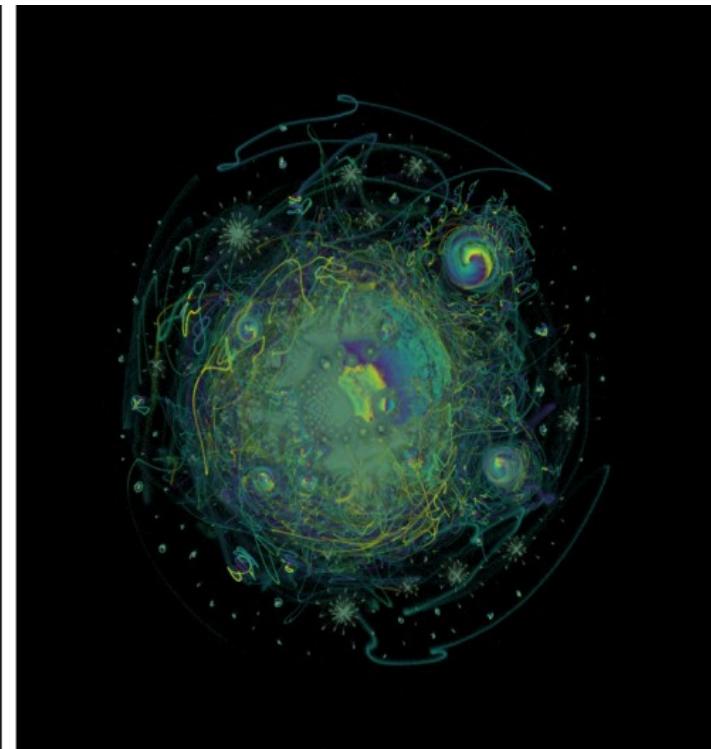
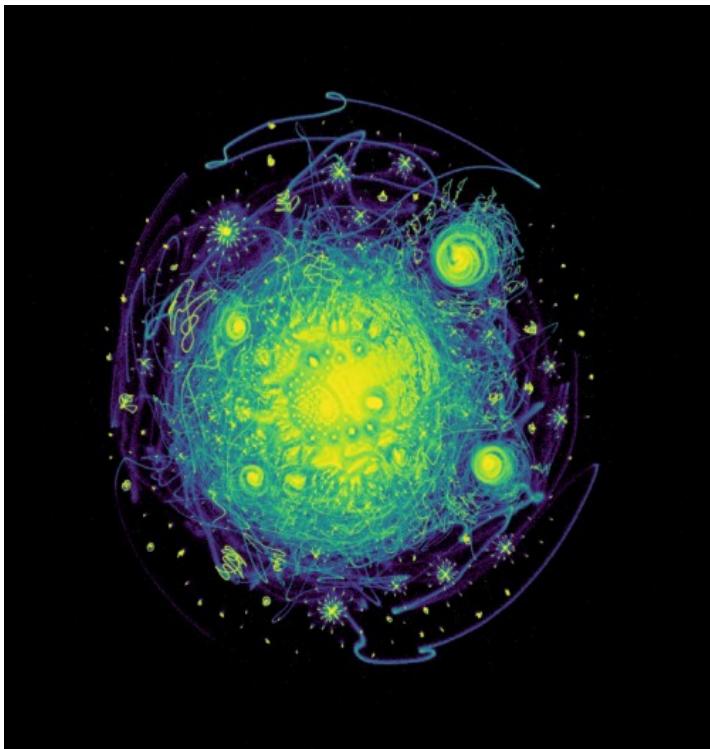


# Dimension Reduction and UMAP for Omics Integration

Physalia course, 15-17 December 2025, University of Barcelona  
Nikolay Oskolkov, MRG Group Leader, LIOS, Riga, Latvia



@NikolayOskolkov



@oskolkov.bsky.social



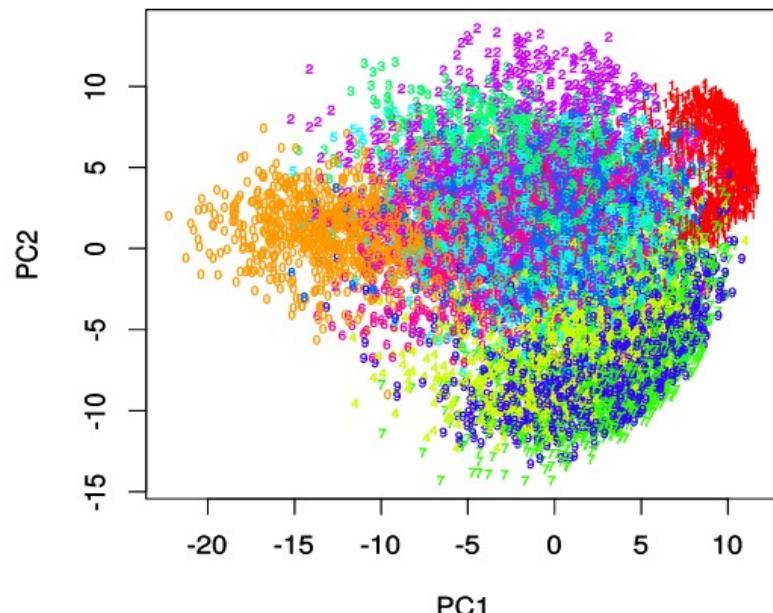
Personal homepage:  
<https://nikolay-oskolkov.com>

Image adapted from McInnes et al. 2018

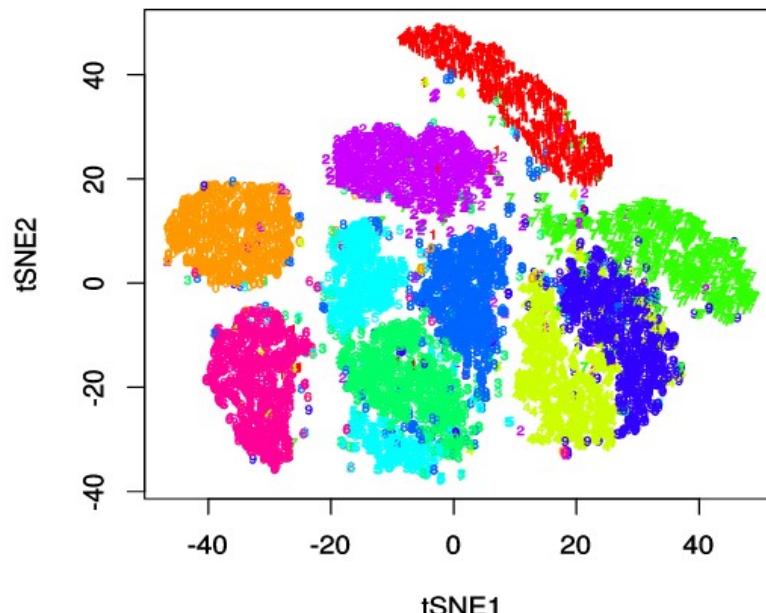
**Dimensionality reduction  
is also supposed to ... reduce dimensions**

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

PCA PLOT WITH PRCOMP

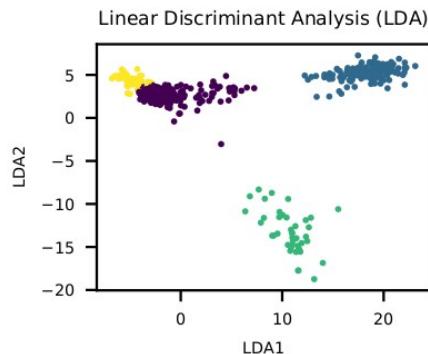
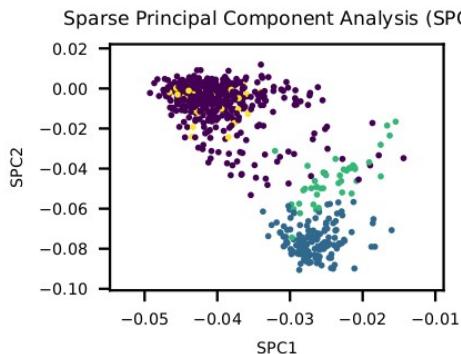
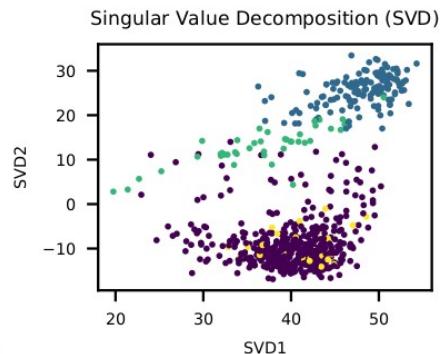
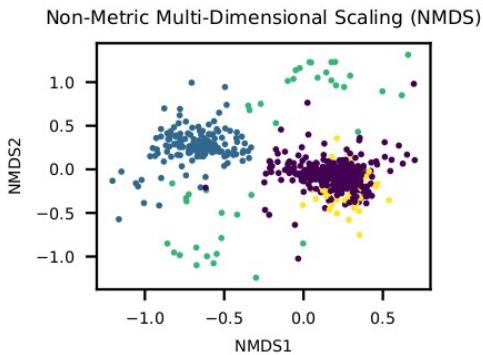
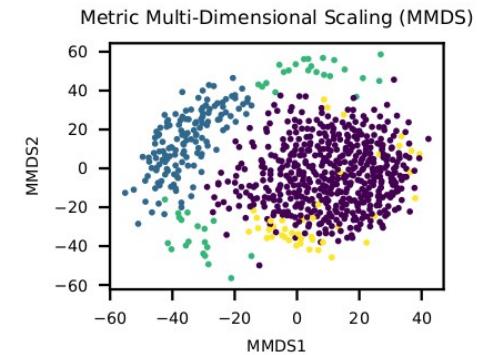
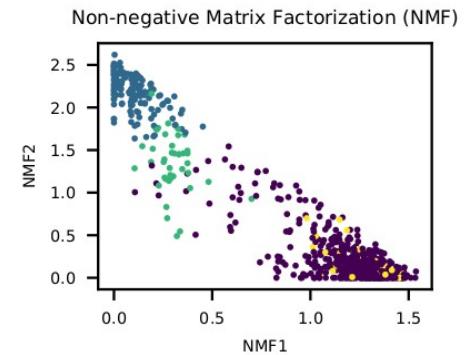
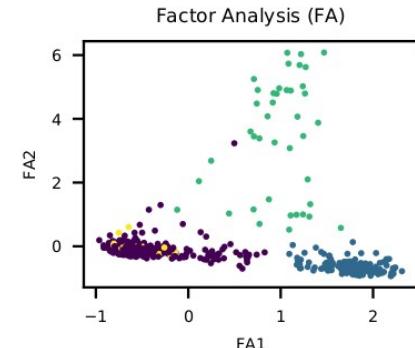
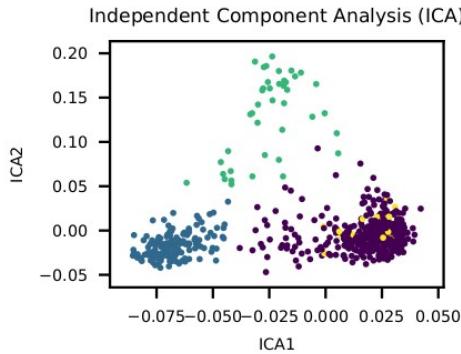
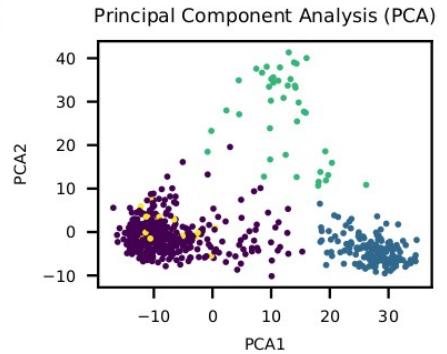


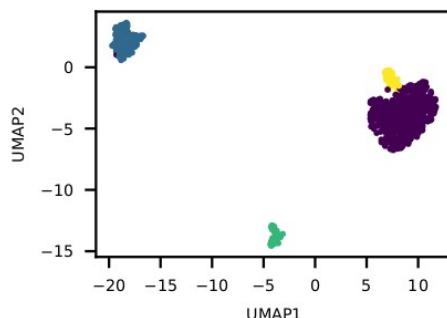
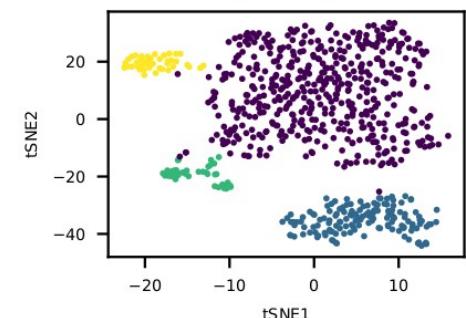
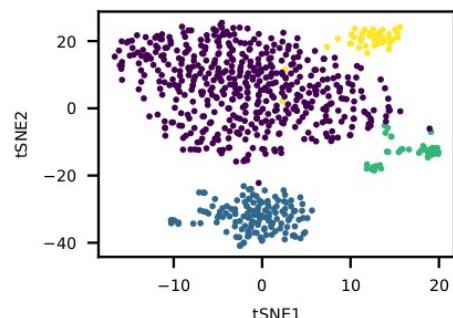
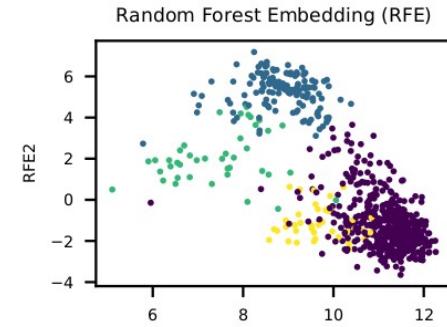
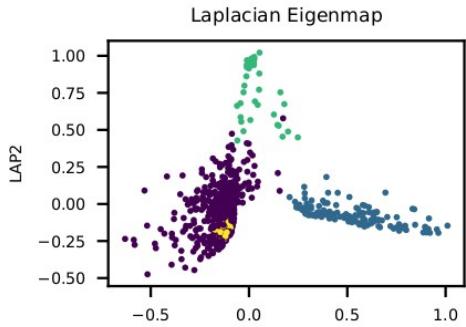
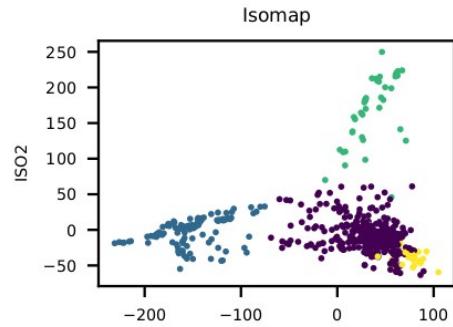
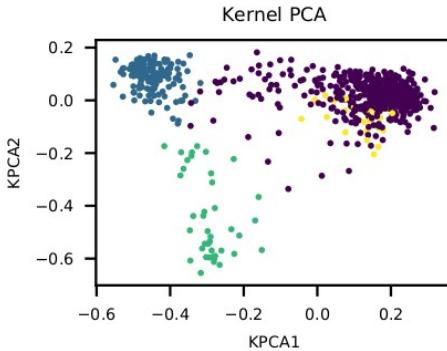
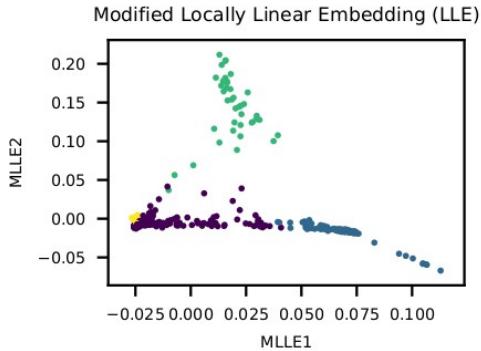
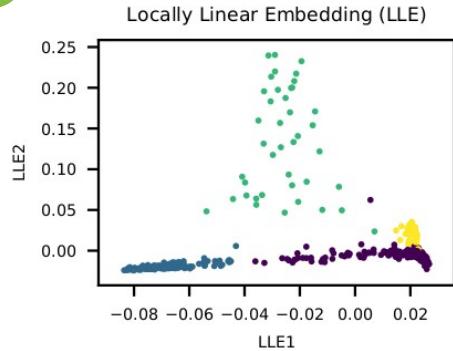
tSNE MNIST



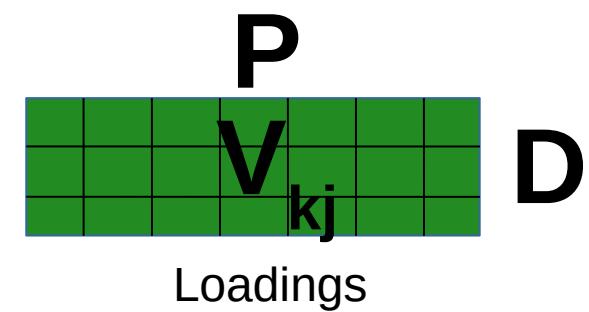
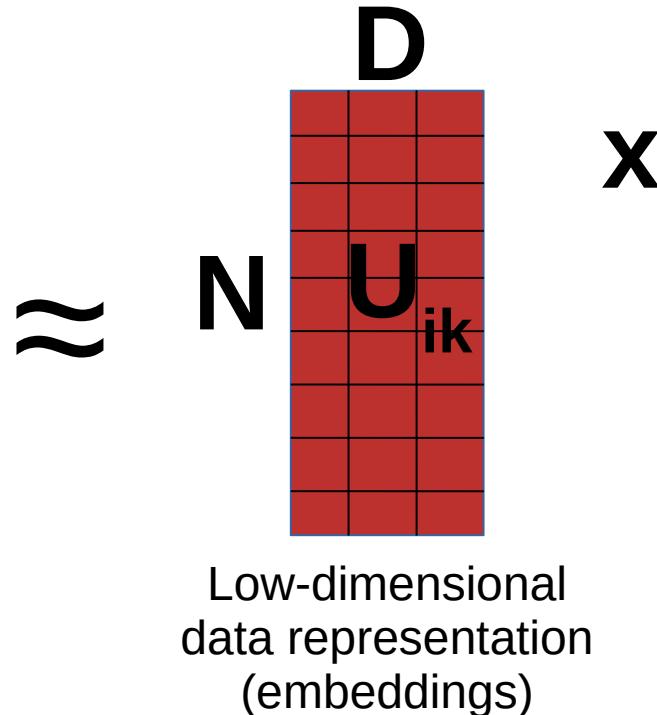
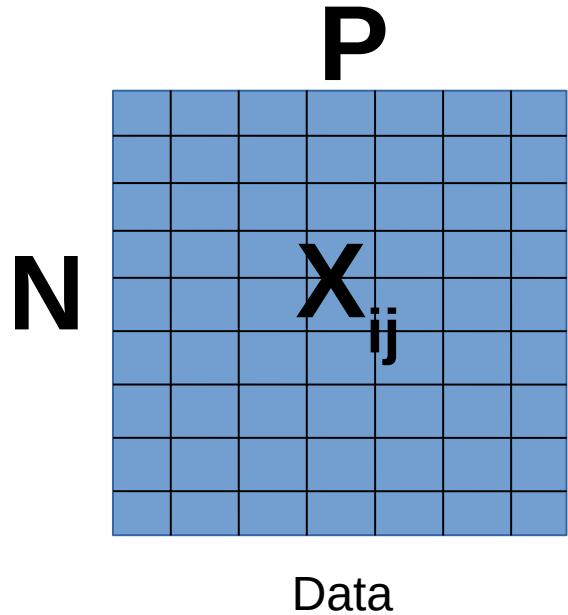
The goal of dimension reduction is not only visualization but also reducing dimensions

# Dimension reduction techniques: linear vs. non-linear





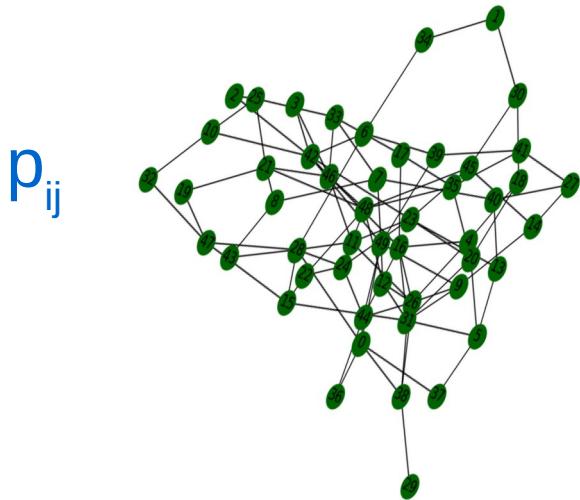
$$\mathbf{X}_{ij} \approx \mathbf{U}_{ik} \mathbf{V}_{kj}$$



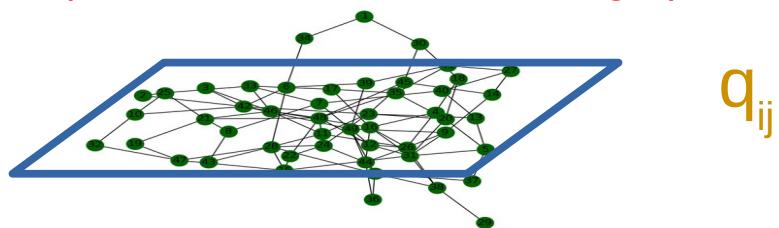
$$\text{Loss} = \sum_{i=1}^N \sum_{j=1}^P (\mathbf{X}_{ij} - \mathbf{U}_{ik} \mathbf{V}_{kj})^2$$

# Non-linear dimension reduction: neighborhood graph

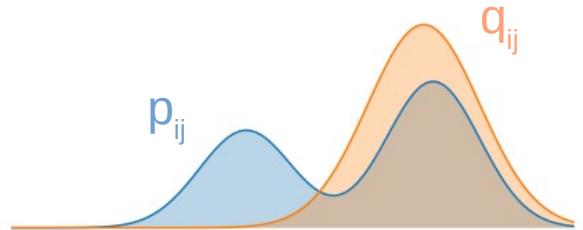
1) Construct high-dimensional graph



2) Construct low-dimensional graph



3) Collapse the graphs together



Kullback-Leibler divergence

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]);
```

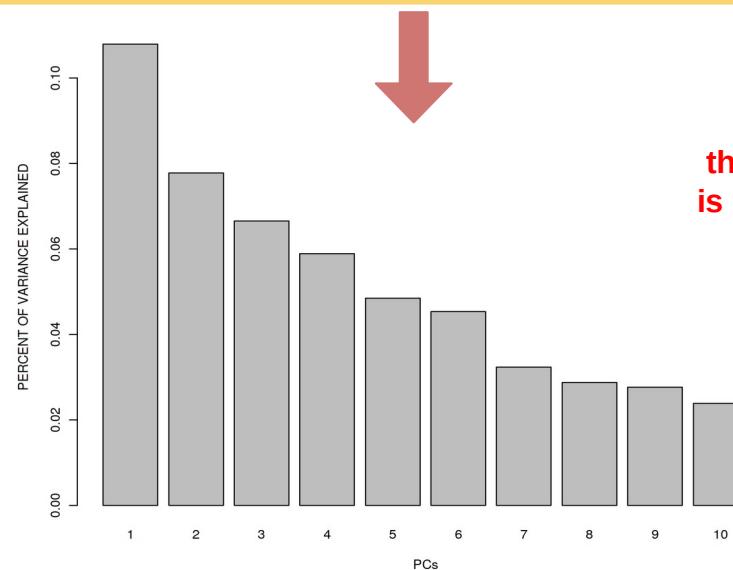
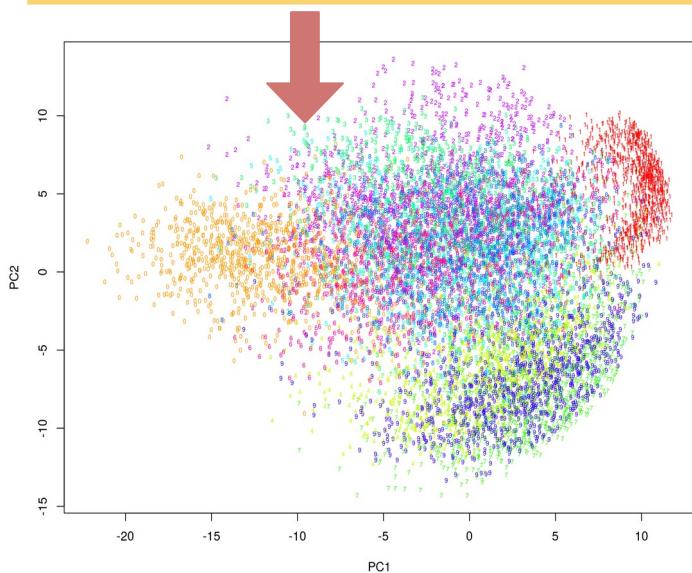
Mathematically:

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

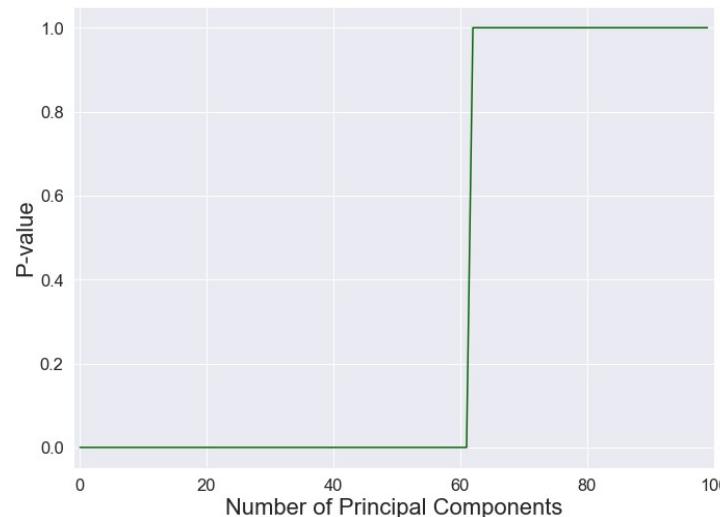
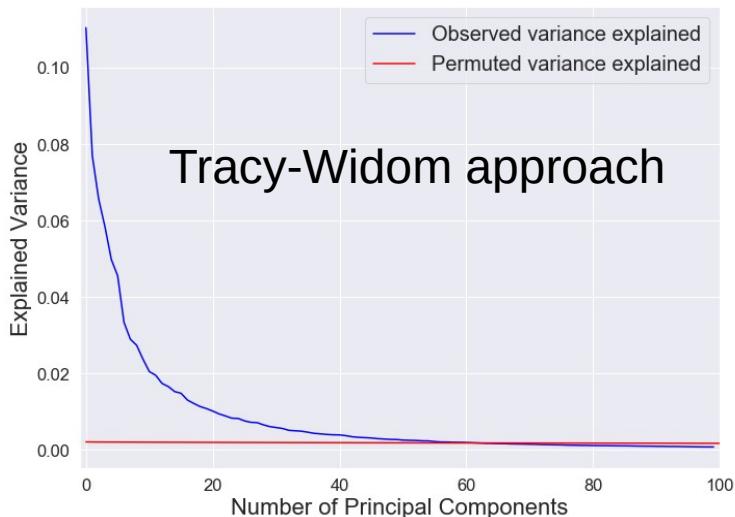
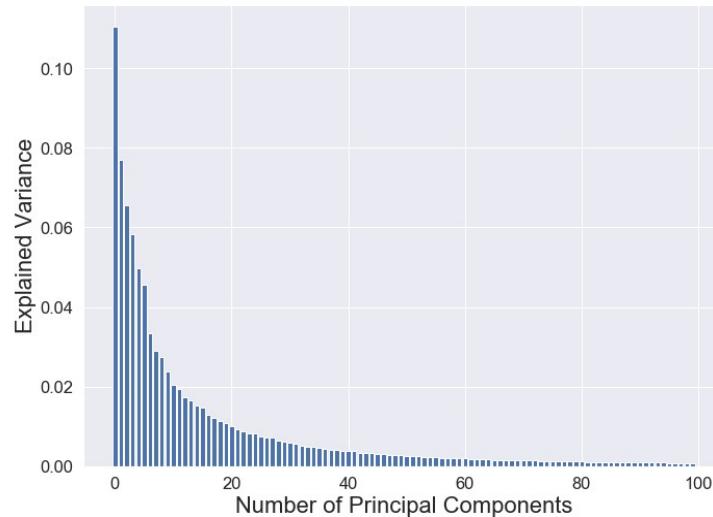
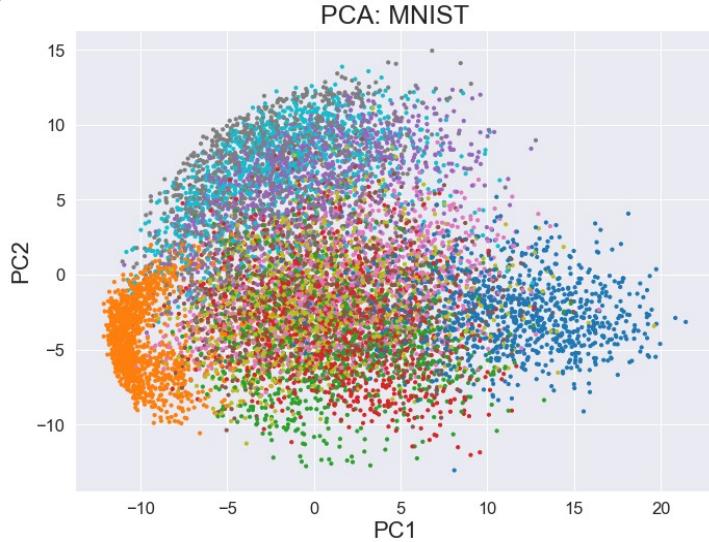
$$A = (1/N)M^T M$$

$$A^*u = \lambda^*u$$

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



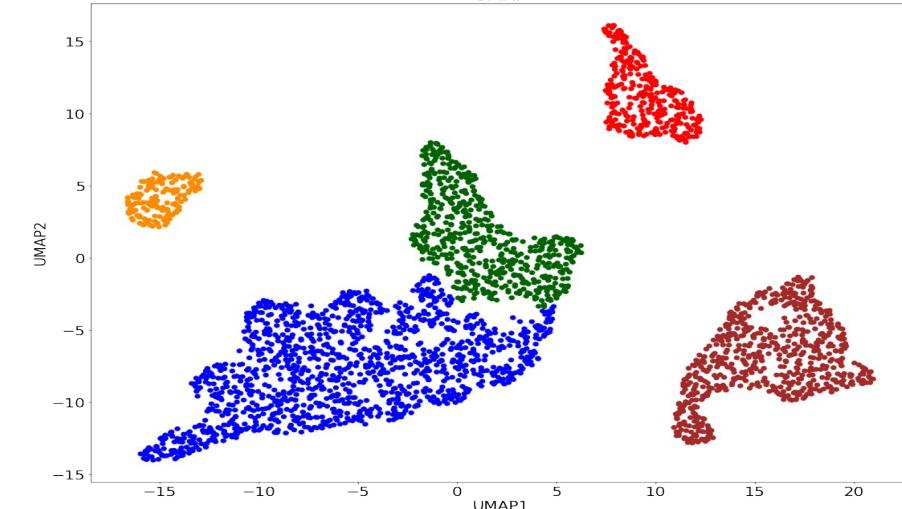
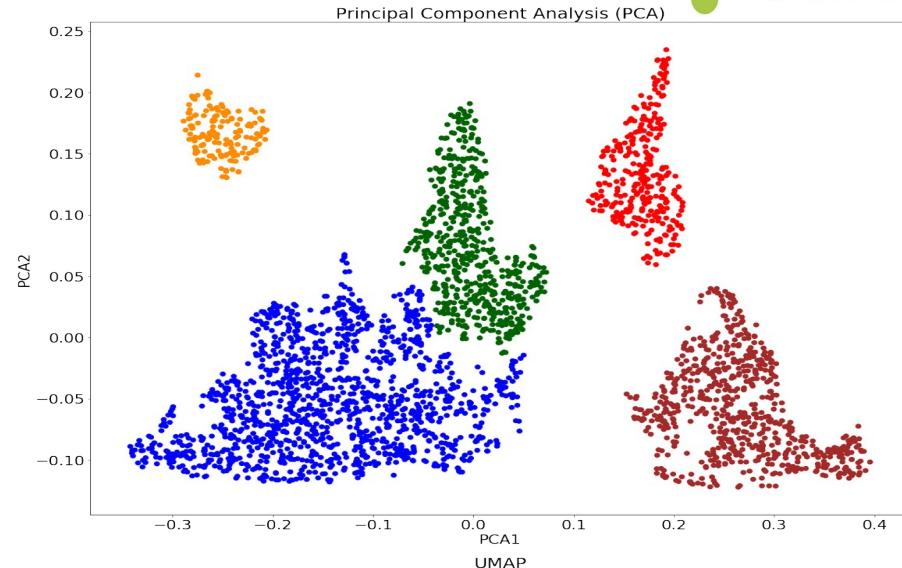
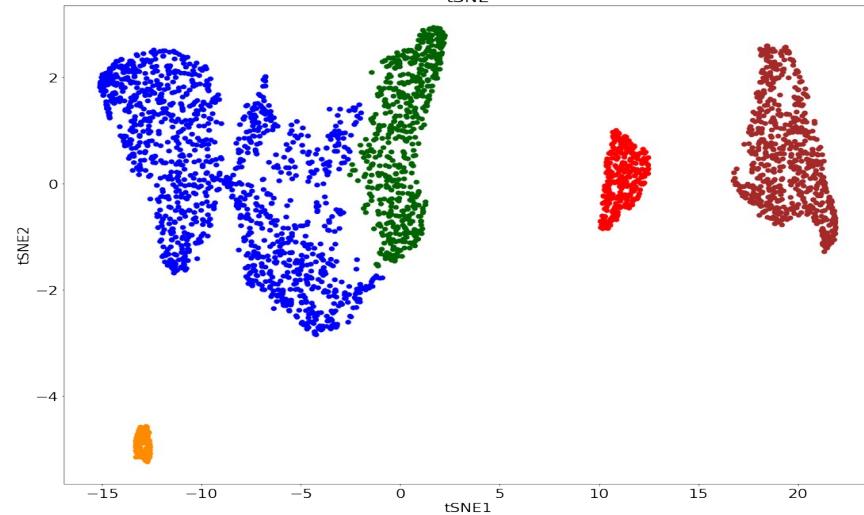
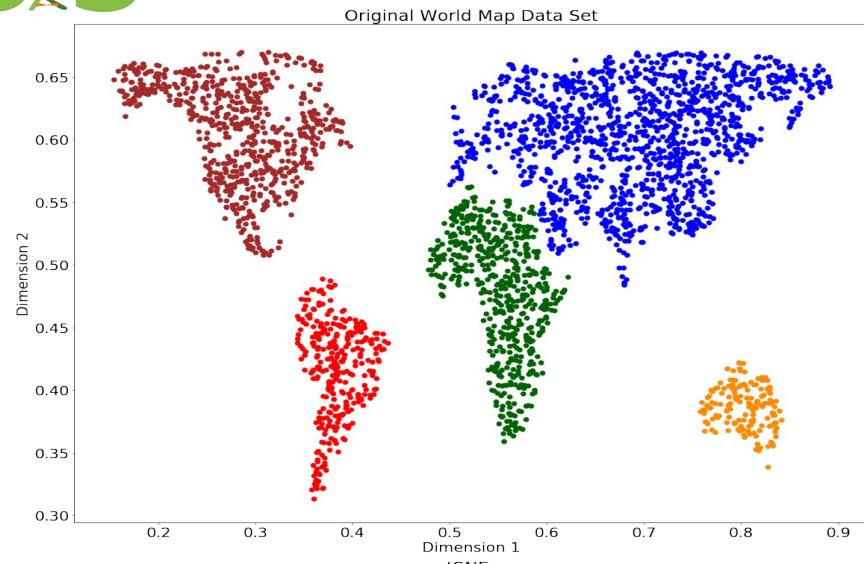
It can be analytically derived that the eigen value decomposition in PCA is equivalent to projecting data on axes of maximal variation in the data

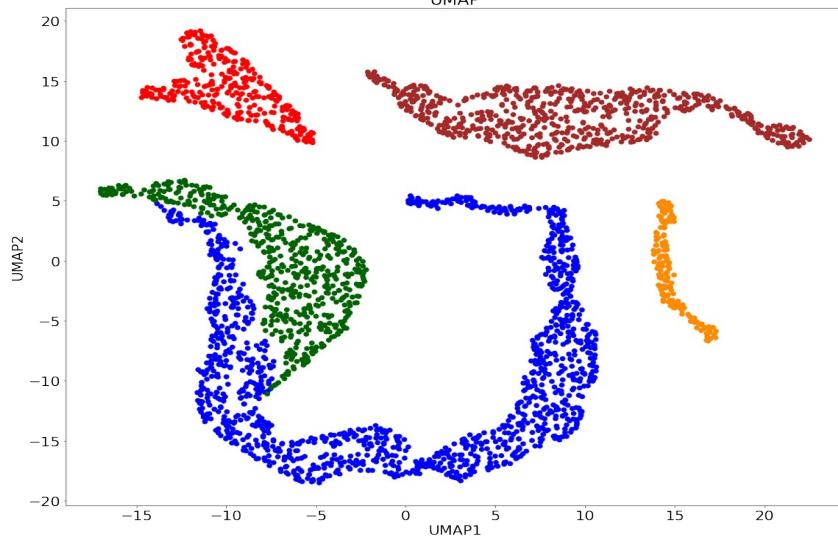
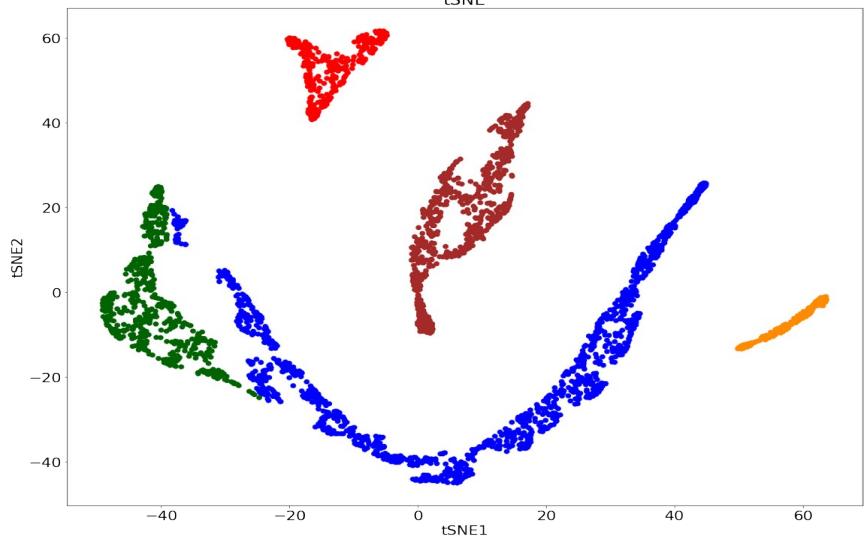
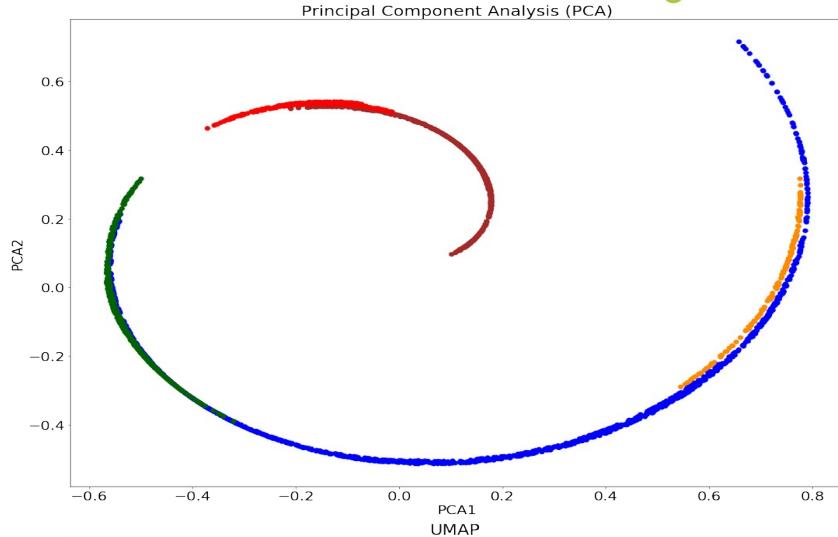
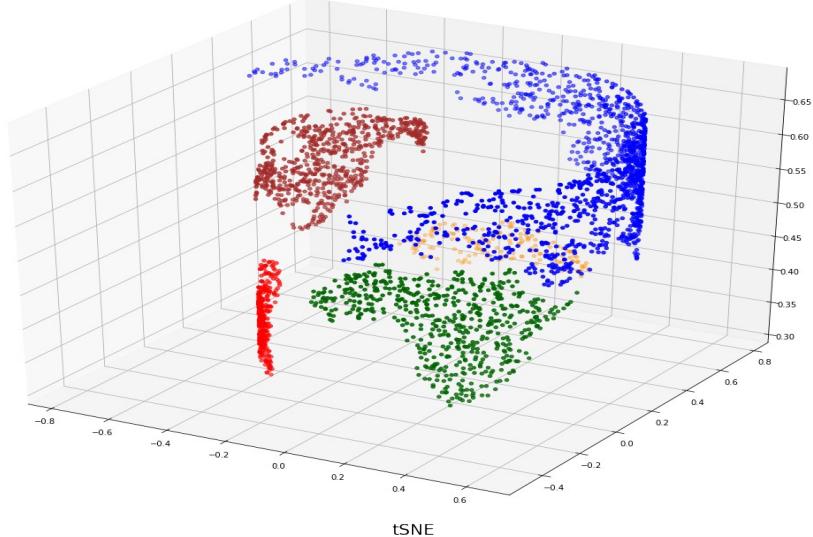


In Seurat:  
JackStraw

# When linear dimension reduction is not good enough

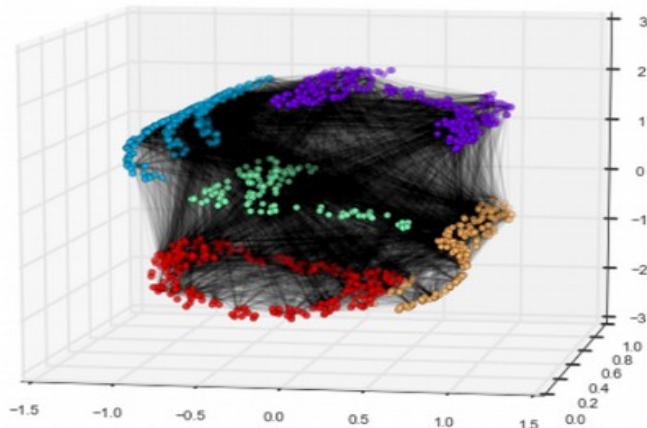
## PCA works fine on a linear manifold



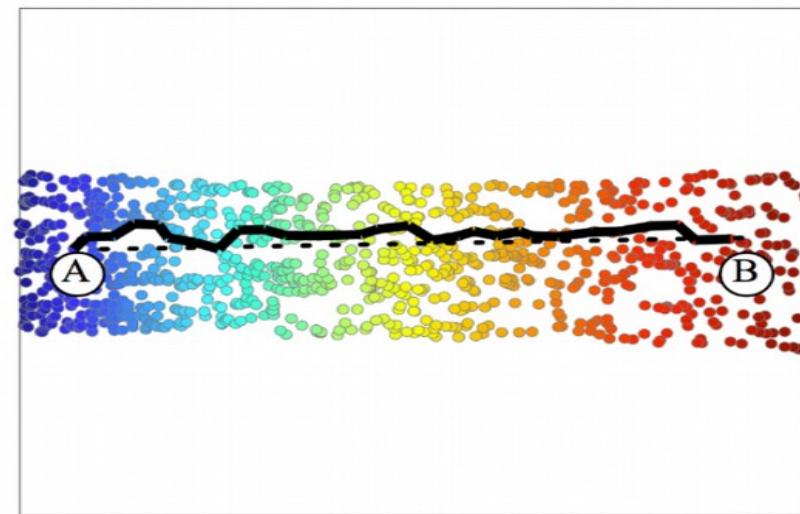
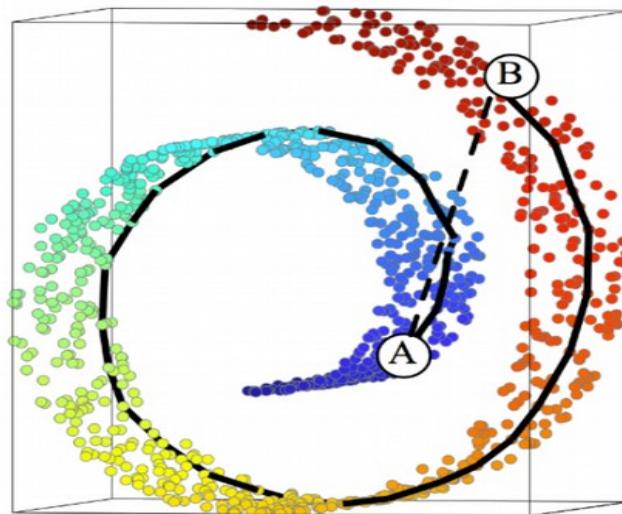
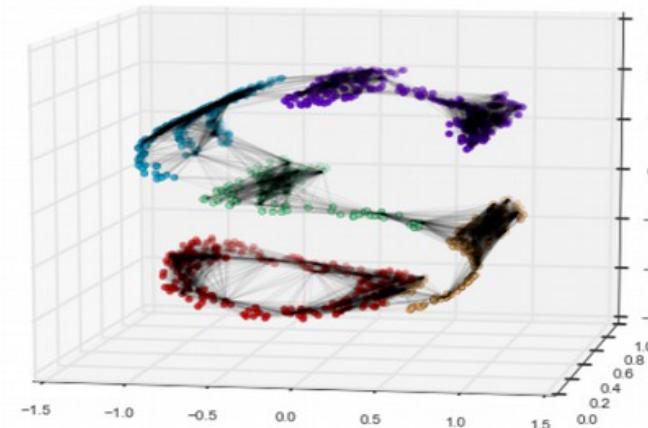


# Why PCA can't unwrap the Swiss Roll

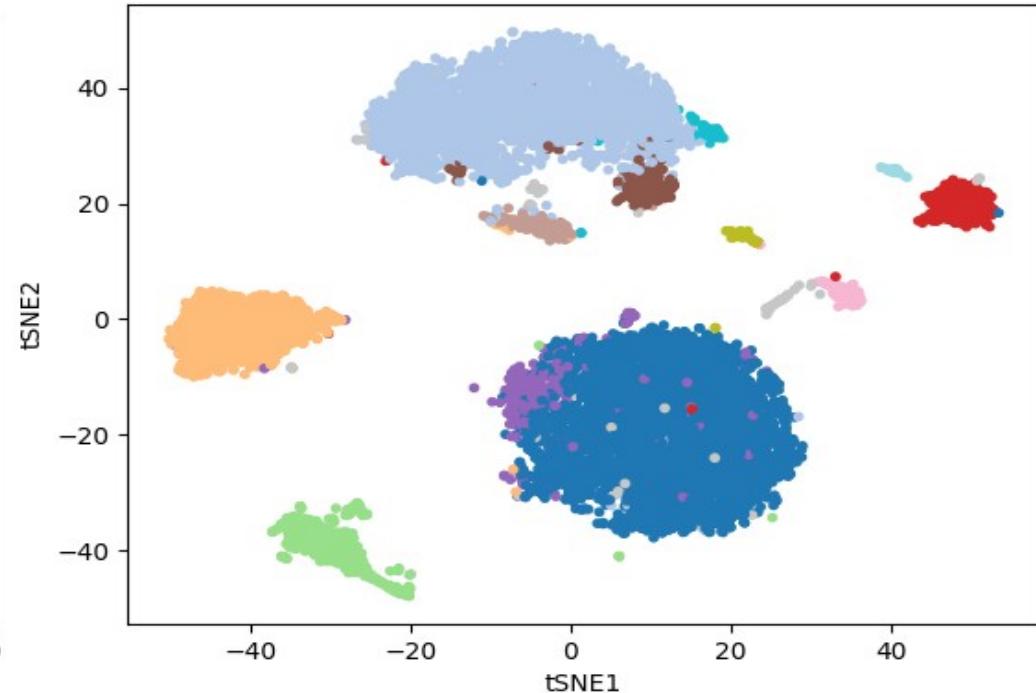
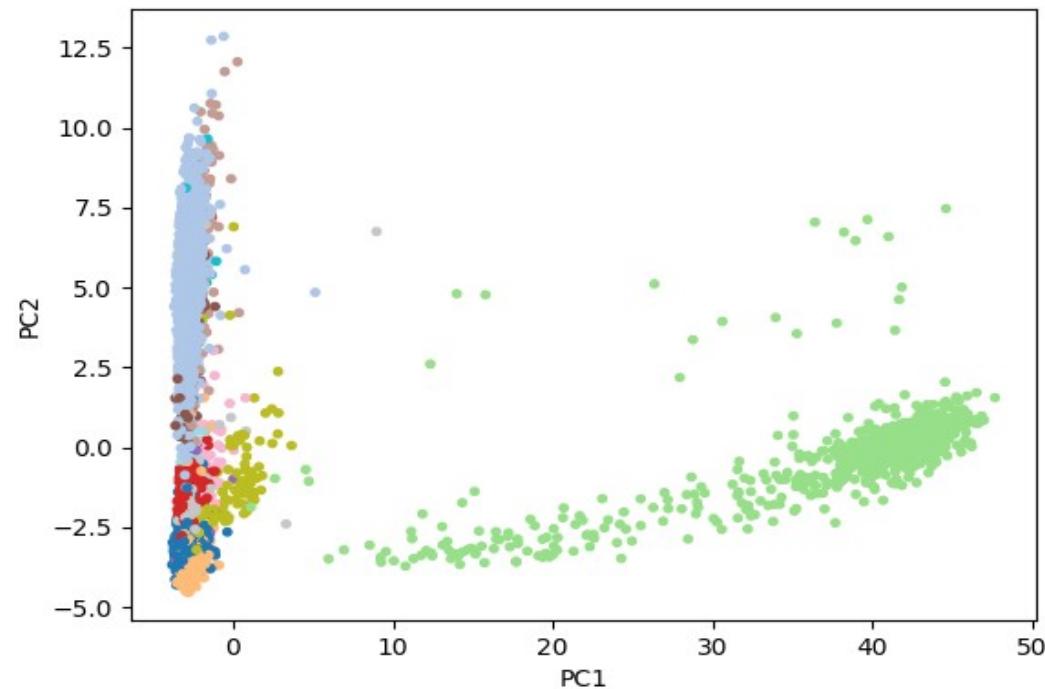
MDS Linkages



LLE Linkages (100 NN)



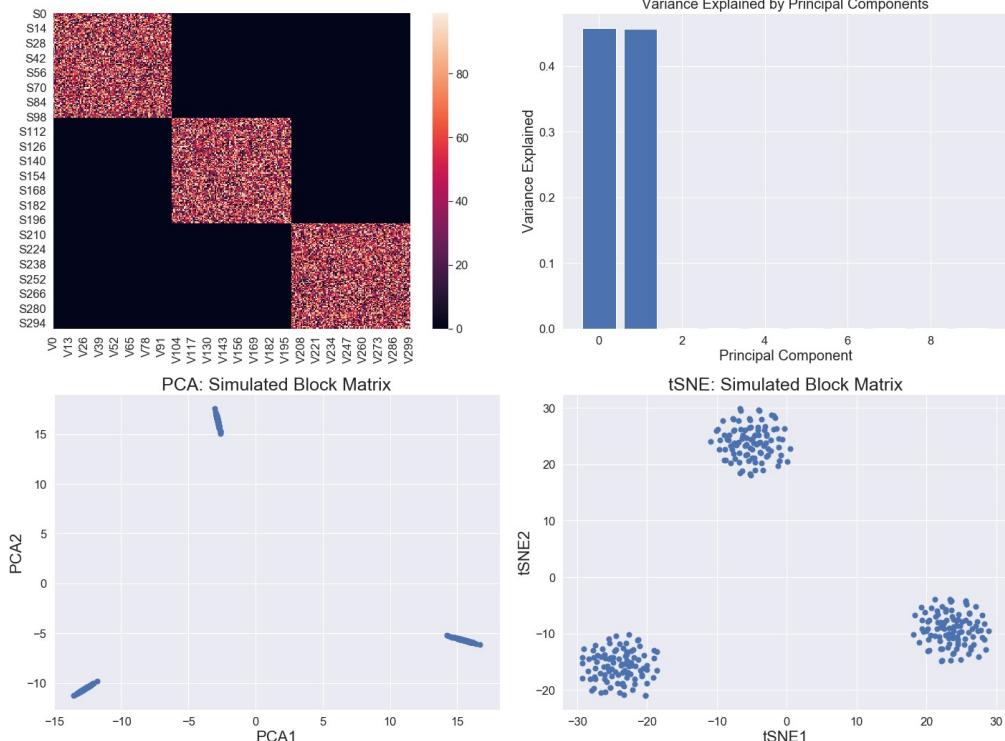
3k Peripheral Blood Mononuclear Cells (PBMC) available from 10X Genomics



Two principal components (PCs) seem to be insufficient to fully reveal heterogeneity in single cell gene expression data.

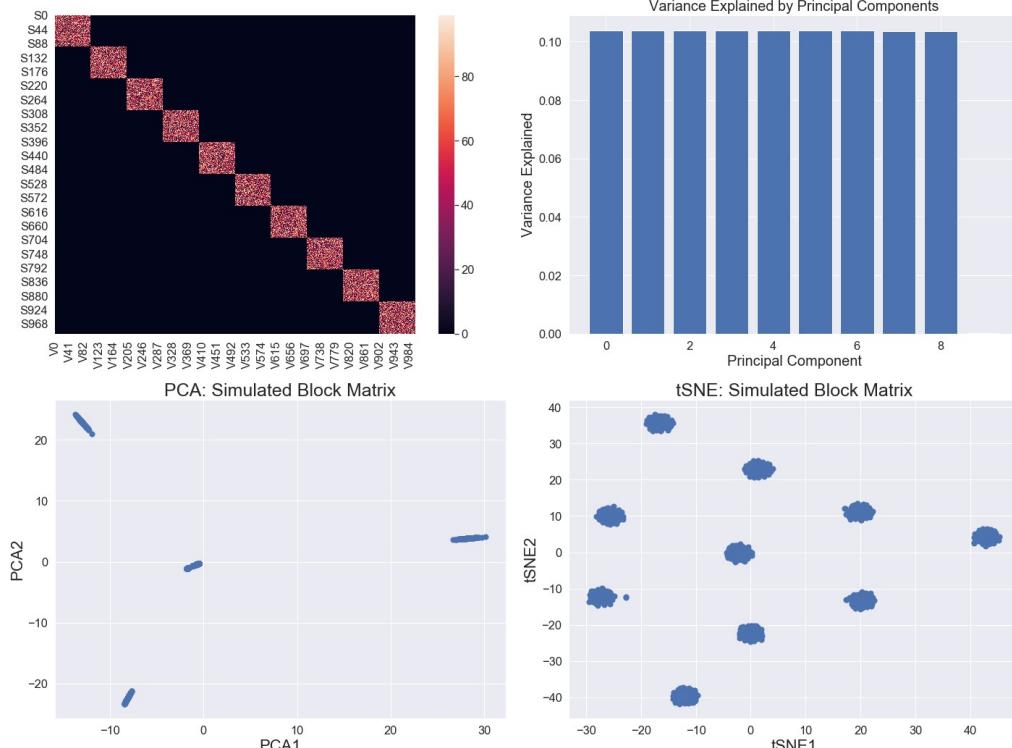
**Solution: use more PCs or tSNE / UMAP**

## Three classes of data points



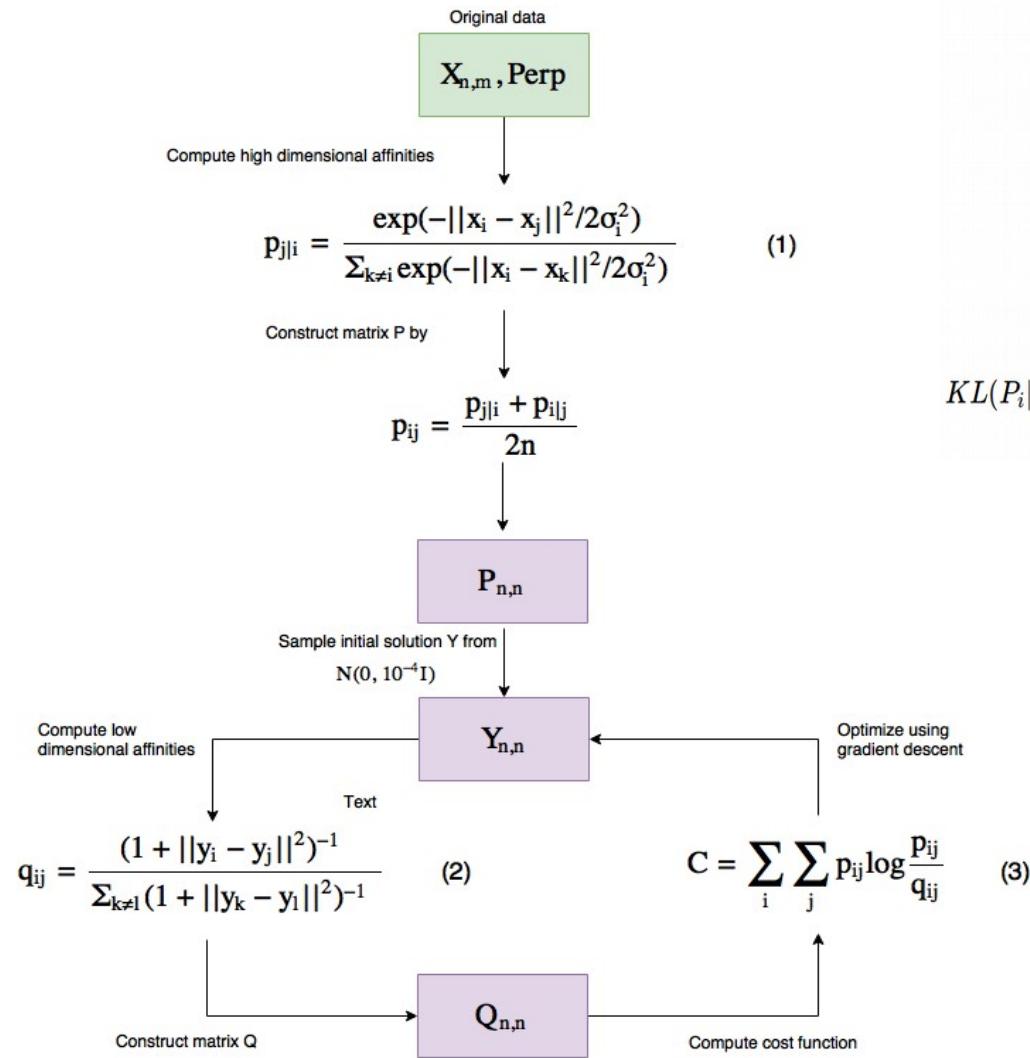
PCA and tSNE tell the same story

## Ten classes of data points



tSNE is more informative than PCA

# Neighborhood graph dimension reduction: tSNE and UMAP

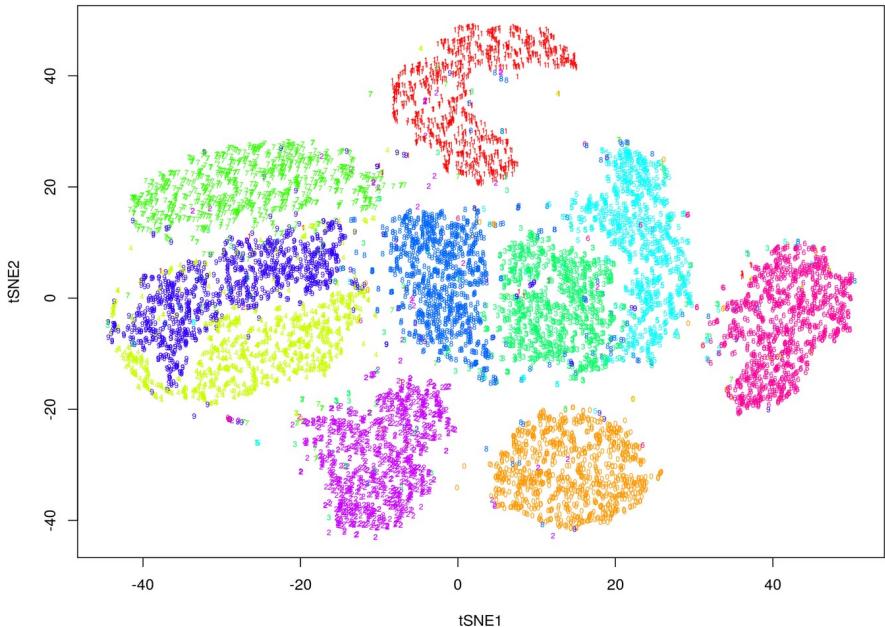


$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}, \quad p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N} \quad (1)$$

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

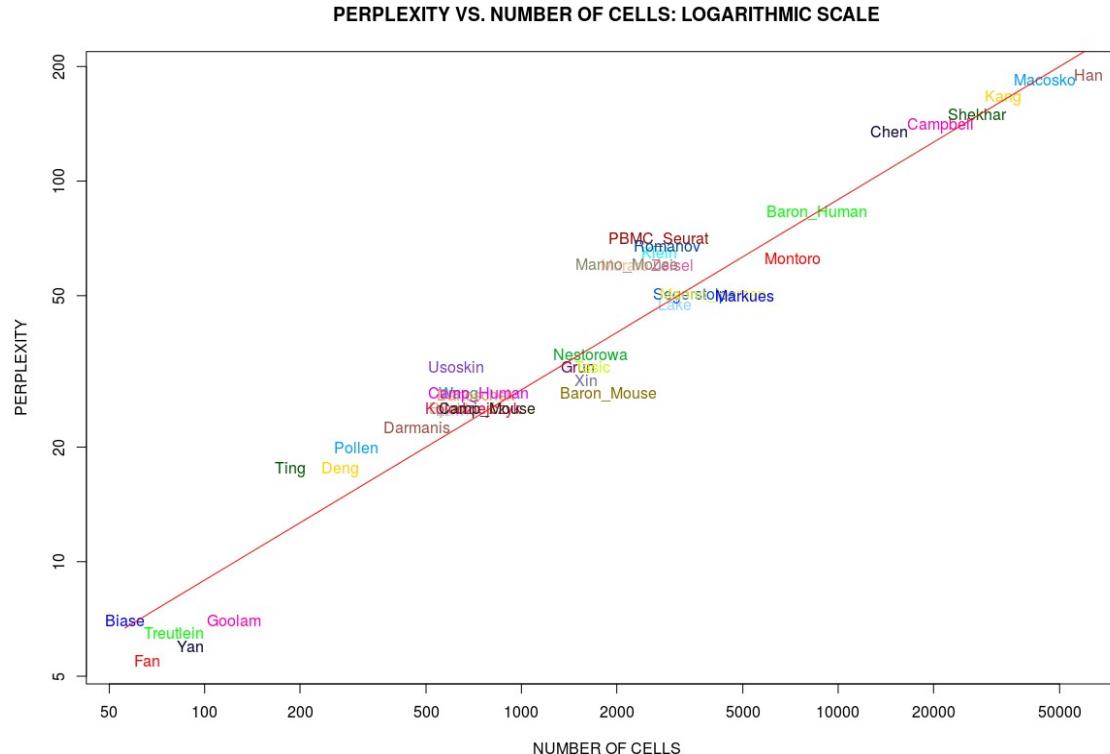
$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \quad (3)$$

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



# How to select optimal perplexity

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



$$\log(\text{Perp}) = -0.179 + 0.51 \cdot \log(N)$$

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

tSNE does not scale for large data sets?

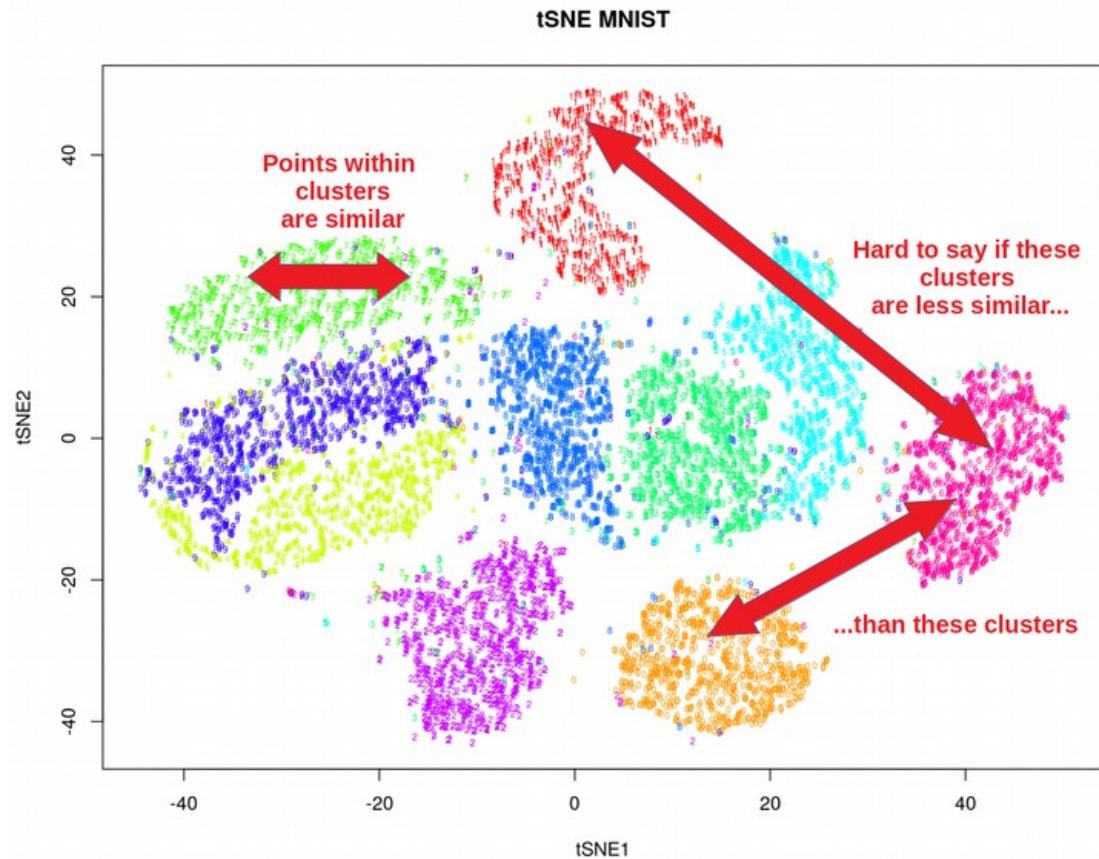
tSNE does not preserve global structure?

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 MNIST

UMAP does not normalize probabilities (speed-up)

UMAP can deliver a number of components for clustering

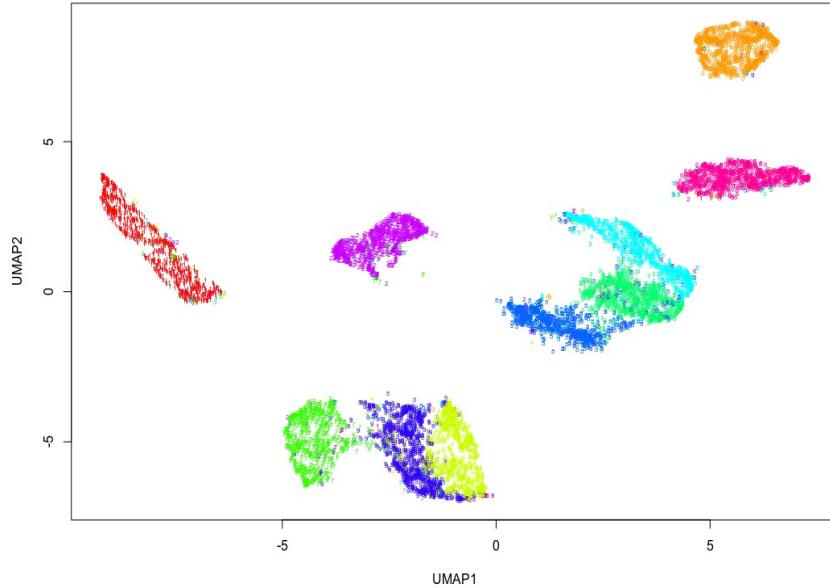
UMAP uses Laplacian Eigenmap for initialization

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

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

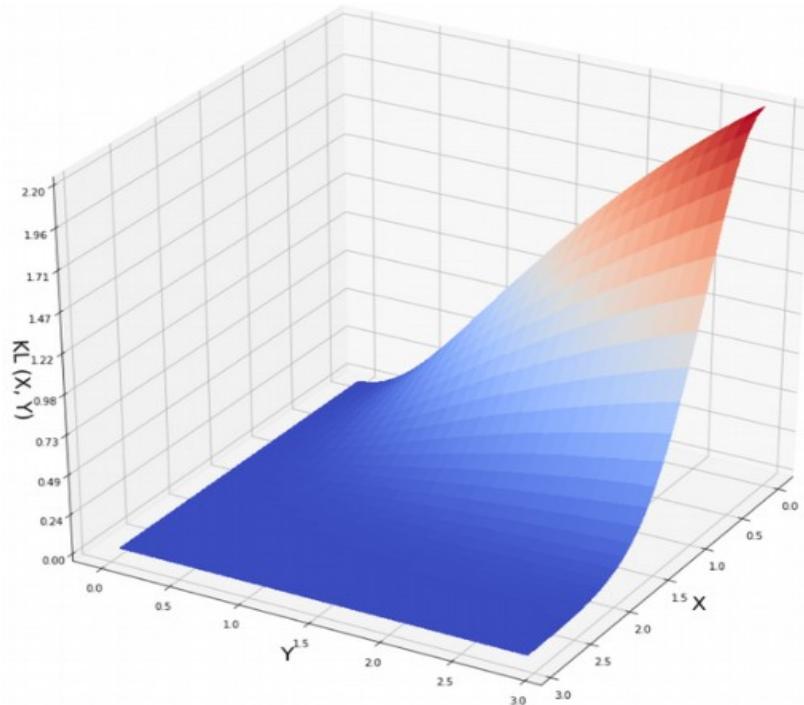
This is similar to tSNE cost function

This term is UMAP specific

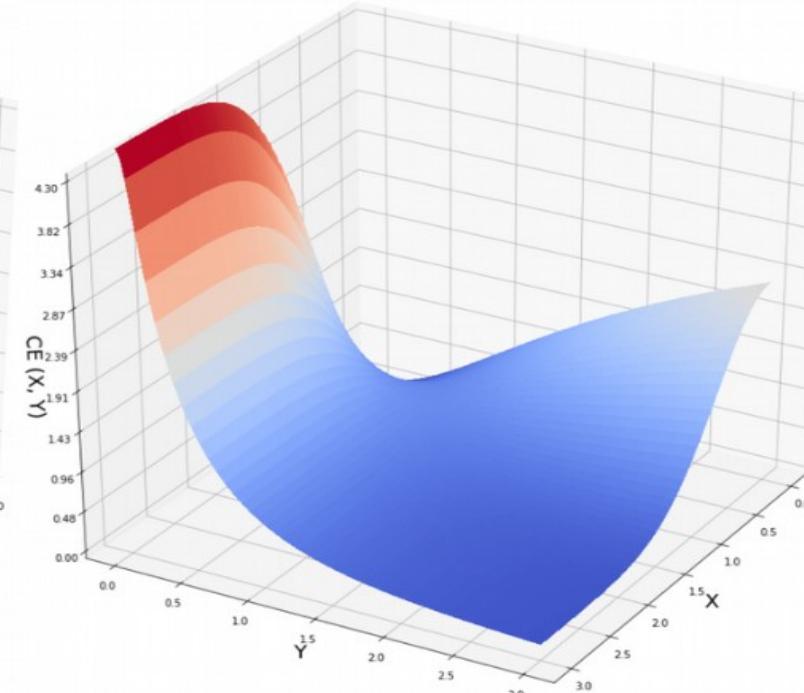


# **tSNE vs. UMAP: global structure preservation**

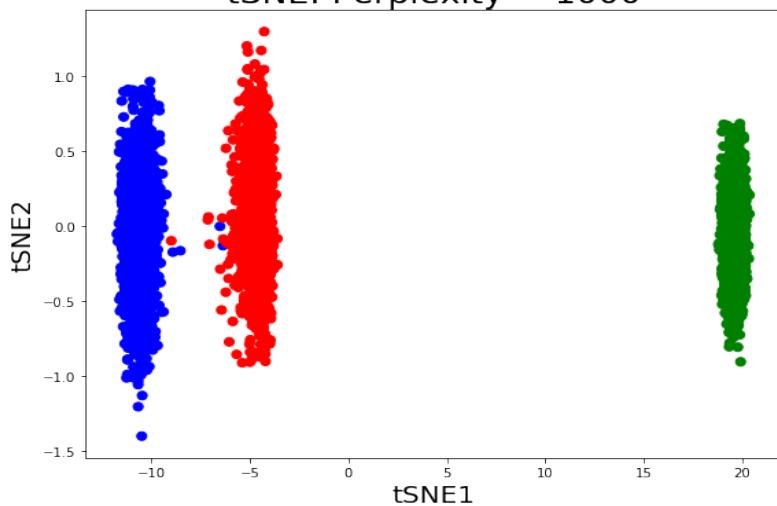
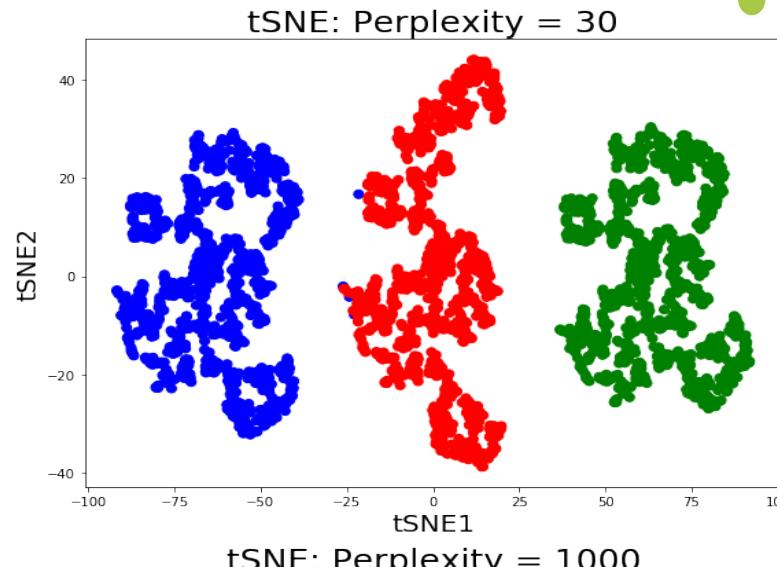
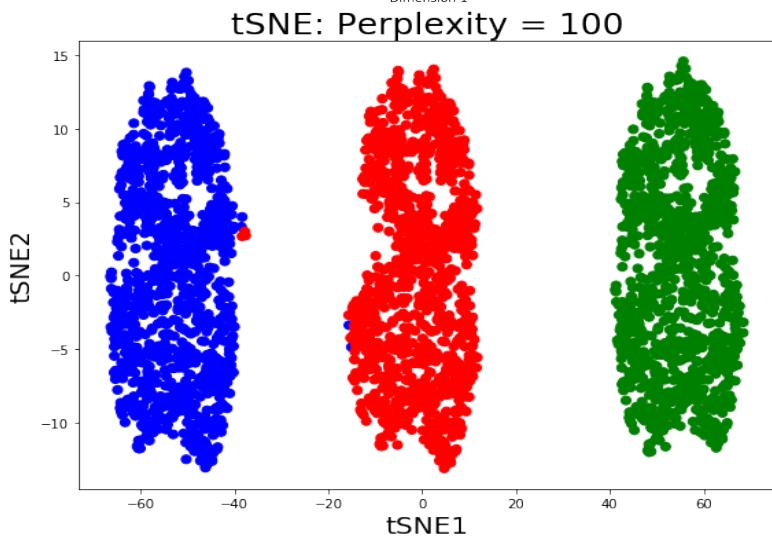
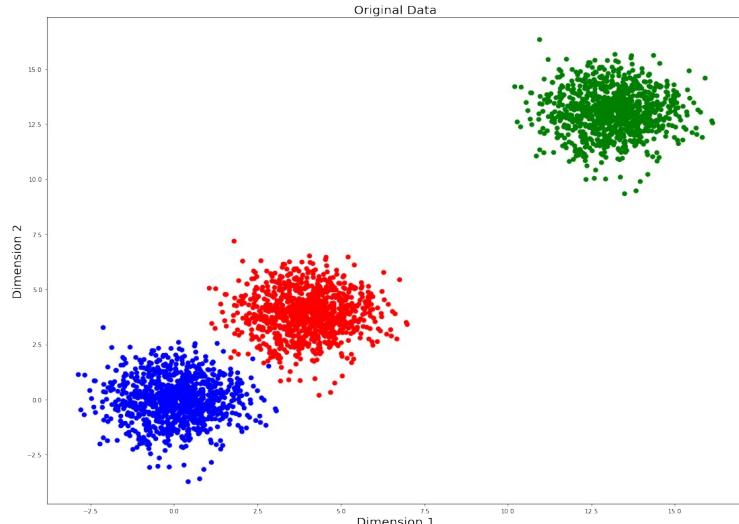
# Cost function seems to make UMAP preserve more of global structure than tSNE



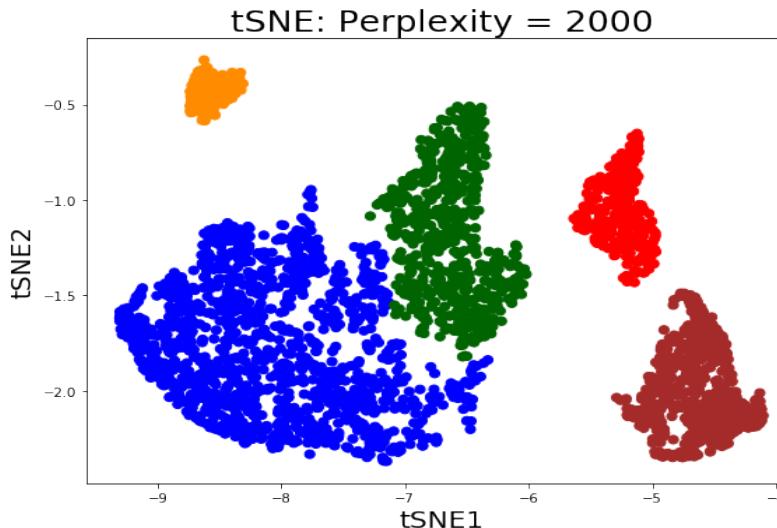
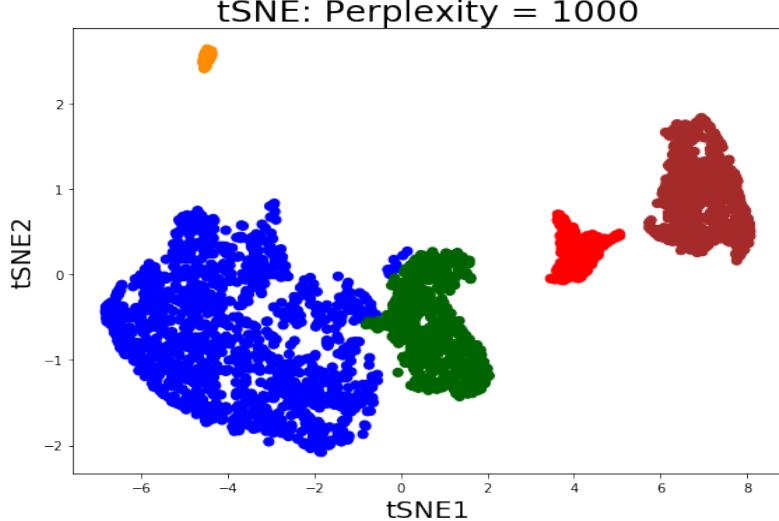
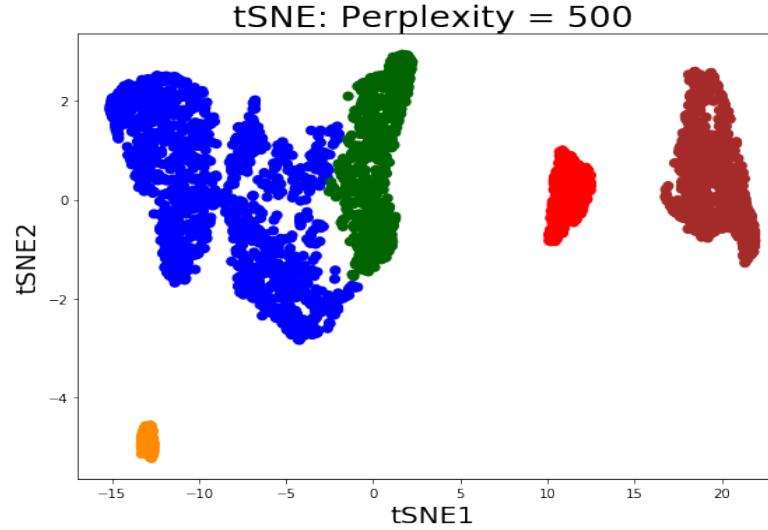
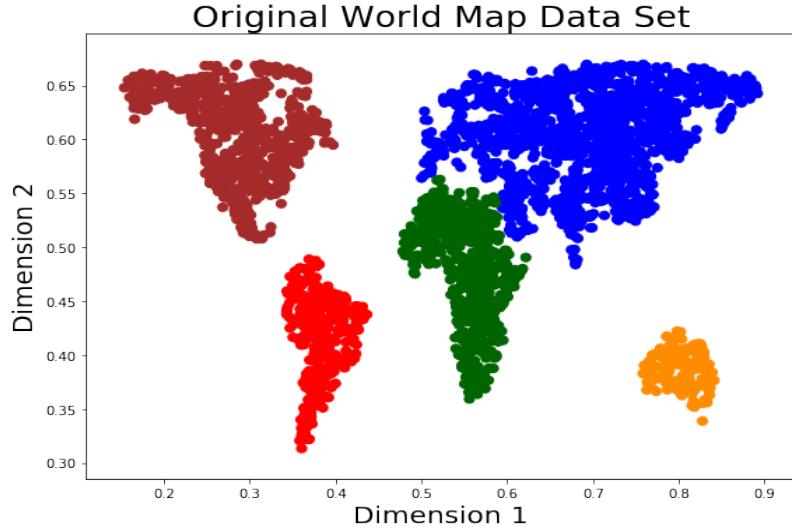
X → infinity, Y can be any



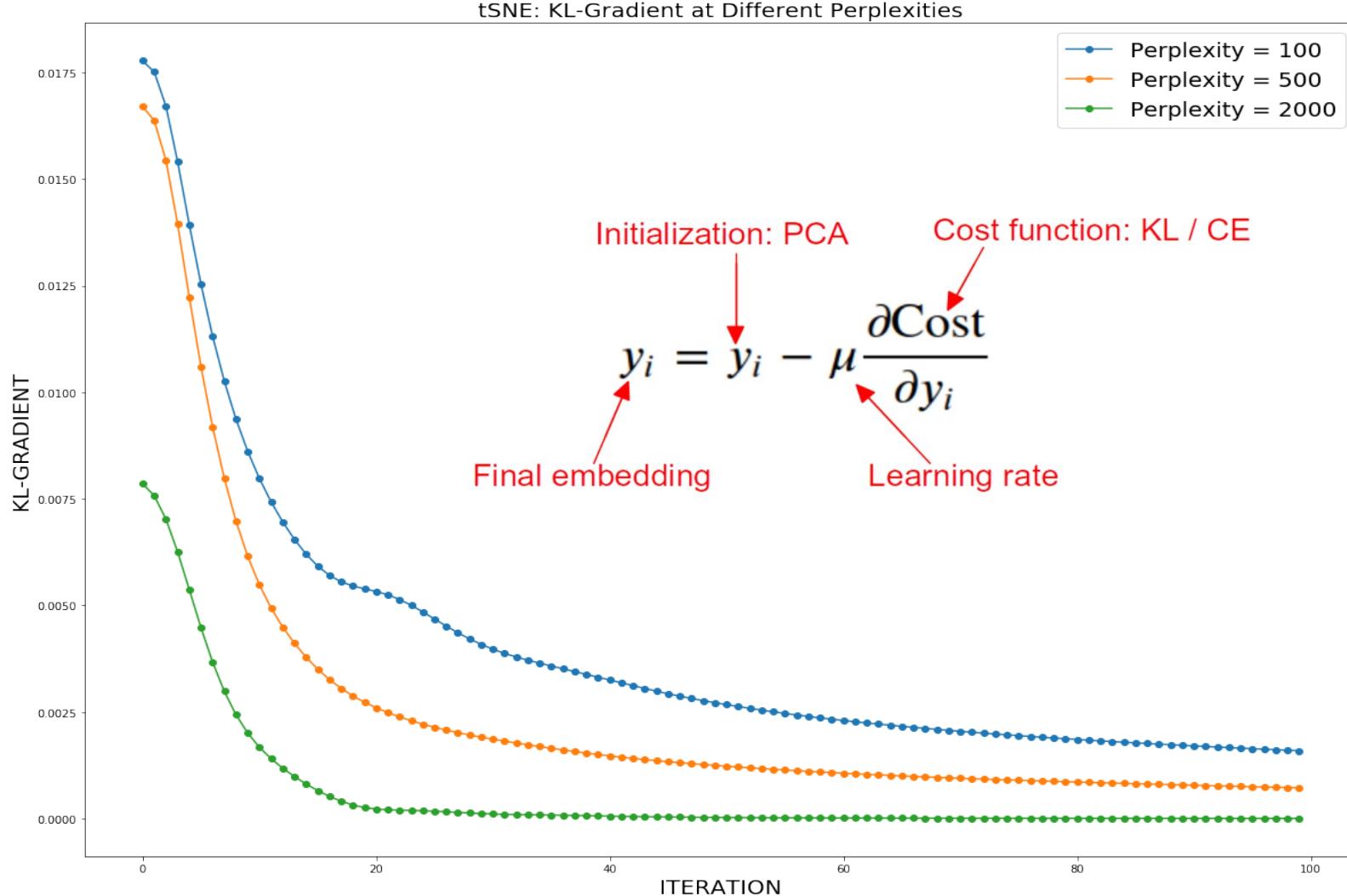
X → infinity, Y → infinity



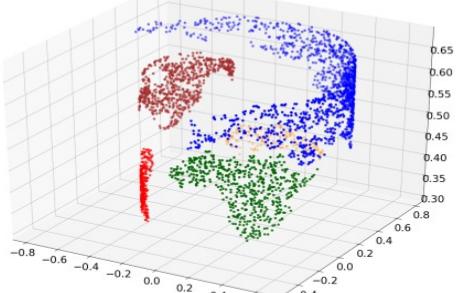
Can large perplexity  
solve the problem  
of global structure  
for tSNE?



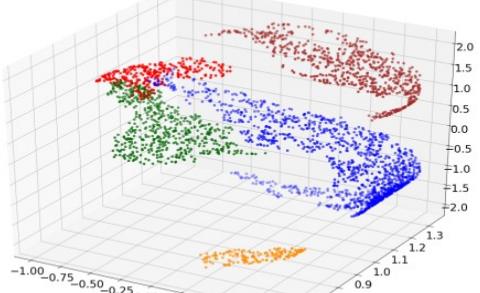
Can large perplexity solve the problem of global structure for tSNE?



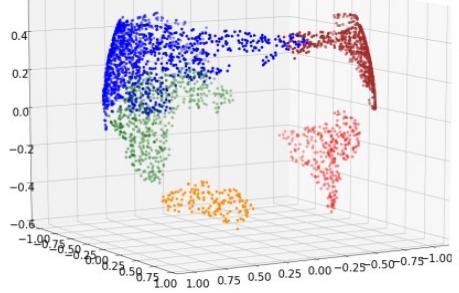
Swiss Roll: 3023 points



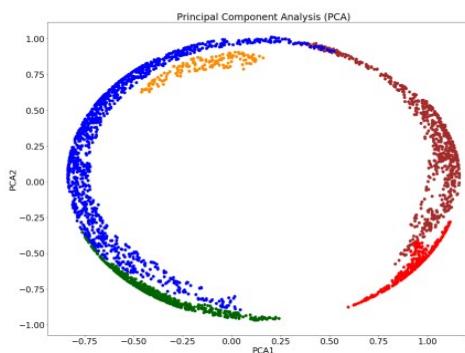
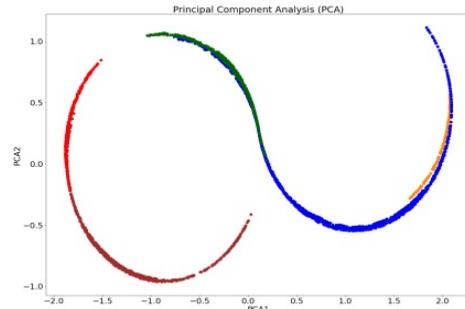
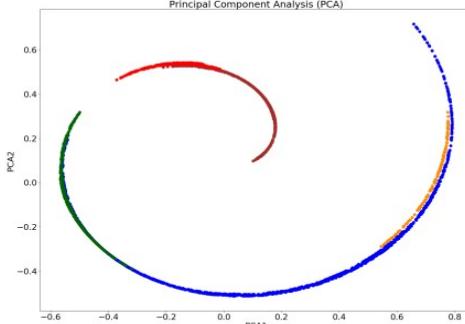
S-shape: 3023 points



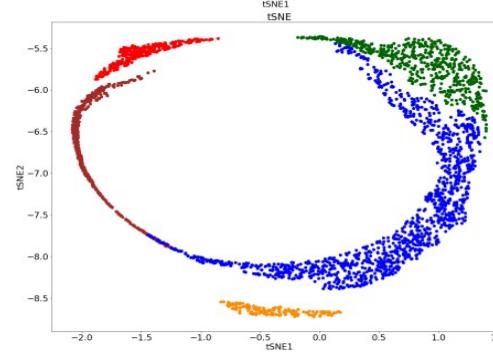
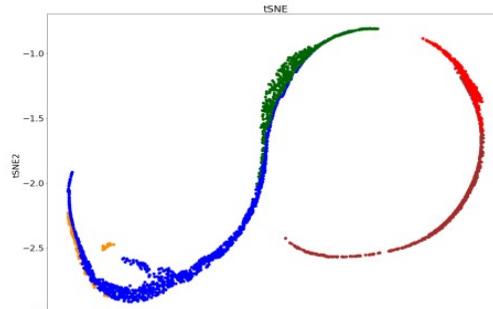
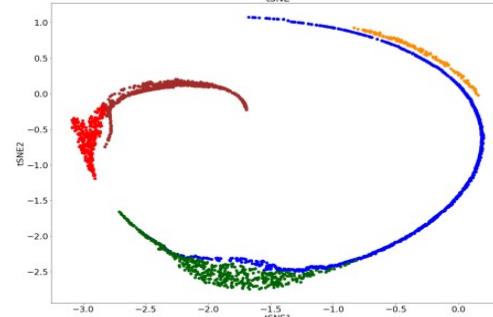
Sphere: 3023 points



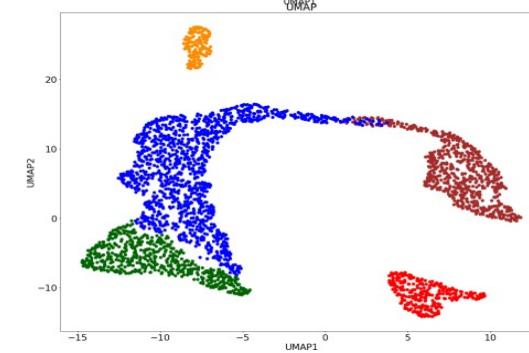
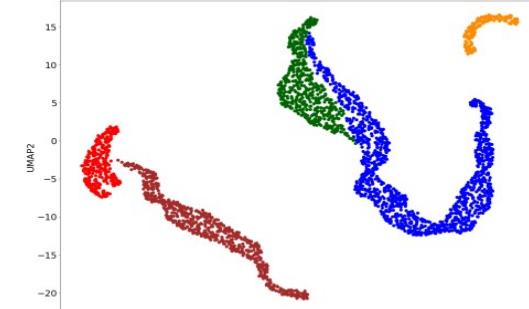
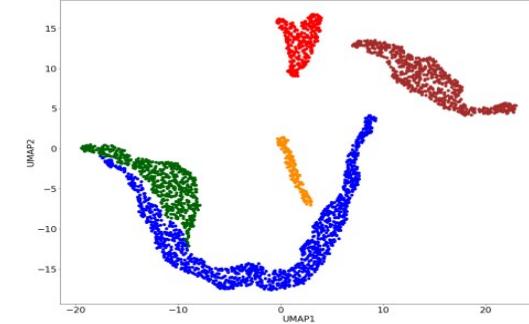
Principal Component Analysis



tSNE: perplexity = 2000

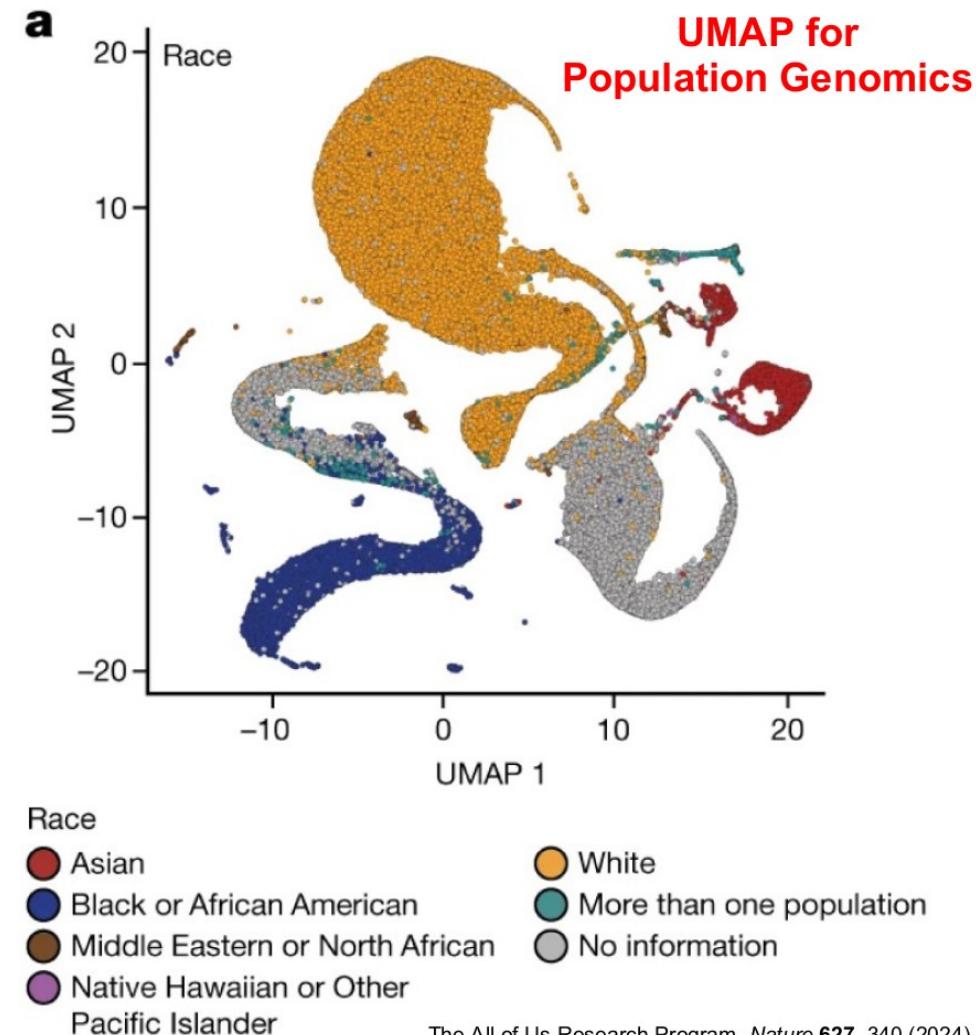
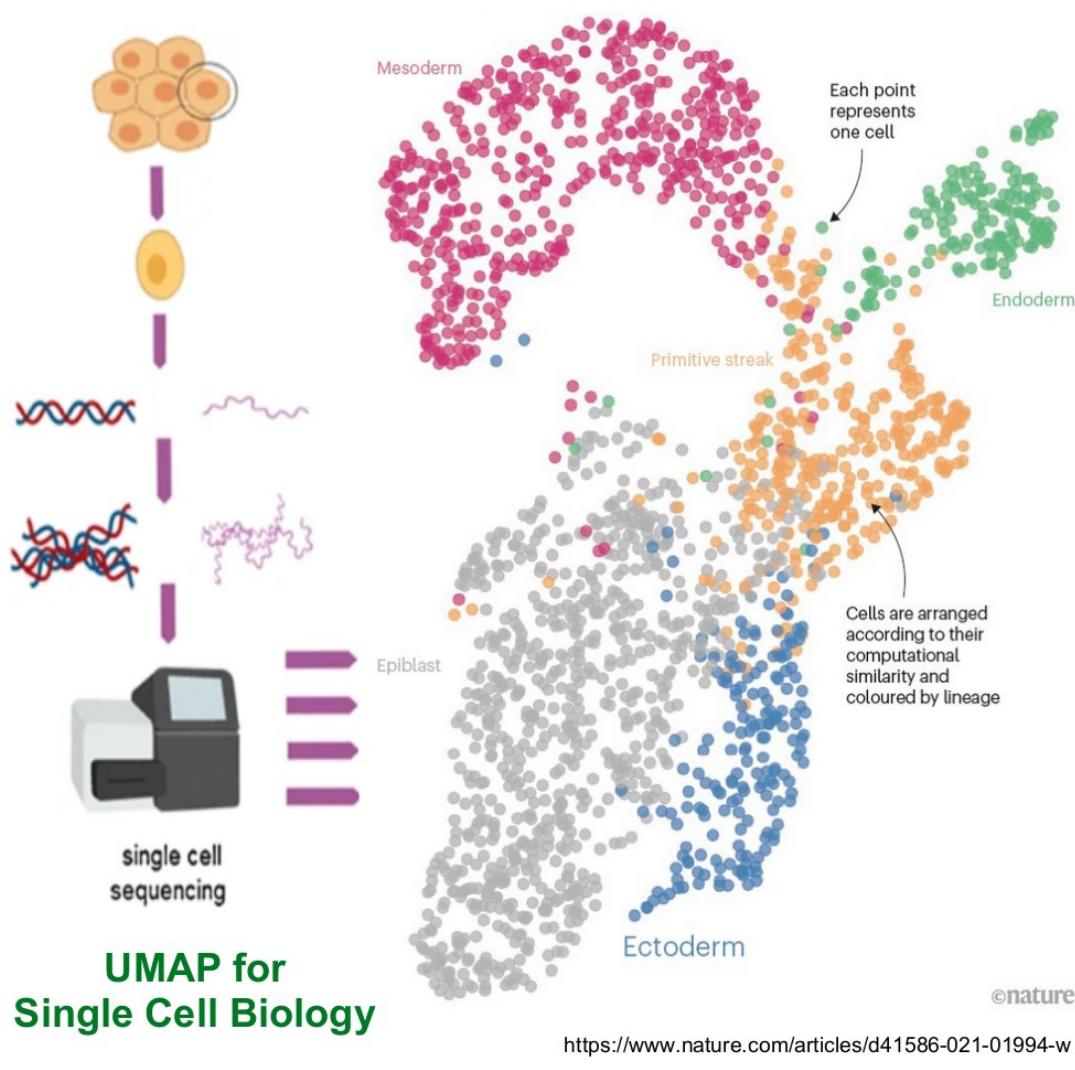


UMAP: n\_neighbor = 2000



# **UMAP for Population Genomics applications: addressing some recent fair and unfair criticism**

## UMAP: Single Cell vs. PopGen





## The specious art of single-cell genomics

Tara Chari<sup>1</sup>, Lior Pachter<sup>1,2\*</sup>

**1** Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, California, United States of America, **2** Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California, United States of America

\* [lpachter@caltech.edu](mailto:lpachter@caltech.edu)

### Abstract

Dimensionality reduction is standard practice for filtering noise and identifying relevant features in large-scale data analyses. In biology, single-cell genomics studies typically begin with reduction to 2 or 3 dimensions to produce “all-in-one” visualizations of the data that are amenable to the human eye, and these are subsequently used for qualitative and quantitative exploratory analysis. However, there is little theoretical support for this practice, and we show that extreme dimension reduction, from hundreds or thousands of dimensions to 2, inevitably induces significant distortion of high-dimensional datasets. We therefore examine the practical implications of low-dimensional embedding of single-cell data and find that extensive distortions and inconsistent practices make such embeddings counter-productive for exploratory, biological analyses. In lieu of this, we discuss alternative approaches for conducting targeted embedding and feature exploration to enable hypothesis-driven biological discovery.

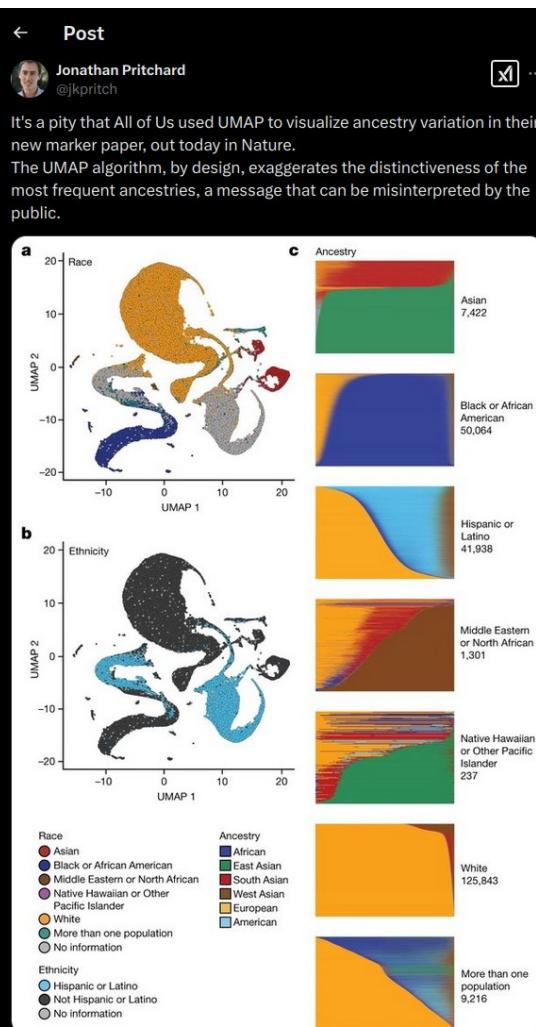
### Introduction

The high-dimensionality of “big data” genomics datasets has led to the ubiquitous application of dimensionality reduction to filter noise, enable tractable computation, and to facilitate exploratory data analysis (EDA). Ostensibly, the goal of this reduction is to preserve and extract local and/or global structures from the data for biological inference [1–3]. Trial and error application of common techniques has resulted in a currently popular workflow combining initial dimensionality reduction to a few dozen dimensions, often using principal component analysis (PCA), with further nonlinear reduction to 2 dimensions using t-SNE [4] or UMAP [1,5,6]. For single-cell genomics in particular, these embeddings are used extensively in qualitative and quantitative EDA tasks that fall into 4 main categories of applications (Fig 1, “Application”):

- Modality-mixing, integration, and reference mapping:

Embeddings are used to visually assess the extent of integration, mixing, or similarities between cells from different batches [7–9] and to compare methods of integration/batch-correction [10]. For query dataset(s) mapped onto reference datasets/embeddings, visuals likewise provide an assessment of merged data similarities or differences [11,12].

- Cluster validation and relationships:



# Biologists, stop putting UMAP plots in your papers

UMAP is a powerful tool for exploratory data analysis, but without a clear understanding of how it works, it can easily lead to confusion and misinterpretation.

**AUTHOR**  
Rafael Irizarry  
**PUBLISHED**  
Dec. 23, 2024

**Rafael A. Irizarry**  
Professor of Biostatistics  
Biostatistics, Harvard T.H. Chan School of Public Health  
Departments  
Department of Biostatistics

```
library(Matrix)
library(ggplot2)
library(dplyr)
library(umap)
set.seed(2024-6-21)
load("rda/pop_gen_sample.RData")
```

## The UMAP craze in single cell RNA-Seq

Single-cell RNA sequencing (scRNA-seq) has become one of the most widely used technologies in basic biology. With the rise of scRNA-seq, the use of UMAP has become ubiquitous in publications. While this dimensionality reduction technique is useful for exploratory data analysis, its overuse and misinterpretation have led to confusion and

**Going through the post of Rafael Irizarry:  
Point1: UMAP makes artificial clusters!**

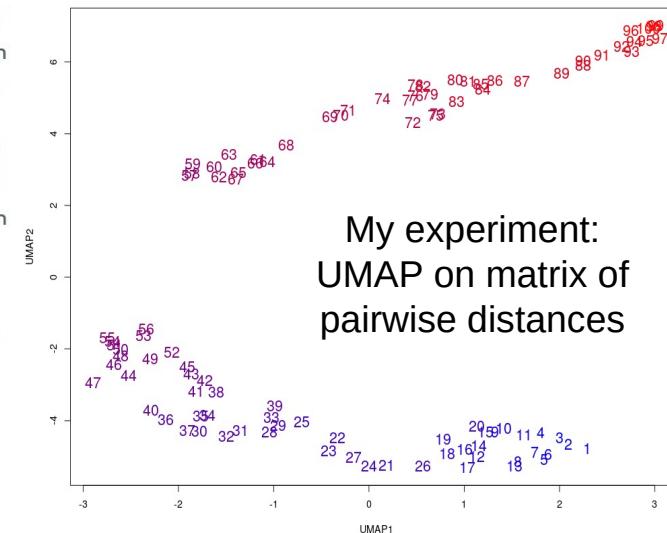
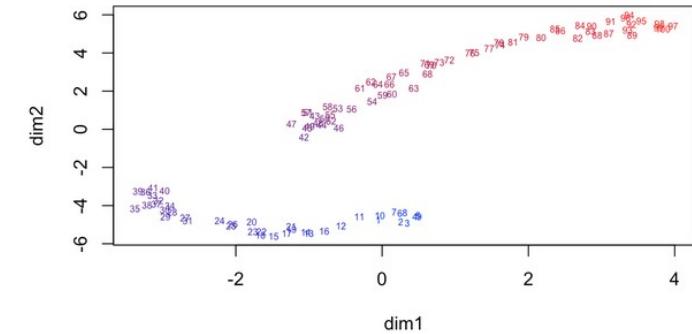
The issue becomes more significant when the underlying mathematics of UMAP is not fully understood. UMAP takes a  $p$ -dimensional vector of numeric values, such as gene expression in scRNA-Seq, and applies a mathematical transformation to produce two values, resulting in the two coordinates shown in the plot. But what exactly is this function? Do the authors who include these plots in papers fully understand the mathematics behind it? What genes are included in the calculation and how? How exactly does distance in the two dimensional summary relate to the actual distance in  $p$ -dimensional space? The actual summary function is rarely if ever explained, leaving readers uncertain about what the plot truly represents.

Additionally, UMAP is highly sensitive and can create separations in data that shouldn't necessarily exist. For example, consider applying UMAP to 100 randomly generated points from a multivariate normal distribution representing three correlated random variables:

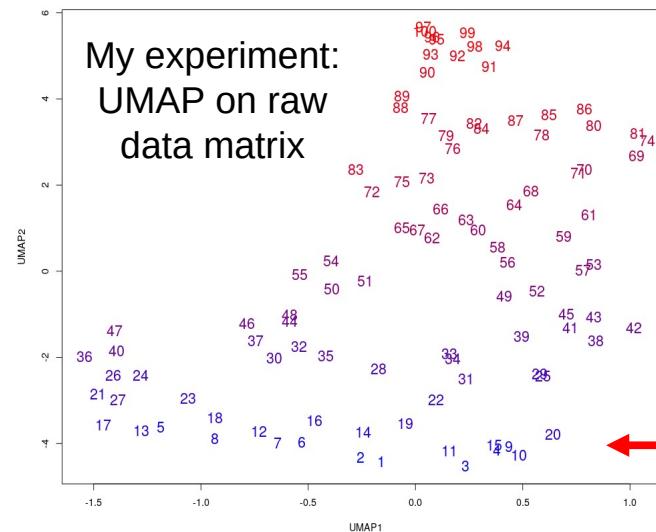
```

Sigma <- matrix(.8, 3, 3); diag(Sigma) <- 1
x <- MASS::mvrnorm(100, rep(0,3), Sigma)
#x <- matrix(rnorm(100), ncol = 1)
u <- umap(as.matrix(dist(x)))
ranks <- rank(rowMeans(x))
colors <- colorRampPalette(c("blue", "red"))(nrow(x))
colormap <- colors[ranks]
plot(u$layout[,1], u$layout[,2], type = "n", xlab = "dim1", ylab = "dim2")
text(u$layout[,1], u$layout[,2], labels = ranks, col = colormap, cex = 0.5)

```



My experiment:  
UMAP on matrix of  
pairwise distances



My experiment:  
UMAP on raw  
data matrix

Post

this is the output (with "dist" on the left, without "dist" on the right)

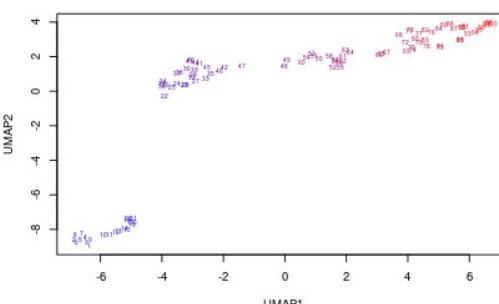
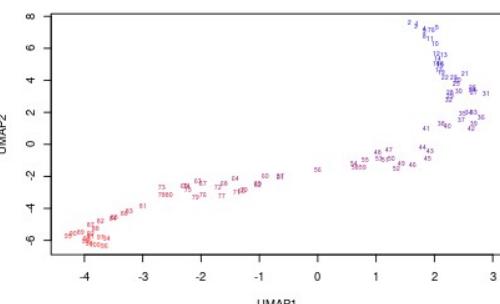
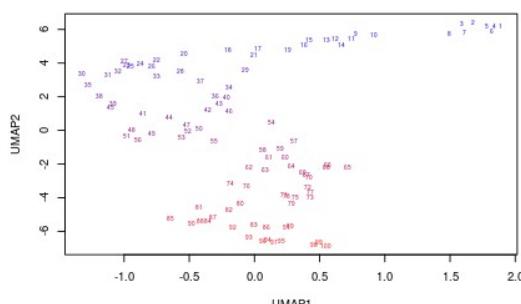
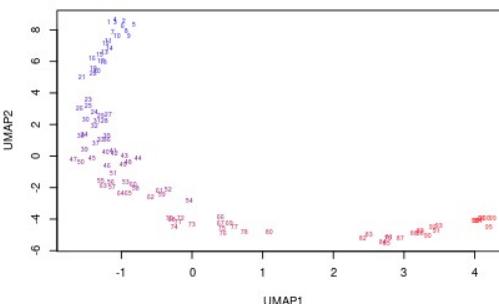
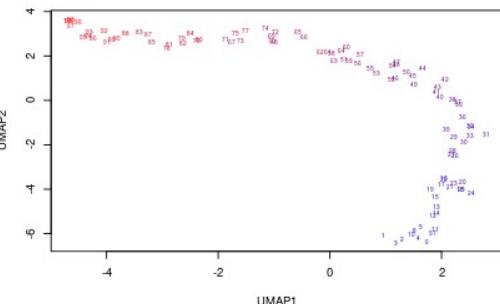
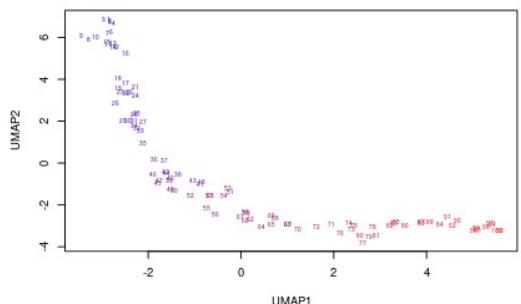
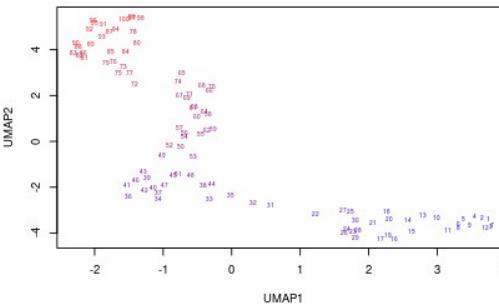
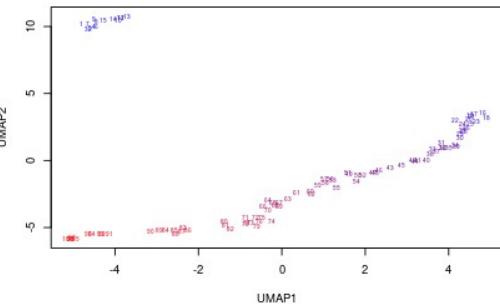
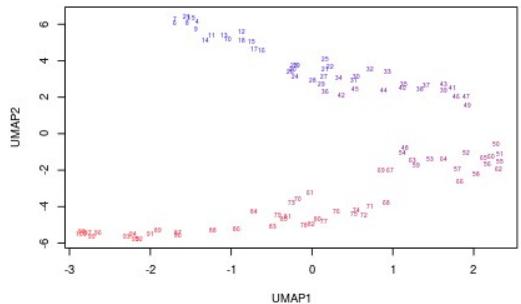
Rafael Irizarry @rafalab · 1h  
My recollection is that the version I was using took distance as input. Maybe I was wrong. So I updated to the latest, changed code to explicitly tell UMAP the input is a distance matrix, clarify that not every simulation results in separation & thank you in the acknowledgements.

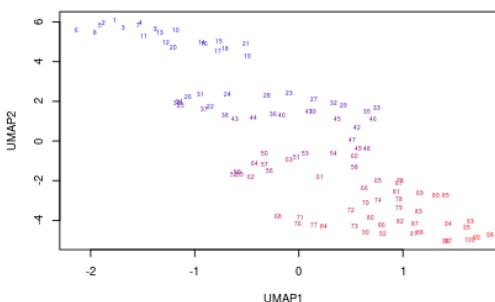
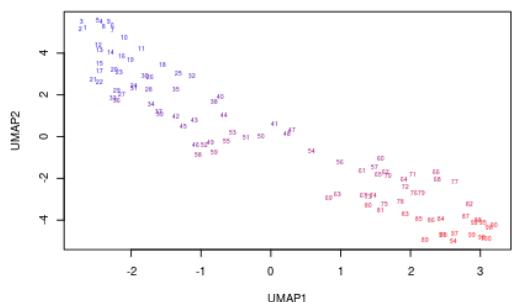
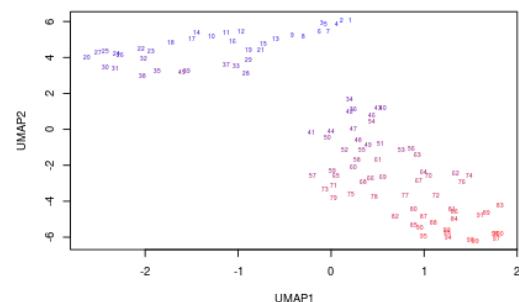
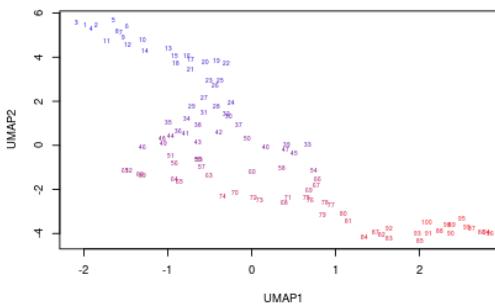
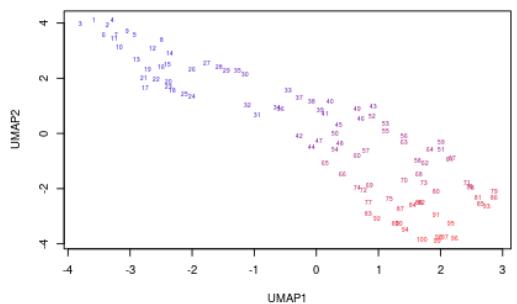
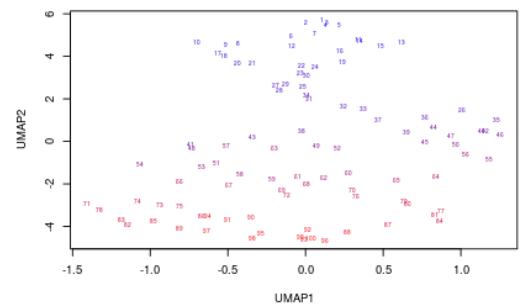
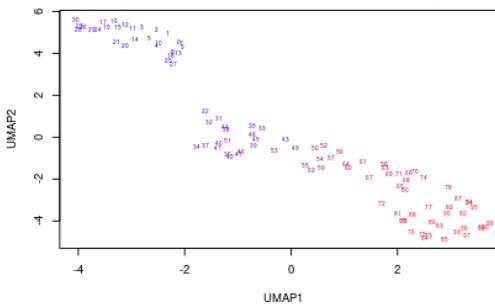
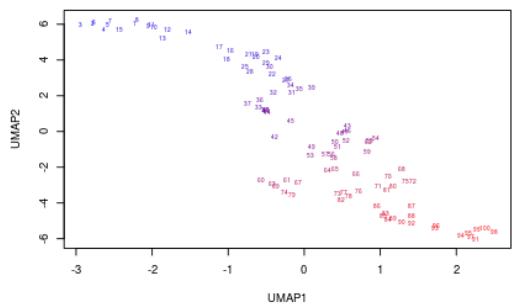
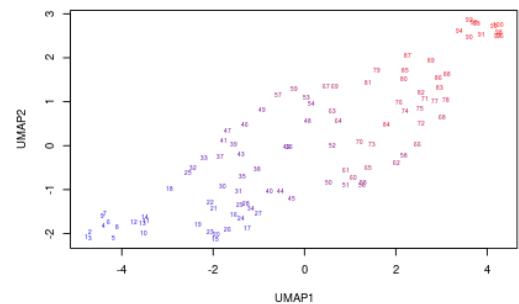
Nikolay Oskolkov @NikolayOskolkov · 10h  
Regarding your code for demonstrating artificial separation of data points, may I ask about the motivation to compute the distance matrix here. "u<-umap(as.matrix(dist(x)))"? Are you using 3-dimensional or 100-dimensional data? In the code above you input 100-dimensional data

```

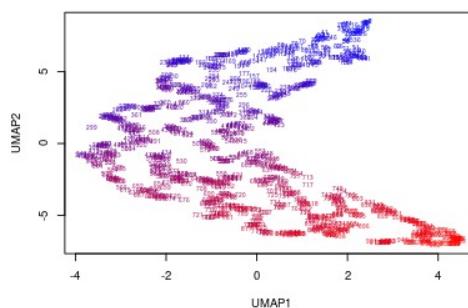
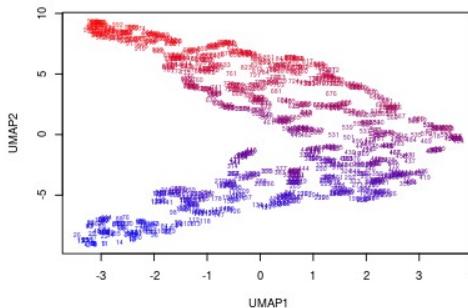
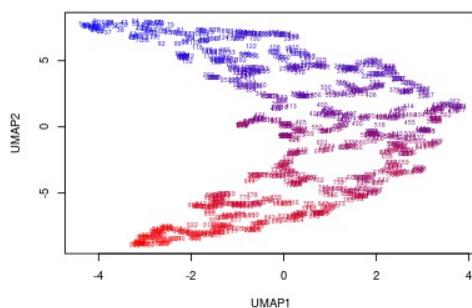
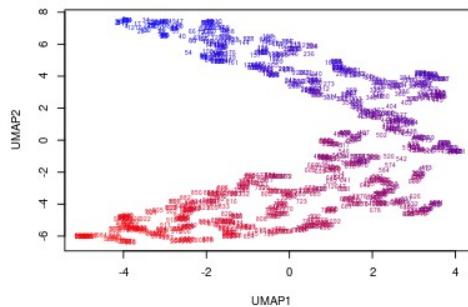
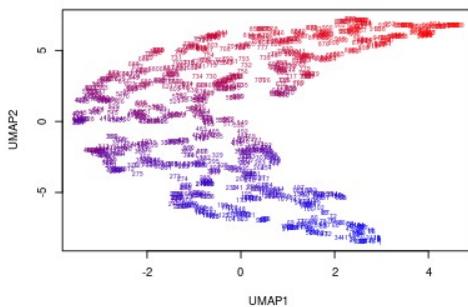
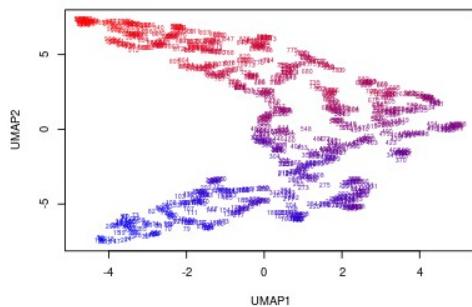
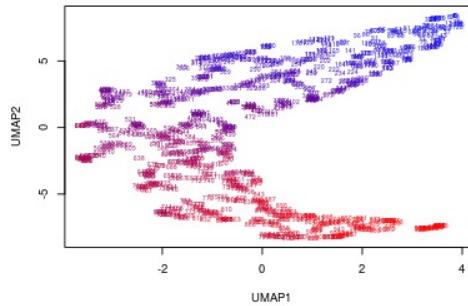
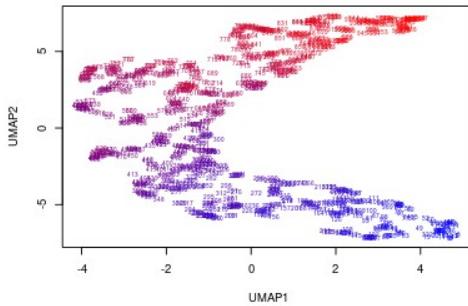
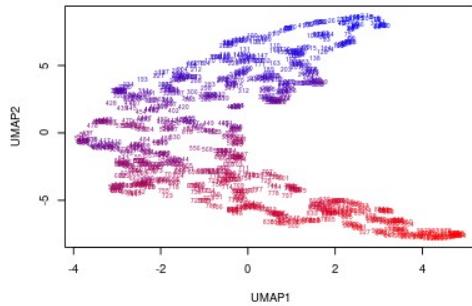
Sigma <- matrix(.8, 3, 3); diag(Sigma) <- 1
x <- MASS::mvrnorm(100, rep(0,3), Sigma)
custom.settings <- umap.defaults
custom.settings$input <- "dist"
u <- umap(as.matrix(dist(x)), config = custom.settings)
ranks <- rank(rowMeans(x))
colors <- colorRampPalette(c("blue", "red"))(nrow(x))
colormap <- colors[ranks]
plot(u$layout[,1], u$layout[,2], type = "n", xlab = "dim1", ylab = "dim2")
text(u$layout[,1], u$layout[,2], labels = ranks, col = colormap, cex = 0.5)

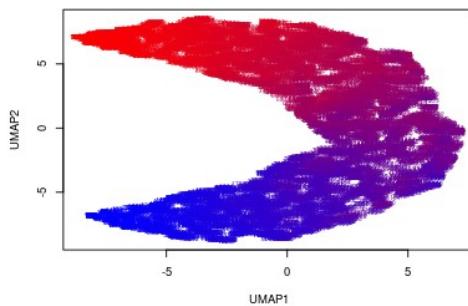
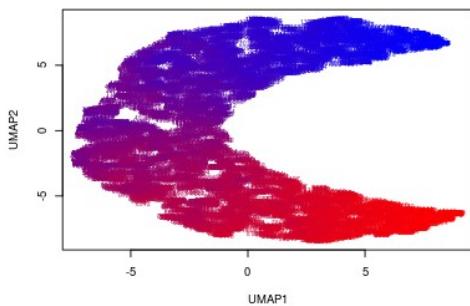
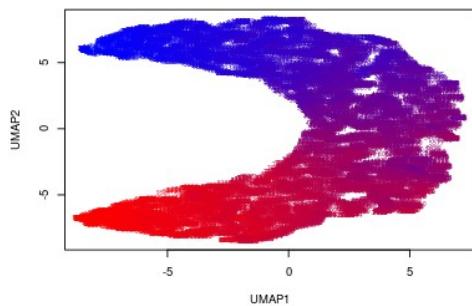
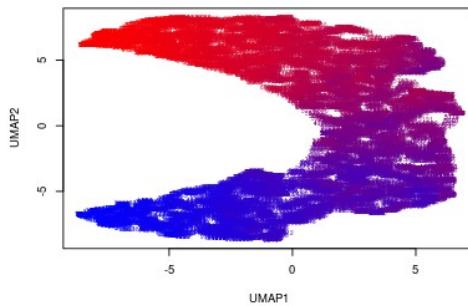
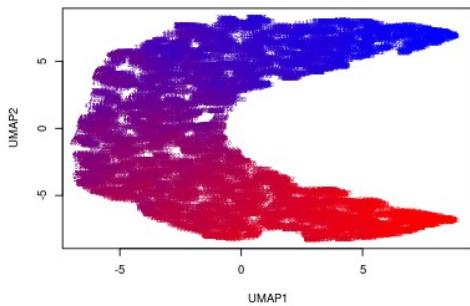
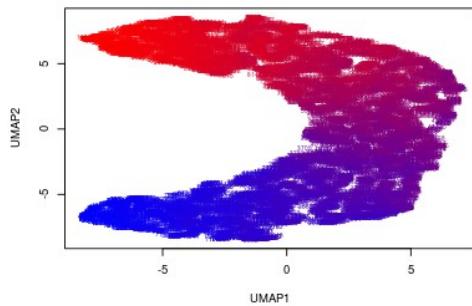
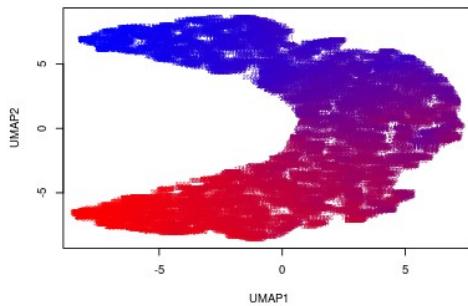
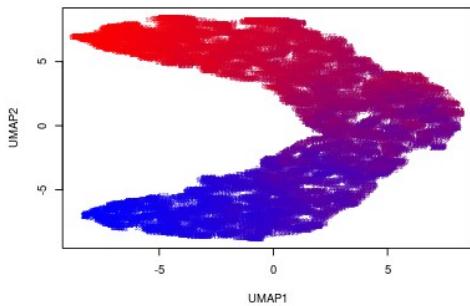
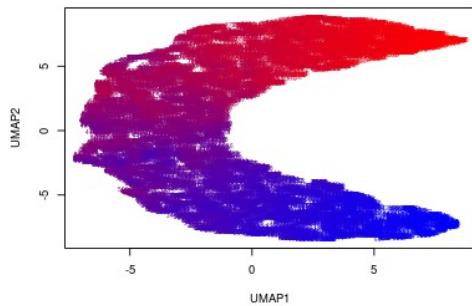
```



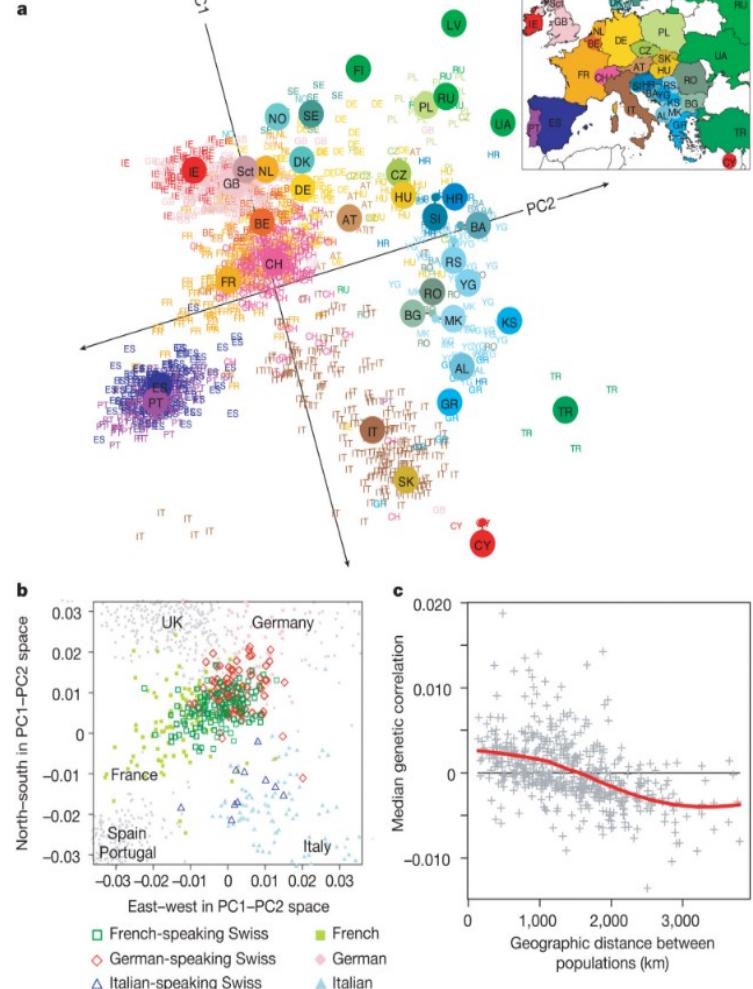


Is N=100  
OK for  
statistics?

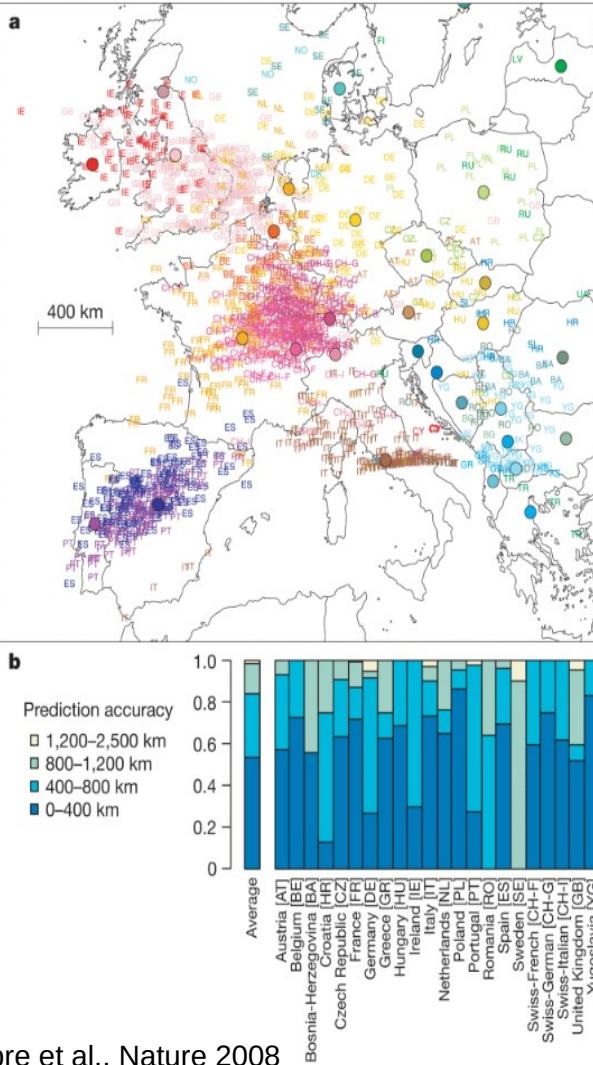




**Going through the post of Rafael Irizarry:  
Point2: PCA better than UMAP for PopGen!**



Novembre et al., Nature 2008



## scientific reports

## OPEN Principal Component Analyses (PCA)-based findings in population genetic studies are highly biased and must be reevaluated

Eran Elhaik

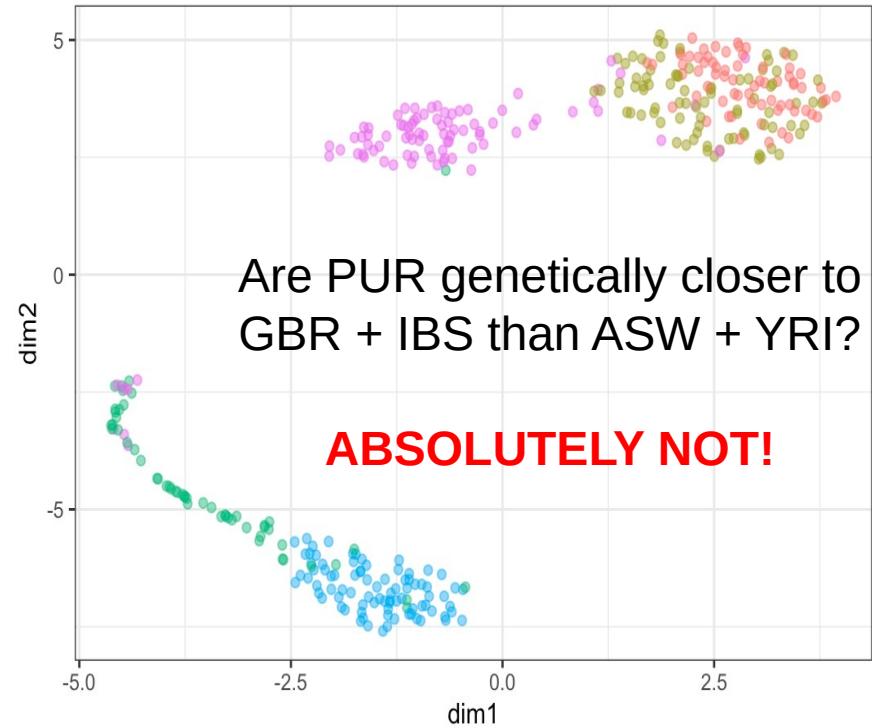
Principal Component Analysis (PCA) is a multivariate analysis that reduces the complexity of datasets while preserving data covariance. The outcome can be visualized on colorful scatterplots, ideally with only a minimal loss of information. PCA applications, implemented in well-cited packages like EIGENSOFT and PLINK, are extensively used as the foremost analyses in population genetics and related fields (e.g., animal and plant or medical genetics). PCA outcomes are used to shape study design, identify, and characterize individuals and populations, and draw historical and ethnobiological conclusions on origins, evolution, dispersion, and relatedness. The replicability crisis in science has prompted us to evaluate whether PCA results are reliable, robust, and replicable. We analyzed twelve common test cases using an intuitive color-based model alongside human population data. We demonstrate that PCA results can be artifacts of the data and can be easily manipulated to generate desired outcomes. PCA adjustment also yielded unfavorable outcomes in association studies. PCA results may not be reliable, robust, or replicable as the field assumes. Our findings raise concerns about the validity of results reported in the population genetics literature and related fields that place a disproportionate reliance upon PCA outcomes and the insights derived from them. We conclude that PCA may have a biasing role in genetic investigations and that 32,000–216,000 genetic studies should be reevaluated. An alternative mixed-admixture population genetic model is discussed.

The ongoing replicability crisis, undermining the foundation of science<sup>1</sup>, raises various concerns ranging from study design to statistical rigor<sup>2,3</sup>. Population genetics is confounded by its utilization of small sample sizes, ignorance of effect sizes, and adoption of questionable study designs. The field is relatively new and may involve financial interests<sup>4,5</sup> and ethical dilemmas<sup>6,7</sup>. Since biases in the field rapidly propagate to related disciplines like medical genetics, biogeography, association studies, forensics, and paleogenomics in humans and non-humans alike, it is imperative to ask whether and to what extent our most elementary tools satisfy risk criteria.

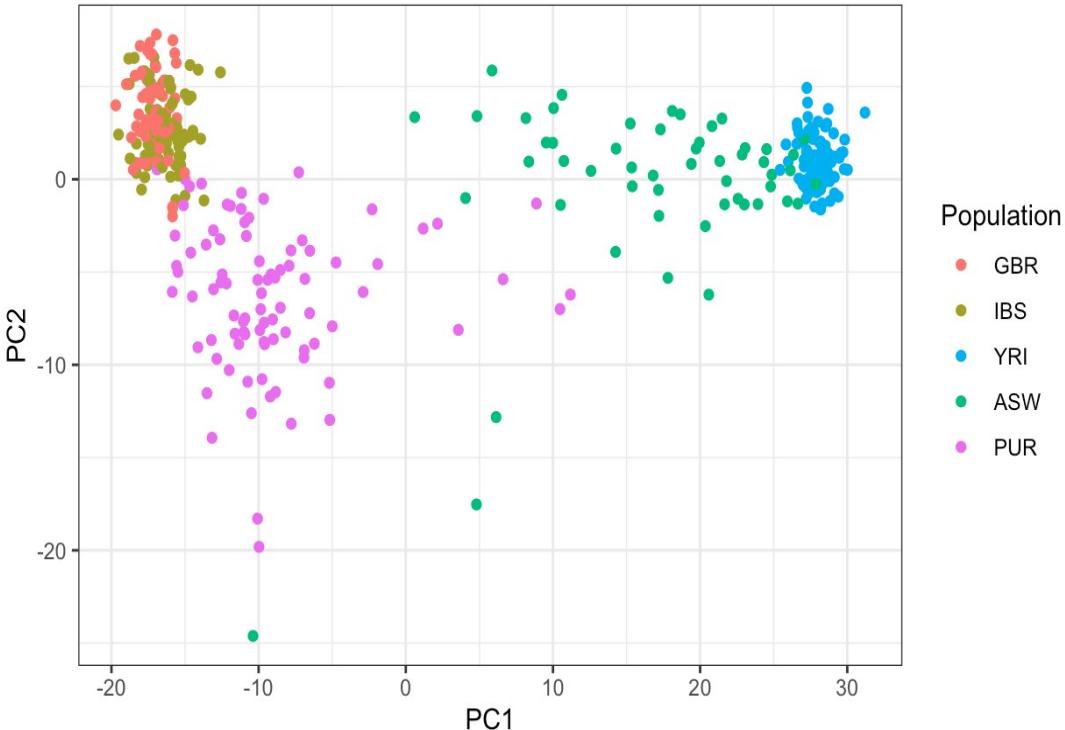
Principal Component Analysis (PCA) is a multivariate analysis that reduces the data's dimensionality while preserving their covariance. When applied to genotype bi-allelic data, typically encoded as AA, AB, and BB, PCA finds the eigenvalues and eigenvectors of the covariance matrix of allele frequencies. The data are reduced to a small number of dimensions termed principal components (PCs); each describes a decreased proportion of the genomic variation. Genotypes are then projected onto space spanned by the PC axes, which allows visualizing the samples and their distances from one another in a colorful scatter plot. In this visualization, sample overlap is considered evidence of identity, due to common ancestry<sup>8,9</sup>. PCA has an attractive property for population geneticists as it that the distances between clusters allegedly reflect the genetic and geographic distances between them. PCA also supports the projection of points onto the components calculated by a different dataset, presumably accounting for insufficient data in the projected dataset. Initially adapted for human genomic data in 1963<sup>10</sup>, the popularity of PCA has slowly increased over time. It was not until the release of the SmartPCA tool (EIGENSOFT package)<sup>11</sup> that PCA was propelled to the front stage of population genetics.

PCA is used as the first analysis of data investigation and data description in most population genetic analyses, e.g., Refs.<sup>12–15</sup>. It has a wide range of applications. It is used to examine the population structure of a cohort or individuals to determine ancestry, analyze the demographic history and admixture, decide on the genetic similarity of samples and exclude outliers, decide how to model the populations in downstream analyses, describe the ancient and modern genetic relationships between the samples, infer kinship, identify ancestral clines in the data, e.g., Refs.<sup>16–19</sup>, detect genomic signatures of natural selection, e.g., Ref.<sup>20</sup> and identify convergent evolution<sup>21</sup>.

UMAP



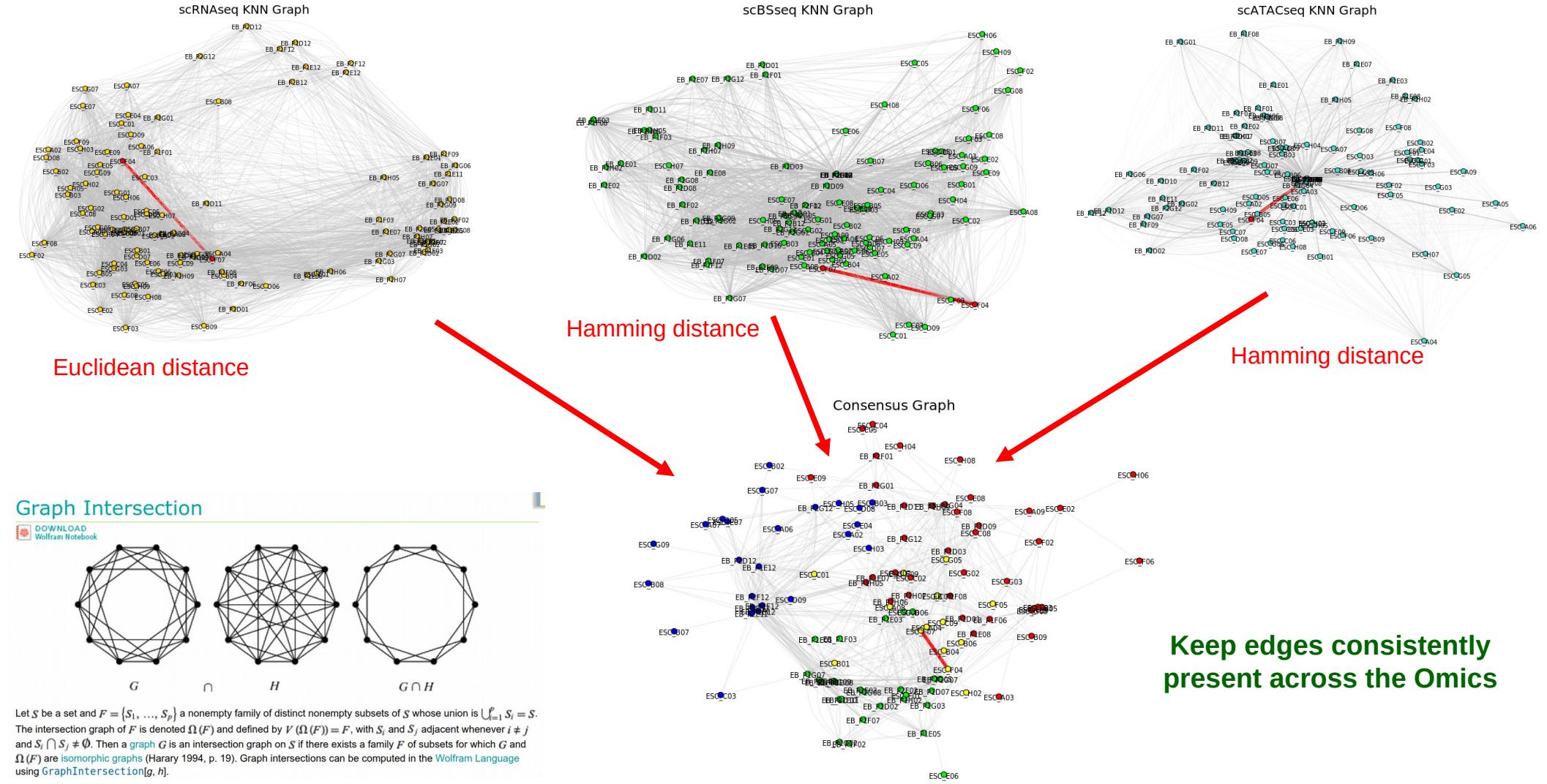
PCA

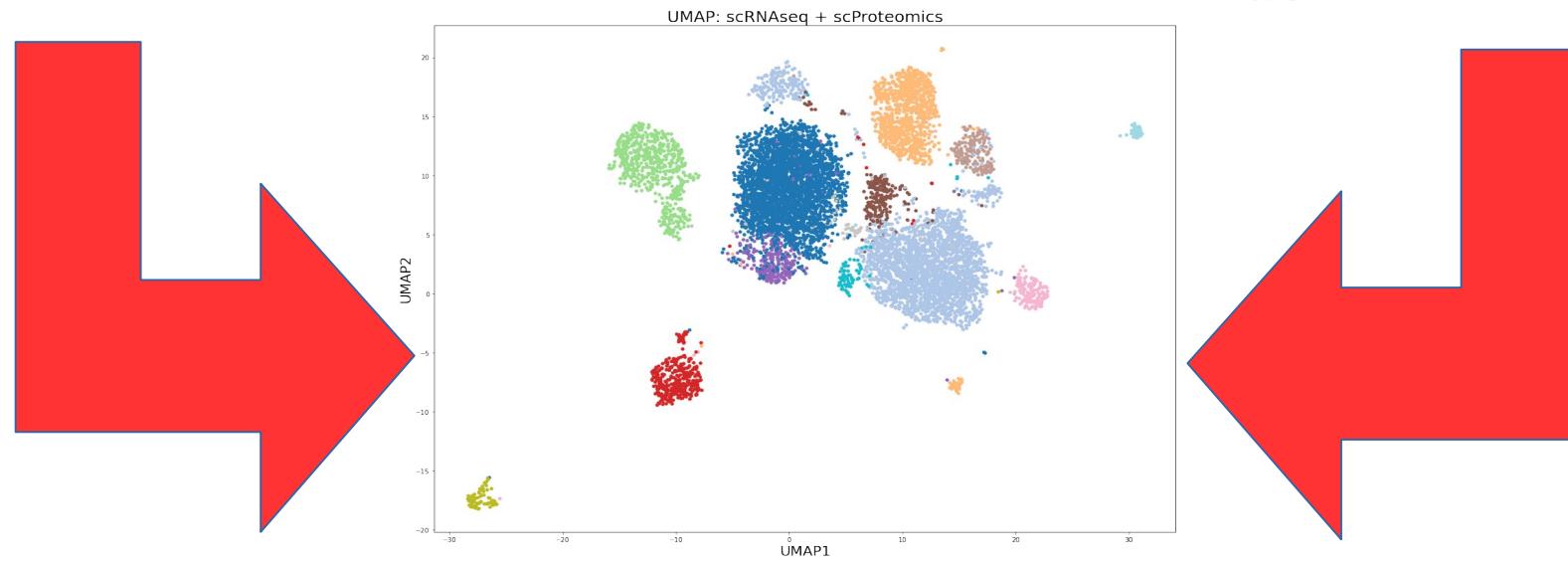
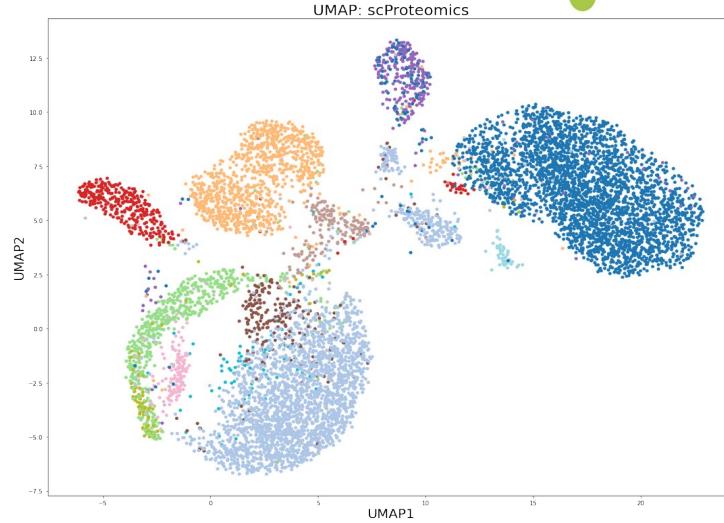
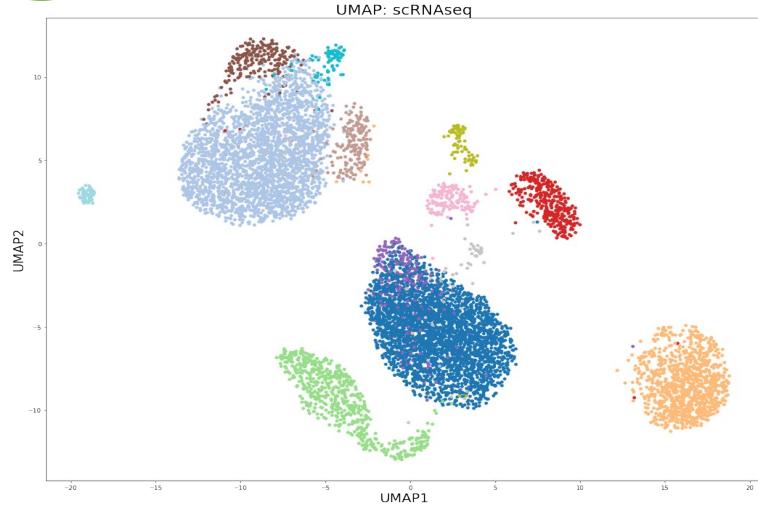


- Because of their meaningless inter-cluster distances tSNE / UMAP are less useful for population genomics than PCA
- The goal of tSNE / UMAP is to **discover clusters**, which is sufficient for Single Cell Biology but not for PopGen.
- In PopGen we generally do not discover clusters, we have an idea about e.g. human populations, and the aim is often to explore the **genetic relatedness** between the populations, a task UMAP can absolutely not solve!

# UMAP for data integration

# Graph Intersection Method







*Knut och Alice  
Wallenbergs  
Stiftelse*



**LUNDS  
UNIVERSITET**