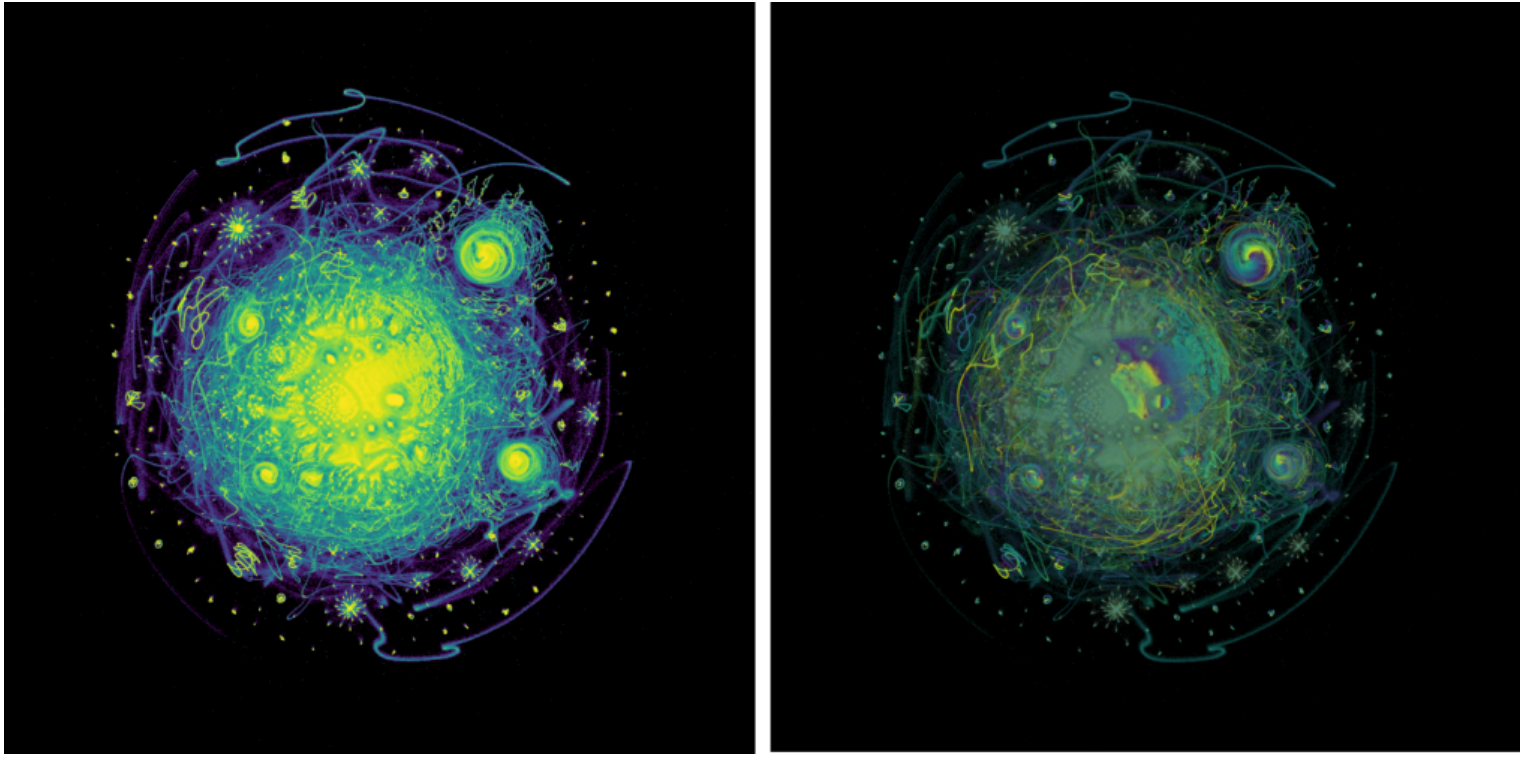
Dimension Reduction for Single Cell Data Analysis

Nikolay Oskolkov, Lund University, NBIS SciLifeLab, Sweden

scRNAseq course, 13.02.2024



@NikolayOskolkov

GitHub

github.com/NikolayOskolkov

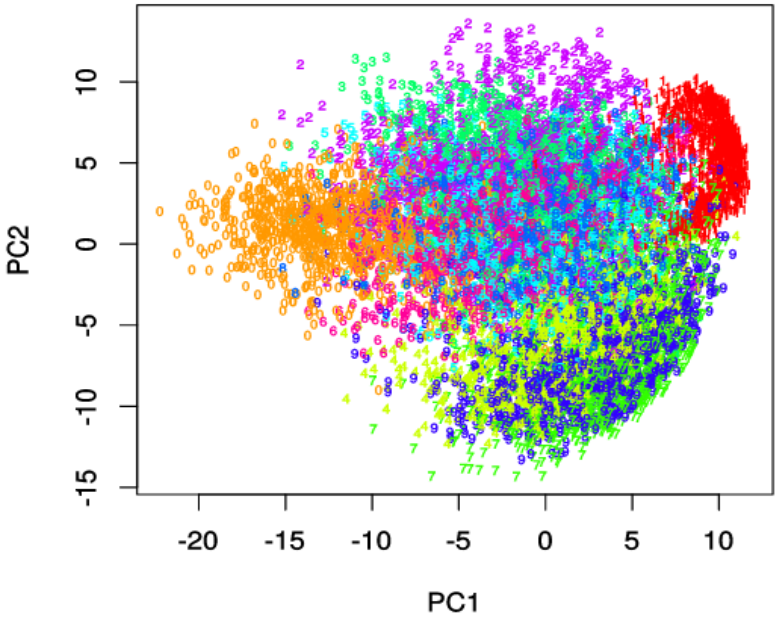
Image adapted from McInnes et al. 2018

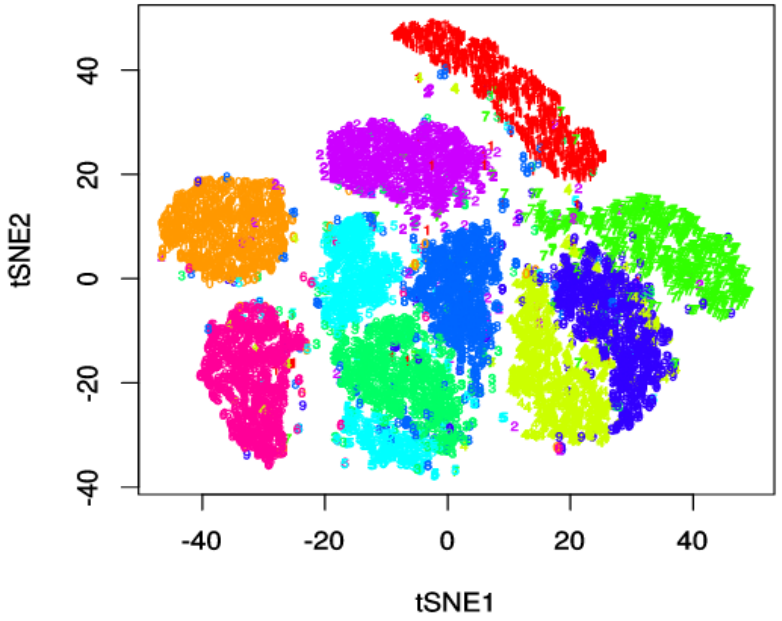
Dimensionality reduction
is also supposed to ... reduce dimensions

Dimension reduction: more than visualization

PCA PLOT WITH PRCOMP

tSNE MNIST





The goal of dimension reduction is not only visualization but also reducing dimensions

Biological data are usually high dimensional

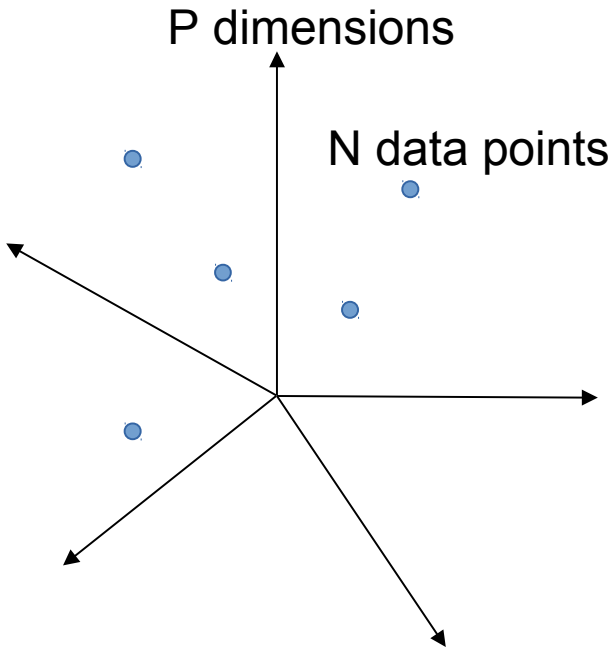
Statistical observations:
e.g. samples, cells etc.

N

P

0	3	1	0	2	3	8	1	1	3
1	1	0	0	7	1	2	2	3	3
1	2	2	0	0	6	7	1	2	2
1	2	3	10	0	4	6	1	0	5
3	2	2	1	4	3	2	1	6	0
7	4	4	5	3	9	6	1	6	1
7	1	1	5	2	8	9	1	3	6
5	0	1	6	2	0	0	0	1	5
1	6	3	3	4	6	2	0	1	1
1	2	2	4	1	1	3	0	8	2

Features: genes, proteins, microbes, metabolites etc.



High Dimensional Data:
P >> N

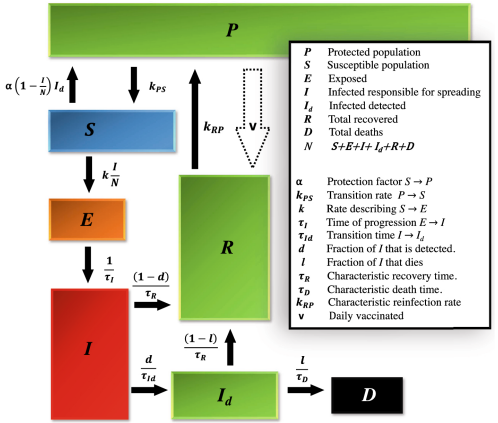
For a robust statistical analysis, one should properly “sample” the P-dimensional space, hence large sample size is required, N >> P

Types of Data Analysis

P is the number of features (genes, proteins, genetic variants etc.)
N is the number of observations (samples, cells, nucleotides etc.)

Biology / Biomedicine

Mathematical modeling



N ⟶ 0

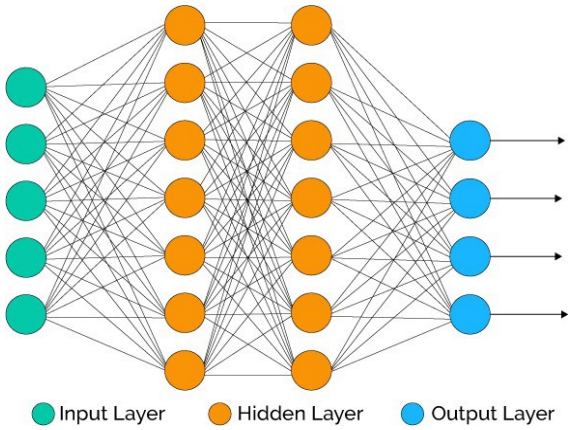
Bayesianism

N << P

Frequentism

N ≈ P

Machine Learning



N >> P

Hypothesis Driven ⟶ **Data Driven**

Amount of Data

$$Y = \alpha + \beta X$$

$$\beta = \left(X^T X\right)^{-1} X^T Y$$

$$\left(X^T X\right)^{-1} \sim \frac{1}{\det\left(X^T X\right)} \cdots \to \infty, \quad n << p$$

The Curse of Dimensionality

We need to reduce dimensions to overcome the Curse of Dimensionality!

Literature on the Curse of Dimensionality

POINTS OF SIGNIFICANCE

The curse(s) of dimensionality

There is such a thing as too much of a good thing.

Naomi Altman and Martin Krzywinski

We generally think that more information is better than less. However, in the 'big data' era, the sheer number of variables that can be collected from a single sample can be problematic. This embarrassment of riches is called the 'curse of dimensionality'[1] (CoD) and manifests itself in a variety of ways. This month, we discuss four important problems of dimensionality as it applies to data sparsity[1,2], multicollinearity[3], multiple testing[4] and overfitting[5]. These effects are amplified by poor data quality, which may increase with the number of variables.

Throughout, we use n to indicate the sample size from the population of interest and p to indicate the number of observed variables, some of which may have missing values for some samples. For example, we may have $n = 1{,}000$ subjects and $p = 200{,}000$ single-nucleotide polymorphisms (SNPs).

First, as the dimensionality p increases, the 'volume' that the samples may occupy

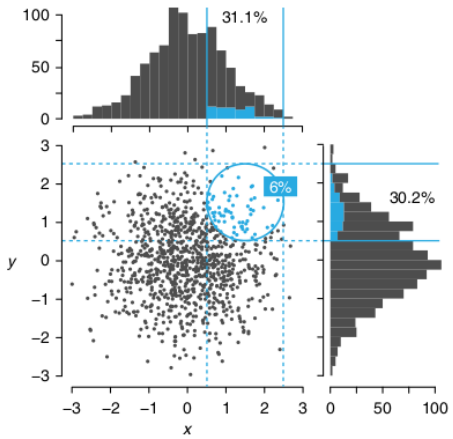


Fig. 1 | Data tend to be sparse in higher dimensions. Among 1,000 (x, y) points in which both x and y are normally distributed with a mean of 0 and s.d. $\sigma = 1$, only 6% fall within σ of $(x, y) = (1.5, 1.5)$ (blue circle). However, when the data are projected into a lower dimension—shown by histograms—about 30% of the points (all bins

A and 100 to have the minor allele a. If we tabulate on two SNPs, A and B, we will expect only ten samples to exhibit both minor alleles with genotype ab. With SNPs A, B and C, we expect only one sample to have genotype abc, and with four or more SNPs, we expect empty cells in our table. We need a much larger sample size to observe samples with all the possible genotypes. As p increases, we may quickly find that there are no samples with similar values of a predictor.

Even with just five SNPs, our ability to predict and classify the samples is impeded because of the small number of subjects that have similar genotypes. In situations where there are many gene variants, this effect is exacerbated, and it may be very difficult to find affected subjects with similar genotypes and hence to predict or classify on the basis of genetic similarity.

If we treat the distance between points (e.g., Euclidian distance) as a measure of similarity, then we interpret greater distance

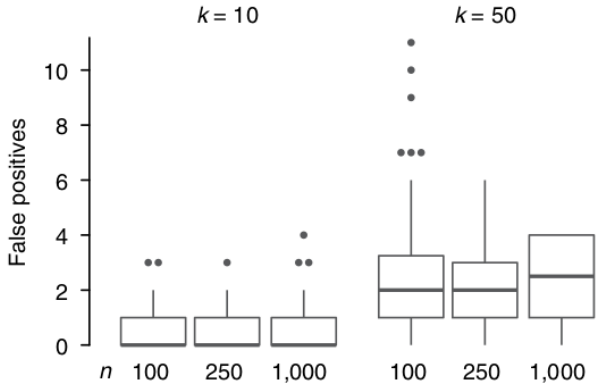


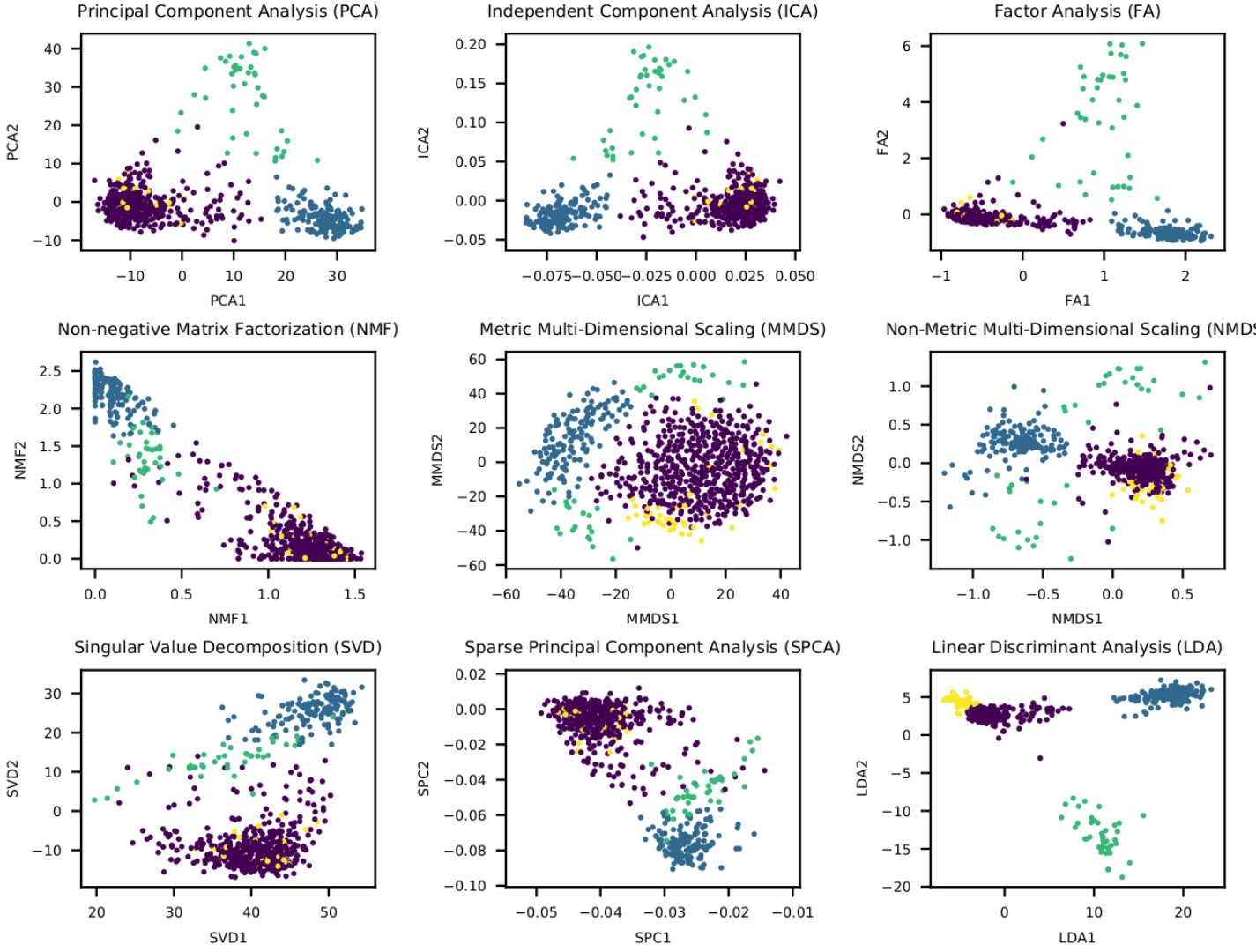
Fig. 3 | The number of false positives increases with each additional predictor. The box plots show the number of false positive regression-fit P values (tested at $\alpha = 0.05$) of 100 simulated multiple regression fits on various numbers of samples ($n = 100$, 250 and 1,000) in the presence of one true predictor and $k = 10$ and 50 extraneous uncorrelated predictors. Box plots show means (black center lines), 25th and 75th percentiles (box edges), and minimum and maximum values (whiskers). Outliers (dots) are jittered.

Correcting for multiple testing does not solve the problem of too many false-positive hits

Altman N, Krzywinski M. The curse(s) of dimensionality. Nat Methods. 2018 Jun;15(6):399-400. doi: 10.1038/s41592-018-0019-x. PMID: 29855577.

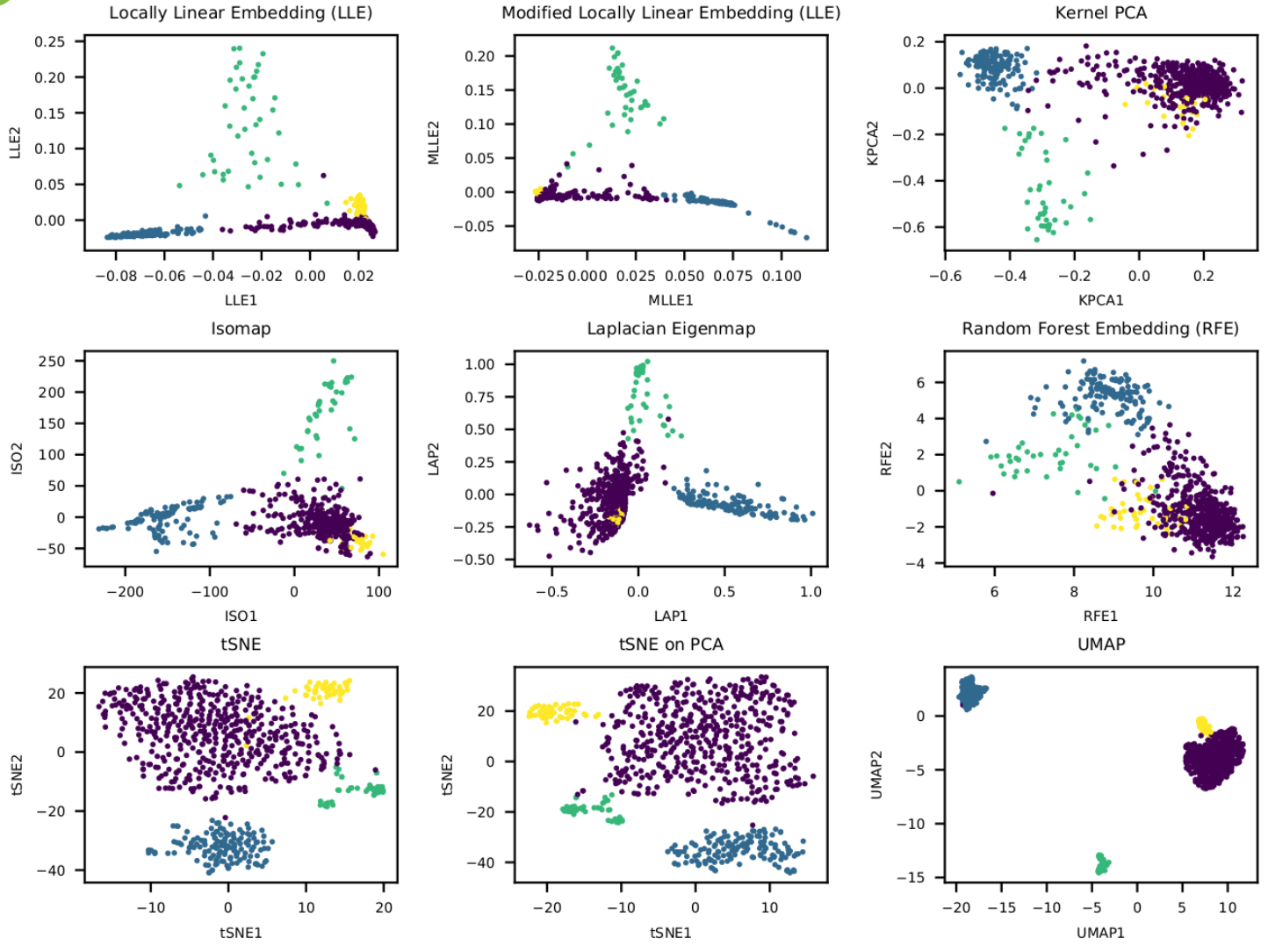
Dimension reduction techniques: linear vs. non-linear

Linear dimensionality reduction



M.Bartoschek, N. Oskolkov et al., Nature Communications 2018

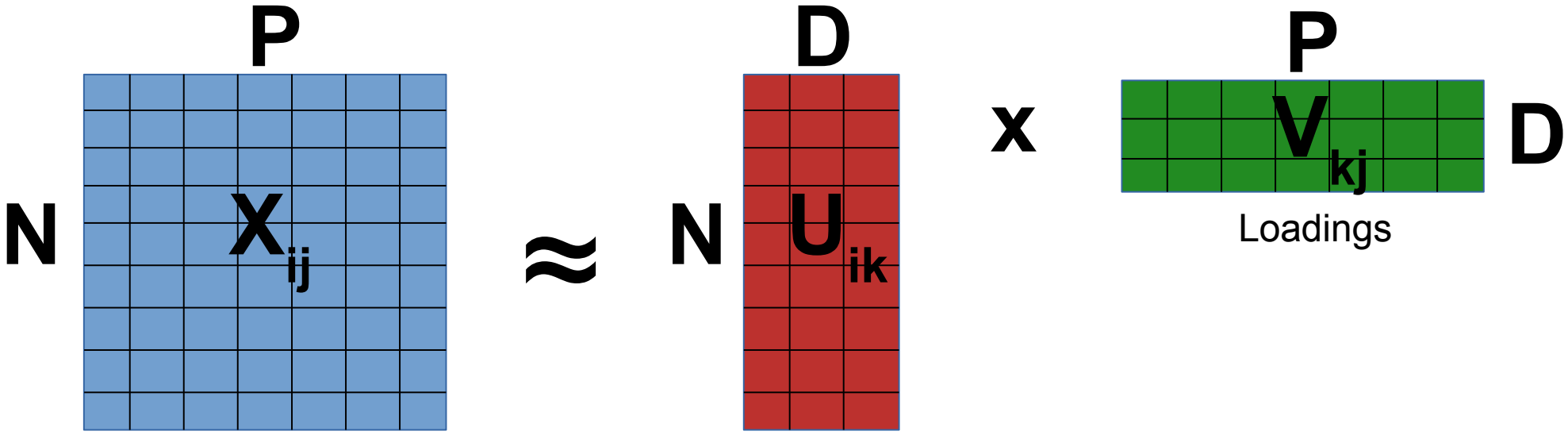
Non-linear dimensionality reduction



M.Bartoschek, N. Oskolkov et al., Nature Communications 2018

Linear dimension reduction: matrix factorization

$$\mathbf{X_{ij}} \approx \mathbf{U_{ik}}\mathbf{V_{kj}}$$



$$\text{Loss} = \sum_{i=1}^{N}\sum_{j=1}^{P}(\mathbf{X_{ij}} - \mathbf{U_{ik}}\mathbf{V_{kj}})^2$$

PCA dimension reduction algorithm

Coding in R:

```
data_centered <- scale(data, center = TRUE, scale = FALSE)

covariance <- t(data_centered) %*% data_centered

eig <- eigen(covariance)

plot(eig$vectors[,1:2]);
```

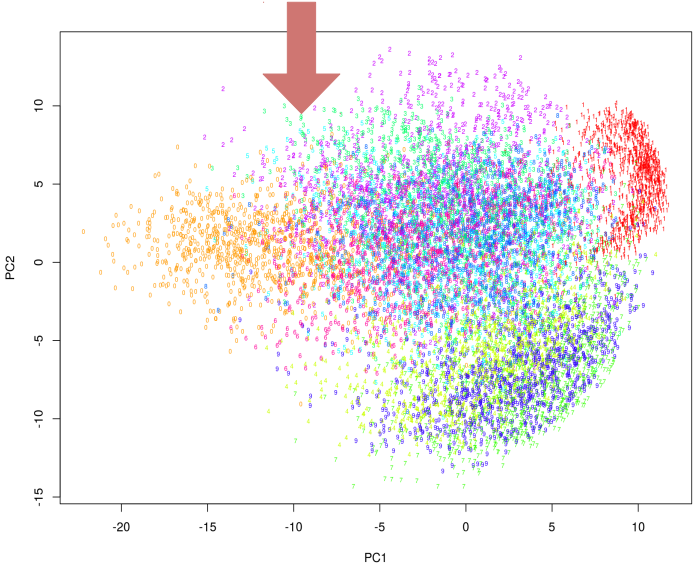
```
barplot(eig$values / sum(eig$values))
```

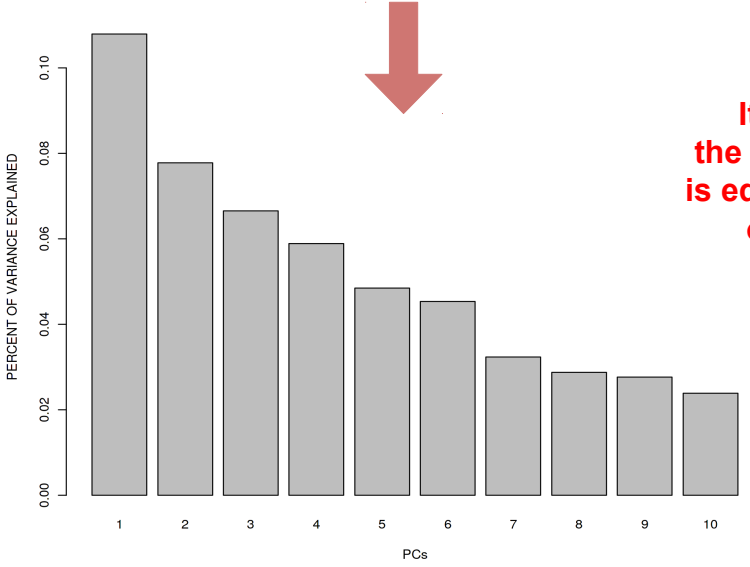
Mathematically:

$M_{ij} = X_{ij} - \mu_j$

$A = (1/N)*M^T*M$

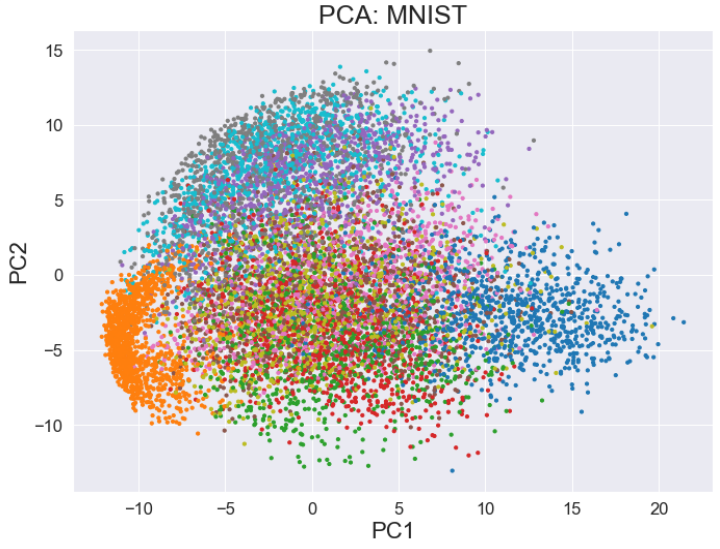
$A*u = \lambda*u$

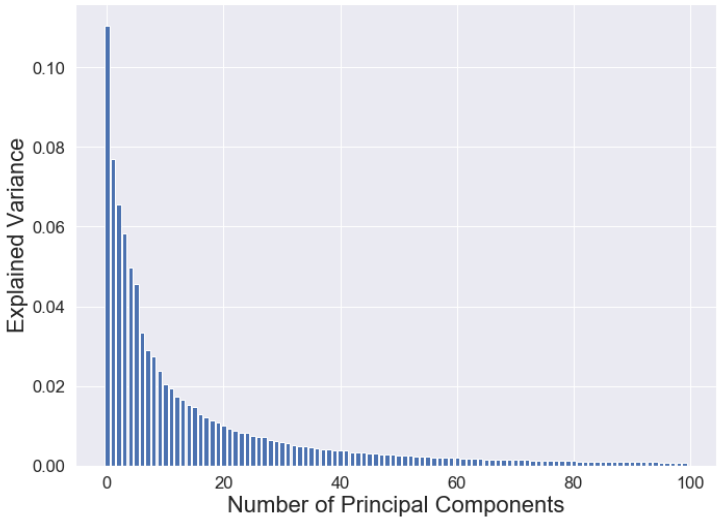


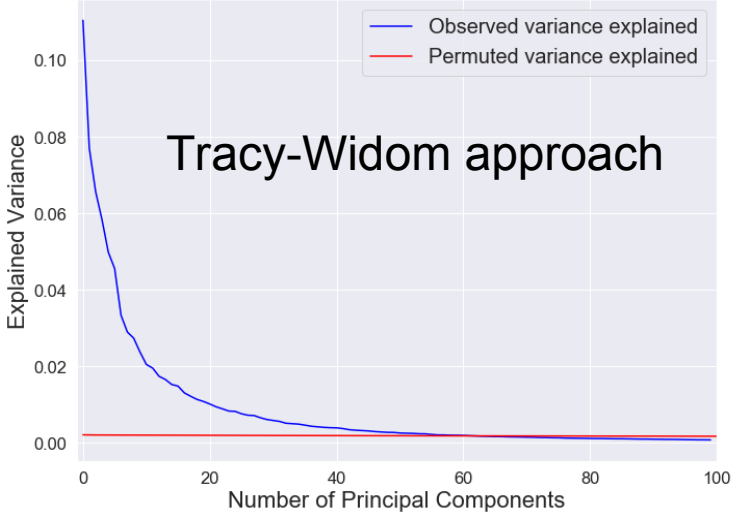


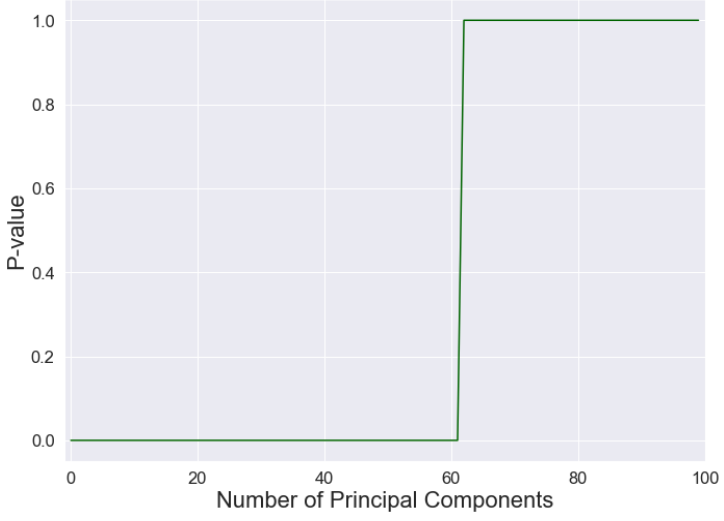
It can be analitically derived that the eigen value decomposition in PCA is equivalent to projecting data on axes of maximal variation in the data

Estimating the number of informative PCs



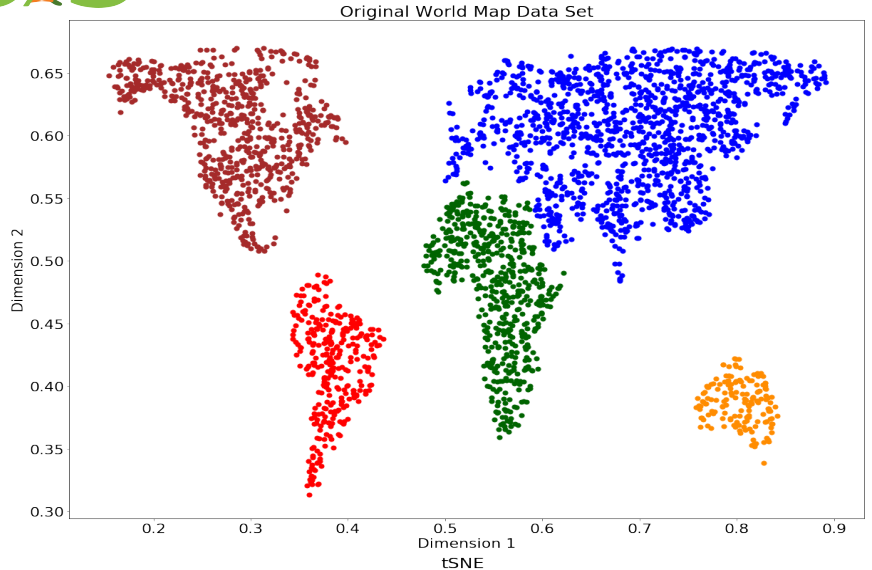


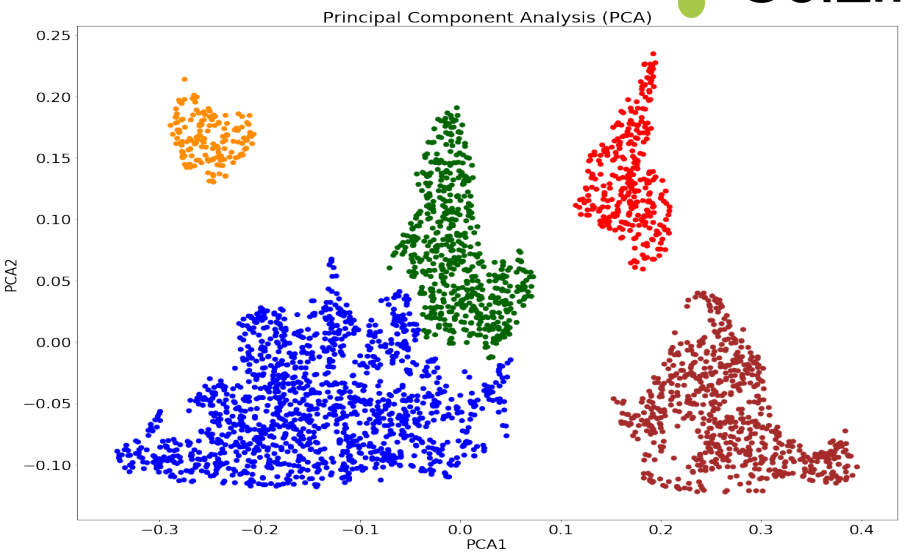


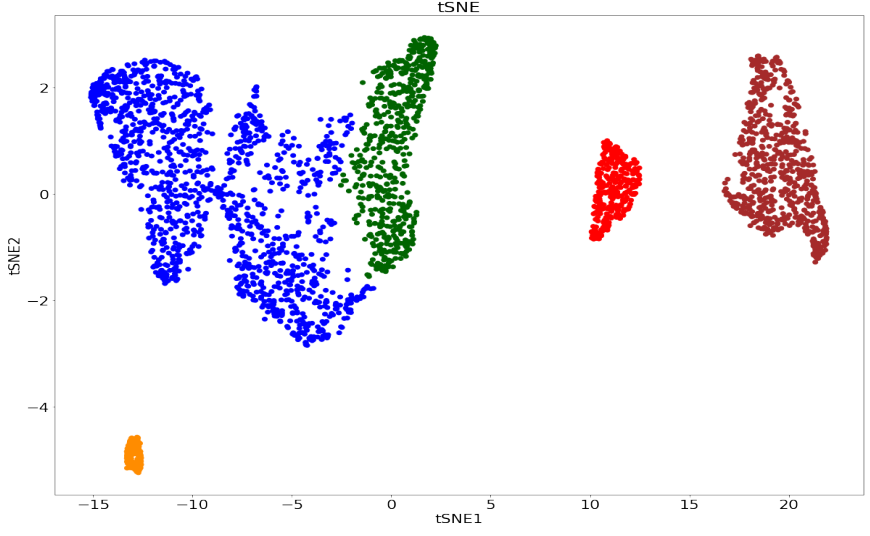


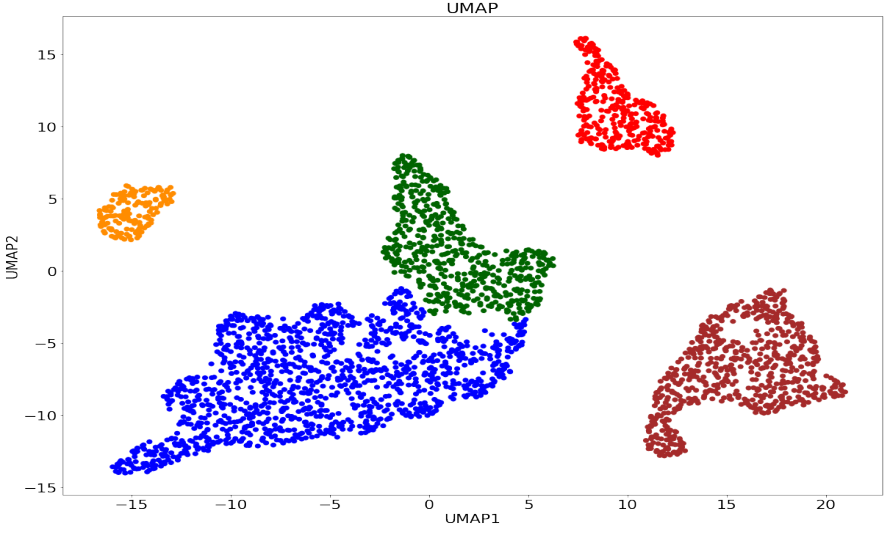
In Seurat: JackStraw

PCA works fine on a linear manifold

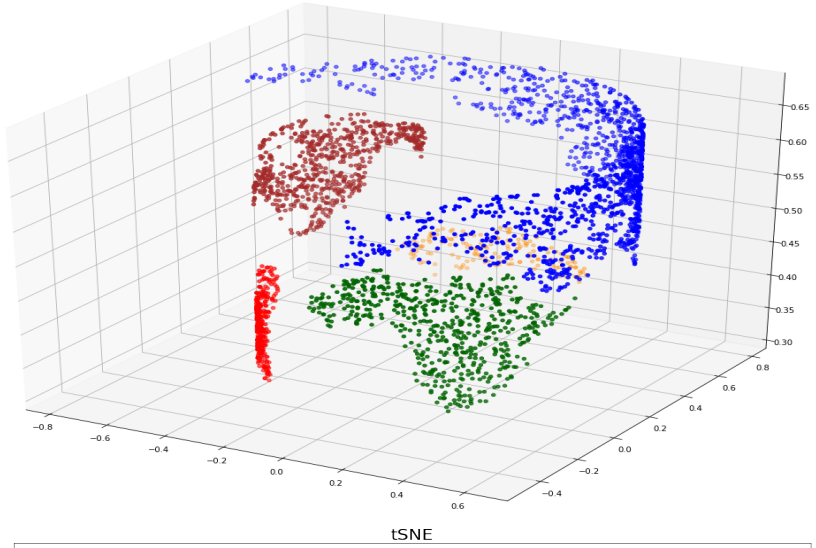


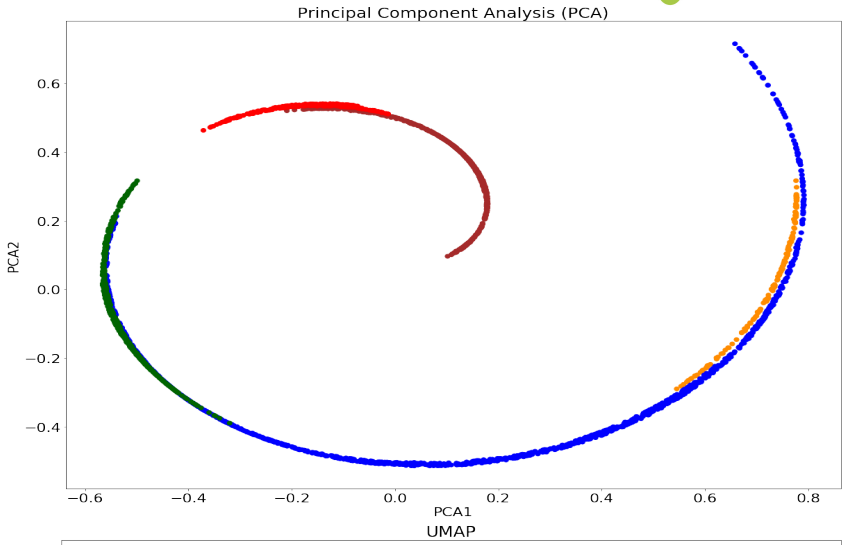




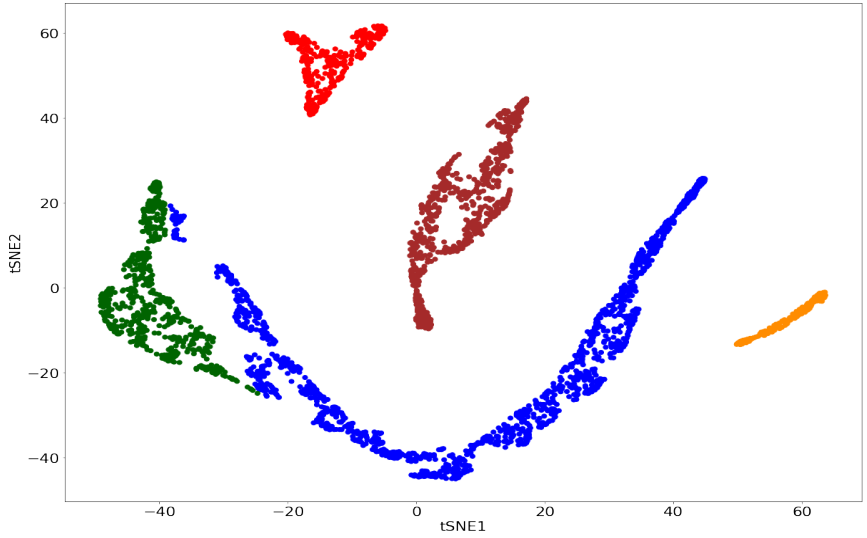


PCA vs. tSNE vs. UMAP on non-linear manifold

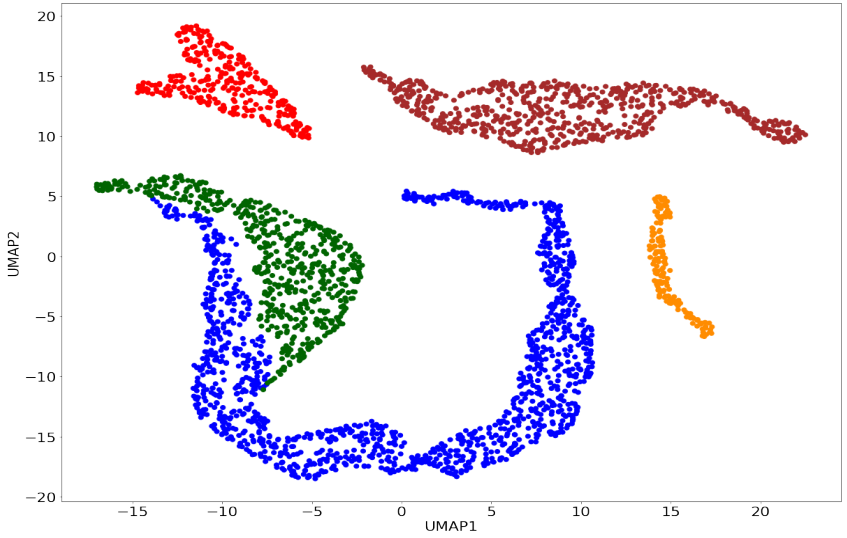




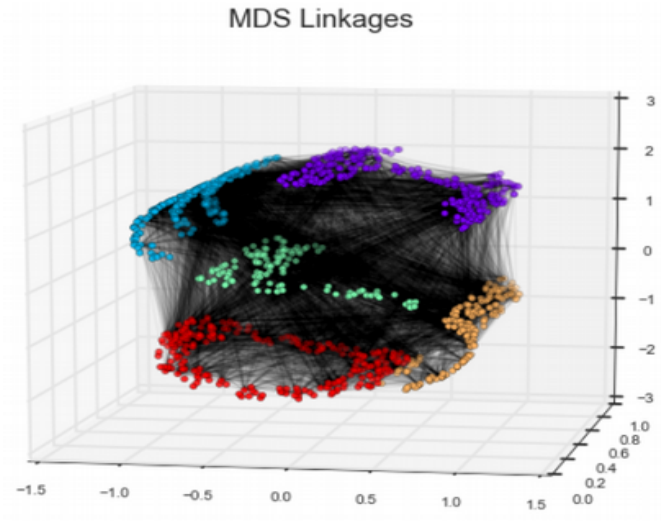
tSNE

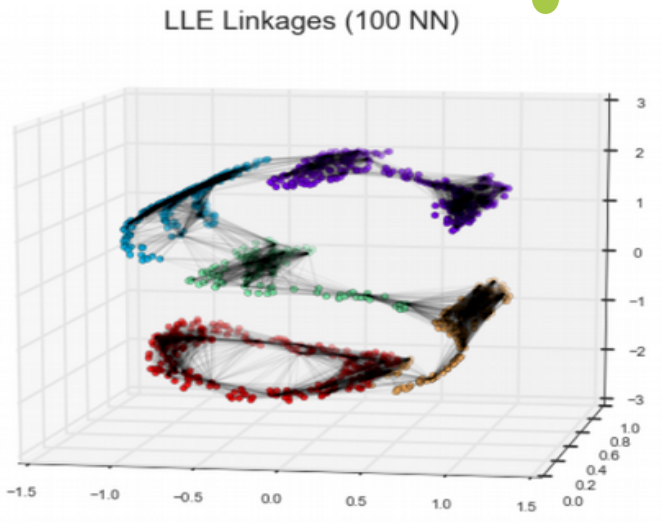


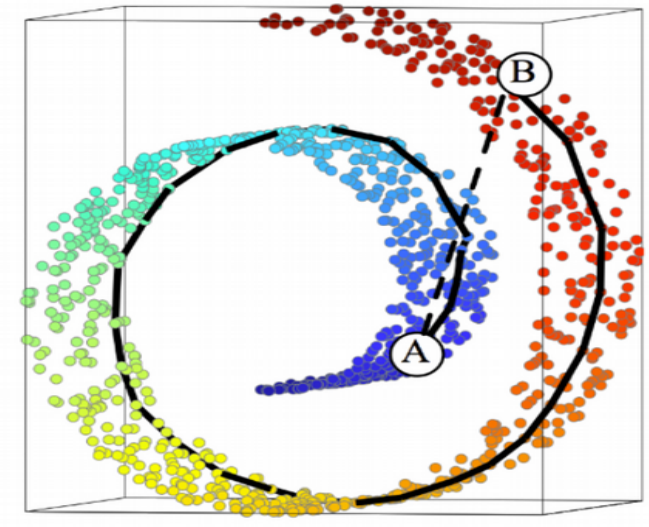
UMAP

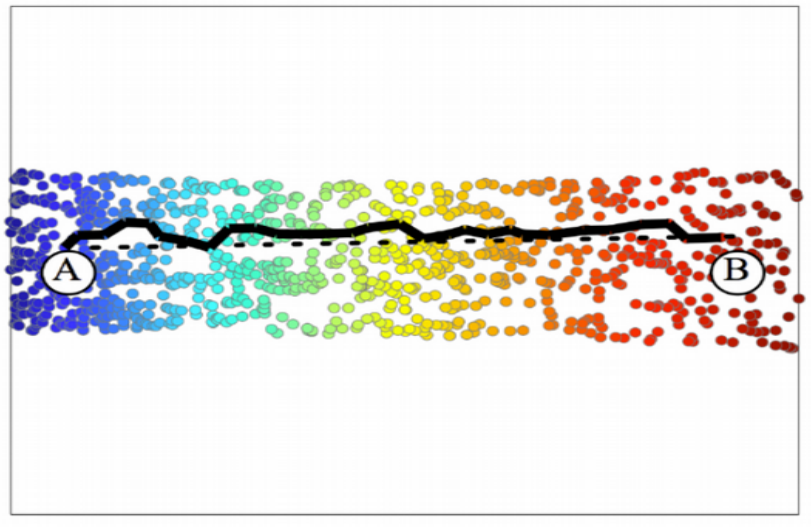


Why PCA can't unwrap the Swiss Roll





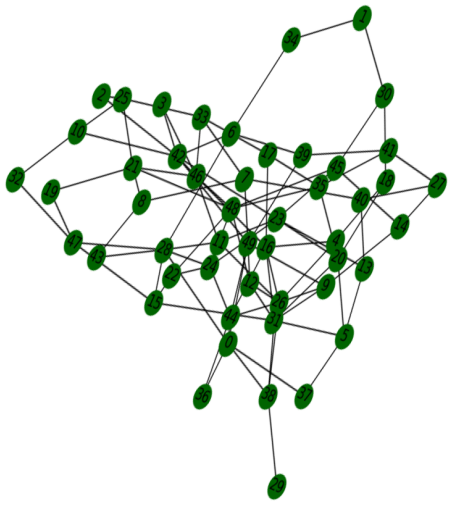




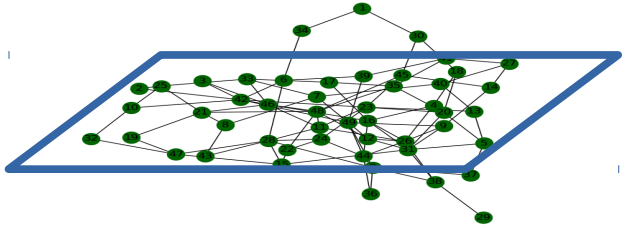
Non-linear dimension reduction: neigborhood graph

1) Construct high-dimensional graph

p_{ij}

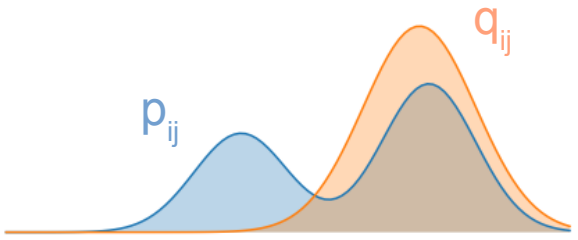


2) Construct low-dimensional graph



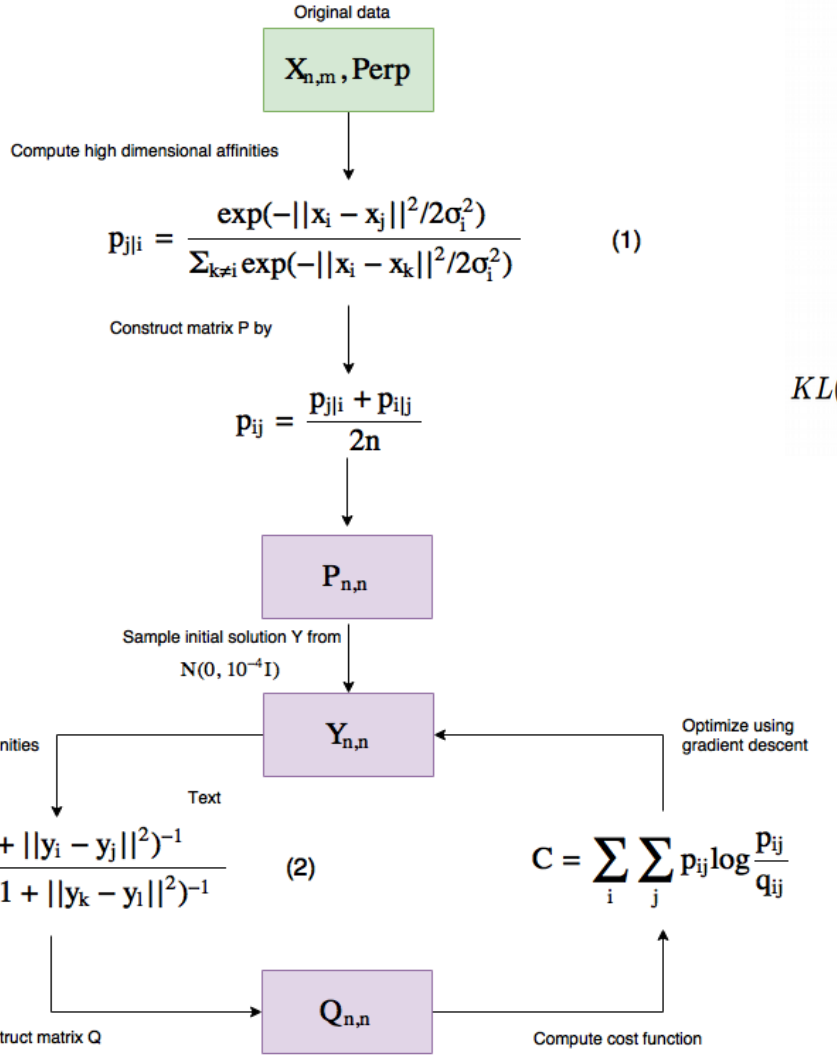
q_{ij}

3) Collapse the graphs together



Kullback-Leibler divergence

tSNE dimension reduction algorithm

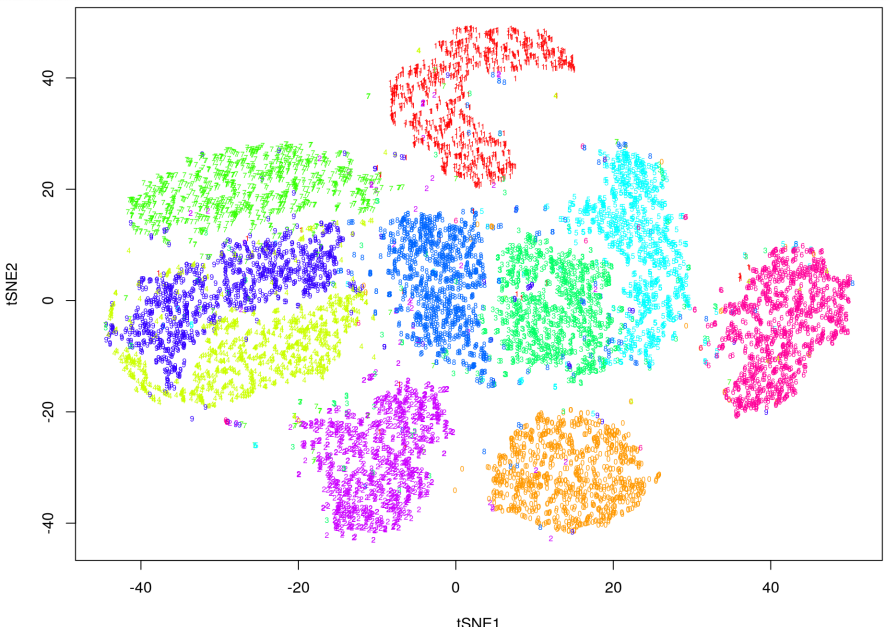


$$p_{j|i} = \frac{\exp\left(-||x_i - x_j||^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-||x_i - x_k||^2/2\sigma_i^2\right)}, \qquad p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N} \qquad (1)$$

$$\text{Perplexity} = 2^{-\sum_j p_{j|i} \log_2 p_{j|i}} \qquad (2)$$

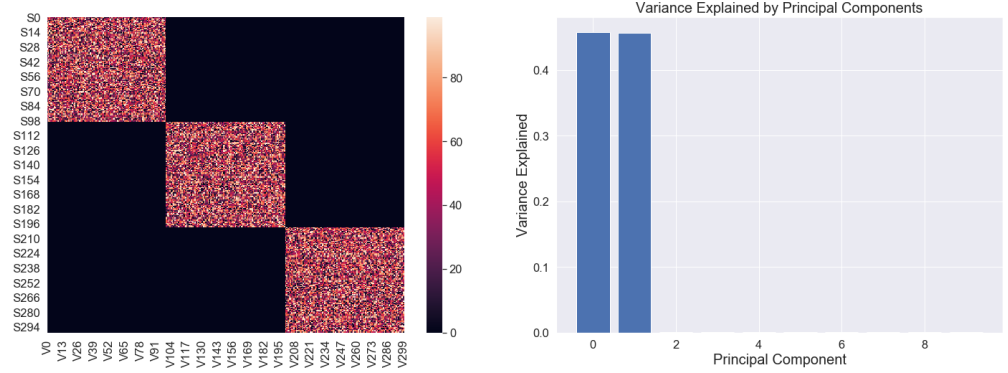
$$q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{k\neq l}\left(1 + ||y_k - y_l||^2\right)^{-1}} \qquad (3)$$

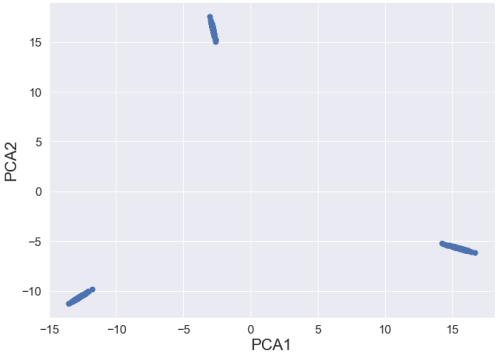
$$KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}, \qquad \frac{\partial KL}{\partial y_i} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + ||y_i - y_j||^2\right)^{-1} \qquad (4)$$

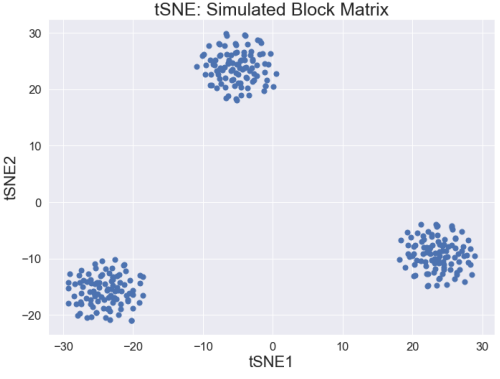


PCA vs. tSNE
when number of populations increases

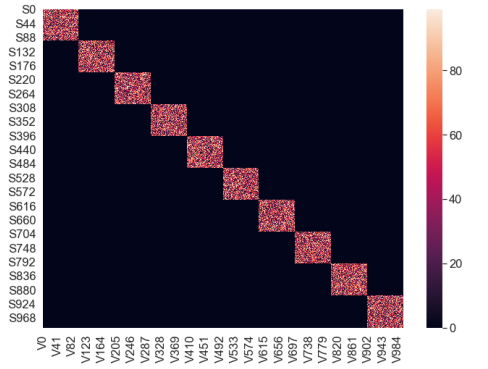
Three classes of data points

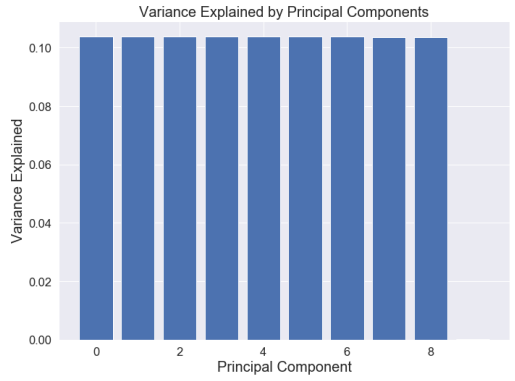


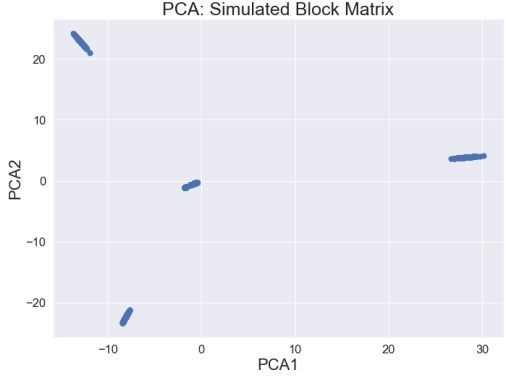


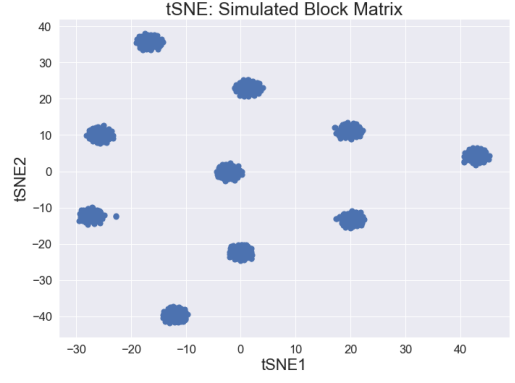


Ten classes of data points



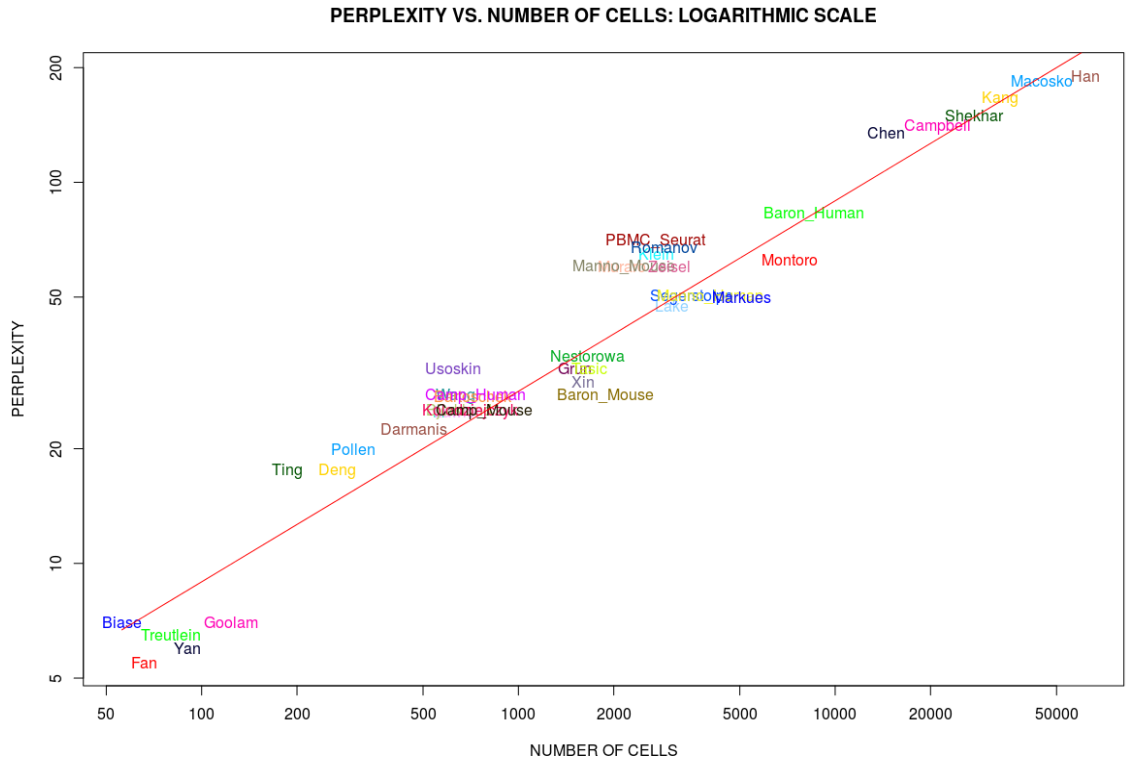






How to select optimal perplexity

Van der Maaten: “Loosely speaking, one could say that a larger / denser dataset requires a larger perplexity.“



$$\log(Perp) = -0.179 + 0.51*\log(N)$$

$$Perp \sim N^{(1/2)}$$

Limitations of tSNE and promise of UMAP

tSNE does not scale for large data sets?

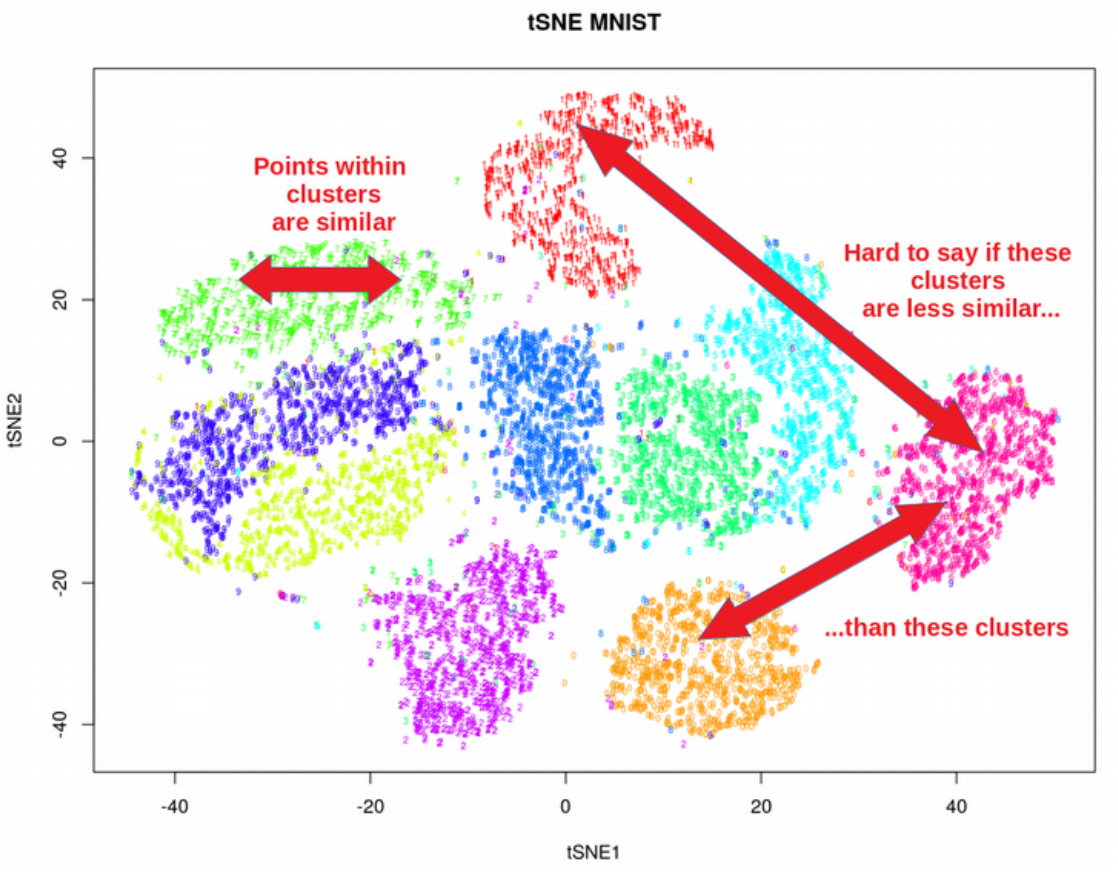
tSNE does not preserve global structure?

tSNE can only embed into 2-3 dims?

tSNE performs non-parametric mapping (no variance explained statistics)?

tSNE can not work with high-dimensional data directly (PCA needed)?

tSNE uses too much RAM at large perp?



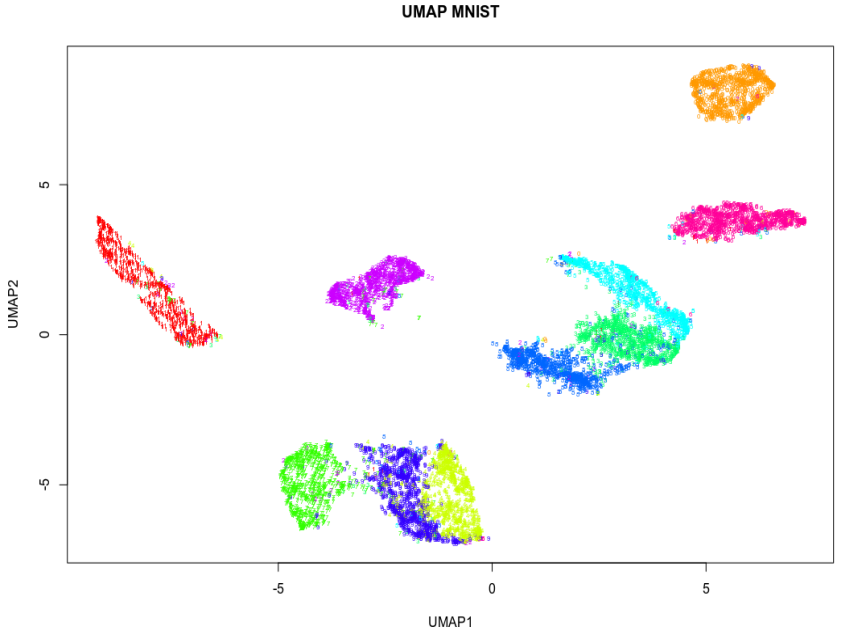
How is UMAP different from tSNE

UMAP uses local connectivity for high-dim probabilities

$$p_{i|j} = e^{-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}}$$

UMAP does not normalize probabilities (speed-up)

UMAP can deliver a number of components for clustering



UMAP uses Laplacian Eigenmap for initialization

UMAP uses Cross-Entropy (not KL) as cost function

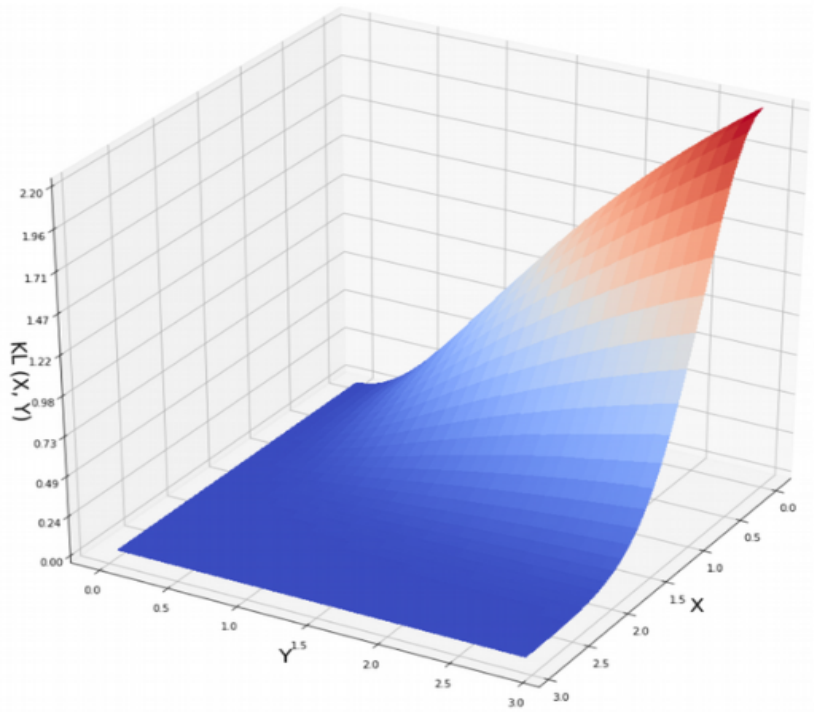
$$CE(X,Y) = \sum_i \sum_j \left[p_{ij}(X) \log\left(\frac{p_{ij}(X)}{q_{ij}(Y)}\right) + (1 - p_{ij}(X)) \log\left(\frac{1 - p_{ij}(X)}{1 - q_{ij}(Y)}\right) \right]$$

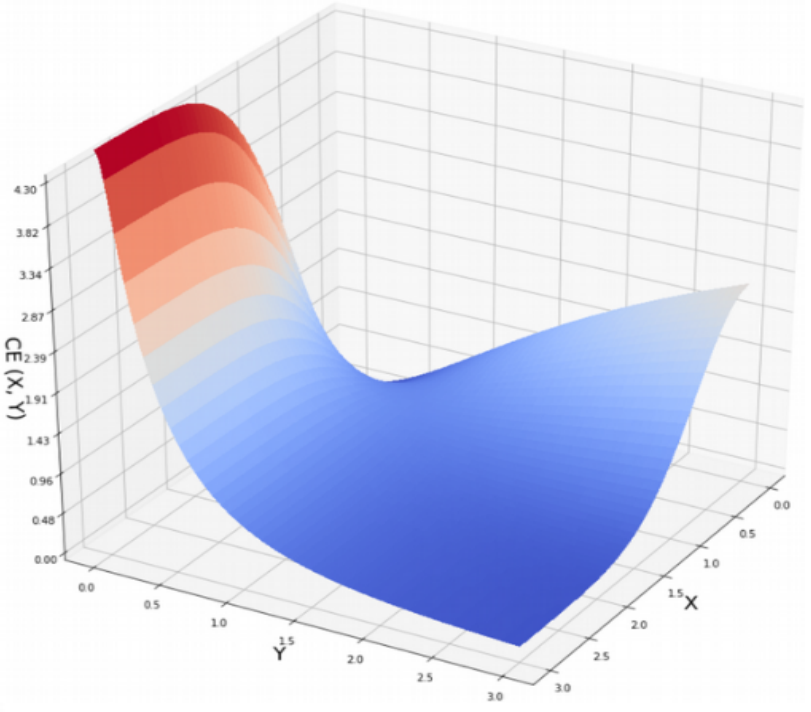
This is similar to tSNE cost function

This term is UMAP specific

tSNE vs. UMAP:
global structure preservation

Cost function seems to make UMAP preserve more of global structure than tSNE

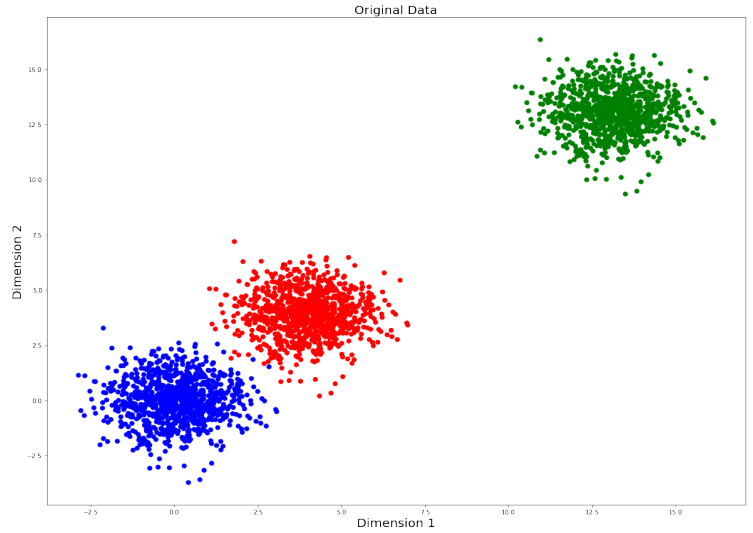


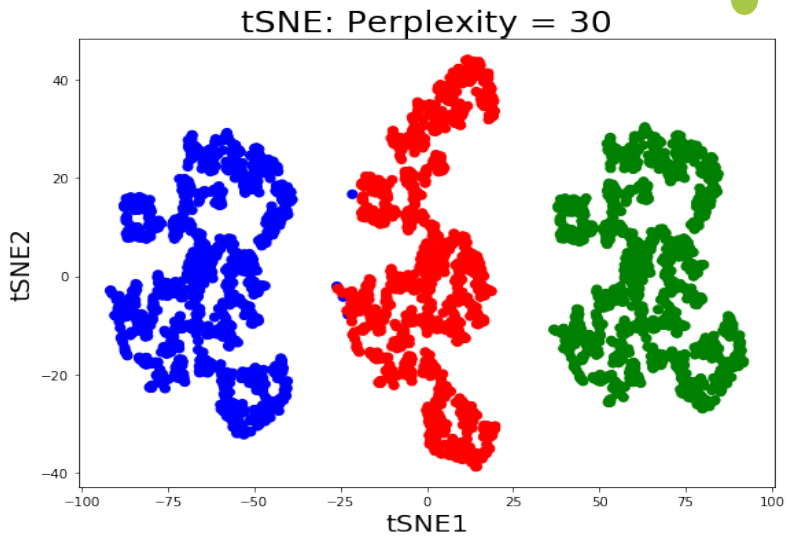


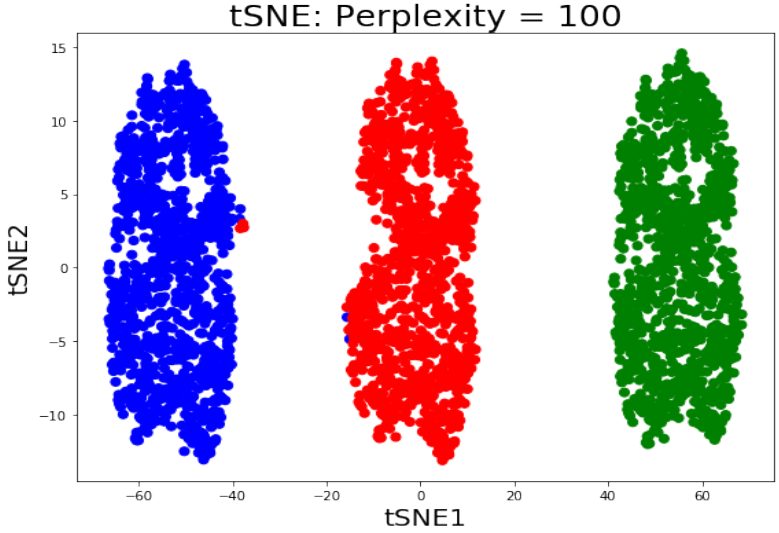
X → infinity, Y can be any

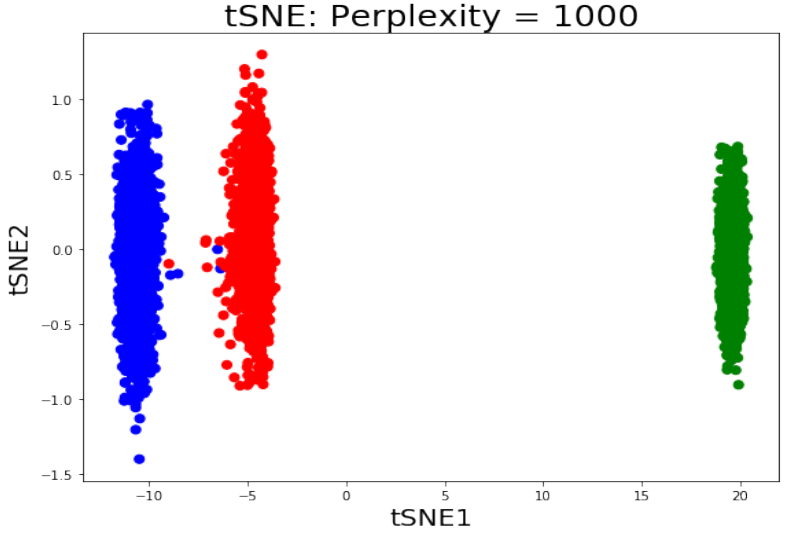
X → infinity, Y → infinity

Why preserving global structure is important



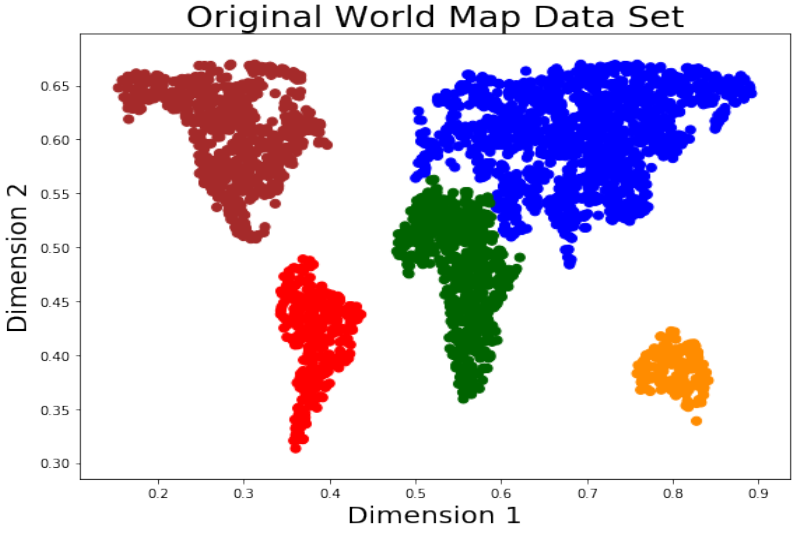


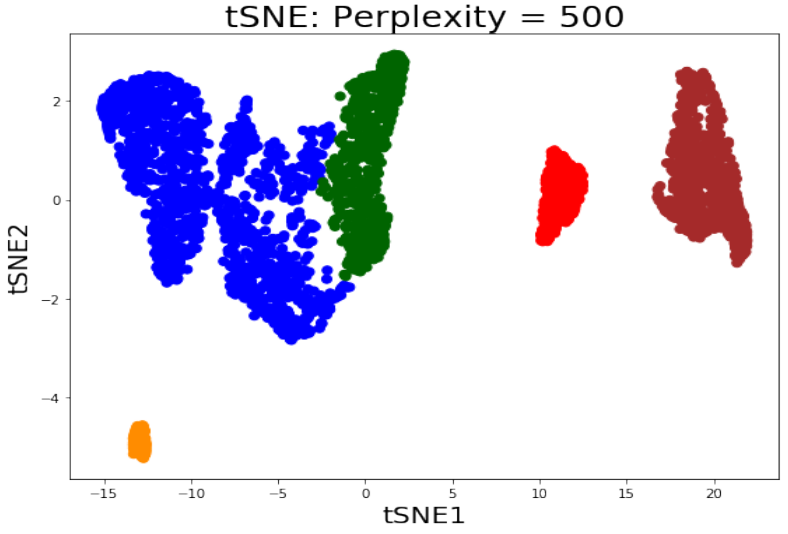


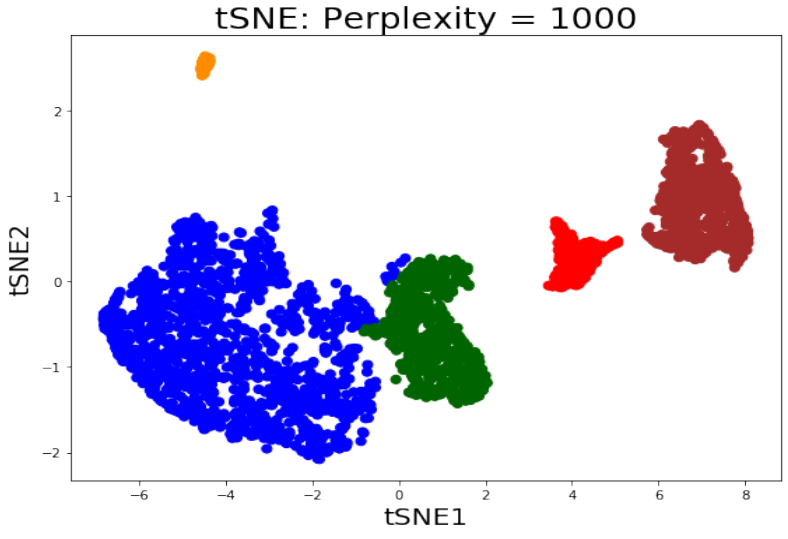


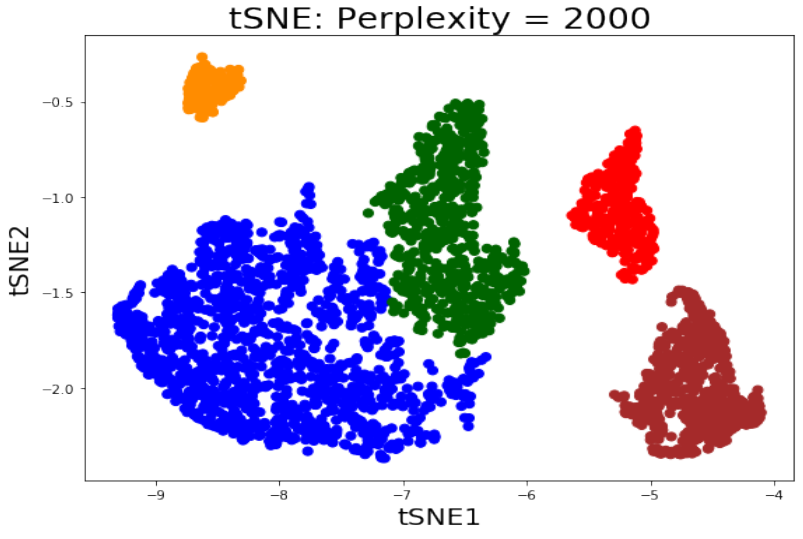
Can large perplexity solve the problem of global structure for tSNE?

Why preserving global structure is important



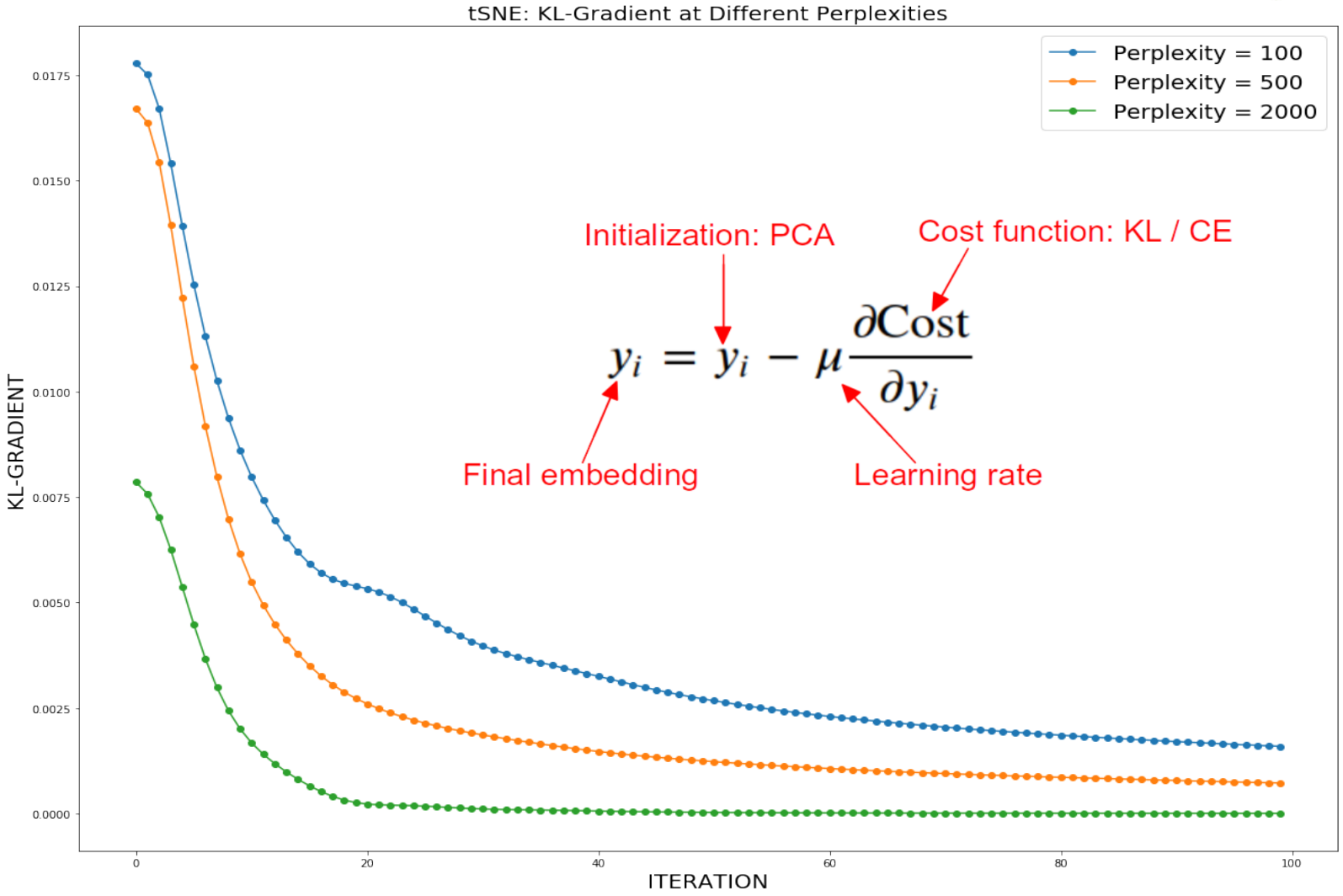




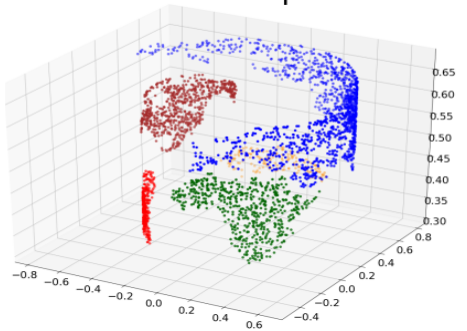
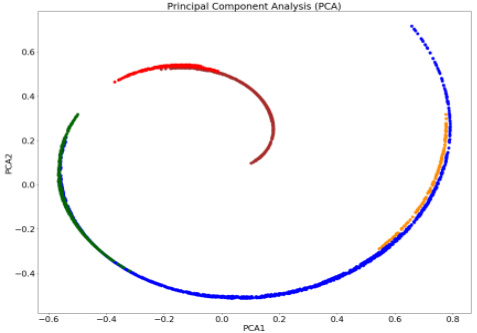
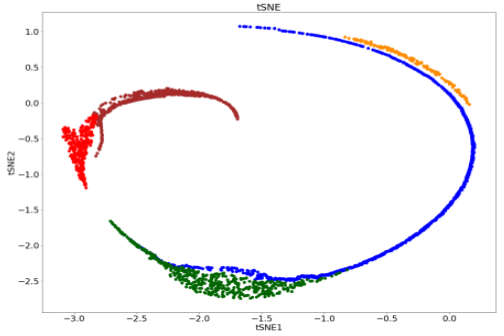
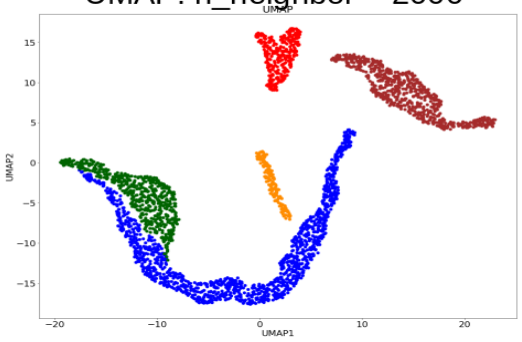
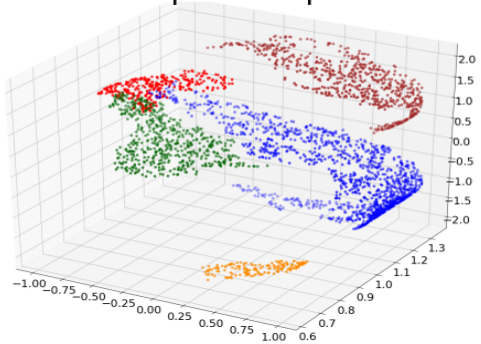
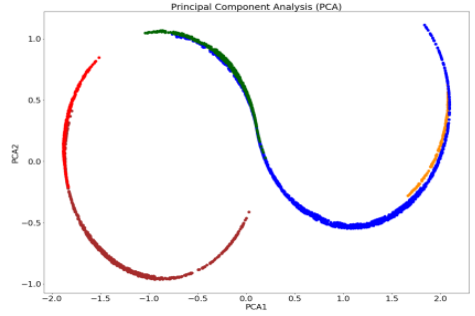
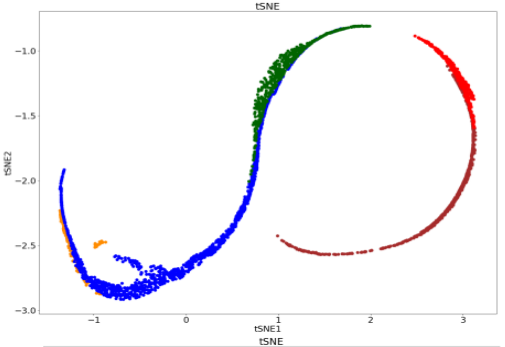
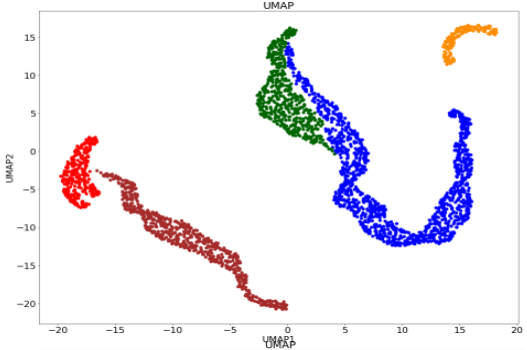
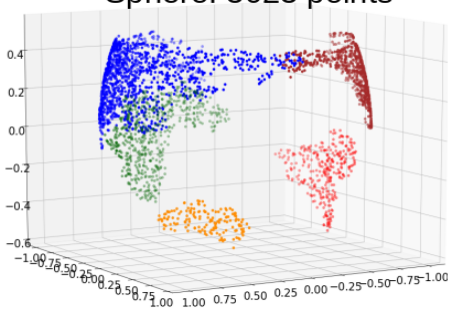
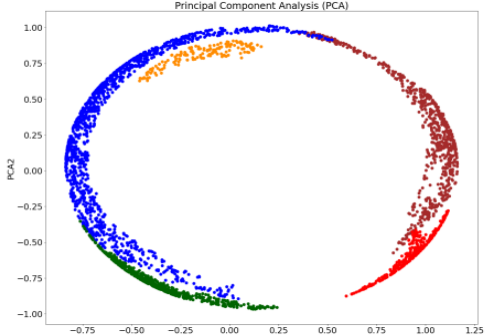
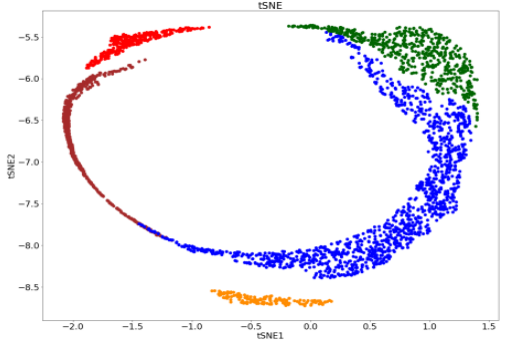
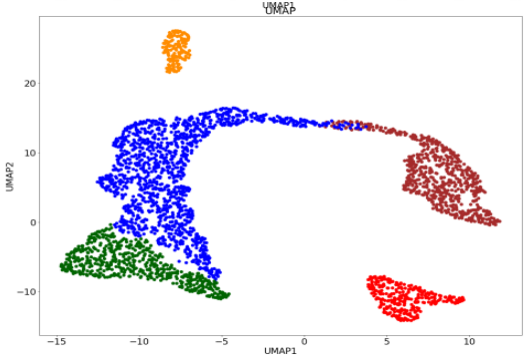


Can large perplexity solve the problem of global structure for tSNE?

KL-gradient goes to zero at large perplexity

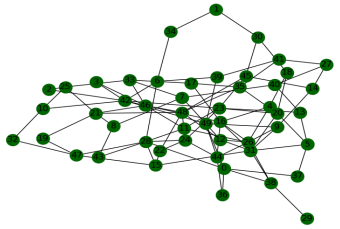


tSNE degrades to PCA on non-linear manifold at large perplexity

Swiss Roll: 3023 points	Principal Component Analysis	tSNE: perplexity = 2000	UMAP: n_neighbor = 2000
			
S-shape: 3023 points			
			
Sphere: 3023 points			
			

Attempts to balance local and global structure

Graph Laplacian, Laplacian Eigenmap, spectral clustering, diffusion maps, spectral dimension reduction methods etc.



$$s(x_i, x_j) = exp(-\alpha \|x_i - x_j\|^2)$$

$$L = I - D^{-1} * S$$
$$L * u = \lambda * u$$

Laplacian Eigenmap

$$P = D^{-1} * S$$
$$P^t * u = \lambda^t * u$$

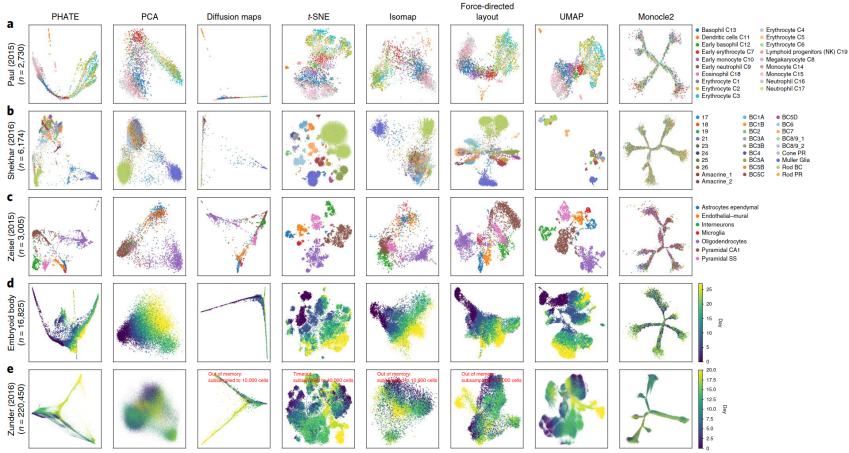
Diffusion Maps

S=

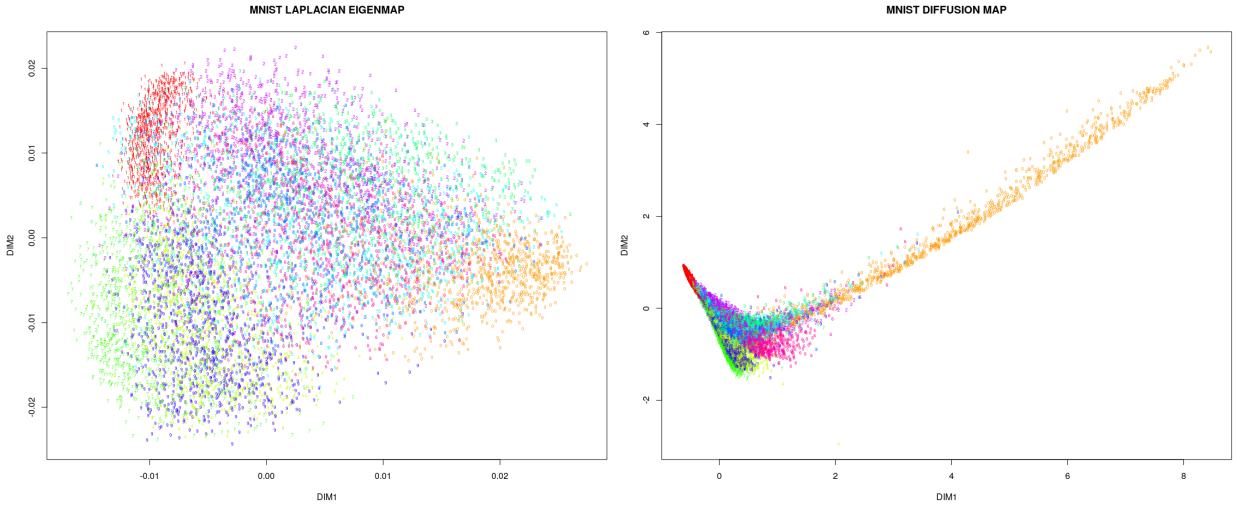
	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	1.0000000	0	0.7429016	0.6319343	0.0000000	0.0000000	0.0000000
[2,]	0.0000000	1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
[3,]	0.7429016	0	1.0000000	0.0000000	0.0000000	0.0000000	0.0657756
[4,]	0.6319343	0	0.0000000	1.0000000	0.0000000	0.0000000	0.7195922
[5,]	0.0000000	0	0.0000000	0.0000000	1.0000000	0.7765565	0.0000000
[6,]	0.0000000	0	0.0000000	0.0000000	0.7765565	1.0000000	0.0000000
[7,]	0.0000000	0	0.6657756	0.7195922	0.0000000	0.0000000	1.0000000
[8,]	0.0000000	0	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

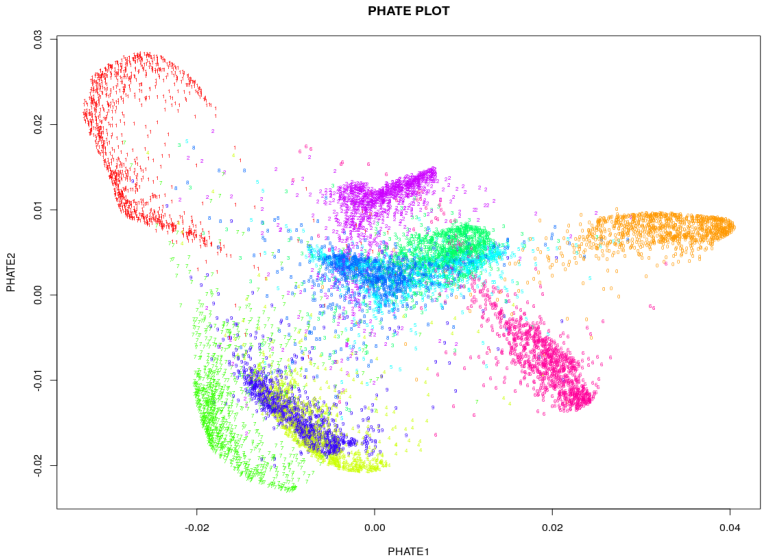
D=

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	2.374836	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
[2,]	0.000000	2.597451	0.000000	0.000000	0.000000	0.000000	0.000000
[3,]	0.000000	0.000000	2.408677	0.000000	0.000000	0.000000	0.000000
[4,]	0.000000	0.000000	0.000000	2.351526	0.000000	0.000000	0.000000
[5,]	0.000000	0.000000	0.000000	0.000000	2.523175	0.000000	0.000000
[6,]	0.000000	0.000000	0.000000	0.000000	0.000000	2.519936	0.000000
[7,]	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.170424
[8,]	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000



Moon et al., Nat Biotechnol. 2019; 37(12):1482-1492



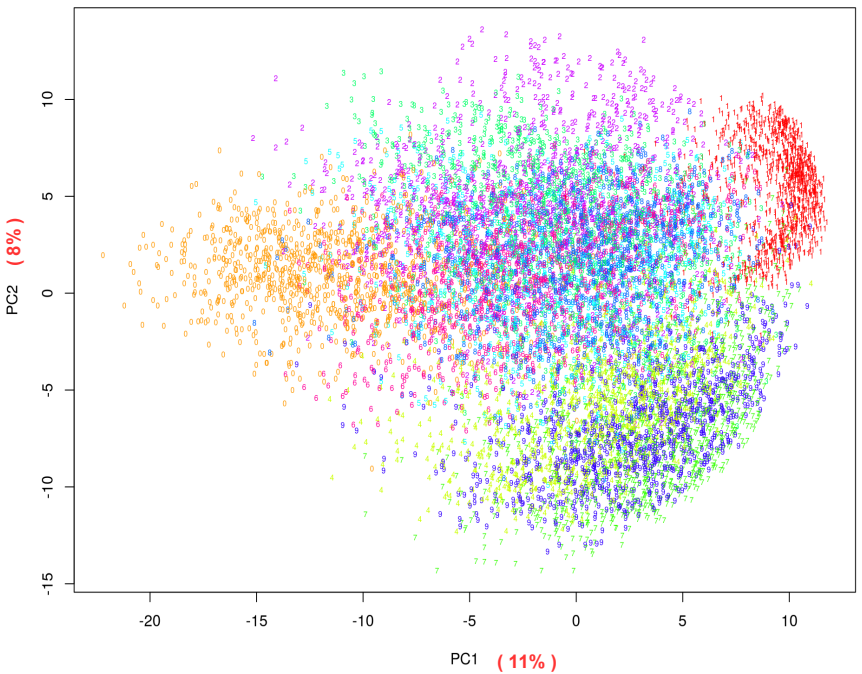


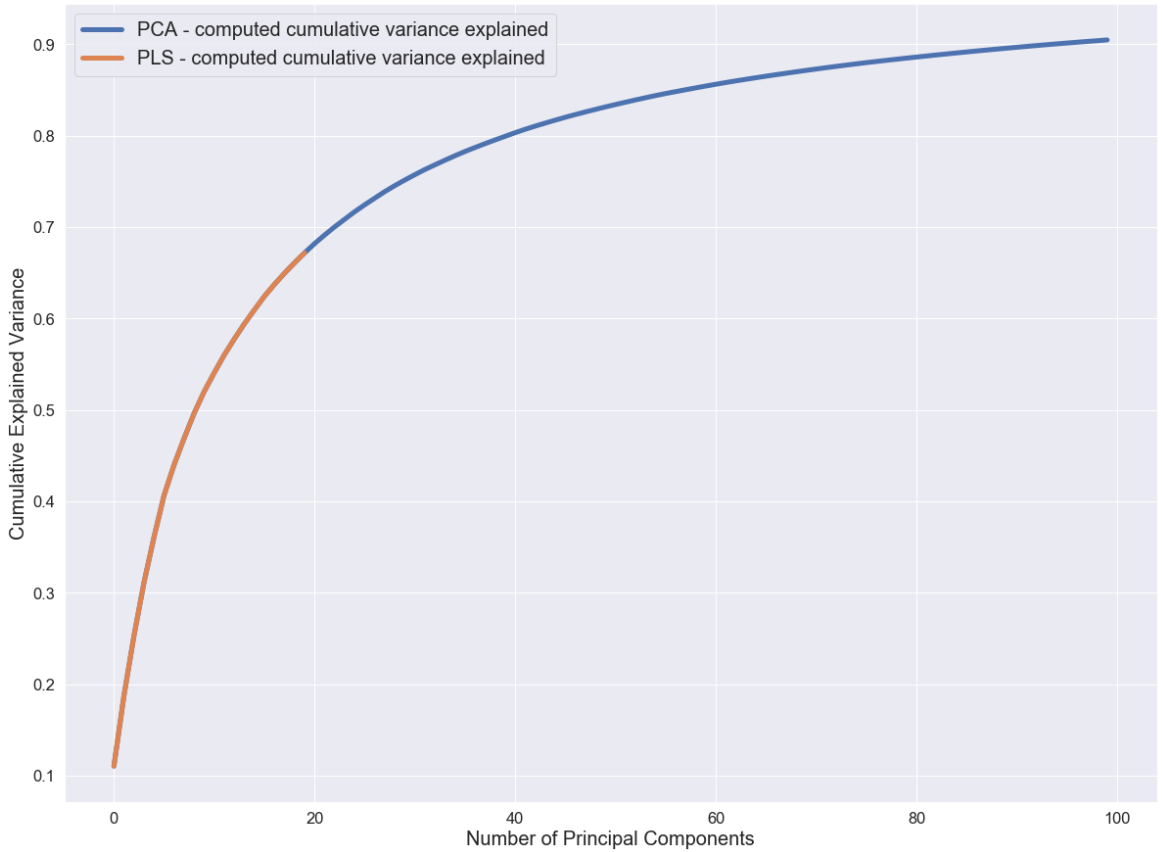
Variance explained by PCA, tSNE and UMAP

Variance explained by PCA components

$$\mathbf{X} = \alpha + \beta\ \mathbf{PCA_{matrix}} + \epsilon$$

$$R^2 = 1 - \frac{||X - B * PCA_{matrix}||^2}{||X||^2}$$

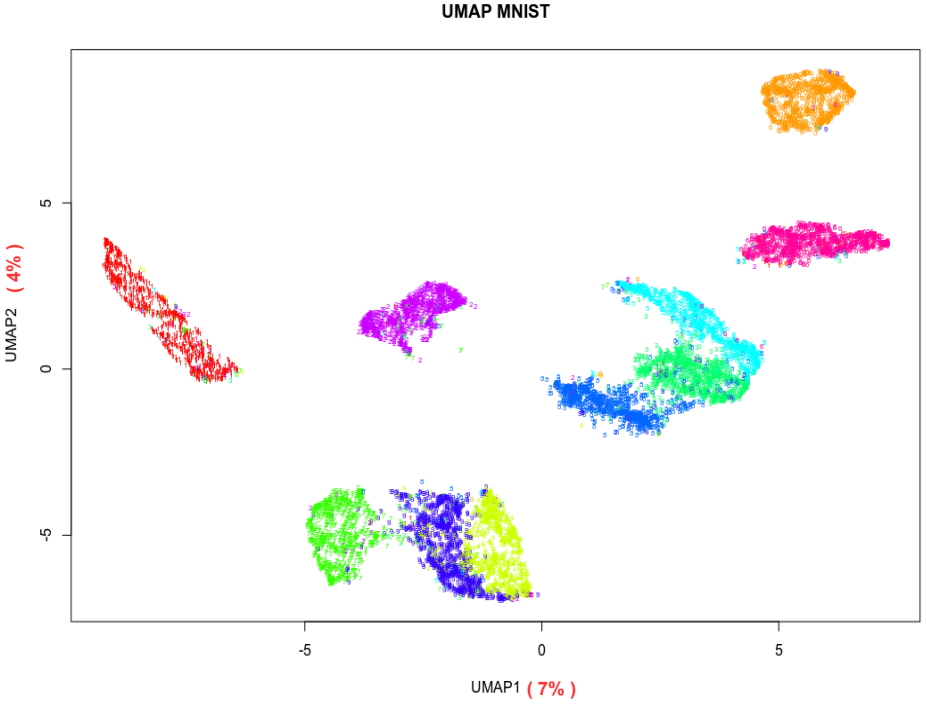


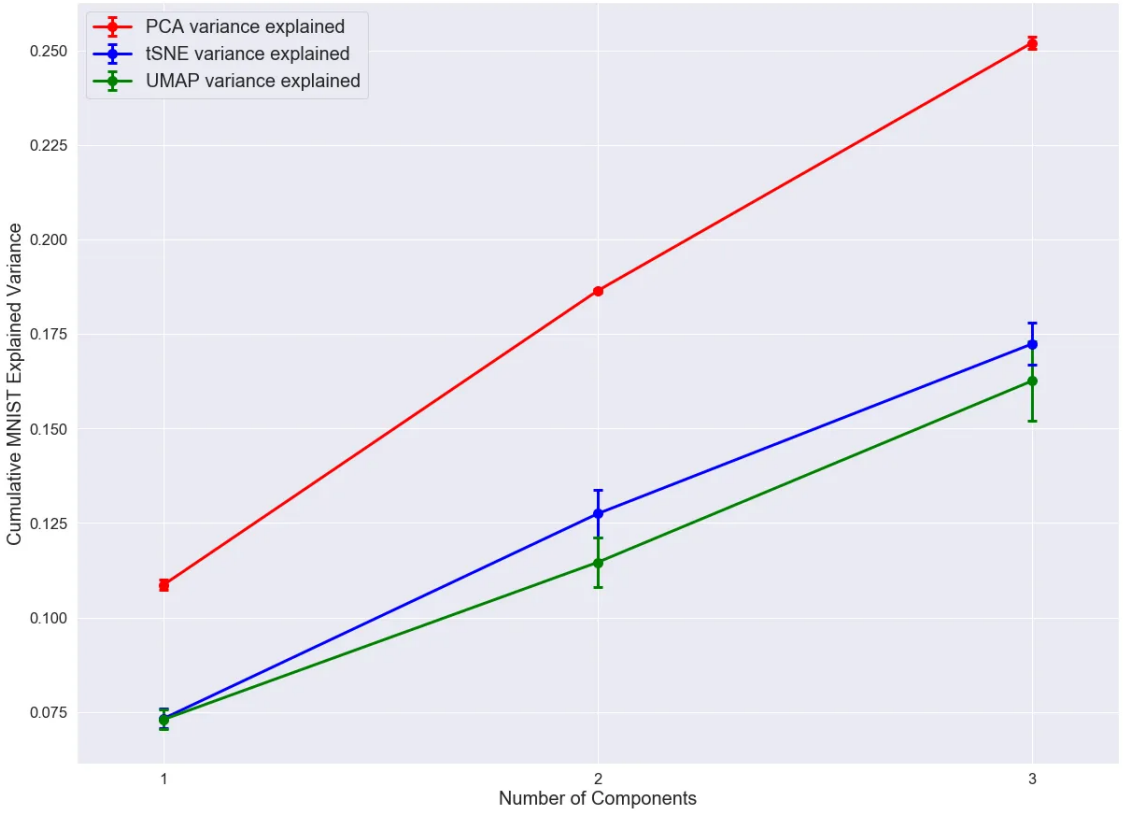


Variance explained by UMAP components

$$X = \alpha + \beta\ \mathbf{UMAP}_{\mathbf{matrix}} + \epsilon$$

$$R^2 = 1 - \frac{||X - B * UMAP_{matrix}||^2}{||X||^2}$$

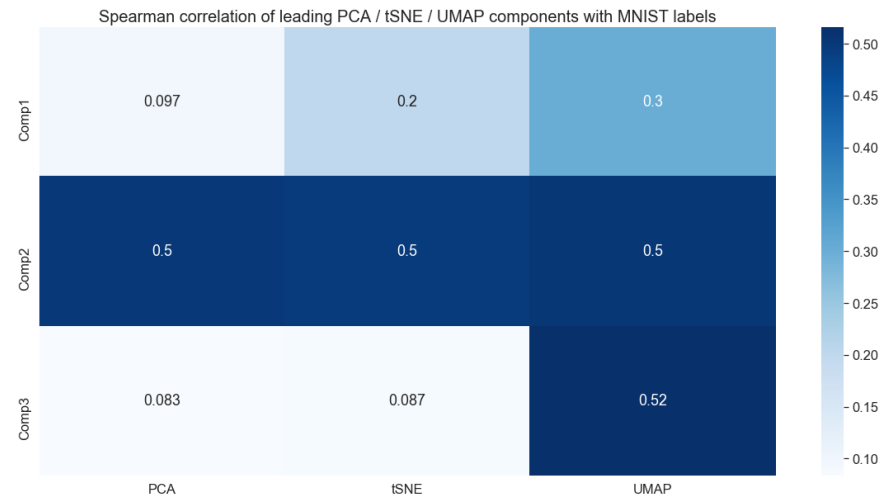


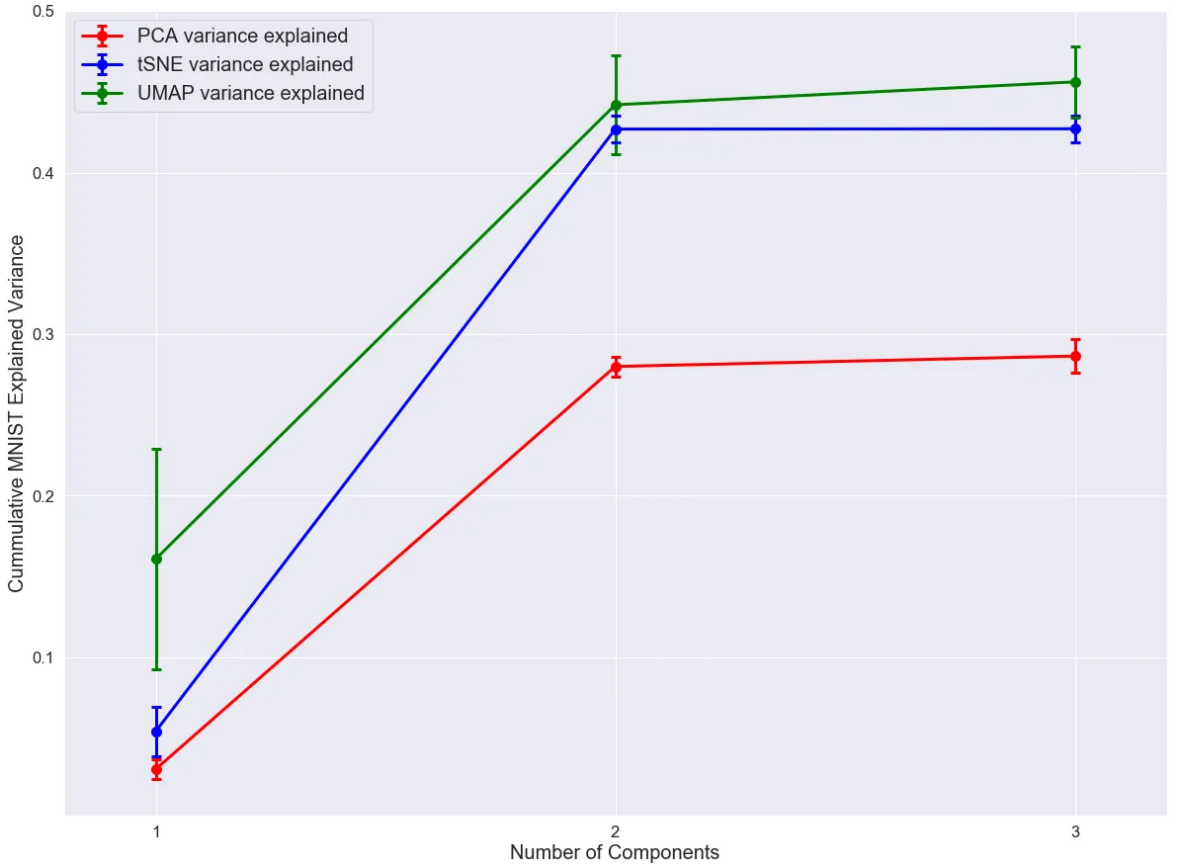


MNIST labels variance explained by UMAP components

$$\mathbf{labels} = \alpha + \beta\ \mathbf{UMAP}_{\mathbf{matrix}} + \epsilon$$

$$R^2 = 1 - \frac{||\mathbf{labels} - \boldsymbol{B} * \boldsymbol{UMAP}_{\boldsymbol{matrix}}||^2}{||\mathbf{labels}||^2}$$





National Bioinformatics Infrastructure Sweden (NBIS)

Knut och Alice Wallenbergs Stiftelse

Vetenskapsrådet

Lunds Universitet