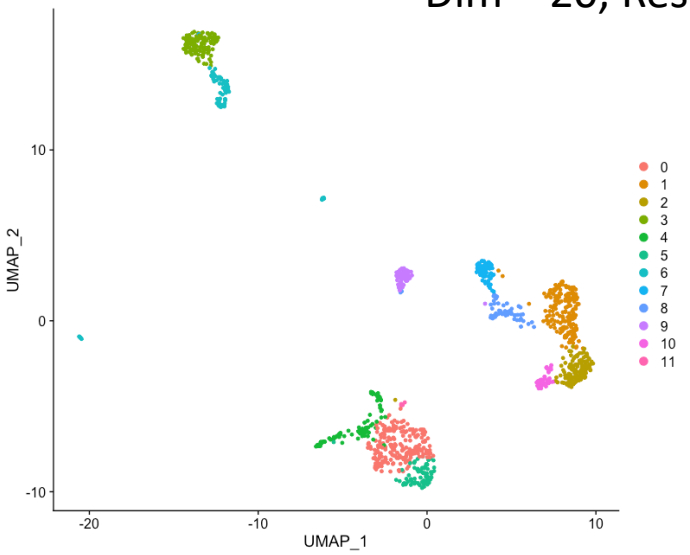
Dim = 20, Res = 0.2



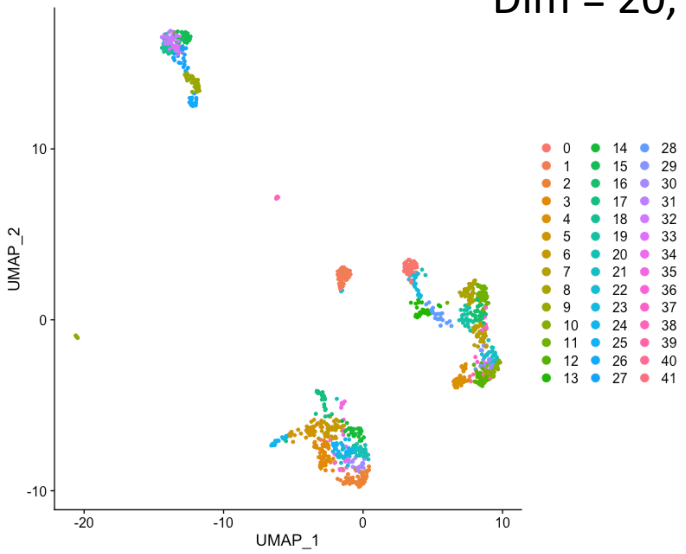
Dim = 20, Res = 0.5

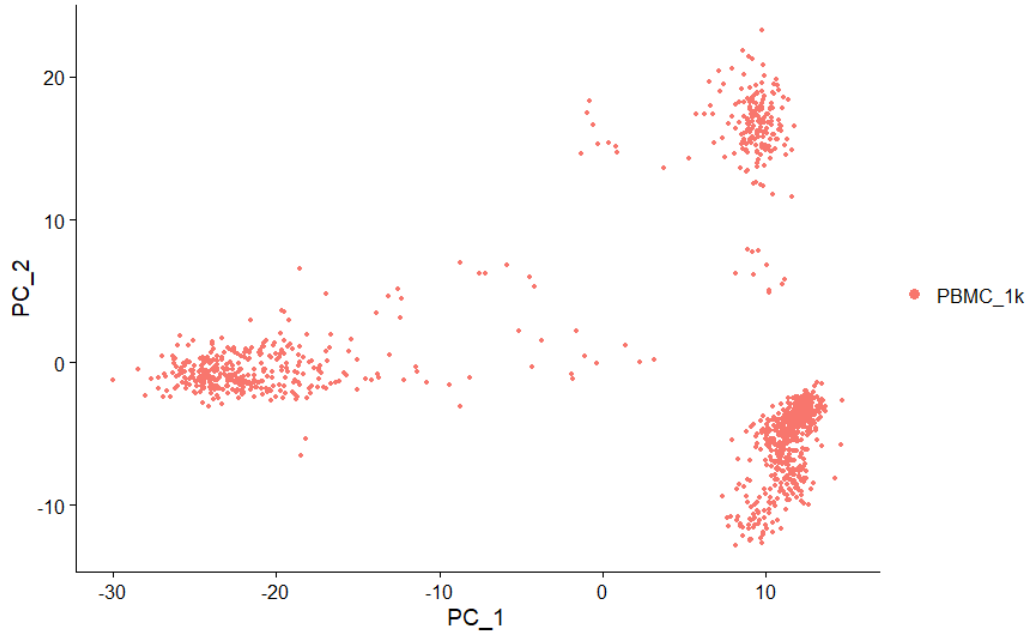


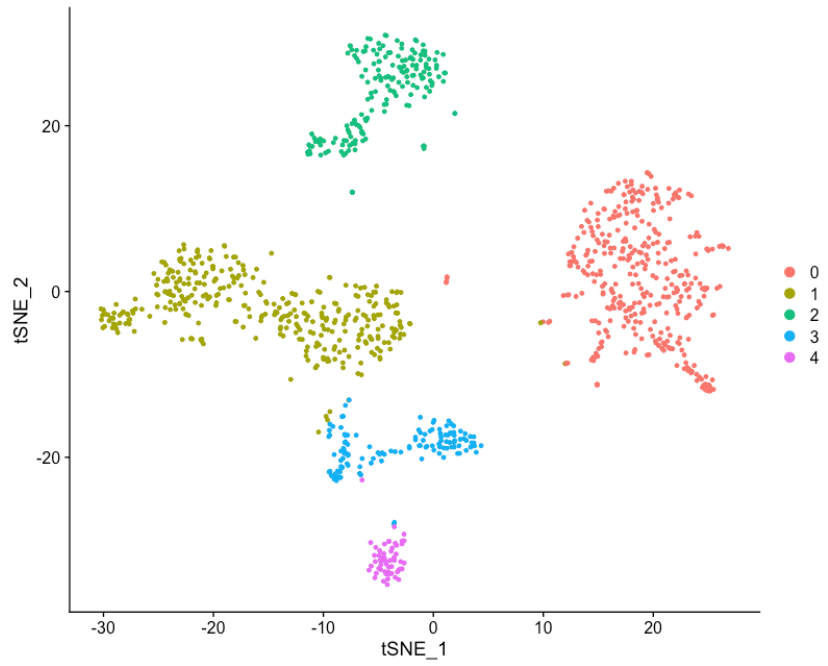
Dim = 20, Res = 1.0

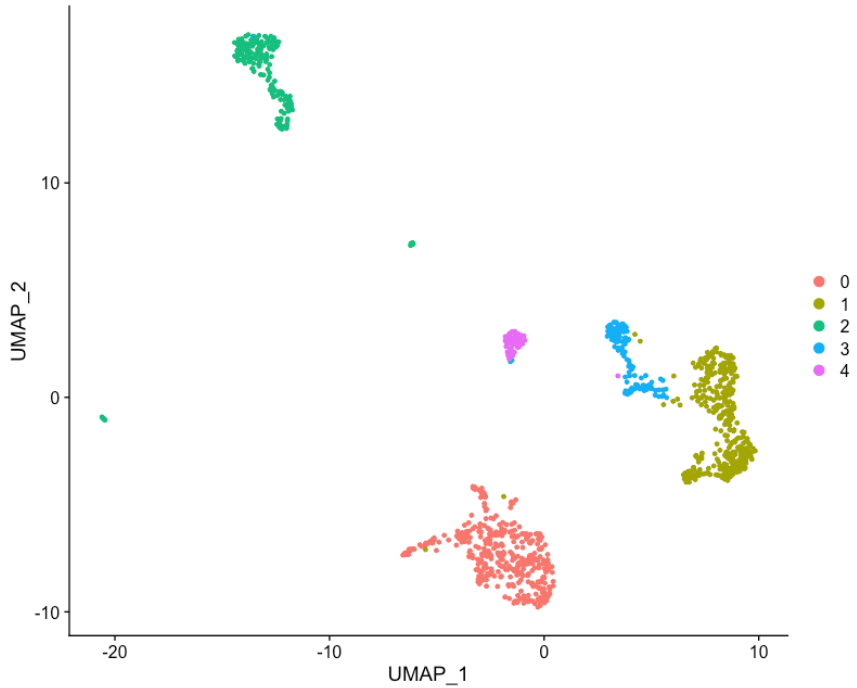


Dim = 20, Res = 10.0




PC_2
PC_1
PBMC_1k


tSNE_2
tSNE_1
0
1
2
3
4


UMAP_2
UMAP_1
0
1
2
3
4

Bootstrapping

How confident can you be about your clustering?

For sanity check, always try to do bootstrapping.

- Take random sets of cells from the same dataset
- Perform the same clustering on each of these sets

Currently not present as a default option in commercial packages, but can be implemented outside.