

Tutorial on Dimensionality Reduction

Some recent approaches in statistics and machine learning

Julien Chiquet

UMR MIA Paris-Saclay, AgroParisTech, INRAE

May 25, 2023

<https://jchiquet.github.io/>



Outline

- 1 Introduction
- 2 Background: Geometric view of PCA
- 3 Reconstruction error approach
- 4 Model-based approaches
- 5 Preserving pairwise relations
- 6 Probabilistic Neighborhood Embedding



Exploratory analysis of (modern) data sets

Assume a table with n individuals described by p features/variables

$$\mathbf{X}_{n \times p} = \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & & & & \\ \hline & & x_{ij} & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array}$$

- genetics: variant j in genome i
- genomics: gene j in cell i
- ecology: species j in site i
- image: pixel j in image i
- etc.

Questions

Look for **patterns** or **structures** to summarize the data by

Challenges

- **Large** (n and p grows) and **high dimensional** (n grows but $\ll p$)
- **Redundancy** many variables may carry the same information
- **Unsupervised**: we don't (necessary) know what we are looking for
- **Discrete**: measures with counts are as common as with intensity

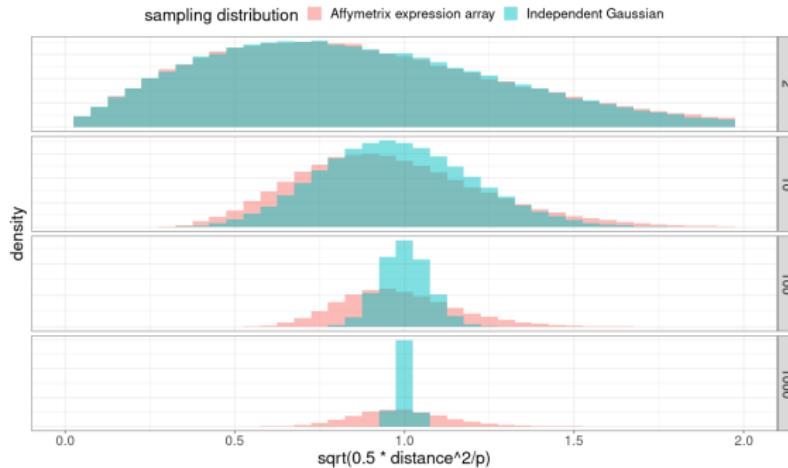
Dimensionality curse

Theorem (Folks theorem)

Let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be in the p -hypercube with i.i.d. coordinates. Then,

$$p^{-1/2} (\max \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2 - \min \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2) = 0 + \mathcal{O}\left(\sqrt{\frac{\log n}{p}}\right)$$

When p is large, all the points are almost equidistant



Hopefully, the data are not really leaving in p dimensions!

Dimension reduction: general goals

Main objective: find a **low-dimensional representation** that captures the “essence” of (high-dimensional) data

Application in Machine Learning

Preprocessing, Regularization

- Compression, denoising, anomaly detection
- Reduce overfitting in supervised learning

Application in Statistics/Data analysis}

Better understanding of the data

- descriptive/exploratory methods
- visualization (difficult to plot and interpret $> 3d!$)

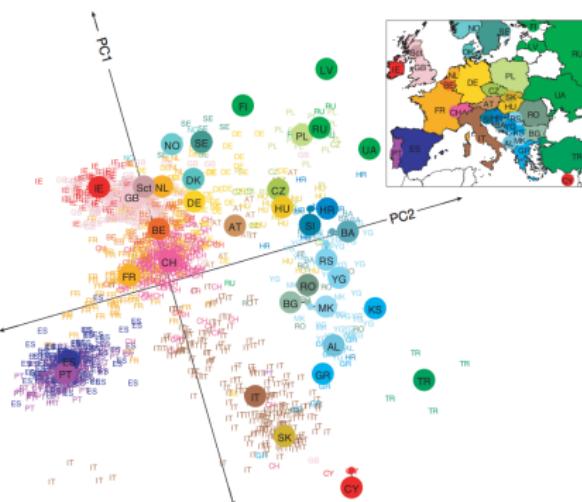
Example in genetics

Genetics variant in European population

500, 000 variants (Single Nucleotide Polymorphism) for 3000 individuals

- SNP: 90 % of human genetic variations
- coded as 0, 1 or 2 (# alleles different against pop. reference)

Summarized with 2 features¹

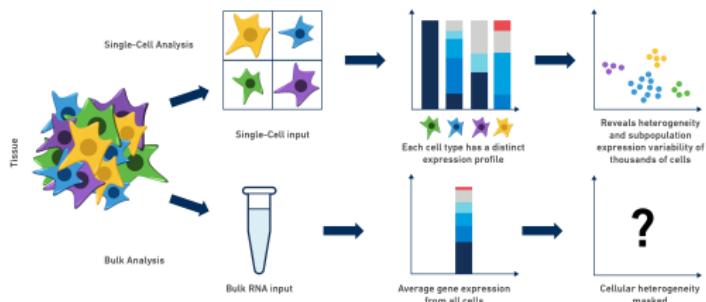


- an extremely strong structure between individuals (**"clustering"**)
- a very simple subspace where it is obvious (**"dimension reduction"**)

¹source: Nature “Gene Mirror Geography Within Europe”, 2008

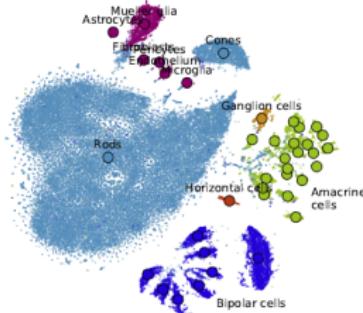
Example in genomics

Genome-wide cell biology with single-cell RNAseq data

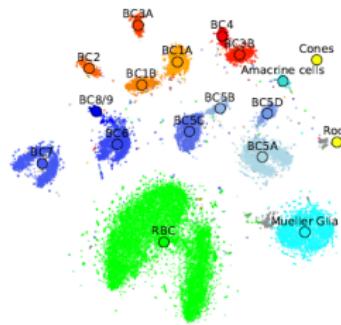


Describe cell population ($n \rightarrow 10^6$)
with high dimensional molecular
features ($p \rightarrow 10^5$)

a Macosko et al. 2015



b Shekhar et al. 2016



c Harris et al. 2018



Figure 1: Successful t-SNE visualizations of sc-RNASeq data

Example in Image: MNIST

Famous database of 60,000 labeled handwritten digits (28 x 28 images)

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

Figure 2: Data Samples

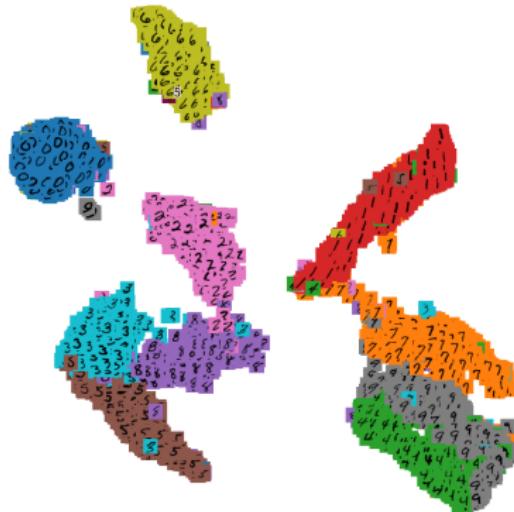


Table 1: Vectorized data

label	X2	X3	X4	X5
5	0	0	0	0
0	0	0	0	0
4	0	0	0	0

UMAP 2-dimensional visualization

Dimension reduction: problem setup

Dimension Reduction Map

- Original data : $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathbb{R}^p$
- Low dimensional data : $\{\mathbf{z}_1, \dots, \mathbf{z}_n\} \in \mathbb{R}^q, q \ll p$
- Space \mathbb{R}^p of possibly high dimension: $n \ll p$

Construct a map Φ from \mathbb{R}^p into a \mathbb{R}^q with $q \ll p$:

$$\Phi : \begin{cases} \mathbb{R}^p \rightarrow \mathbb{R}^q, q \ll p \\ \mathbf{x} \mapsto \Phi(\mathbf{x}) \triangleq \mathbf{z} \end{cases}$$

⤱ How should we design/construct Φ ?

Criterion

- Geometrical approach
- Reconstruction error
- Relationship preservation

Form of the map Φ

- Linear or non-linear?
- interpretability and versatility?
- high or low computational resource?



Outline

- 1 Introduction
- 2 Background: Geometric view of PCA
- 3 Reconstruction error approach
- 4 Model-based approaches
- 5 Preserving pairwise relations
- 6 Probabilistic Neighborhood Embedding

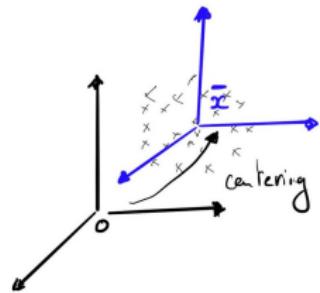


Cloud of observation in \mathbb{R}^p and Inertia

Individuals in the variable space \mathbb{R}^p

Cloud \mathbf{X} is centered around^a $\bar{\mathbf{x}} = \sum_{i=1}^n \mathbf{x}_i / n$

$$\mathbf{X}^c = \begin{pmatrix} x_{11} - \bar{x}_1 & \dots & x_{1j} - \bar{x}_j & \dots & x_{1p} - \bar{x}_p \\ \vdots & & \vdots & & \vdots \\ x_{i1} - \bar{x}_1 & \dots & x_{ij} - \bar{x}_j & \dots & x_{ip} - \bar{x}_p \\ \vdots & & \vdots & & \vdots \\ x_{n1} - \bar{x}_1 & \dots & x_{nj} - \bar{x}_j & \dots & x_{np} - \bar{x}_p \end{pmatrix}$$



^aempirical mean, barycentrum, center of inertia

Figure 3: Example in \mathbb{R}^3

Total Inertia I_T as a measure of information

Distances to the center of the cloud \propto the total empirical variance

$$I_T = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p (x_{ij} - \bar{x}_j)^2 = \frac{1}{n} \sum_{i=1}^n \text{dist}^2(\mathbf{x}_i, \bar{\mathbf{x}}) = \sum_{j=1}^p \mathbb{V}(\mathbf{x}^j) = \text{trace}(\hat{\Sigma})$$

⇒ Good representation has large inertia (much variability)

Geometric view in a nutshell

Consider collection of orthogonal axes (with dimension =1), then

$$I_T = I_{\Delta_1} + I_{\Delta_2} + \cdots + I_{\Delta_p}$$

PCA is matrix factorisation

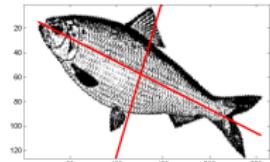
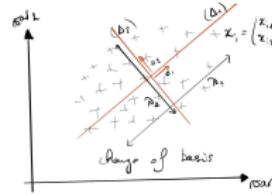
$$\hat{\Sigma} = \mathbf{V}\Lambda\mathbf{V}^T, \quad \mathbf{V} = (\mathbf{v}_1 \quad \mathbf{v}_2, \quad \dots \quad \mathbf{v}_p), \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_p)$$

\mathbf{V} are known as the **loadings**

Interpretation in \mathbb{R}^p

\mathbf{V} describes a new orthogonal basis and a rotation of data in this basis

$$\left\{ \begin{array}{l} \Delta_1 \oplus \dots \oplus \Delta_p \\ \mathbf{v}_1 \perp \dots \perp \mathbf{v}_p \\ \lambda_1 > \dots > \lambda_p \\ I_{\Delta_1^\perp} > \dots > I_{\Delta_p^\perp} \end{array} \right.$$



~> PCA is an appropriate rotation on axes that maximizes the variance

Unifying view of variables and individuals

In the new basis $\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$, coordinates of i (a.k.a. **scores**) are

$$\mathbf{c}_i^\top = (\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{V} = \mathbf{X}_i^c \mathbf{V}, \quad \mathbf{c}_i \in \mathbb{R}^p.$$

In the variable space \mathbb{R}^n , new variables (factors) are formed by linear combinations of the original variables: the **principal components** (PC)

$$\mathbf{f}_k = \sum_{j=1}^p v_{kj} (\mathbf{x}^j - \bar{\mathbf{x}}_j) = \mathbf{X}^c \mathbf{v}_k, \quad \mathbf{f}_k \in \mathbb{R}^n$$

The matrix of PC connects individual coordinates to latent factors:

$$\text{PC} = \mathbf{X}^c \mathbf{V} = (\mathbf{f}_1 \quad \mathbf{f}_2 \quad \dots \quad \mathbf{f}_p) = \begin{pmatrix} \mathbf{c}_1^\top \\ \mathbf{c}_2^\top \\ \vdots \\ \mathbf{c}_n^\top \end{pmatrix}$$

↔ Everything can be interpreted on a single plot, called the biplot



Reconstruction formula

Recall that $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_p)$ is the matrix of Principal components. Then,

- $\mathbf{f}_k = \mathbf{X}^c \mathbf{v}_k$ for projection on axis k
- $\mathbf{F} = \mathbf{X}^c \mathbf{V}$ for all axis.

Using orthogonality of \mathbf{V} , we get back the original data as follows, without loss (\mathbf{V}^T performs the inverse rotation of \mathbf{V}):

$$\mathbf{X}^c = \mathbf{F} \mathbf{V}^T$$

We obtain an approximation $\hat{\mathbf{X}}^c$ (compression) of the data \mathbf{X}^c by considering a subset \mathcal{S} of PC, typically $\mathcal{S} = 1, \dots, q$ with $q \ll p$.

$$\hat{\mathbf{X}}^c = \mathbf{F}_q \mathbf{V}_q^T = \mathbf{X}^c \mathbf{V}_q \mathbf{V}_q^T$$

↔ This is a rank- q approximation of \mathbf{X} (captured by the first q axes).



Single-Cell data analysed with PCA

Toy single-cell RNA data set (https://github.com/LuyiTian/sc_mixology/)

The dataset scRNA contains the counts of the 500 most varying transcripts (tens of thousands) in the mixtures of 5 cell lines for a total of 3918 cells in human liver (obtained with standard 10x scRNaseq Chromium protocol).

	KRT81	AKR1B10	LCN2	AKR1C2	ALDH1A1	AGR2	AKR1C3	GPX2
Lib90_00000	6	2	43	4	2	4	3	0
Lib90_00001	38	16	175	30	8	19	5	25
Lib90_00002	5	6	3	3	1	0	3	4

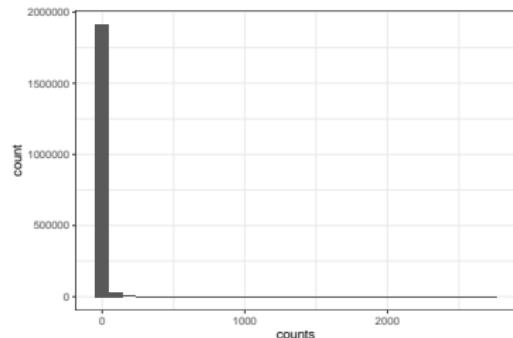


Figure 4: raw counts

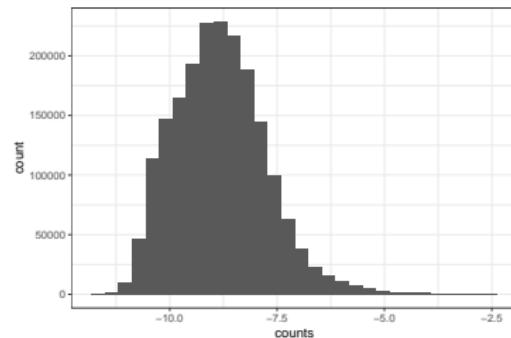
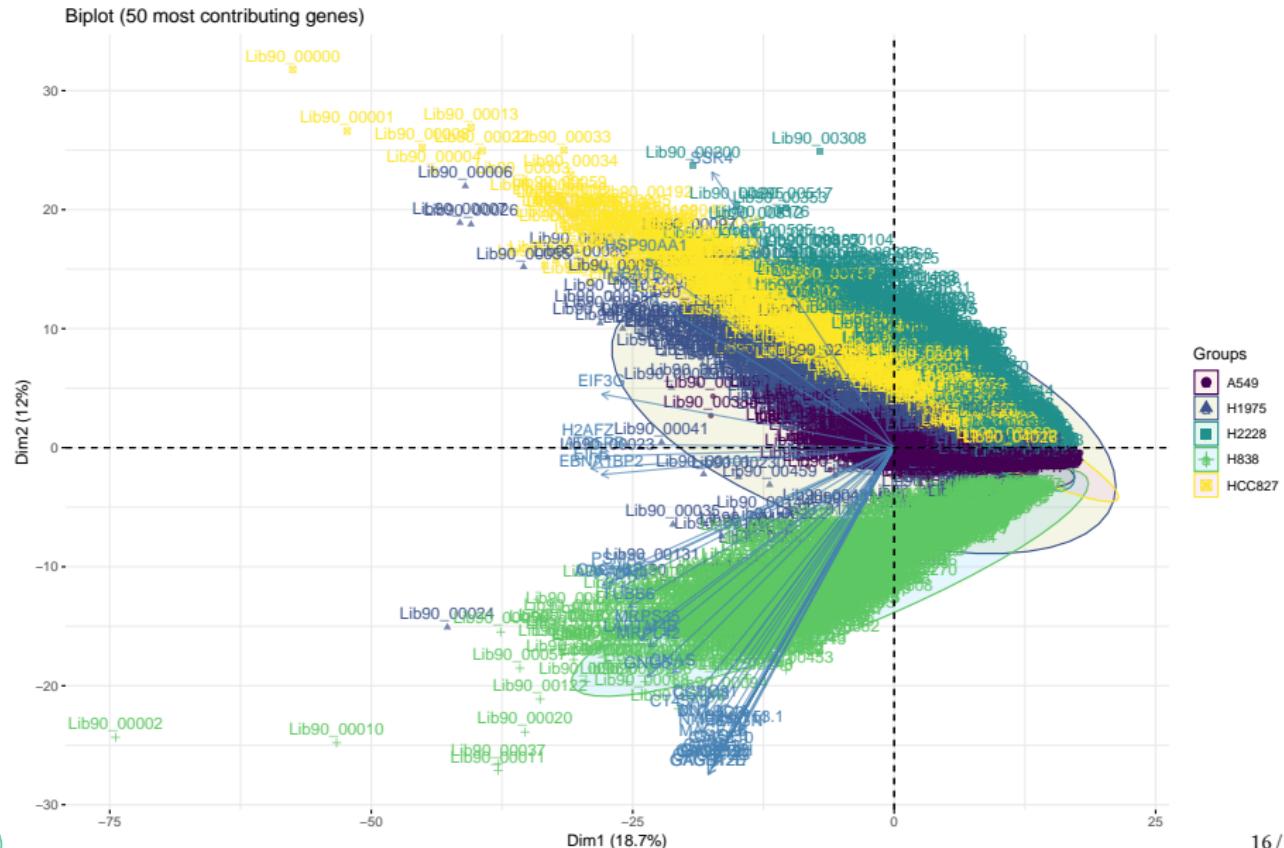


Figure 5: log/total-counts normalization

Single-Cell data analysed with PCA

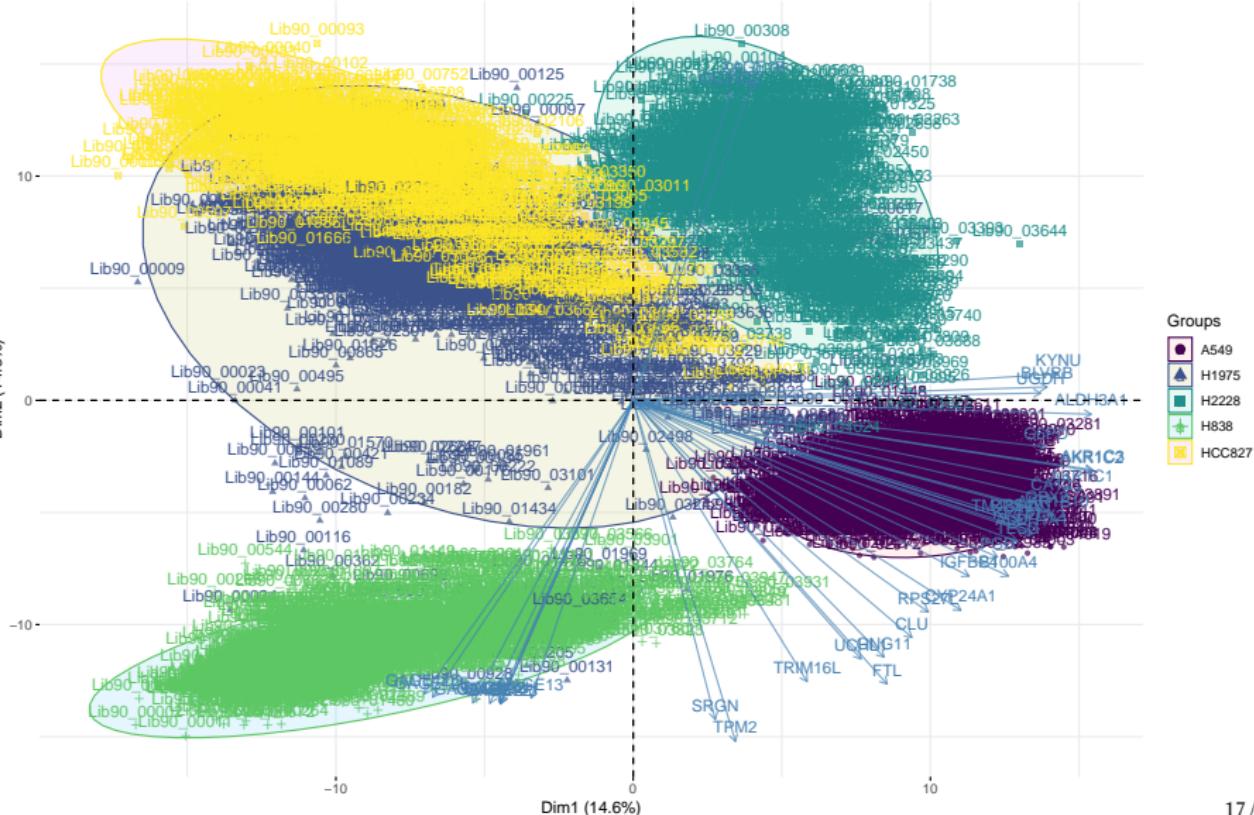
Raw data



Single-Cell data analysis with PCA

Normalized data

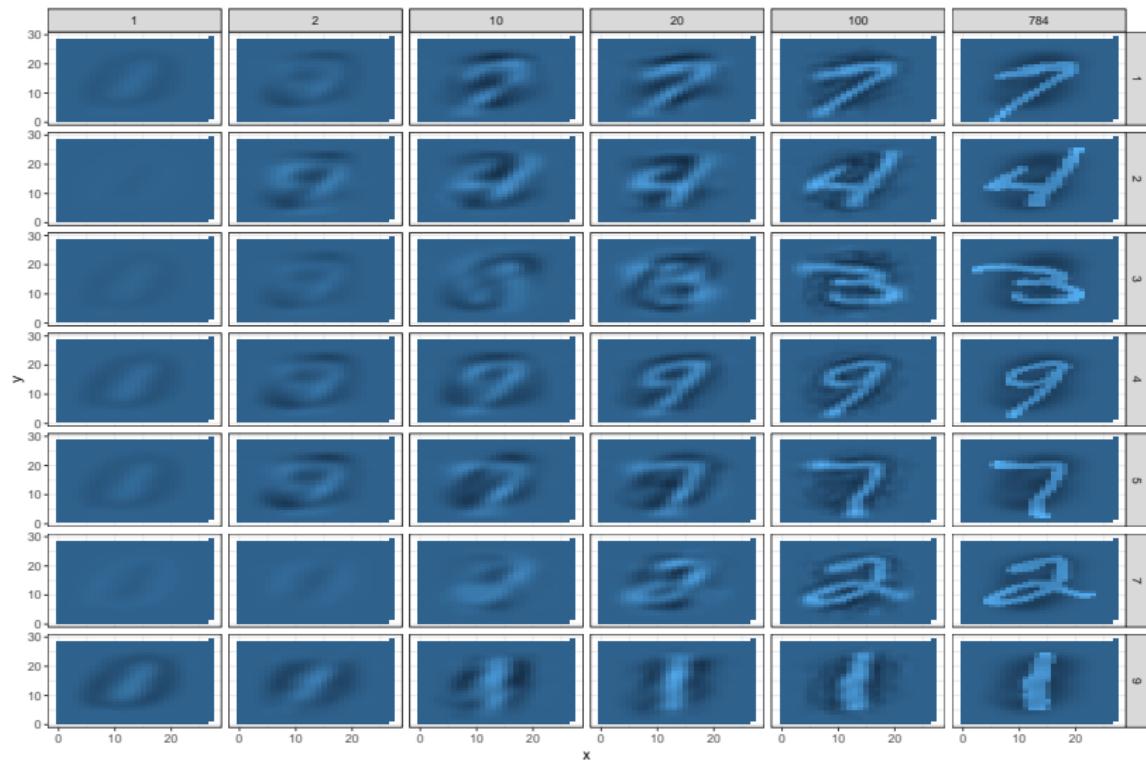
Biplot (50 most contributing genes)



MNIST data analysed with PCA

Compression/projection

Project 5 samples on the first $\{1, 2, 10, 20, 100, 784\}$ axes



Beyond PCA and linear methods

Limitations

Robust but,

- badly shaped for complex geometries (like multiscale properties)
- Fails with **Count** or **Skew** data (hidden Gaussian assumption)

Ideas

- Modify the model by playing with the **reconstruction error**
- Gain in versatility with **probabilistic/model-based approaches**
- Focus on **relationship preservation** to keep local characteristics
- Go **non-linear** by transforming the input space or amending the map
 $\Phi : \mathbb{R}^p \rightarrow \mathbb{R}^q$

Challenges

- tradeoff between interpretability and **versatility**
- tradeoff between **high** or low computational resource



Outline

- 1 Introduction
- 2 Background: Geometric view of PCA
- 3 Reconstruction error approach
- 4 Model-based approaches
- 5 Preserving pairwise relations
- 6 Probabilistic Neighborhood Embedding



Reinterpretation of PCA

PCA model

Let \mathbf{V}_q be a $p \times q$ matrix whose columns are of q orthonormal vectors.

$$\Phi(\mathbf{x}) = \mathbf{V}_q^\top (\mathbf{x} - \boldsymbol{\mu}) = \mathbf{z}, \quad \hat{\mathbf{x}} = \tilde{\Phi}(\mathbf{z}) = \boldsymbol{\mu} + \mathbf{V}_q \mathbf{z}.$$

Model with **Linear assumption + ortho-normality constraints**

PCA reconstruction error

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^p, \mathbf{V}_q \in \mathcal{O}_{p,q}}{\text{minimize}} \sum_{i=1}^n \|(\mathbf{x}_i - \boldsymbol{\mu}) - \mathbf{V}\mathbf{V}^\top(\mathbf{x}_i - \boldsymbol{\mu})\|^2 = \left(\underset{\mathbf{F}_q \in \mathcal{M}_{n,q}, \mathbf{V}_q \in \mathcal{O}_{p,q}}{\text{minimize}} \|\mathbf{X}^c - \mathbf{F}_q \mathbf{V}_q^\top\|_F^2 \right)$$

Solution (explicit)

- $\boldsymbol{\mu}$ is the empirical mean, \mathbf{V}_q eigenvectors of the empirical covariance
- In practice: SVD of the centered matrix $\mathbf{X}^c = \mathbf{U}_q \mathbf{D}_q \mathbf{V}_q^\top = \mathbf{F}_q \mathbf{V}_q^\top$



Reconstruction error approach

Find maps Φ and $\tilde{\Phi}$ in a given family (e.g, linear, constraint on parameters, etc.), minimizing an error between \mathbf{x} and $\hat{\mathbf{x}} = \tilde{\Phi}(\Phi(\mathbf{x}))$

- **Distance** between \mathbf{X} and $\hat{\mathbf{X}}$, e.g, sum of squares:

$$\epsilon^{\text{SSQ}}(\mathbf{X}, \hat{\mathbf{X}}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 = \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i))\|^2$$

- **Divergence** between distributions $\hat{p}_{\mathbf{X}}$ and $\hat{p}_{\hat{\mathbf{X}}}$ of \mathbf{X}_i and $\hat{\mathbf{X}}_i$

$$D_{\text{KL}} (\hat{p}_{\mathbf{X}}, \hat{p}_{\hat{\mathbf{X}}}) = - \sum_i \hat{p}_{\mathbf{X}_i} \log \left(\frac{\hat{p}_{\mathbf{X}_i}}{\hat{p}_{\hat{\mathbf{X}}_i}} \right)$$

- **Log-likelihood** of a parametric model p_{θ} , with $\hat{\mathbf{X}} = f(\theta)$:

$$-\log p_{\theta}(\mathbf{X}) = - \sum_{i=1}^n \log p_{\theta}(\mathbf{X}_i)$$



Example: Non-negative Matrix Factorization

Assume that \mathbf{X} contains only non-negative entries (i.e. ≥ 0).

Model: Linearity of Φ plus non-negativity constraints:

$$\hat{\mathbf{X}} \approx \underbrace{\mathbf{X}\mathbf{V}_q}_{\mathbf{F}_q} \mathbf{V}_q^\top, \text{ s.c. } \mathbf{F}_q, \mathbf{V}_q \text{ has non-negative entries.}$$

- Least-squares loss:

$$\hat{\mathbf{X}}^{\text{ls}} = \underset{\begin{array}{l} \mathbf{F} \in \mathcal{M}(\mathbb{R}_+)^{n,q} \\ \mathbf{V} \in \mathcal{M}(\mathbb{R}_+)^{p,q} \end{array}}{\arg \min} \|\mathbf{X} - \mathbf{F}\mathbf{V}^\top\|_F^2,$$

- Poisson likelihood for \mathbf{X}_{ij} with intensity $\lambda_{ij}^q = (\mathbf{F}_q \mathbf{V}_q^\top)_{ij} \geq 0$:

$$\hat{\mathbf{X}}^{\text{poisson}} = \underset{\begin{array}{l} \mathbf{F} \in \mathcal{M}(\mathbb{R}_+)^{n,q} \\ \mathbf{V} \in \mathcal{M}(\mathbb{R}_+)^{p,q} \end{array}}{\arg \max} \sum_{i,j} x_{ij} \log(\lambda_{ij}^q) - \lambda_{ij}^q.$$



Kernel-PCA

Principle: non linear transformation of \mathbf{x} prior to linear PCA

- ① Project the data into a higher space where it is linearly separable
- ② Apply PCA to the transformed data

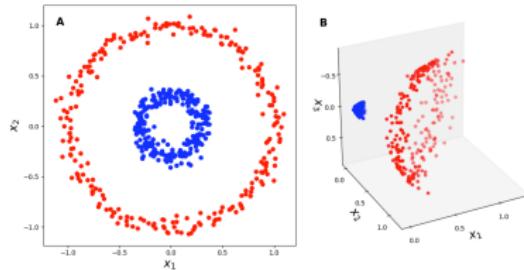


Figure 6: Transformation $\Psi : \mathbf{x} \rightarrow \Psi(\mathbf{x})$ (illustration in presence of existing labels)

Model

Assume a non linear transformation $\Psi(\mathbf{x}_i)$ where $\Psi : \mathbb{R}^p \rightarrow \mathbb{R}^n$, then perform PCA, with \mathbf{V}_q a $n \times q$ orthonormal matrix

$$\Phi(\mathbf{x}) = \mathbf{V}_q^\top \Psi(\mathbf{x} - \mu) = \mathbf{z}$$

Choice of the transformation

All relationships are described in terms of scalar products between $(\mathbf{x}_i, \mathbf{x}_{i'})$:

$$K = k(\mathbf{x}_1, \mathbf{x}_2) = (\Psi(\mathbf{x})_i, \Psi(\mathbf{x})_{i'}) = \Psi(\mathbf{x}_i)^\top \Psi(\mathbf{x}_{i'}),$$

where the kernel K is a symmetric positive definite function.

Some common kernels

Polynomial: $k(\mathbf{x}_i, \mathbf{x}_{i'}) = (\mathbf{x}_i^\top \mathbf{x}_{i'} + c)^d$

Gaussian: $k(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp \frac{-\|\mathbf{x}_i - \mathbf{x}_{i'}\|^2}{2\sigma^2}$

Laplacian kernel: $k(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp \frac{-\|\mathbf{x}_i - \mathbf{x}_{i'}\|}{\sigma}$

Kernel PCA suffers from the choice of the Kernel



Outline

- 1 Introduction
- 2 Background: Geometric view of PCA
- 3 Reconstruction error approach
- 4 Model-based approaches
- 5 Preserving pairwise relations
- 6 Probabilistic Neighborhood Embedding



Probabilistic Gaussian PCA

Generative model [1]

pPCA is a special factor model with parameter $\theta = (\mathbf{C}, \sigma)$:

$$\begin{array}{lll} \text{latent space} & \mathbf{Z}_i & \text{i.i.d.} \quad \mathbf{W}_i \sim \mathcal{N}(\mathbf{0}_q, \mathbf{I}_q) \\ \text{observation space} & \mathbf{X}_i | \mathbf{Z}_i & \text{indep.} \quad \mathbf{X} | \mathbf{Z}_i \sim \mathcal{N}(\boldsymbol{\mu} + \mathbf{C}\mathbf{Z}, \sigma^2 \mathbf{I}_n) \end{array}$$

By direct integration², the marginal distribution of the observation is

$$p_{\theta}(\mathbf{X}_i) = \int_{\mathbb{R}^q} p_{\theta}(\mathbf{X}_i | \mathbf{Z}_i) p(\mathbf{Z}_i) d\mathbf{Z}_i = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^T + \sigma^2 \mathbf{I}_n$$

↗ rank- q decomposition of the covariance matrix + noise.

²easy since everything is Gaussian



Estimation

Criterion: negative log-likelihood

$$-\sum_{i=1}^n \log p_\theta(\mathbf{X}_i) = \log |\Sigma| + \text{tr}(\Sigma^{-1} \hat{\Sigma}), \quad \hat{\Sigma} = \frac{1}{n} \sum_i (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$$

Maximum likelihood estimator

$$\hat{\mathbf{C}}^{\text{mle}} = \mathbf{V}_q (\Lambda_q - \hat{\sigma}^2 \mathbf{I}_n)^{1/2}, \quad \hat{\sigma}^2 = \frac{1}{p-q} \sum_{i=q+1}^p \lambda_i, \quad \hat{\Sigma} = \mathbf{V} \Lambda \mathbf{V}^\top$$

Latent position: posterior distribution

$$\mathbf{Z}_i | \mathbf{X}_i \sim \mathcal{N}(\mathbf{S}^{-1} \hat{\mathbf{C}}^\top (\mathbf{X}_i - \bar{\mathbf{x}}), \mathbf{S}^{-1} \hat{\sigma}^2), \quad \mathbf{S} = (\hat{\mathbf{C}} \hat{\mathbf{C}}^T + \hat{\sigma}^2)$$

When $\sigma^2 \rightarrow 0$, $\mathbb{E}(\mathbf{Z}_i | \mathbf{X}_i) \equiv \text{orthogonal projection in the latent space.}$



Estimation: alternative

Expectation-Maximization

With $\mathcal{H}(p) = -\mathbb{E}_p(\log(p))$ the entropy of p , decompose

$$\log p_\theta(\mathbf{X}) = \mathbb{E}[\log p_\theta(\mathbf{X}, \mathbf{Z}) | \mathbf{X}; \theta] + \mathcal{H}[p_\theta(\mathbf{Z} | \mathbf{X}; \theta)]$$

EM requires to evaluate (some moments of) $p_\theta(\mathbf{Z} | \mathbf{X}; \theta)$

- E-step: evaluate $Q(\theta | \theta') = \mathbb{E}(\log \ell(\mathbf{X}, \mathbf{W}; \theta) | \mathbf{X}; \theta')$
- M-step: update θ by maximizing $Q(\theta | \theta')$

EM for pPCA

- E-step: update the latente position means $\mathbb{E}(\mathbf{Z} | \mathbf{X})$
- M-step: update the model parameters \mathbf{C}, σ^2

→ can be faster than MLE when $p \gg q$



PCA for counts: poisson lognormal PCA

Generative Model [2]

latent space	\mathbf{Z}_i	i.i.d.	$\mathbf{Z}_i \sim \mathcal{N}(\mathbf{0}_q, \mathbf{I}_q)$
observation space	$\mathbf{X}_i \mathbf{Z}_i$	indep.	$\mathbf{X} \mathbf{Z}_i \sim \mathcal{P}(\exp\{\boldsymbol{\mu} + \mathbf{C}^\top \mathbf{Z}_i\})$

Estimation: Issues

- The marginal distribution is hard to compute, even numerically

$$p_\theta(\mathbf{X}_i) = \int_{\mathbb{R}^p} \prod_{j=1}^p p_\theta(X_{ij} | Z_{ij}) p_\theta(Z_i) dZ_i$$

⤟ no direct MLE possible

- Posterior distribution of Z_i has no close form

⤟ no genuine application of EM possible

Variational inference

Variational approximation [4]

- Use a proxy q_ψ of $p_\theta(\mathbf{Z} | \mathbf{X})$ minimizing a divergence in a class \mathcal{Q}

$$q_\psi(\mathbf{Z})^* = \arg \min_{q \in \mathcal{Q}} KL(q(\mathbf{Z}), p(\mathbf{Z} | \mathbf{Y})), \quad KL(., .) = \mathbb{E}_{q_\psi} \left[\log \frac{q(z)}{p(z)} \right].$$

- maximize the ELBO (Evidence Lower BOund)

$$J(\theta, \psi) = \log p_\theta(\mathbf{Y}) - KL[q_\psi(\mathbf{Z}) || p_\theta(\mathbf{Z} | \mathbf{Y})] = \mathbb{E}_\psi [\log p_\theta(\mathbf{Y}, \mathbf{Z})] + \mathcal{H}[q_\psi(\mathbf{Z})]$$

Variational EM for Poisson-lognormal [5]

Consider \mathcal{Q} the class of diagonal multivariate Gaussian distributions.
The ELBO $J(\theta, \psi)$ has close-form and is bi-concave.

- E-step: solve in ψ for given θ
- M-step: solve in θ for given ψ



Model selection and Visualization

Number of components/rank k needs to be chosen.

$\log p_{\hat{\beta}}(\mathbf{X})$ intractable: use variational “likelihood” $J(\hat{\beta}, \hat{\psi})$

- BIC \rightsquigarrow vBIC $_k = J(\hat{\beta}, \tilde{p}) - \frac{1}{2}p(d+k)\log(n)$
- ICL \rightsquigarrow vICL $_k = \text{vBIC}_k - \mathcal{H}(\tilde{p})$

$$\hat{k} = \arg \max_k \text{vBIC}_k \quad \text{or} \quad \hat{k} = \arg \max_k \text{vICL}_k$$

Visualization

- Gaussian PCA: Optimal subspaces nested when q increases.
- PLN-pPCA: Non-nested subspaces.

For the selected dimension dimension \hat{k} :

- Compute the estimated latent positions $\mathbb{E}_q(\mathbf{Z}_i)$
- Perform PCA on these estimated positions
- Display result in any dimension $k \leq \hat{k}$



PCA: Goodness of fit

pPCA: Cumulated sum of the eigenvalues = % of variance preserved on the first q components.

PLN-pPCA

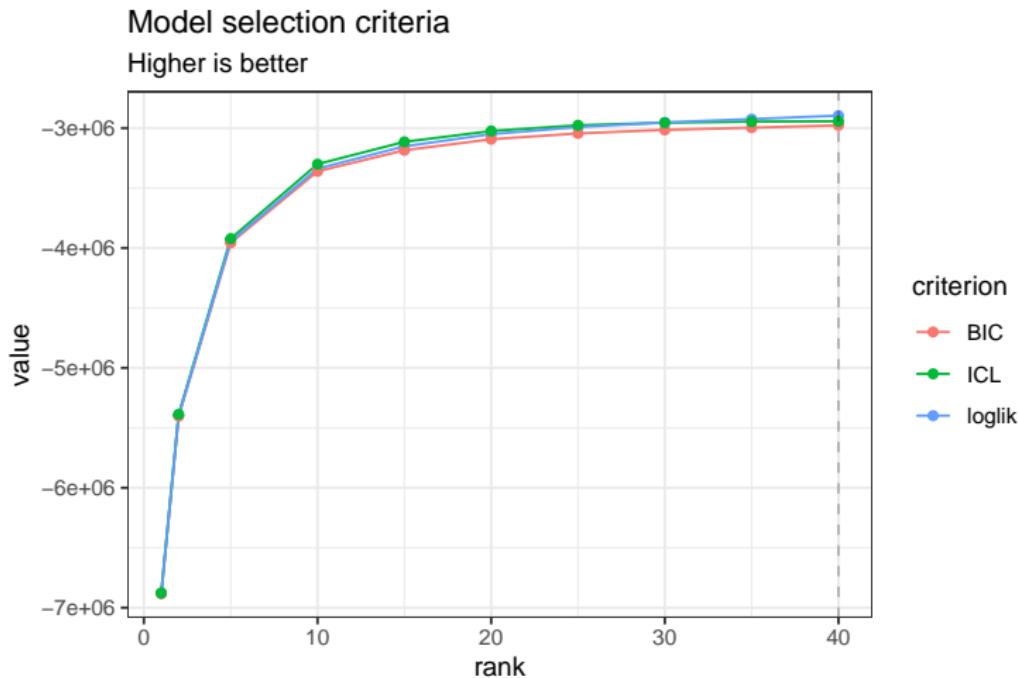
Deviance based criterion.

- Take $\lambda_{ij}^{(k)} = \exp(\mathbb{E}_q(Z_{ij}^{(k)}))$
- Define $\lambda_{ij}^{\min} = \exp(\tilde{Z}_{ij}^0)$ and $\lambda_{ij}^{\max} = Y_{ij}$
- Compute the Poisson log-likelihood $\ell_k = \log \mathbb{P}(\mathbf{X}; \lambda^{(k)})$

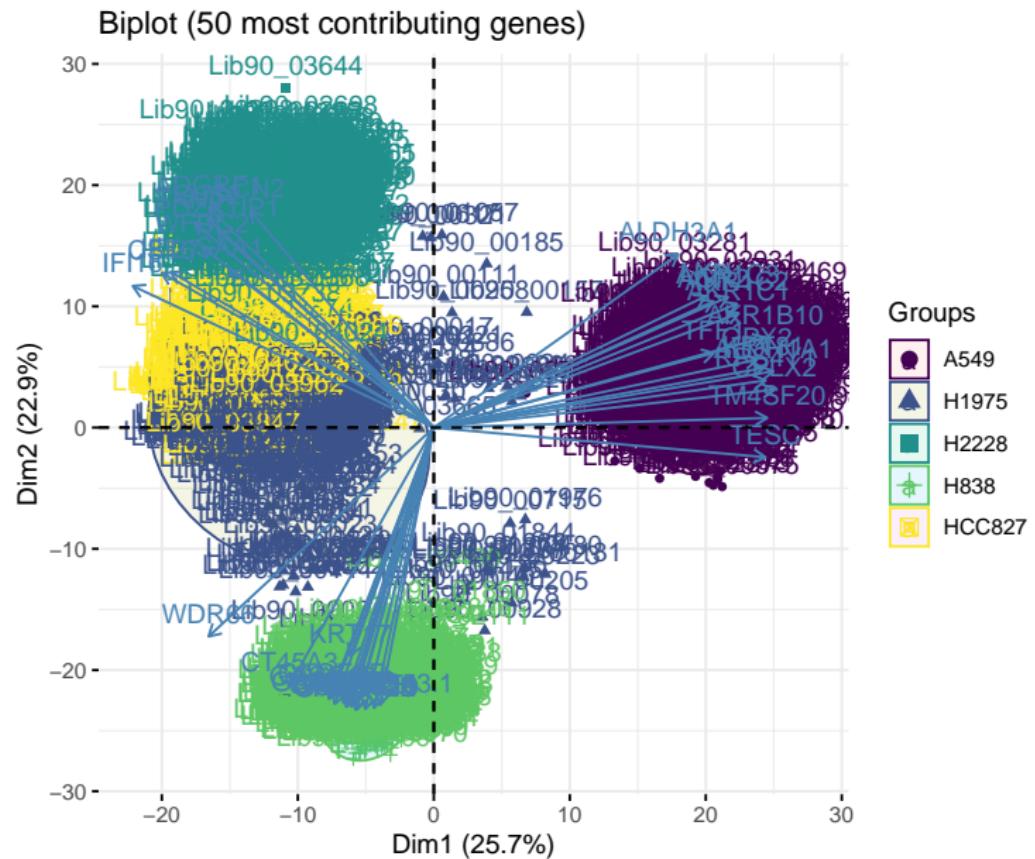
Pseudo R²

$$R_k^2 = \frac{\ell_k - \ell_{\min}}{\ell_{\max} - \ell_{\min}}$$

Poisson-lognormal PCA for the scRNA data set



Poisson-lognormal PCA for the scRNA data set



Variational Auto-Encoders

Highly non-linear model

Find Φ and $\tilde{\Phi}$ with **two** neural-networks, controlling the error.

$$\epsilon(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i))\|^2 + \text{regularization}(\Phi, \tilde{\Phi})$$

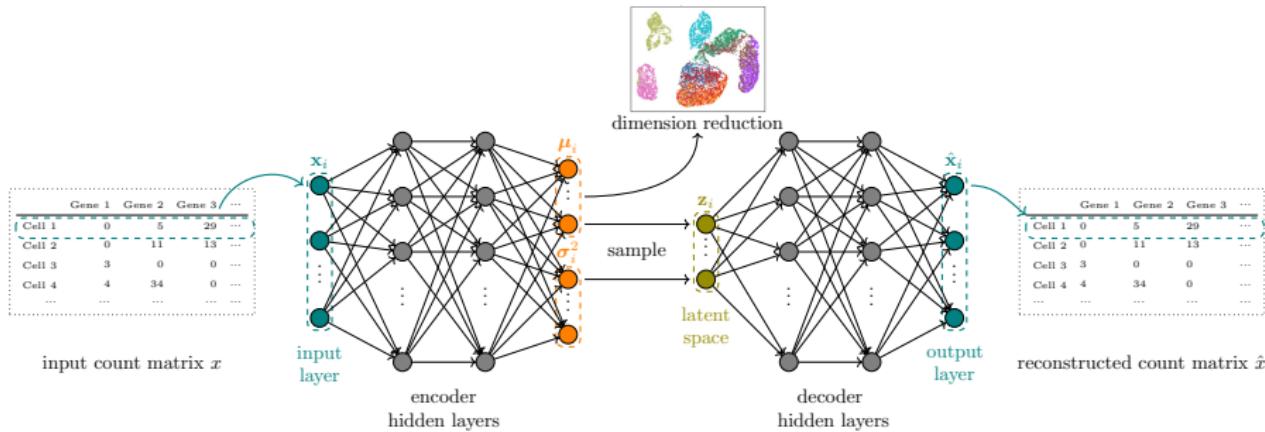


Figure 7: Figure by Hugo Gangloff



Variational Auto-Encoders

Decoder: Generative model

$$p_{\theta}(\mathbf{X}_i, \mathbf{Z}_i) = p_{\theta}(\mathbf{Z}_i)p_{\theta}(\mathbf{X}_i|\mathbf{Z}_i), \text{ with } \begin{cases} p_{\theta}(\mathbf{Z}_i) &= \mathcal{N}(0, \mathbf{I}_q), \\ p_{\theta}(\mathbf{X}_i|\mathbf{Z}_i) &\text{cond. likelihood.} \end{cases}$$

Encoder: Variational Inference model

The encoder approximate the posterior distribution with $q_{\psi}, \psi = \{\boldsymbol{\mu}_i, \boldsymbol{\sigma}^2\}$:

$$q_{\psi}(\mathbf{Z}_i|\mathbf{X}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 \mathbf{I}_q) \approx p_{\theta}(\mathbf{Z}_i|\mathbf{X}_i)$$

Optimization/training

Maximize a lower bound of the marginal log $p_{\theta}(\mathbf{X})$ (a.k.a the ELBO):

$$\log p_{\theta}(\mathbf{X}_i) \geq \mathcal{E}_{\theta, \psi}(\mathbf{X}_i) = \mathbb{E}_{q_{\psi}(\mathbf{Z}_i|\mathbf{X}_i)} [\log p_{\theta}(\mathbf{X}_i|\mathbf{Z}_i)] - D_{KL}(q_{\psi}(\mathbf{Z}_i|\mathbf{X}_i) \| p_{\theta}(\mathbf{Z}_i))$$



Variational Auto-Encoders

Likelihoods relevant for count data

- Data scaled to $[0,1]$ + Continuous Bernoulli (CB) likelihood [6]
- (Zero Inflated) Negative Binomial (ZINB) likelihood [7]
- **(Zero Inflated) Poisson likelihood** (tried this with Hugo Gangloff)

Let $\lambda \in (\mathbb{R}_*)^p$ and $\rho \in [0, 1]^p$ be the outputs of the decoder,

$$p_{\theta}(\mathbf{x}|\mathbf{z}) = \prod_{j=1}^p \begin{cases} \rho_j + (1 - \rho_j)p_{\theta}^{Poiss}(x_{m,n}|\lambda_n), & x_{ij} = 0, \\ (1 - \rho_j)p_{\theta}^{Poiss}(x_{ij}|\lambda_n), & x_{ij} > 0. \end{cases}$$

Promising works and questions

- [8]: Gaussian Mixture VAE + Zero inflated likelihoods
- [9]: Semi-supervised VA
- Us: Connexion with traditional variational inference

Variational Auto-Encoders on scRNA data³

- encoder dimensions: [256, 128, 64]
- decoder dimensions: [64, 128, 256]
- ADAM with learning rate = 1e-3

Negative-Binomial distribution

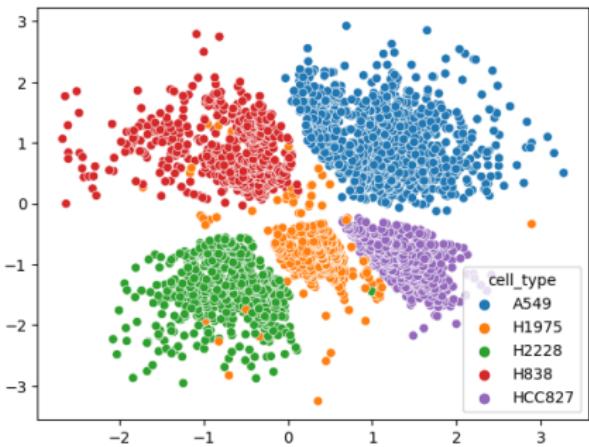


Figure 8: Negative Binomial

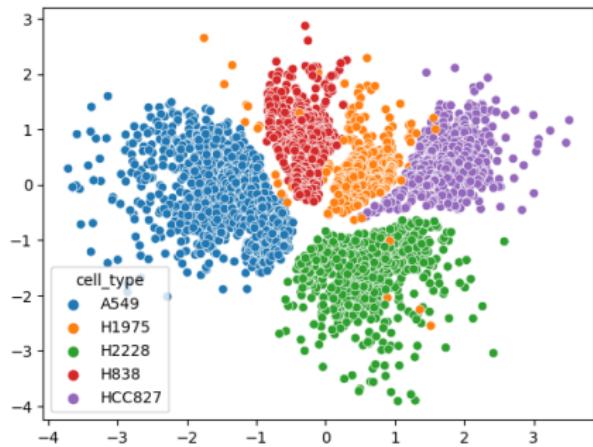


Figure 9: Zero-Inflated Negative Binomial

³based on code by Hugo Gangloff



Variational Auto-Encoders on scRNA data⁴

- encoder dimensions: [256, 128, 64]
- decoder dimensions: [64, 128, 256]
- ADAM with learning rate = 1e-3

Poisson distribution

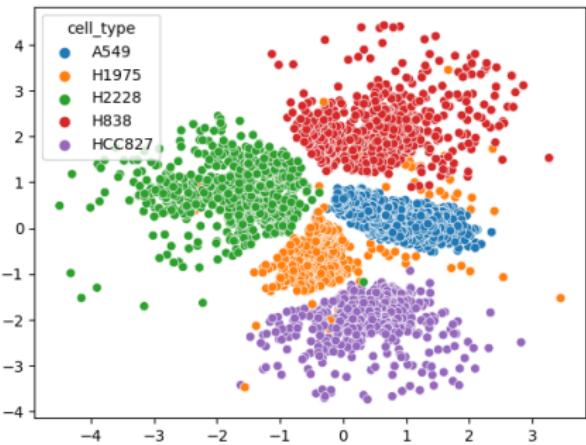


Figure 10: Poisson

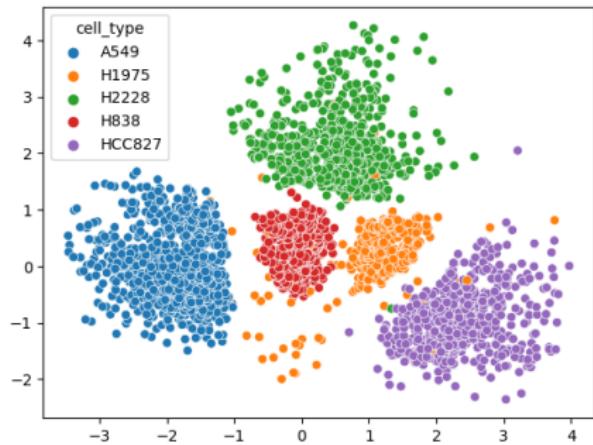


Figure 11: Zero-Inflated Poisson

⁴based on code by Hugo Gangloff

Outline

- 1 Introduction
- 2 Background: Geometric view of PCA
- 3 Reconstruction error approach
- 4 Model-based approaches
- 5 Preserving pairwise relations
- 6 Probabilistic Neighborhood Embedding



Pairwise Relation

Focus on pairwise relation $\mathcal{R}(\mathbf{x}_i, \mathbf{x}_{i'})$.

Distance Preservation

Construct a map Φ from the space \mathbb{R}^p into a space \mathbb{R}^q of **smaller dimension**:

$$\begin{aligned}\Phi : \quad & \mathbb{R}^p \rightarrow \mathbb{R}^q, q \ll p \\ & \mathbf{x} \mapsto \Phi(\mathbf{x})\end{aligned}$$

such that $\mathcal{R}(\mathbf{x}_i, \mathbf{x}_{i'}) \sim \mathcal{R}'(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_{i'})$

Multidimensional scaling

A.k.a PCoA (Principal Coordinates Analysis)

Try to preserve inner product related to the distance (e.g. Euclidean)

t-SNE – Stochastic Neighborhood Embedding



Try to preserve relations with close neighbors with Gaussian kernel

Multidimensional scaling

a.k.a Principale Coordinates Analysis

Consider a $n \times n$ (dis)similarity matrix associated to points $\mathbf{x}_i \in \mathbb{R}^p$

Goal: find $\mathbf{z}_i \in \mathbb{R}^q$ while preserving the (dis)similarities in the latent space

Classical MDS model

Measure similarities with the (centered) **inner product** and minimize

$$\text{Stress}^{cMDS}(\mathbf{z}_i) = \sum_{i \neq i'} \left((\mathbf{x}_i - \boldsymbol{\mu})^\top (\mathbf{x}_i - \boldsymbol{\mu}) - \mathbf{z}_i^\top \mathbf{z}_{i'} \right)^2,$$

Assuming a linear model $\mathbf{z} = \Phi(\mathbf{x}) = \mathbf{V}^\top (\mathbf{x}_i - \boldsymbol{\mu})$, with $\mathbf{V} \in \mathcal{O}_{p \times q}$, minimizing $\text{Stress}^{cMDS}(\mathbf{z}_i)$ is dual to PCA and leads to

$$\mathbf{z} = \mathbf{X}^c \mathbf{V} = \mathbf{U} \mathbf{D} \mathbf{V}^\top \mathbf{V} = \mathbf{U} \mathbf{D}.$$

☞ The principal coordinates in \mathbb{R}^q correspond to the scores of the n individuals projected on the first q principal components.



Metric Multidimensional Scalings

Idea to generalize classical MDS:

preserving similarities in term of **inner product** amounts to preserve dissimilarity in terms of Euclidean distance

Least-squares/Kruskal-Shephard scaling

Use a distance base formulation with the following loss (Stress) function:

$$\text{Stress}^{KS} = \sum_{i \neq i'} (d_{ii'} - \|\mathbf{z}_i - \mathbf{z}_{i'}\|)^2,$$

- Almost equivalent to classical MDS when d is the Euclidean distance
- Generalize to any **quantitative** dissimilarity/distance d

Sammong mapping - Variant of the loss (Stress) function

$$\text{Stress}^{SM} = \sum_{i \neq i'} \frac{(d_{ii'} - \|\mathbf{z}_i - \mathbf{z}_{i'}\|)^2}{d_{ii'}}.$$



Stochastic Neighbor Embedding (SNE)

High dimensional space

Let $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ be the original points in \mathbb{R}^p , and measure similarities by

$$p_{ij} = (p_{j|i} + p_{i|j})/2n, \quad \text{with } p_{j|i} = \frac{\exp(-\|\mathbf{x}_j - \mathbf{x}_i\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_k - \mathbf{x}_i\|^2/2\sigma_i^2)}$$

- preserves relations with **close neighbors**
- σ_i adjusts to local densities (neighborhood of i)

Perplexity

A smoothed effective number of neighbors:

$$Perp(p_i) = 2^{H(p_i)}, \quad H(p_i) = - \sum_{j=1}^n p_{j|i} \log_2 p_{j|i}$$

☞ σ_i found by binary search to match a user-defined perplexity for p_i



tSNE and Student / Cauchy kernels

Similarities in the low dimension space

Let $(\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n)$ be the points in the the low-dimensional space $\mathbb{R}^{q=2}$

$$(\text{SNE}) \quad q_{ij} = \frac{\exp(-\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2)}{\sum_{k \neq i} \exp(-\|\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_j\|^2)}$$

$$(\text{t-SNE}) \quad q_{ij} = \frac{(1 + \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_k\|^2)^{-1}}$$

☞ t-SNE robustifies Gaussian kernel by using Student(1) (Cauchy) kernels

Optimization

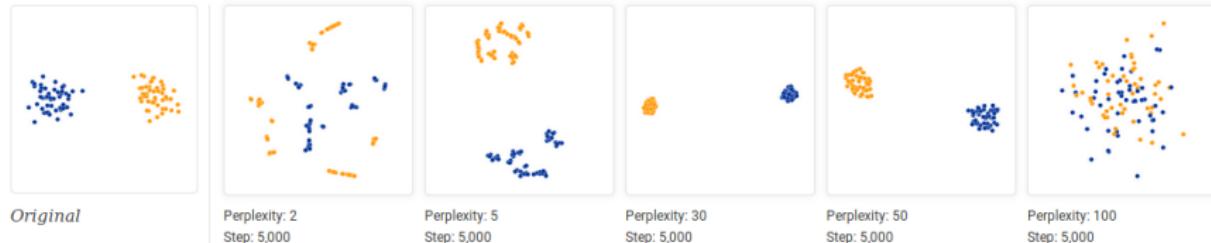
Criterion – Kullback-Leibler between p and q : $C(\tilde{\mathbf{x}}) = \sum_{ij} KL(p_{ij}, q_{ij})$

Algorithm – adaptive stochastic gradient initialized by $\mathcal{N}(0, \epsilon I_q)$

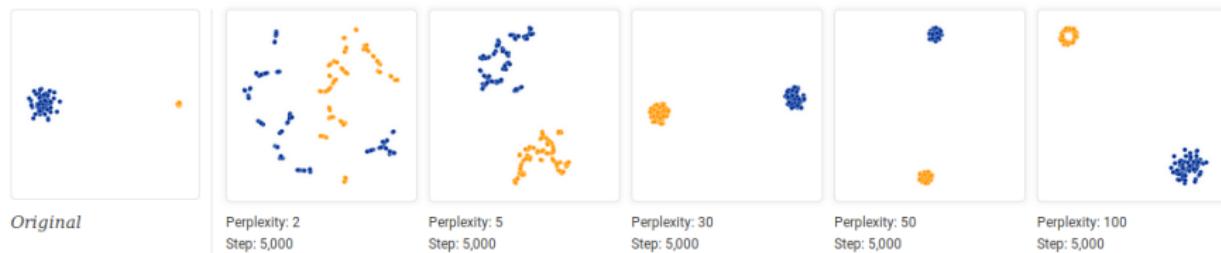
Initialization – reduce original data with PCA then initialized by $\mathcal{N}(0, \epsilon I_q)$

Empirical properties of tSNE (1)

Effect of Hyperparameters : Perplexity

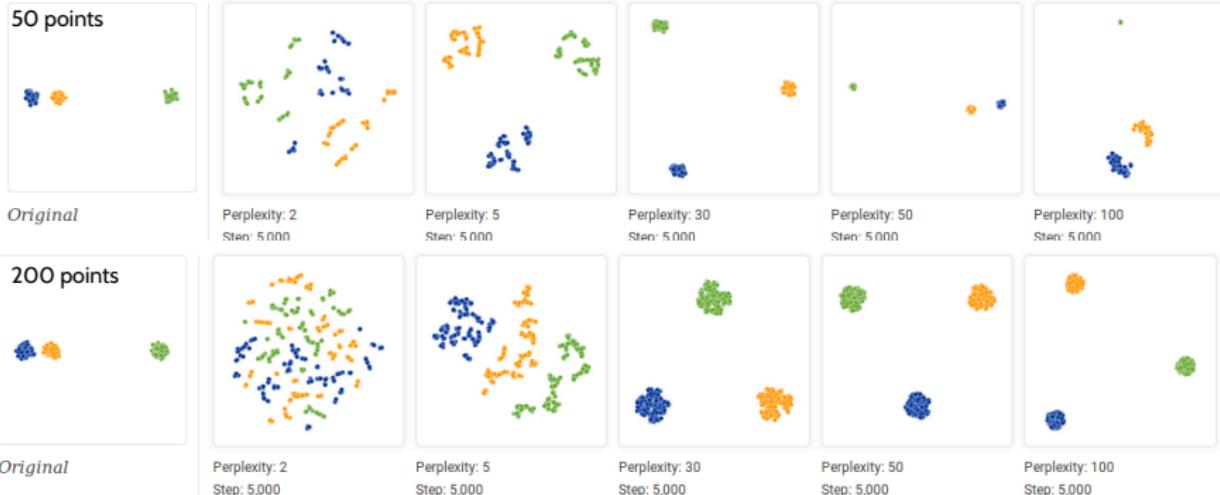


tSNE does not account for heteroscedasticity

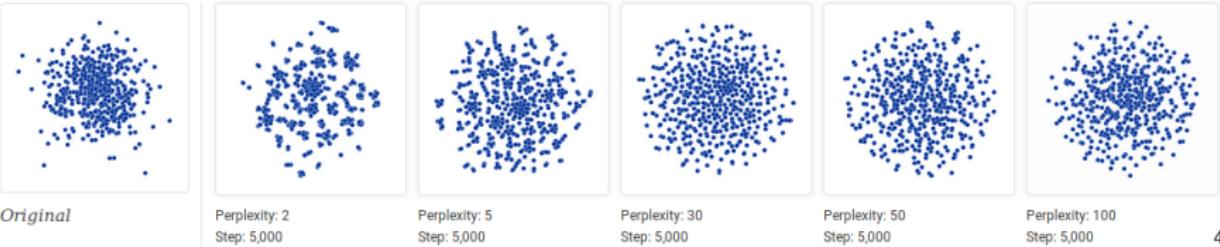


Empirical properties of tSNE (2)

tSNE does not account for between-cluster distance

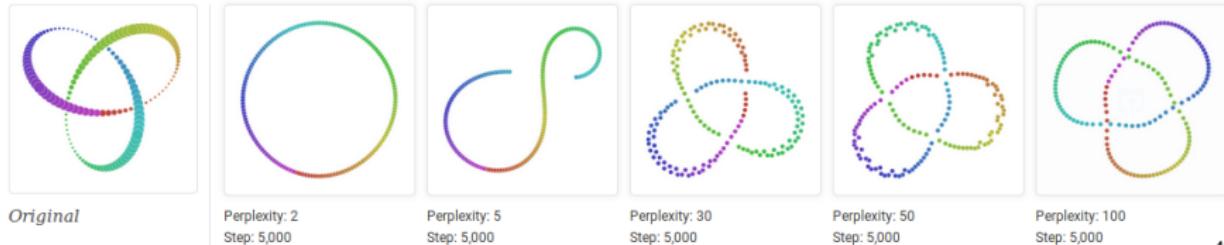
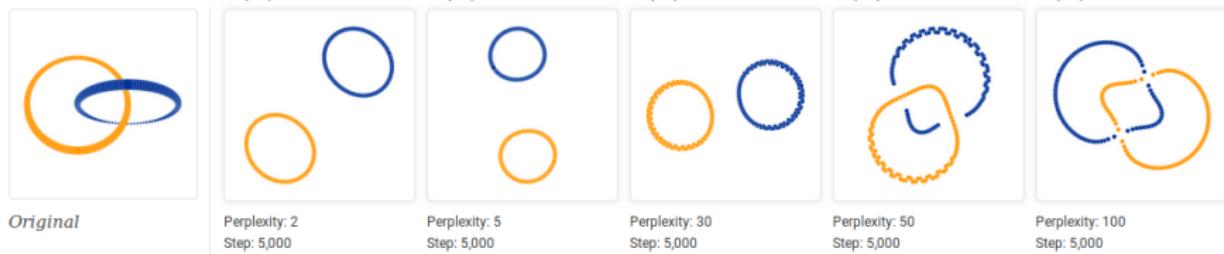


What about random noise ?



Empirical properties of tSNE (3)

Catching Complex Geometries



t-SNE: pros/cons

Properties

- good at preserving local distances (intra-cluster variance)
- not so good for global representation (inter-cluster variance)
- good at creating clusters of close points, bad at positioning clusters wrt each other

Limitations

- importance of preprocessing: initialize with PCA and feature selection plus log transform (non linear transform)
- percent of explained variance ? interpretation of the q distribution ?
- Lack of reproducibility due to stochastic optimization

Uniform Manifold Approximation and Projection [10]

For j in the k -neighborhood of i , define the conditional distribution

$$p_{j|i} = \exp\left(-\frac{\|X_i - X_j\|_2^2 - \rho_i}{\sigma_i}\right) \quad \text{with } \rho_i = \min_{j \neq i} \|X_i - X_j\|^2$$

and its symmetrized version

$$p_{ij} = p_{j|i} + p_{i|j} - p_{j|i}p_{i|j}.$$

Rely on a generalized Student-distribution with a, b fitted on the data:

$$q_{ij} = \left(1 + a\|Z_i - Z_j\|_2^{2b}\right)^{-1}$$

UMAP solves the following problem:

$$\min_{Z \in \mathbb{R}^{n \times d}} - \sum_{i < j} p_{ij} \log q_{ij} + (1 - p_{ij}) \log(1 - q_{ij})$$



tSNE and UMAP scRNA data⁵

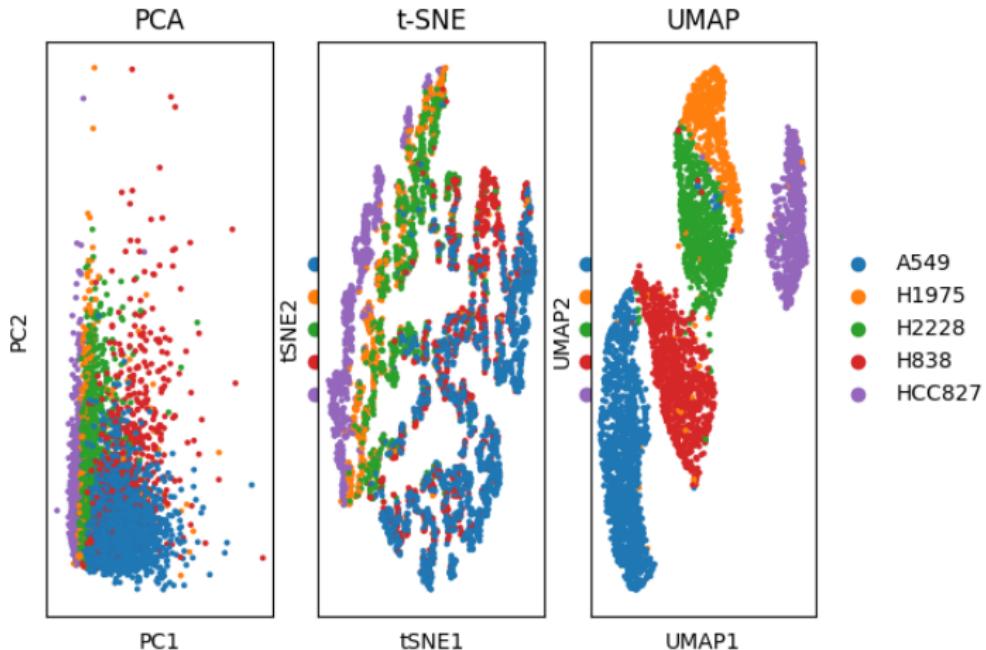


Figure 12: tSNE + UMAP on raw data

⁵using the Python module scanpy

tSNE and UMAP scRNA data⁶

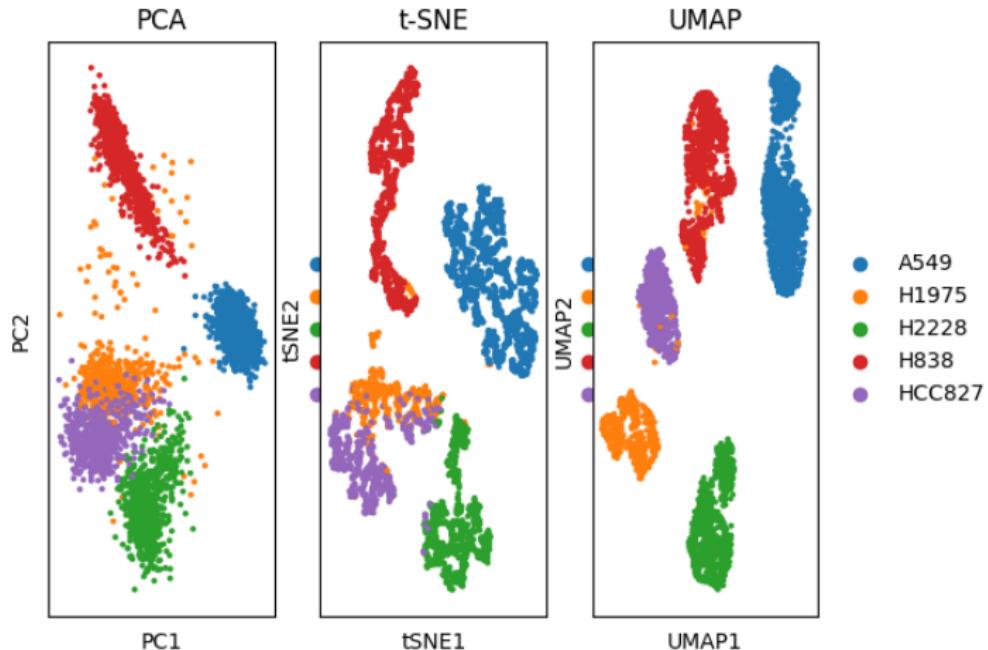


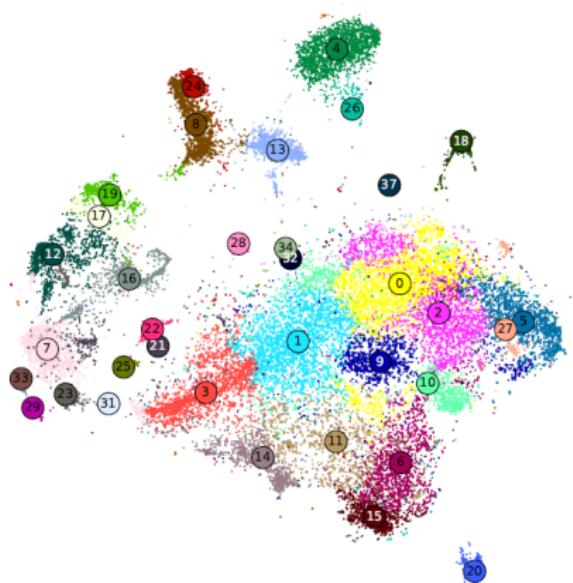
Figure 13: tSNE + UMAP on log-transformed data

⁶using the Python module `scipy`

tSNE on large scRNA Gene Expression [11]

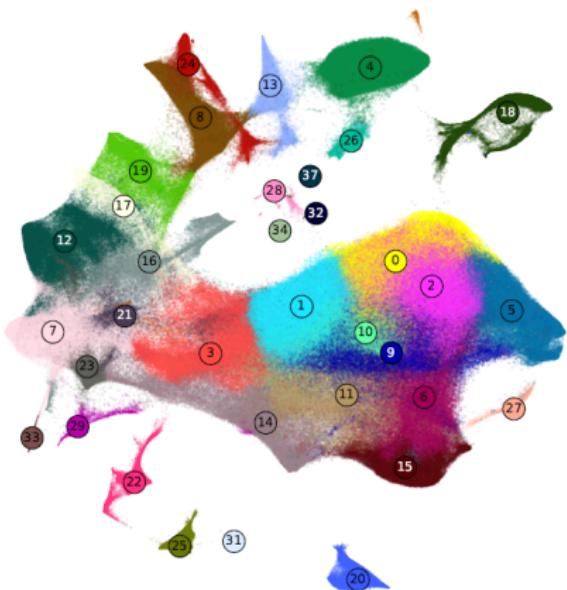
a

$N = 25\,000$



b

$N = 1\,306\,127$



Outline

- 1 Introduction
- 2 Background: Geometric view of PCA
- 3 Reconstruction error approach
- 4 Model-based approaches
- 5 Preserving pairwise relations
- 6 Probabilistic Neighborhood Embedding



Hidden Graph to structure observations

Consider W the adjacency matrix of a hidden random graph⁷

The graph Laplacian operator is the map L such that

$$L(\mathbf{W})_{ij} = \begin{cases} -W_{ij} & \text{if } i \neq j \\ \sum_{k \in [n]} W_{ik} & \text{otherwise.} \end{cases}$$

$L = L(\mathbf{W})$ has the following property:

$$\forall X \in \mathbb{R}^{n \times p}, \quad \sum_{i,j} W_{ij} \|X_i - X_j\|^2 = \text{tr}(X^T L X).$$

⁷we start with one connected component

Conditional distribution of X on a graph W_X

Consider a Matrix Normal model with row and column dependencies

$$X | W_X \sim \mathcal{MN}\left(0, L_X^{-1}, \Sigma^{-1}\right),$$

The conditional density relates to the Gaussian kernel

$$k(X_i - X_j) = \exp\left(-\frac{1}{2}\|X_i - X_j\|_{\Sigma}^2\right),$$

which can be generalized to translation invariant kernels:

$$\mathbb{P}(X | W_X) \propto \prod_{(i,j) \in [n]^2} k(\mathbf{X}_i - \mathbf{X}_j)^{W_{X,ij}}.$$

Conditional distribution of Z on a graph W_Z

Consider that the low-dimensional representation is also structured according to a graph

$$Z \mid W_Z \sim \mathcal{MN}\left(0, L_Z^{-1}, I_q\right),$$

with the Gaussian kernel for Z

$$k(Z_i - Z_j) = \exp\left(-\frac{1}{2}\|Z_i - Z_j\|_{I_q}^2\right),$$

The Conditional distribution of $Z \mid W_Z$ is

$$\mathbb{P}(Z \mid W_Z) \propto \prod_{(i,j) \in [n]^2} k(Z_i - Z_j)^{W_{Z,ij}}$$



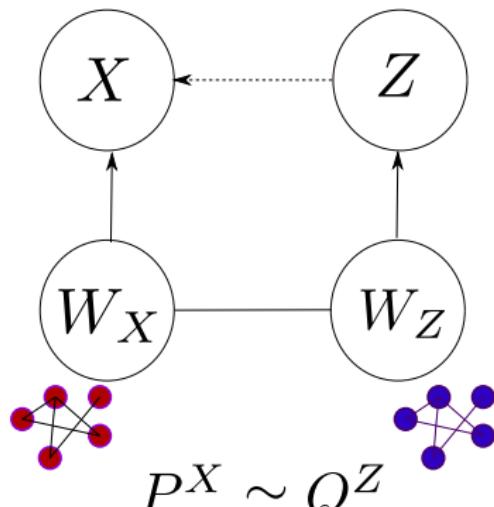
Embedding with Graph Coupling

Couple the 2 hidden graphs W_X and W_Z in a probabilistic way by matching their posterior distributions:

$$P^X = \mathbb{P}(W_X | X)$$

$$Q^Z = \mathbb{P}(W_Z | X; Z)$$

⤱ Z becomes a parameter to be estimated



Probabilistic Coupling

Graph Coupling with Z as a parameter

Consider the cross entropy between posteriors

$$\mathcal{H}(\mathbf{P}^X, \mathbf{Q}^Z) = -\mathbb{E}_{W_X \sim \mathbf{P}^X} \left(\log \mathbb{P}(W_Z = W_X \mid X; Z) \right)$$

Find the best low-dimensional representation such that the two graphs match

$$Z(X) = \arg \min_Z \left\{ \mathcal{H}(\mathbf{P}^X, \mathbf{Q}^Z) \right\}$$

Connection with the KL between posteriors

$$\text{KL}(\mathbf{P}^X, \mathbf{Q}^Z) = \mathcal{H}(\mathbf{P}^X, \mathbf{Q}^Z) - \mathcal{H}(\mathbf{P}^X, \mathbf{P}^X)$$



Conjugate priors and posteriors for hidden graphs

Consider a prior distribution for the hidden graph in the general form

$$\mathbb{P}_{\mathcal{P}}(\mathbf{W}; \boldsymbol{\pi}) \propto \underbrace{\mathcal{C}_k(W)^\alpha}_{\alpha=0} \Omega_{\mathcal{P}}(\mathbf{W}) \prod_{(i,j) \in [n]^2} \pi_{ij}^{W_{ij}}$$

For the following priors family, we derive the posterior $\mathbb{P}_{\mathcal{P}}(\mathbf{W} \mid X; \boldsymbol{\pi}, k)$

\mathcal{P}	$\Omega_{\mathcal{P}}(\mathbf{W})$	Prior for W
\mathcal{B} Bernoulli	$\prod_{ij} \mathbf{1}_{W_{ij} \leq 1}$	$\mathcal{B}\left(\frac{\pi_{ij}}{1+\pi_{ij}}\right)$
\mathcal{D} Unitary Fixed degree	$\prod_i \mathbf{1}_{W_{i+} = 1}$	$\mathcal{M}\left(1, \frac{\boldsymbol{\pi}_i}{\pi_{i+}}\right)$
\mathcal{E} Fixed Number of edges	$\prod_{ij} (W_{ij}!)^{-1}$	$\mathcal{M}\left(n, \frac{\boldsymbol{\pi}}{\pi_{++}}\right)$

$\pi_{ij}k_{ij} = \pi_{ij}k(X_i - X_j)$ is the posterior strength of edges (normalized or not)

Mixing Prior distributions for coupling

Priors for W_X, W_Z induce posteriors $\mathbf{P}^{\mathcal{P}_X}, \mathbf{Q}^{\mathcal{P}_Z}$ matched with cross entropy $\mathcal{H}(\mathbf{P}^{\mathcal{P}_X}, \mathbf{Q}^{\mathcal{P}_Z})$



Model-based Neighbor Embedding

Choosing $\mathcal{P}_X = \mathcal{P}_Z = \mathcal{D}$ lead us to $\mathcal{H}_{D,D} = - \sum_{i \neq j} P_{ij}^D \log Q_{ij}^D$ and

$$P_{ij}^D = \frac{\pi_{ij} k(X_i - X_j)}{\sum_{\ell=1}^n \pi_{i\ell} k(X_i - X_\ell)}, \quad Q_{ij}^D = \frac{\pi_{ij} k(Z_i - Z_j)}{\sum_{\ell=1}^n \pi_{i\ell} k(Z_i - Z_\ell)}.$$

We defined the generative model for SNE! Similarly,

Algorithm	Input Similarity	Latent Similarity	Loss Function
SNE	$P_{ij}^D = \frac{k_x(\mathbf{x}_i - \mathbf{x}_j)}{\sum_{\ell} k_x(\mathbf{x}_i - \mathbf{x}_{\ell})}$	$Q_{ij}^D = \frac{k_z(\mathbf{z}_i - \mathbf{z}_j)}{\sum_{\ell} k_z(\mathbf{z}_i - \mathbf{z}_{\ell})}$	$-\sum_{i \neq j} P_{ij}^D \log Q_{ij}^D$
Sym-SNE	$\bar{P}_{ij}^D = P_{ij}^D + P_{ji}^D$	$Q_{ij}^E = \frac{k_z(\mathbf{z}_i - \mathbf{z}_j)}{\sum_{\ell} k_z(\mathbf{z}_i - \mathbf{z}_{\ell})}$	$-\sum_{i < j} \bar{P}_{ij}^D \log Q_{ij}^E$
LargeVis	$\bar{P}_{ij}^D = P_{ij}^D + P_{ji}^D$	$Q_{ij}^B = \frac{k_z(\mathbf{z}_i - \mathbf{z}_j)}{1 + k_z(\mathbf{z}_i - \mathbf{z}_j)}$	$-\sum_{i < j} \bar{P}_{ij}^D \log Q_{ij}^B + \left(2 - \bar{P}_{ij}^D\right) \log(1 - Q_{ij}^B)$
UMAP	$\widetilde{P}_{ij}^B = P_{ij}^B + P_{ji}^B - P_{ij}^B P_{ji}^B$	$Q_{ij}^B = \frac{k_z(\mathbf{z}_i - \mathbf{z}_j)}{1 + k_z(\mathbf{z}_i - \mathbf{z}_j)}$	$-\sum_{i < j} \widetilde{P}_{ij}^B \log Q_{ij}^B + \left(1 - \widetilde{P}_{ij}^B\right) \log(1 - Q_{ij}^B)$



References I

- [1] TIPPING, M. E. and BISHOP, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **61** 611–22.
- [2] CHIQUET, J., MARIADASSOU, M. and ROXIN, S. (2018). Variational inference for probabilistic poisson PCA. *The Annals of Applied Statistics* **12** 2674–98.
- [3] WAINWRIGHT, M. J. and JORDAN, M. I. (2008). Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.* **1** 1–305.
- [4] BLEI, D. M., KUCUKELBIR, A. and McAULIFFE, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association* **112** 859–77.
- [5] CHIQUET, J., MARIADASSOU, M. and ROBIN, S. (2021). The poisson-lognormal model as a versatile framework for the joint analysis of species abundances. *Frontiers in Ecology and Evolution* **9**.



References II

- [6] WANG, D. and GU, J. (2018). VASC: Dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics* **16** 320–31.
- [7] DONY, L., KÖNIG, M., FISCHER, D. and THEIS, F. J. (2020). Variational autoencoders with flexible priors enable robust distribution learning on single-cell RNA sequencing data. In *ICML 2020 workshop on computational biology (WCB) proceedings paper* vol 37.
- [8] GRØNBECH, C. H., VORDING, M. F., TIMSHEL, P. N., SØNDERBY, C. K., PERS, T. H. and WINTHER, O. (2020). scVAE: Variational auto-encoders for single-cell gene expression data. *Bioinformatics* **36** 4415–22.
- [9] SENINGE, L., ANASTOPOULOS, I., DING, H. and STUART, J. (2021). VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature communications* **12** 1–9.

References III

- [10] McINNES, L., HEALY, J. and MELVILLE, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. *Arxiv* 1–63.
- [11] KOBAK, D. and BERENS, P. (2018). The art of using t-SNE for single-cell transcriptomics. *bioRxiv*.