

# Dimension Reduction and Life Sciences

## Panorama and Probabilistic View of some Recent Approaches

Julien Chiquet

UMR MIA Paris-Saclay, AgroParisTech, INRAE

May 31, 2023

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



# Outline

- 1 Introduction
- 2 Reconstruction error approach
- 3 Preserving pairwise relations



# Exploratory analysis of (modern) data sets

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

$\mathbf{X}_{n \times p} =$

			$x_{ij}$			

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

## Challenges

- **Large** ( $n$  and  $p$  grows) and **high dimensional** ( $n$  grows but  $\ll p$ )
- **Redundancy** many variables may carry the same information
- **Discrete**: measures with counts are as common as with intensity

## Dimension reduction: general goals

Find a **low-dimensional representation** that captures the “essence” (local and/or global structure, signal) of the original data

- **ML**: preprocessing, denoising, compression
- **Stat.**: descriptive/exploratory methods, visualization

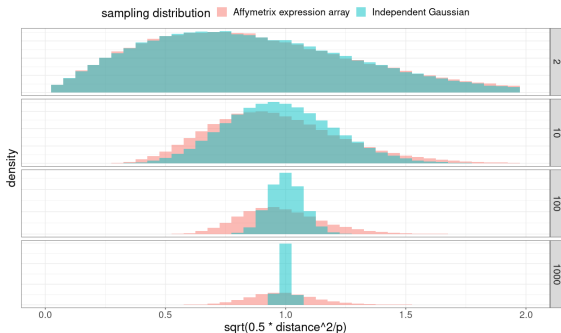
# Dimensionality curse

## Theorem (Folks theorem)

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

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

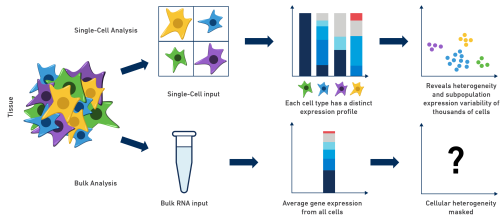
➡ When  $p$  is large, all the points are almost equidistant



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

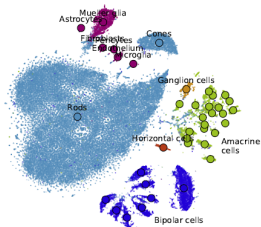
# Example in genomics

## Genome-wide cell biology with single-cell RNAseq data

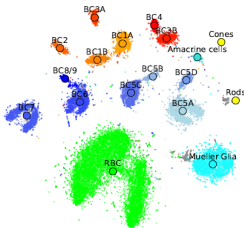


Describe cell population ( $n \rightarrow 10^6$ )  
with high dimensional molecular  
features ( $p \rightarrow 10^5$ )

**a** Macosko et al. 2015



**b** Shekhar et al. 2016



**c** Harris et al. 2018

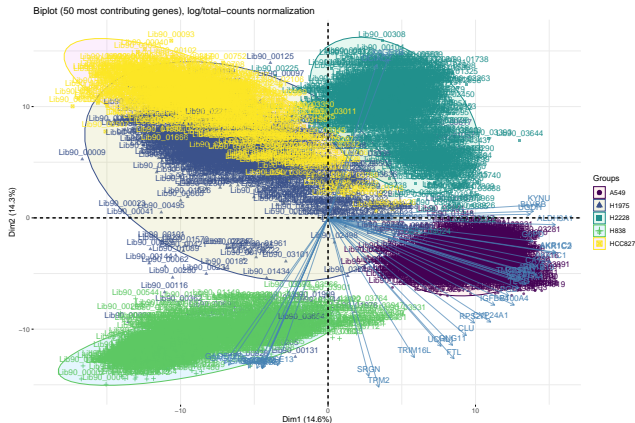


**Figure 1:** Successful t-SNE visualizations of sc-RNAseq data

# Single-Cell data analysed with PCA

Toy single-cell RNA data set ([https://github.com/LuyiTian/sc\\_mixology/](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).



# Beyond PCA and linear methods

## **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**
- Focus on **relationship preservation** to keep local characteristics

➡ Gain in versatility with **probabilistic/model-based approaches**

## Challenges

With, non-linear transformations...

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

# Outline

- 1 Introduction
- 2 Reconstruction error approach**
- 3 Preserving pairwise relations





# Principle

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

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

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

- **Log-likelihood** of a parametric model  $p_\theta$ , with  $\hat{\mathbf{X}} = \mathbb{E}_{\hat{\theta}}(\cdot)$ :

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



# PCA and Reconstruction error

## 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**

## Reconstruction error

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

## Solution (explicit)

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



# Other methods with same rational

## Linear models with other constraints

$$(\hat{\boldsymbol{\mu}}, \hat{\mathbf{V}}, \hat{\mathbf{Z}}) = \arg \min \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i))\|^2, \quad \hat{\mathbf{x}} = \tilde{\Phi}(\mathbf{z}) = \boldsymbol{\mu} + \mathbf{V}_q \mathbf{z}$$

- **sparse PCA**:  $\mathbf{V}_q$  sparse, possibly orthogonal
- **Dictionary learning**:  $\mathbf{Z}$  sparse
- **Independent Component Analysis** ( $z^j, z^{j'}$ ) independent

**Kernel-PCA**: non linear transformation of the input  $\Psi(\mathbf{x}_i)$ , then PCA:

$$\Phi(\mathbf{x}) = \mathbf{V}_q^\top \Psi(\mathbf{x} - \boldsymbol{\mu}) = \mathbf{z}, \quad \Psi : \mathbb{R}^p \rightarrow \mathbb{R}^n$$

## Non Linear Matrix Factorization

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

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



# Probabilistic Gaussian PCA (Tipping and Bishop 1999)

## Generative model

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

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

By direct integration<sup>1</sup>, the marginal distribution of the observation is

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

↪ rank- $q$  decomposition of the covariance matrix + noise.

---

<sup>1</sup>easy 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

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

Latent position: posterior distribution

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

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



# Estimation: alternative

## Expectation-Maximization

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

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

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  $\theta$  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  $\mathbf{C}, \sigma^2$

**On-going:** Fast JAX implementation by Hugo Gangloff, mixture of pPCA with Pierre Barbillon and MsC intern Pierre Brand



# PCA for counts: Poisson lognormal PCA

Generative Model (Chiquet, Mariadassou, and Robin 2018)

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

## Estimation: Issues

- The marginal distribution is hard to compute, even numerically

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

↪ no direct MLE possible

- Posterior distribution of  $\mathbf{Z}_i$  has no close form

↪ no genuine application of EM possible

# Variational inference (Chiquet, Mariadassou, and Robin 2021)

## Variational approximation (Blei, Kucukelbir, and McAuliffe 2017)

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

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

- 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 PCA (PLN-PCA)

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  $\psi$  for given  $\theta$
- M-step: solve in  $\theta$  for given  $\psi$





# Model selection and Visualization for PLN-PCA

## Selection of number of components (rank $k$ )

Use likelihood lower bound in information criteria, e.g,

$$\hat{k} = \arg \max_k \text{vBIC}_k \quad \text{with } \text{vBIC}_k = J(\hat{\beta}, \tilde{p}) - \frac{1}{2}p(d+k)\log(n)$$

## Visualization: non-nested subspaces ( $\neq$ Gaussian PCA)

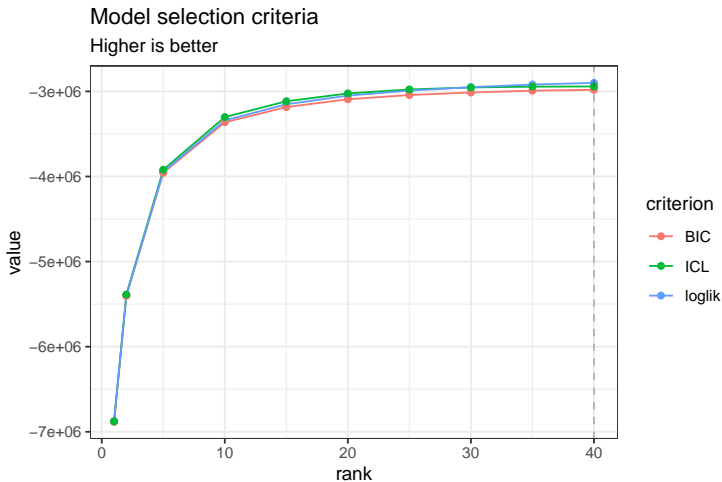
For the selected dimension  $\hat{k}$ , compute the estimated latent positions  $\mathbb{E}_q(\mathbf{Z}_i)$  and perform PCA

## Goodness of fit: deviance based criterion

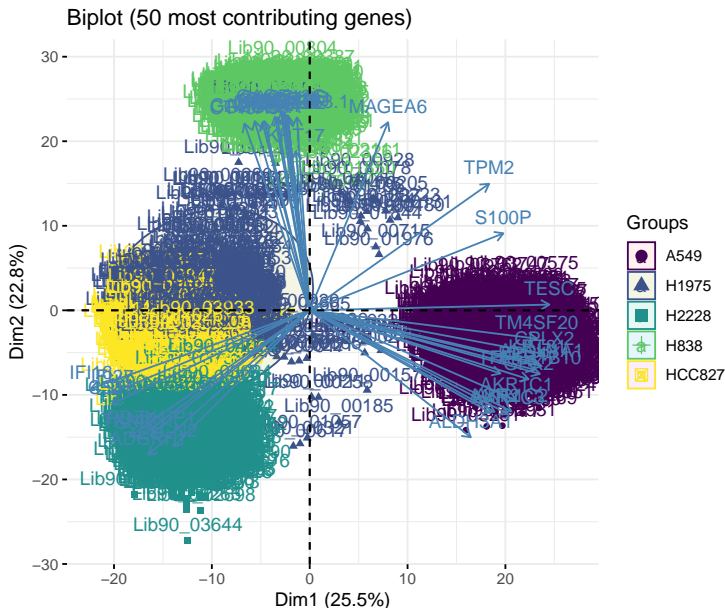
For  $\ell_k = \log \mathbb{P}(\mathbf{X}; \lambda^{(k)})$  the Poisson likelihood,

$$R_k^2 = \frac{\ell_k - \ell_0}{\ell_{\max} - \ell_0}, \quad \text{with } \lambda_{ij}^{(k)} = \exp\left(\mathbb{E}_q(Z_{ij}^{(k)})\right), \quad \lambda_{ij}^{\max} = Y_{ij}.$$

# Poisson-lognormal PCA for the scRNA data set



# Poisson-lognormal PCA for the scRNA data set



Gaussian mixture in the **common** latent  $q$ -dimensional subspace

$$\mathbf{G}_i \sim \mathcal{M}(1, \boldsymbol{\pi} = (\pi_1, \dots, \pi_K)) \quad (\text{clustering})$$

$$\mathbf{Z}_i \sim \mathcal{N}_q(\mathbf{0}, \mathbf{I}_q) \quad (\text{subspace})$$

$$\mathbf{W}_i \mid \mathbf{G}_{ik} = 1 \sim \mu_k + \sigma_k \mathbf{Z}_i \quad (\text{linear transform})$$

$$\mathbf{X}_i \mid \mathbf{W}_i, \mathbf{G}_i \sim \mathcal{P}(\exp(\mathbf{C}\mathbf{W}_i)) \quad (\text{emission})$$

With parameters  $\{\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, \mathbf{C}, \boldsymbol{\pi}\}$ : use variational inference.

## More general models

- Add covariate effects
- Use diagonal variance (rather than spherical)  $\sigma_k^2 \mathbf{I}_q \rightarrow \mathbf{D}_k$
- Use different  $\mathbf{C}_k \rightsquigarrow$  no common projection

# Variational Auto-Encoders (Kingma and Welling 2013)

## Highly non-linear model

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

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

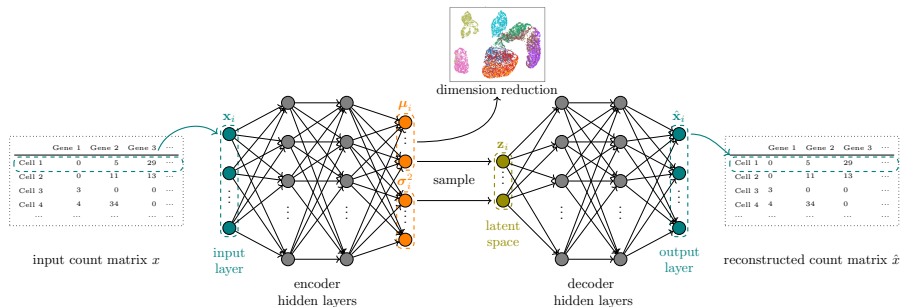


Figure 2: Figure by Hugo Gangloff

# 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 (Wang and Gu 2018)
- (Zero Inflated) Negative Binomial (ZINB) likelihood (Dony et al. 2020)
- **(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,

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

## Promising works and questions

- Grønbech et al. (2020): Gaussian Mixture VAE
- Seninge et al. (2021): Semi-supervised VA
- Us: Connexion with traditional variational inference
- Us: Use as block in wider model-based approaches



# Variational Auto-Encoders on scRNA data<sup>2</sup>

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

## Negative-Binomial distribution

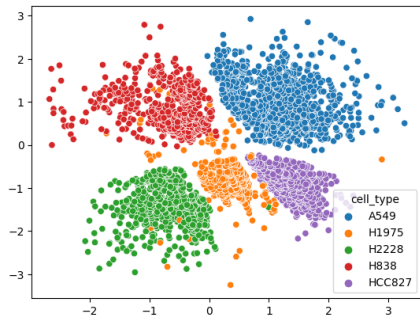


Figure 3: Negative Binomial

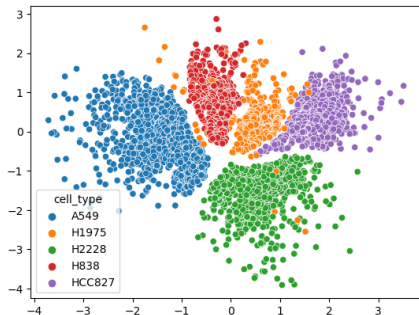


Figure 4: Zero-Inflated Negative Binomial

<sup>2</sup>based on code by Hugo Gangloff



# Variational Auto-Encoders on scRNA data<sup>3</sup>

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

## Poisson distribution

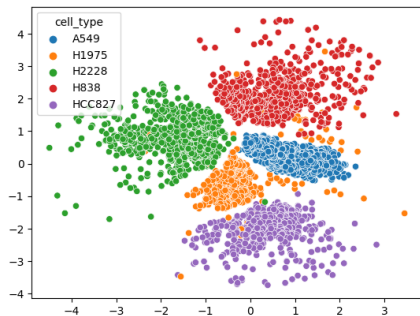


Figure 5: Poisson

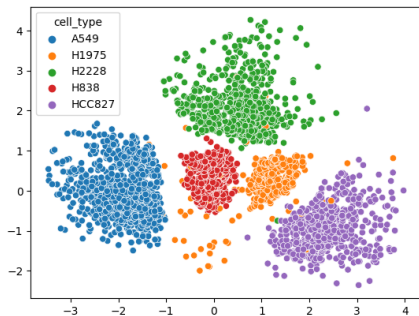


Figure 6: Zero-Inflated Poisson

<sup>3</sup>based on code by Hugo Gangloff

# Outline

- 1 Introduction
- 2 Reconstruction error approach
- 3 Preserving pairwise relations**



# Principle

Consider an  $n \times n$  (dis)similarity matrix associated to  $\mathbf{x}_i \in \mathbb{R}^p$ , measuring pairwise relations  $\mathcal{R}(\bullet, \bullet')$ , using one among

- distances,
- kernels,
- inner products,
- probability distributions.

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

Preserve local properties

Find a map  $\Phi$  from  $\mathbb{R}^p \rightarrow \mathbb{R}^q$  such that

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

↪ preserve  $\mathcal{R}$  both in high and low dimensional spaces to catch complex geometries



# Multidimensional scaling

a.k.a Principale Coordinates Analysis

## Classical Multidimensional Scalings

Preserve similarities in terms **inner product**:

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

## Metric Multidimensional Scalings

Remarking that cMDS amount to preserve dissimilarities in terms of Euclidean distance, use

$$\text{Stress}(\mathbf{z}_1, \dots, \mathbf{z}_n) = \sum_{i \neq i'} (d_{ii'} - \|\mathbf{z}_i - \mathbf{z}_{i'}\|)^2,$$

↪ Generalize to other dissimilarities/distances or stress functions



# Some Embedding methods

## Isomap (Balasubramanian and Schwartz 2002)

- Build a  $k$ -nearest neighbor graph with adjacency matrix  $\mathbf{W}$
- Weight edges by  $W_{ii'} = \|\mathbf{x}_i - \mathbf{x}_{i'}\|$
- Compute the shortest path distance
- Embeds the distance with MDS.

## Laplacian Eigenmaps (Belkin and Niyogi 2003)

- Build a  $k$ -nearest neighbor graph with adjacency matrix  $\mathbf{W}$
- Weight edges with Gaussian kernel  $W_{ii'} = \exp(\|\mathbf{x}_i - \mathbf{x}_{i'}\|^2 / \sigma^2)$
- Compute the graph Laplacian  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  with  $\mathbf{D}$  diagonal with degrees
- Embeddings are obtained with the first eigenvectors associated to positive eigenvalues of  $\mathbf{L}$ .



# Classical embeddings on scRNA data set<sup>4</sup>

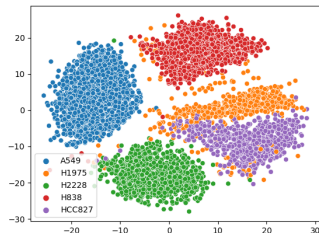


Figure 7: Multidimensional Scaling

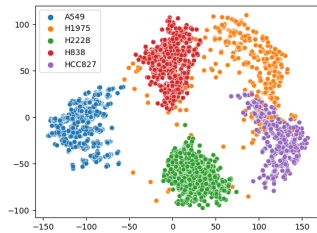


Figure 8: Isomap

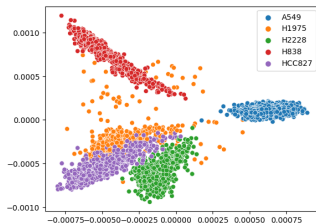


Figure 9: Laplacian Eigenmap

<sup>4</sup>using `sklearn.manifold`

# Stochastic Neighbor Embedding (SNE) (Hinton and Roweis 2002)

## 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**
- $\sigma_i$  adjusts to local densities (neighborhood of  $i$ )

## Perplexity

A smoothed effective number of neighbors:

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

↗  $\sigma_i$  found by binary search to match a user-defined perplexity for  $p_i$



# tSNE and Student / Cauchy kernels (Maaten and Hinton 2008)

## Similarities in the low dimension space

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

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

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

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

## Optimization

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

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

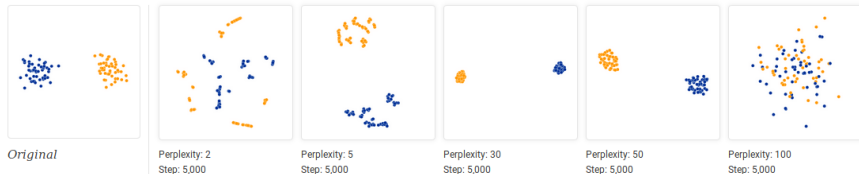
**Initiatization** – reduce original data with PCA then initialized by  $\mathcal{N}(0, \epsilon I_q)$



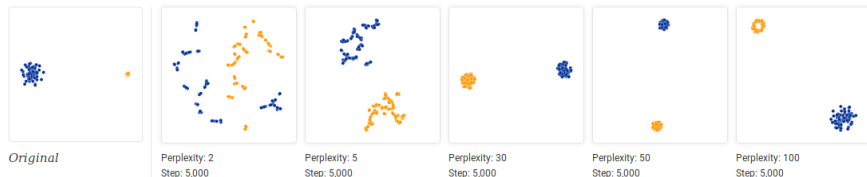


# Empirical properties of tSNE (1)

## Effect of Hyperparameters : Perplexity

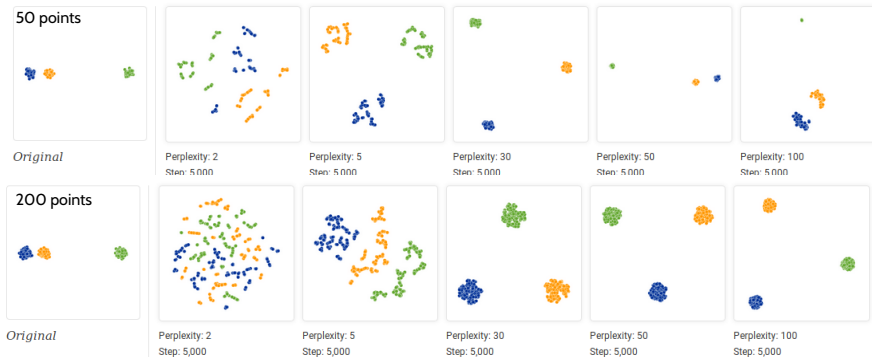


## tSNE does not account for heteroscedasticity

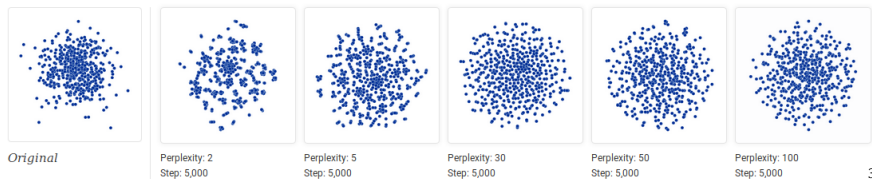


# Empirical properties of tSNE (2)

## tSNE does not account for between-cluster distance



## What about random noise ?

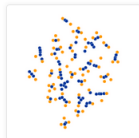


# Empirical properties of tSNE (3)

## Catching Complex Geometries



*Original*



Perplexity: 2  
Step: 5,000



Perplexity: 5  
Step: 5,000



Perplexity: 30  
Step: 5,000



Perplexity: 50  
Step: 5,000



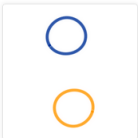
Perplexity: 100  
Step: 5,000



*Original*



Perplexity: 2  
Step: 5,000



Perplexity: 5  
Step: 5,000



Perplexity: 30  
Step: 5,000



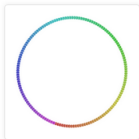
Perplexity: 50  
Step: 5,000



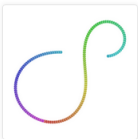
Perplexity: 100  
Step: 5,000



*Original*



Perplexity: 2  
Step: 5,000



Perplexity: 5  
Step: 5,000



Perplexity: 30  
Step: 5,000



Perplexity: 50  
Step: 5,000



Perplexity: 100  
Step: 5,000

# tSNE and UMAP scRNA data<sup>5</sup>

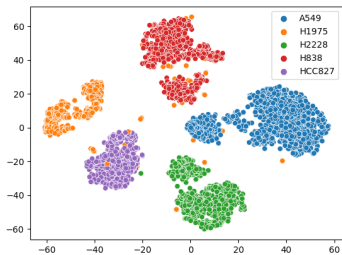


Figure 10: tSNE on log-transformed data

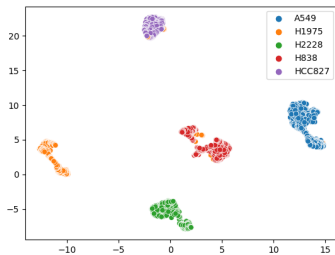


Figure 11: UMAP on log-transformed data

---

<sup>5</sup>using Python modules sklearn, umap-larn

# Probabilistic Neighborhood Embedding (Van Assel et al. 2022)

Unifying probabilistic view of t-SNE like approaches

Hidden graph to structure observations

Let  $\mathbf{W}$  describe a hidden random graph with Laplacian  $\mathbf{L} = \mathbf{D} - \mathbf{W}$

Conditional distribution structured by graphs

Consider matrix Normal models with **row** and column dependencies

$$\mathbf{X} \mid \mathbf{W}_X \sim \mathcal{MN}\left(0, \mathbf{L}_X^{-1}, \boldsymbol{\Sigma}^{-1}\right), \quad \mathbf{Z} \mid \mathbf{W}_Z \sim \mathcal{MN}\left(0, \mathbf{L}_Z^{-1}, \mathbf{I}_q\right).$$

The conditional densities relates to the Gaussian kernel:

$$k(\mathbf{X}_i - \mathbf{X}_j) = \exp\left(-\frac{1}{2}\|\mathbf{X}_i - \mathbf{X}_j\|_{\boldsymbol{\Sigma}^{-1}}^2\right), \quad k(\mathbf{Z}_i - \mathbf{Z}_j) = \exp\left(-\frac{1}{2}\|\mathbf{Z}_i - \mathbf{Z}_j\|^2\right).$$

which can be generalized to translation invariant kernels:

$$\mathbb{P}(\mathbf{X} \mid \mathbf{W}_X) \propto \prod_{(i,j)} k(\mathbf{X}_i - \mathbf{X}_j)^{W_{X,ij}}, \quad \mathbb{P}(\mathbf{Z} \mid \mathbf{W}_Z) \propto \prod_{(i,j)} k(\mathbf{Z}_i - \mathbf{Z}_j)^{W_{Z,ij}}.$$

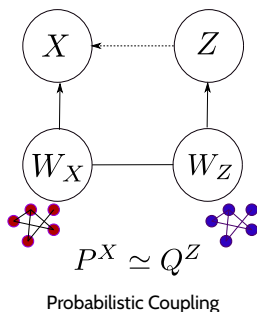


# Embedding with Graph Coupling

Couple the hidden graphs  $W_X$  and  $W_Z$  in by matching their posterior distributions:

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

$\rightsquigarrow Z$  becomes a parameter to be estimated



Optimize cross-entropy between posteriors

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

Find the best low-dimensional representation such that the 2 graphs match

$$\hat{\mathbf{Z}}(X) = \arg \min_{\mathbf{Z}} \mathcal{H}(\mathbf{P}^X, \mathbf{Q}^Z) = \arg \min_{\mathbf{Z}} D_{\text{KL}}(\mathbf{P}^X, \mathbf{Q}^Z)$$

# Conjugate priors and posteriors for hidden graphs

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

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

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 $W$	$\approx$ Posterior for $W$
$\mathcal{B}$	Bernoulli	$\prod_{ij} \mathbf{1}_{W_{ij} \leq 1}$	$\mathcal{B}\left(\frac{\pi_{ij}}{1+\pi_{ij}}\right)$	$\mathcal{B}\left(\frac{\pi_{ij}k_{ij}}{1+\pi_{ij}k_{ij}}\right)$
$\mathcal{D}$	Unitary Fixed degree	$\prod_i \mathbf{1}_{W_{i+}=1}$	$\mathcal{M}\left(1, \frac{\pi_i}{\pi_{i+}}\right)$	$\mathcal{M}\left(1, \frac{[\pi k]_i}{[\pi k]_{i+}}\right)$
$\mathcal{E}$	Fixed Number of edges	$\prod_{ij} (W_{ij}!)^{-1}$	$\mathcal{M}\left(n, \frac{\pi}{\pi_{++}}\right)$	$\mathcal{M}\left(n, \frac{\pi k}{[\pi k]_{++}}\right)$

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

## Mixing Prior distributions for coupling

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



# 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	$P_{ij}^D = \frac{k_x(X_i - X_j)}{\sum_{\ell} k_x(X_i - X_{\ell})}$	$Q_{ij}^D = \frac{k_z(Z_i - Z_j)}{\sum_{\ell} k_z(Z_i - Z_{\ell})}$	$-\sum_{i \neq j} P_{ij}^D \log Q_{ij}^D$
Sym-SNE	$\bar{P}_{ij}^D = P_{ij}^D + P_{ji}^D$	$Q_{ij}^E = \frac{k_z(Z_i - Z_j)}{\sum_{\ell, t} k_z(Z_{\ell} - Z_t)}$	$-\sum_{i < j} \bar{P}_{ij}^D \log Q_{ij}^E$
LargeVis	$\bar{P}_{ij}^D = P_{ij}^D + P_{ji}^D$	$Q_{ij}^B = \frac{k_z(Z_i - Z_j)}{1 + k_z(Z_i - Z_j)}$	$-\sum_{i < j} \bar{P}_{ij}^D \log Q_{ij}^B + \left(2 - \bar{P}_{ij}^D\right) \log(1 - Q_{ij}^B)$
UMAP	$\tilde{P}_{ij}^B = P_{ij}^B + P_{ji}^B - P_{ij}^B P_{ji}^B$	$Q_{ij}^B = \frac{k_z(Z_i - Z_j)}{1 + k_z(Z_i - Z_j)}$	$-\sum_{i < j} \tilde{P}_{ij}^B \log Q_{ij}^B + \left(1 - \tilde{P}_{ij}^B\right) \log(1 - Q_{ij}^B)$





# Conclusion

**Thank you for your attention**

## Co-authors on this topic

- Poisson log-normal PCA: Stéphane Robin, Mahendra Maridassou, Bastien Batardière, Nicolas Jouvin
- Probabilistic t-SNE: Hugues van Assel, Franck Picard, Thibault Espinasse, Eddie Aamari

## Some code

- R/C++ package PLNmodels is on <https://cran.r-project.org/>
- Python/Pytorch package pyplnmodels is on <https://pypi.org/>
- Github repos of this presentation is available at [https://github.com/jchiquet/dimred\\_intro](https://github.com/jchiquet/dimred_intro)

## Advertissing

<https://computo.sfds.asso.fr/>, an open diamond academic journal promoting reproducibility



# References I

- Balasubramanian, Mukund, and Eric L Schwartz. 2002. “The Isomap Algorithm and Topological Stability.” *Science* 295 (5552): 7–7.
- Belkin, Mikhail, and Partha Niyogi. 2003. “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation.” *Neural Computation* 15 (6): 1373–96.
- Blei, David M, Alp Kucukelbir, and Jon D McAuliffe. 2017. “Variational Inference: A Review for Statisticians.” *Journal of the American Statistical Association* 112 (518): 859–77.
- Chiquet, Julien, Mahendra Mariadassou, and Stéphane Robin. 2018. “Variational Inference for Probabilistic Poisson PCA.” *The Annals of Applied Statistics* 12: 2674–98. <http://dx.doi.org/10.1214/18-AOAS1177>.
- . 2021. “The Poisson-Lognormal Model as a Versatile Framework for the Joint Analysis of Species Abundances.” *Frontiers in Ecology and Evolution* 9. <https://doi.org/10.3389/fevo.2021.588292>.



## References II

- Dony, Leander, Martin König, D Fischer, and Fabian J Theis. 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.
- Grønbech, Christopher Heje, Maximillian Fornitz Vording, Pascal N Timshel, Casper Kaae Sønderby, Tune H Pers, and Ole Winther. 2020. “scVAE: Variational Auto-Encoders for Single-Cell Gene Expression Data.” *Bioinformatics* 36 (16): 4415–22.
- Hinton, Geoffrey E, and Sam Roweis. 2002. “Stochastic Neighbor Embedding.” *Advances in Neural Information Processing Systems* 15.
- Kingma, Diederik P, and Max Welling. 2013. “Auto-Encoding Variational Bayes.” *arXiv Preprint arXiv:1312.6114*.
- Maaten, Laurens van der, and Geoffrey Hinton. 2008. “Visualizing Data Using t-SNE.” *Journal of Machine Learning Research* 9: 2579–2605.



# References III

- Seninge, Lucas, Ioannis Anastopoulos, Hongxu Ding, and Joshua Stuart. 2021. “VEGA Is an Interpretable Generative Model for Inferring Biological Network Activity in Single-Cell Transcriptomics.” *Nature Communications* 12 (1): 1–9.
- Tipping, M. E, and C. M Bishop. 1999. “Probabilistic Principal Component Analysis.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61 (3): 611–22.
- Van Assel, Hugues, Thibault Espinasse, Julien Chiquet, and Franck Picard. 2022. “A Probabilistic Graph Coupling View of Dimension Reduction.” In *Advances in Neural Information Processing Systems*.
- Wang, Dongfang, and Jin Gu. 2018. “VASC: Dimension Reduction and Visualization of Single-Cell RNA-Seq Data by Deep Variational Autoencoder.” *Genomics, Proteomics & Bioinformatics* 16 (5): 320–31.

