Tutorial on Dimensionality Reduction

Some recents approches in statistics and machine learning

Julien Chiquet

UMR MIA Paris-Saclay, AgroParisTech, INRAE

May 25, 2023

https://jchiquet.github.io/

MIA
PARIS-SACLAY

INRAE
la science pour la vie, l'humain, la terre

Outline

Exploratory analysis of (modern) data sets

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

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

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

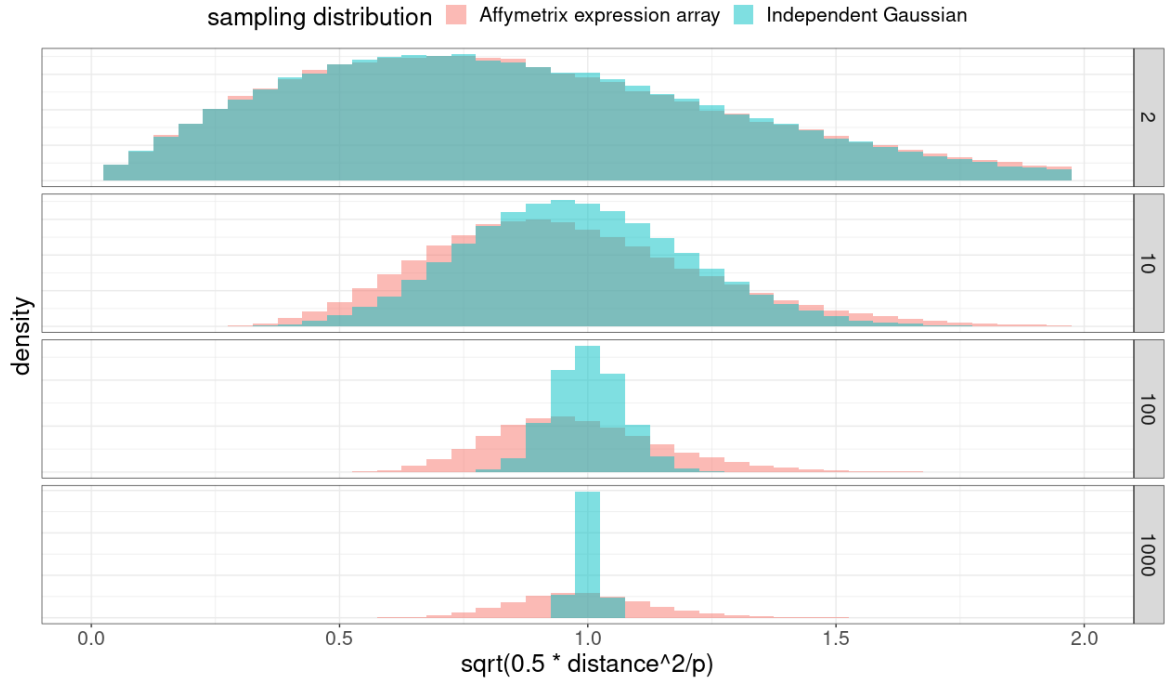
Dimensionality curse

Theorem (Folks theorem)

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

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

↝ *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 $> 3d$!)

See Chapter 20 in Murphy (2022) for a nice, recent introduction and Chapter 14 in Hastie et al. (2009) for reference.

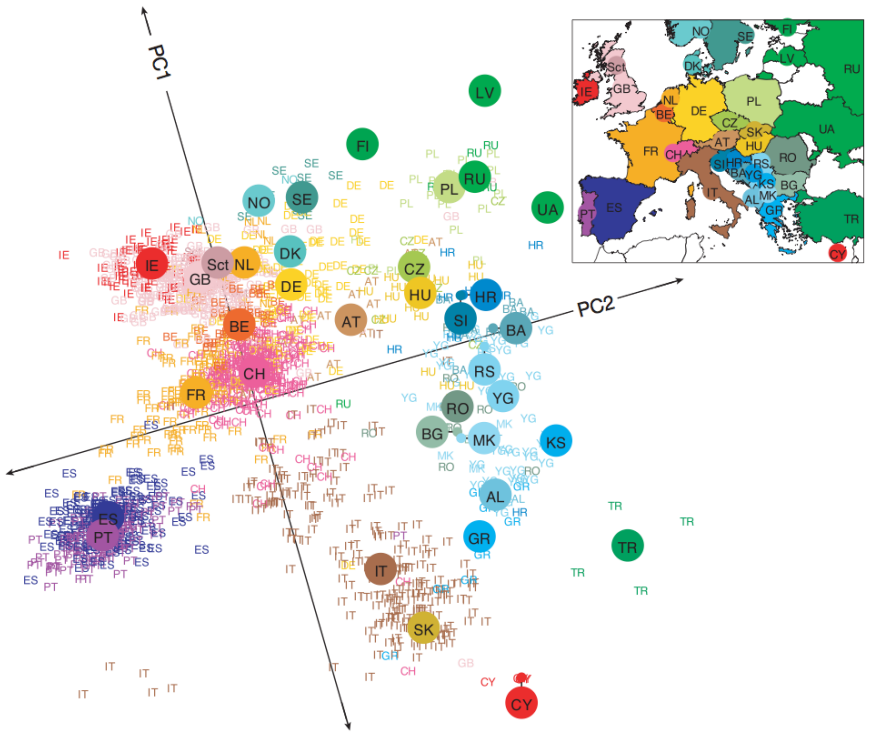
Example in genetics

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 (# allels different against pop. reference)

Summarized with 2 features[1]

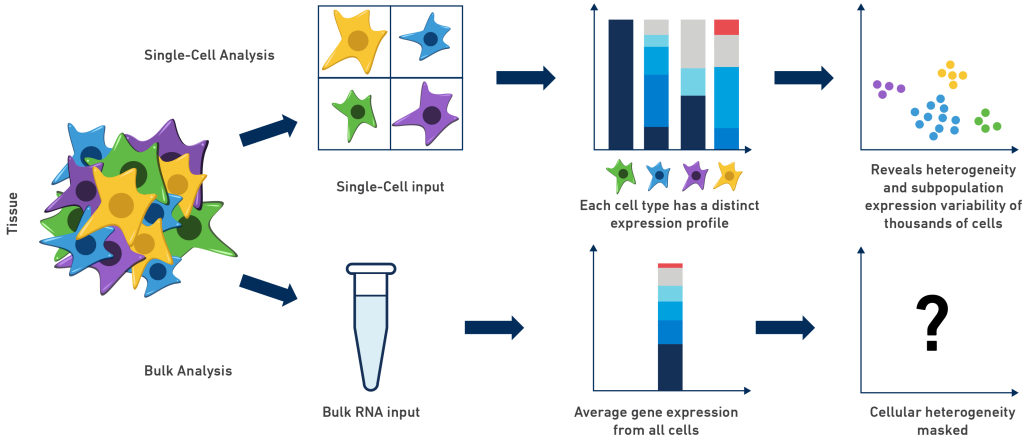


- an extremely strong structure between individuals (”**clustering**”)
- a very simple subspace where it is obvious (”**dimension reduction**”)

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

Example in genomics

Genome-wide cell biology with single-cell RNAseq data



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

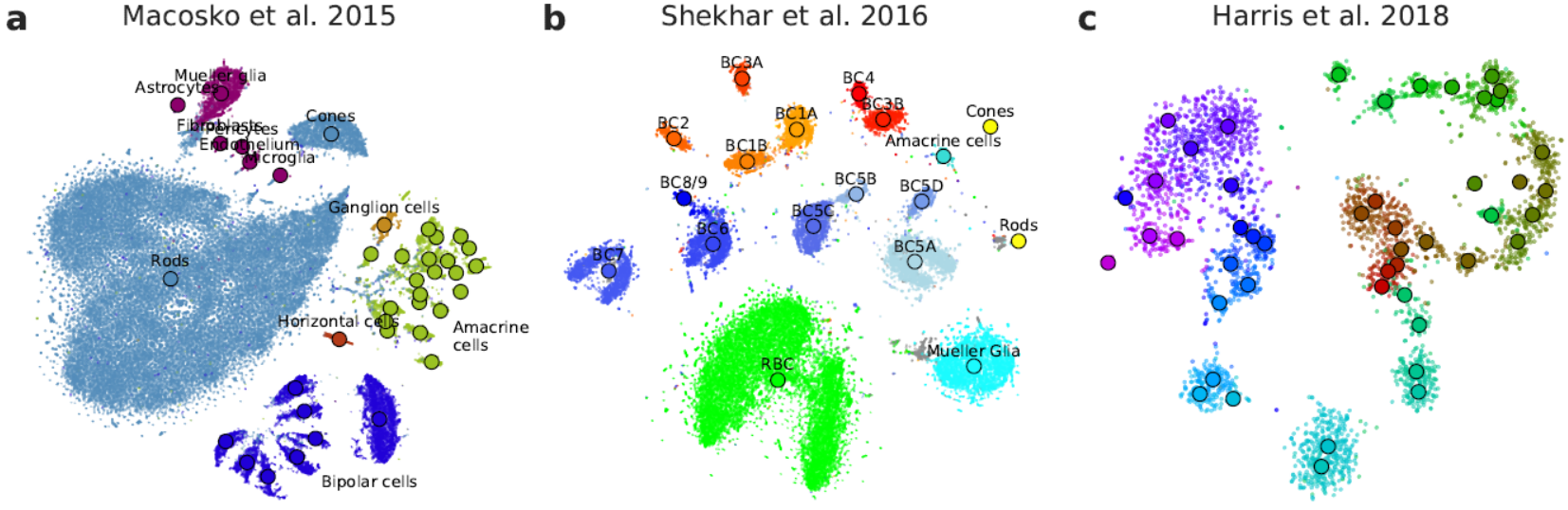


Figure 1: Successful t-SNE visualizations of sc-RNAseq data

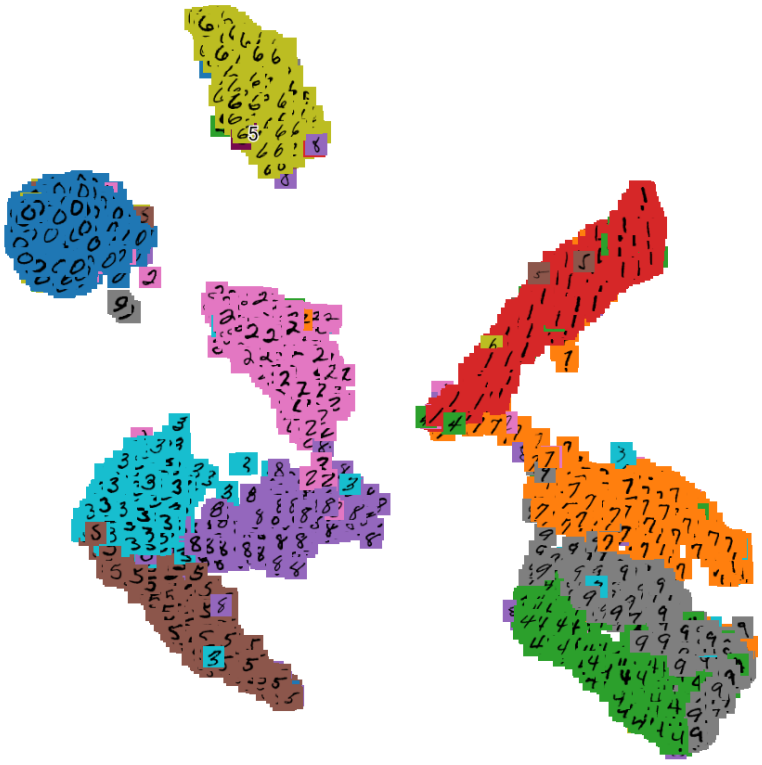
Example in Image: MNIST

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

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

Obtained via
https://projector.tensorflow.org/, **try it!**

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 $q \ll p$:

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

↠ 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

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

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

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

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

[a] empirical mean, barycentrum, center of inertia

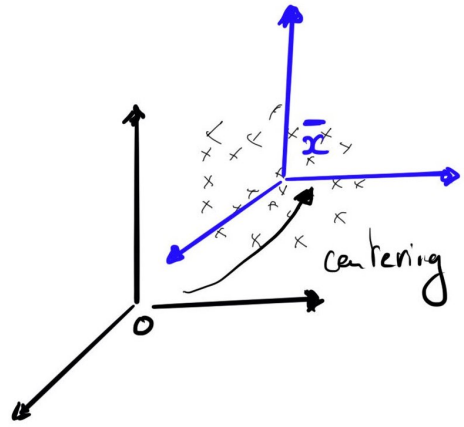


Figure 3: Example in $\mathbb{R}^3$

Total Inertia I_T as a measure of information

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

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

⇝ 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 (Hotelling 1933)

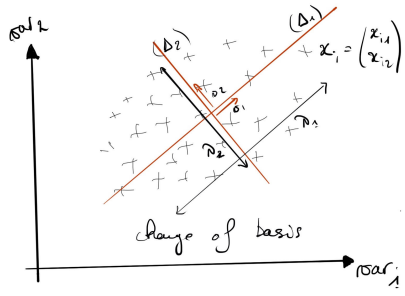
$$\hat{\mathbf{\Sigma}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top, \quad \mathbf{V} = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2, & \dots & \mathbf{v}_p \end{pmatrix}, \quad \mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$$

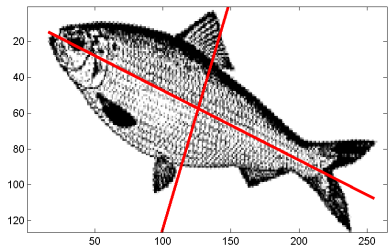
$\mathbf{V}$ are known as the **loadings**

Interpretation in $\mathbb{R}^p$

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

$$\begin{cases} \Delta_1 & \oplus & \dots & \oplus & \Delta_p \\ \mathbf{v}_1 & \perp & \dots & \perp & \mathbf{v}_p \\ \lambda_1 & > & \dots & > & \lambda_p \\ I_{\Delta_1^\perp} & > & \dots & > & I_{\Delta_p^\perp} \end{cases}$$





⇝ 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 orginal variables: the **principal components** (PC)

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

The matrix of PC connects individual coordinates to latent factors:

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

⇝ 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 $\mathbf{V}$):

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

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 $q \ll p$.

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

⇝ 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

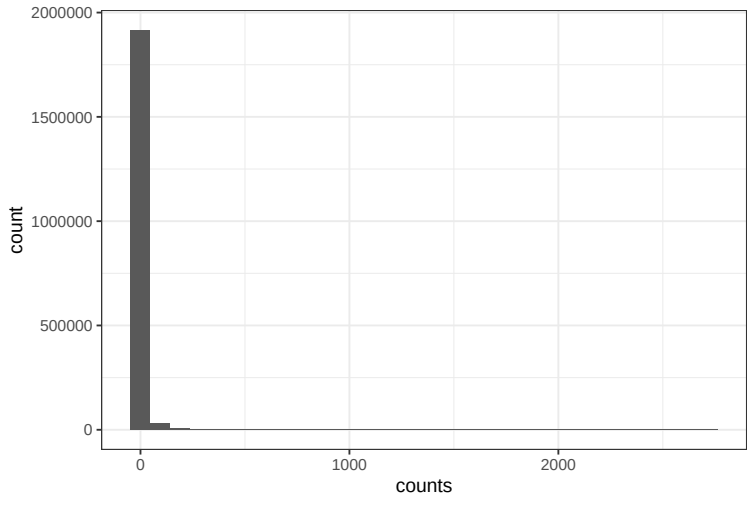


Figure 4: raw counts

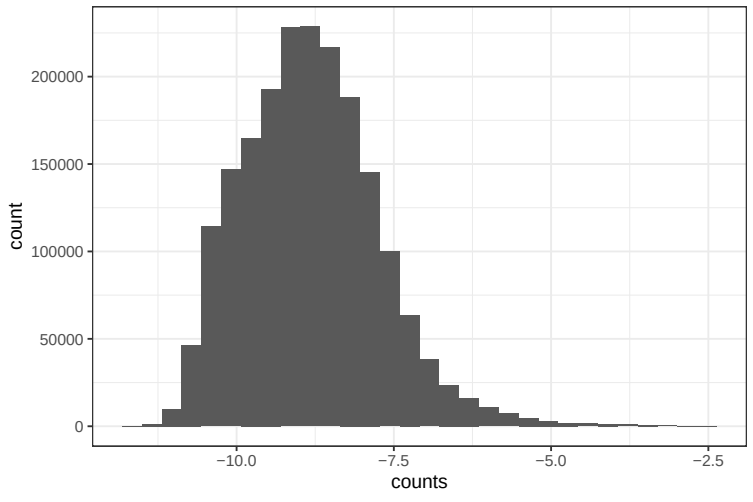
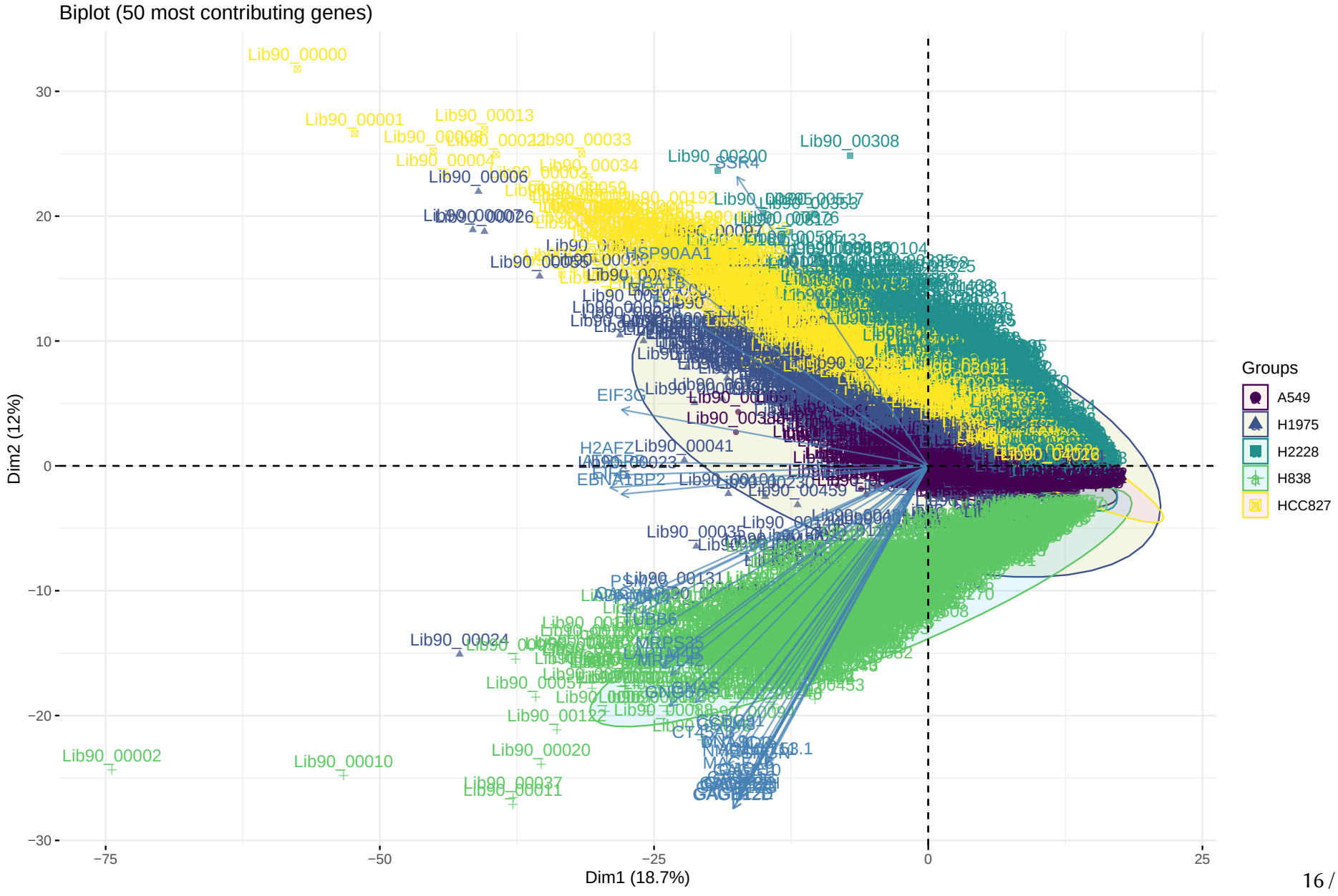


Figure 5: log/total-counts normalization

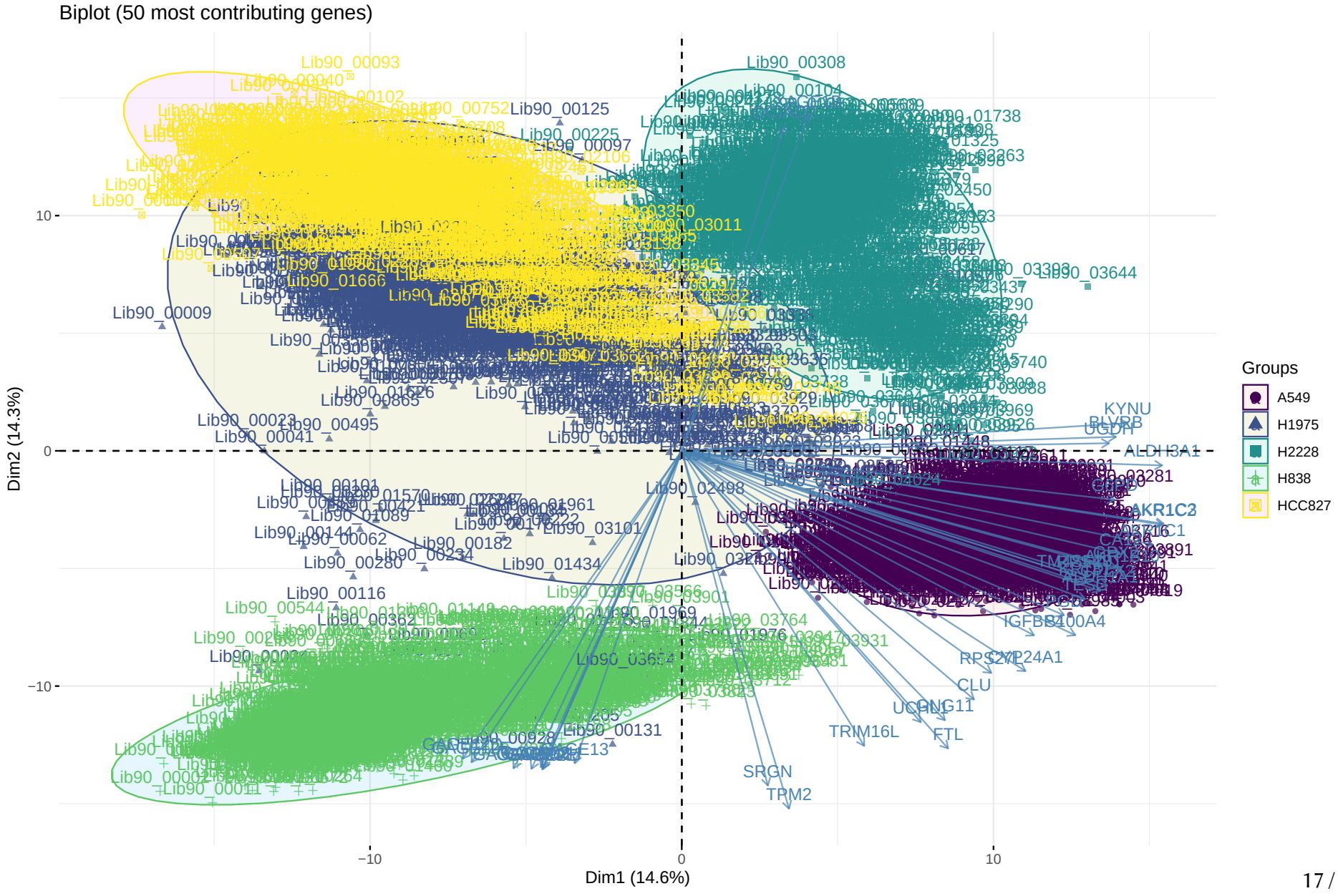
Single-Cell data analysed with PCA

Raw data



Single-Cell data analysis with PCA

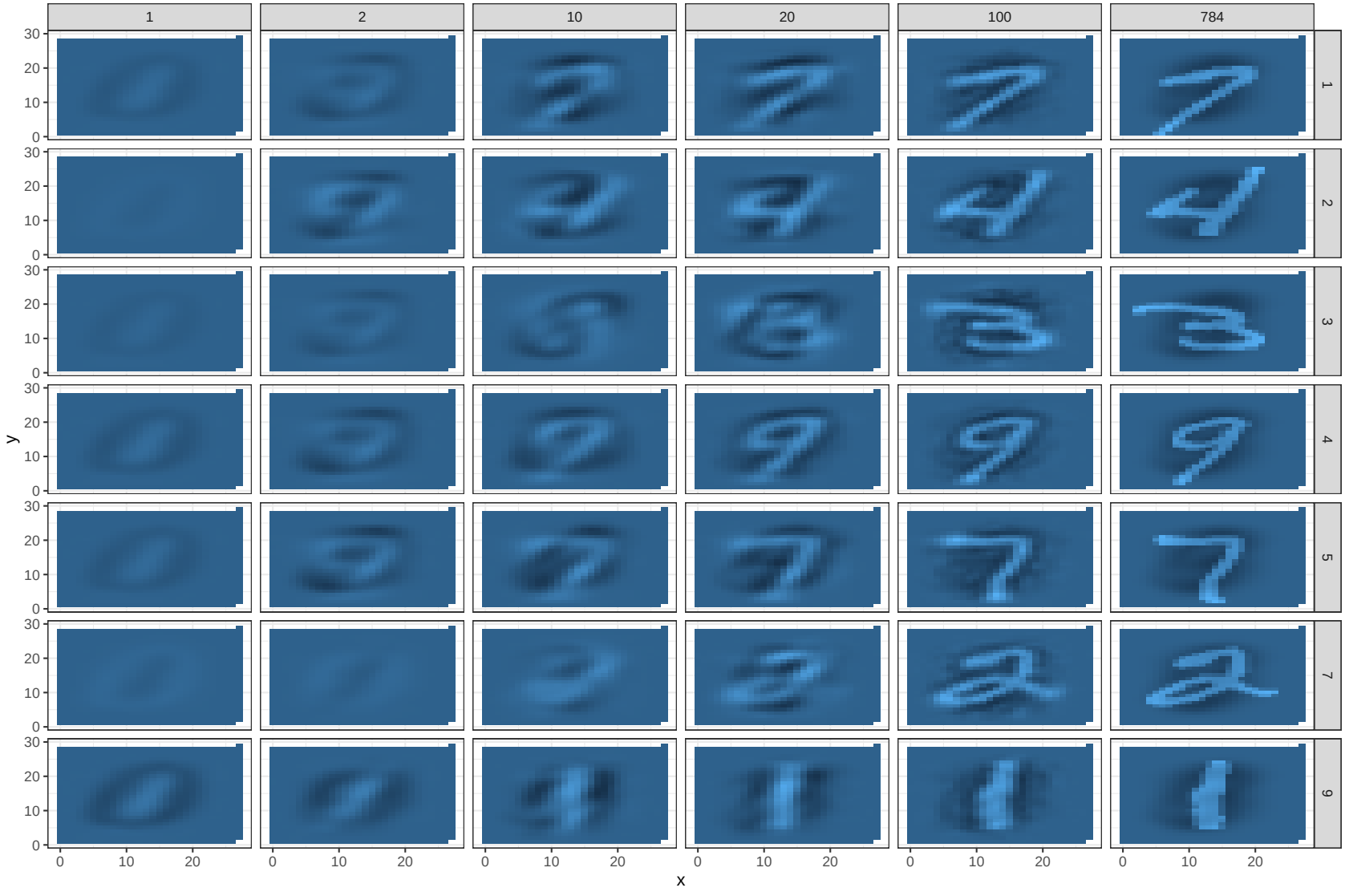
Normalized data



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 $\Phi : \mathbb{R}^p \to \mathbb{R}^q$

Challenges

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

Outline

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:

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

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

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

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

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

Another interpretation 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

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

Solution (explicit)

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

Non-negative Matrix Factorization (Sra and Dhillon 2005)

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:

$$\hat{\mathbf{X}}^{\text{ls}} = \underset{\substack{\mathbf{F} \in \mathscr{M}(\mathbb{R}_+)_{n,q} \\ \mathbf{V} \in \mathscr{M}(\mathbb{R}_+)_{p,q}}}{\arg\min} \left\| \mathbf{X} - \mathbf{F}\mathbf{V}^\top \right\|_F^2,$$

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

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

Kernel-PCA (Schölkopf, Smola, and Müller 1998)

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

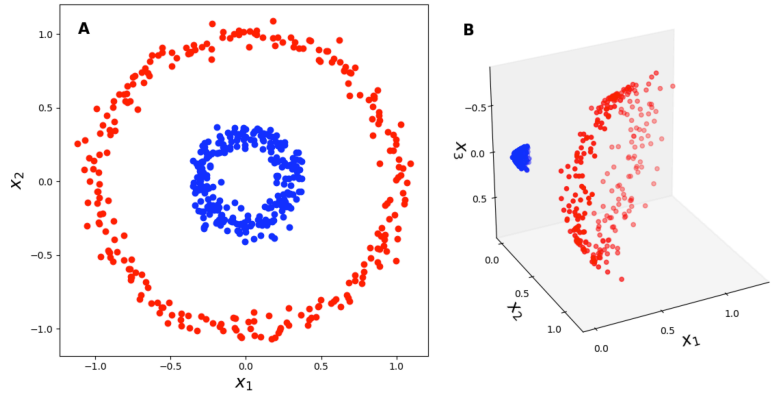


Figure 6: Transformation $\Psi : \mathbf{x} \to \Psi(\mathbf{x})$ (illustration in presence of existing labels)

Model

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

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

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

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

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

Other methods

Linear models with other constraints

Let $\mathbf{V}_q$ be a $p \times q$ matrix and $\mathbf{z} \in \mathbb{R}^q$

$$\hat{\mathbf{x}} = \tilde{\Phi}(\mathbf{z}) = \boldsymbol{\mu} + \sum_{j=1}^{q} \tilde{z}^j \mathbf{V}^j = \boldsymbol{\mu} + \mathbf{V}_q \mathbf{z}$$

Apply other constraints on $\mathbf{V}$ and or the factor/representation $\mathbf{z}$

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

↝ optimize square-loss $\left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F^2$ to fit $\mu, \mathbf{V}, \mathbf{z}$

MNIST: original

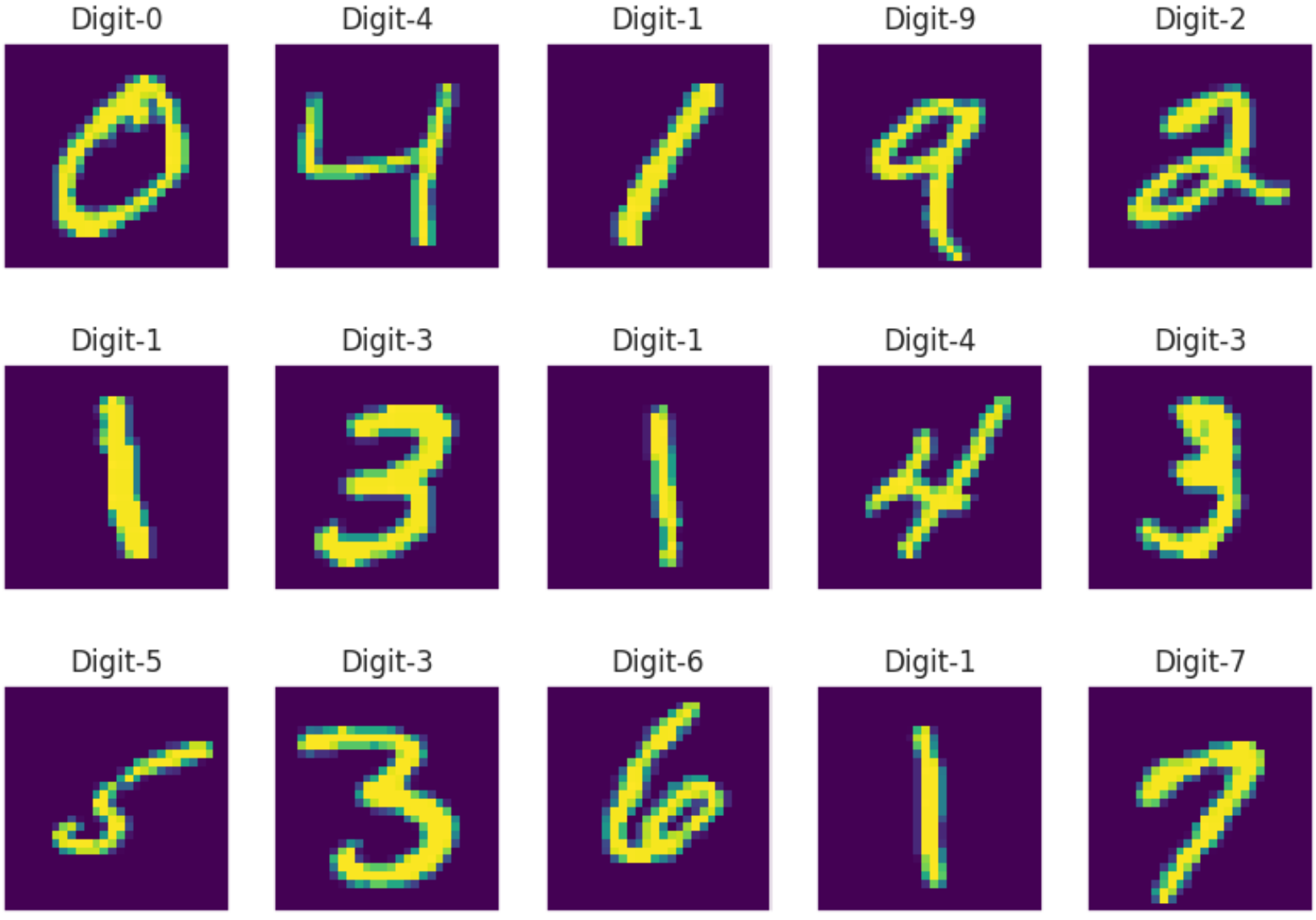


Figure 7: Original data: Subsample of 2,000 labeled handwritten digits

MNIST: PCA compression

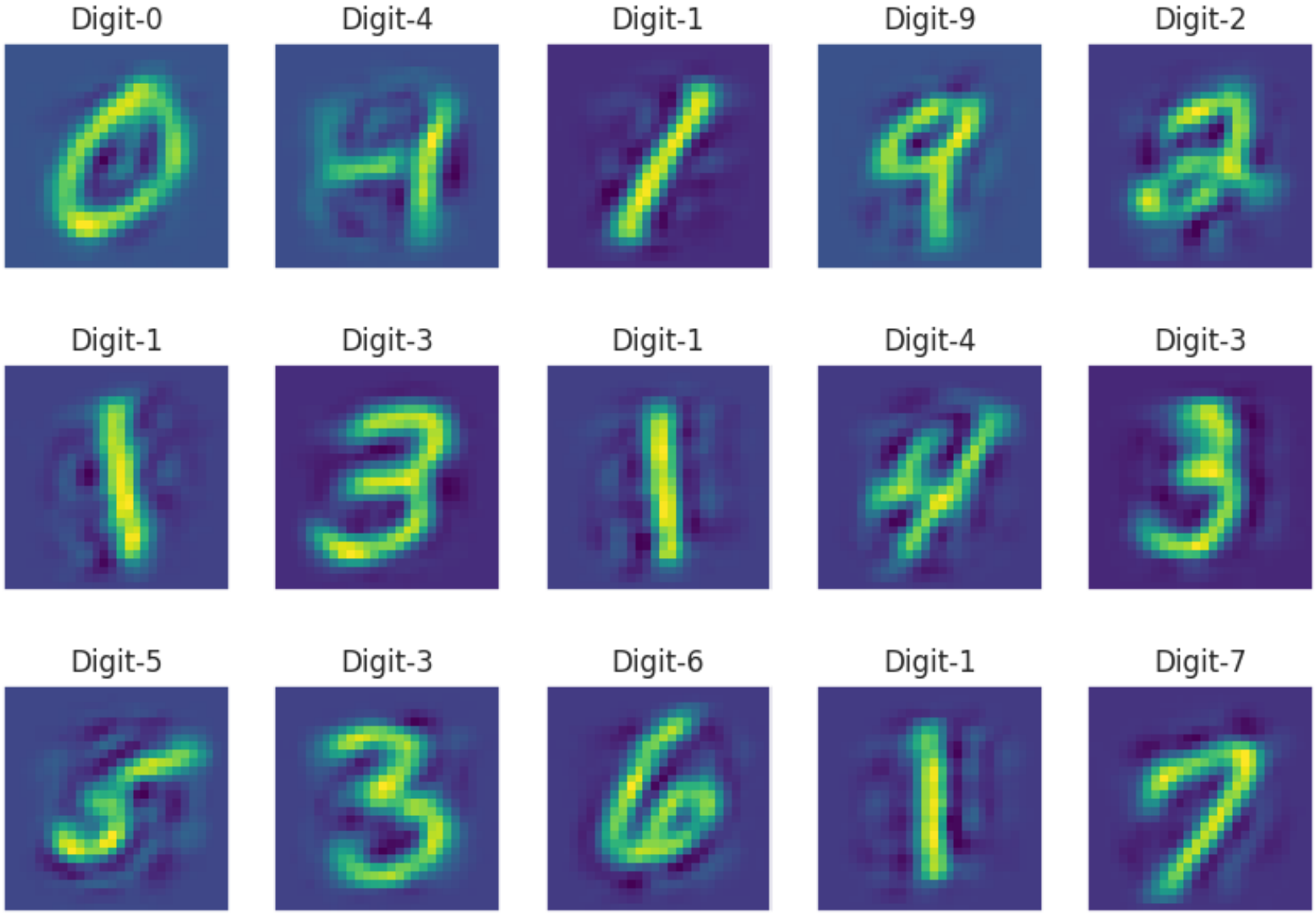


Figure 8: PCA with 40 components

MNIST: NMF compression

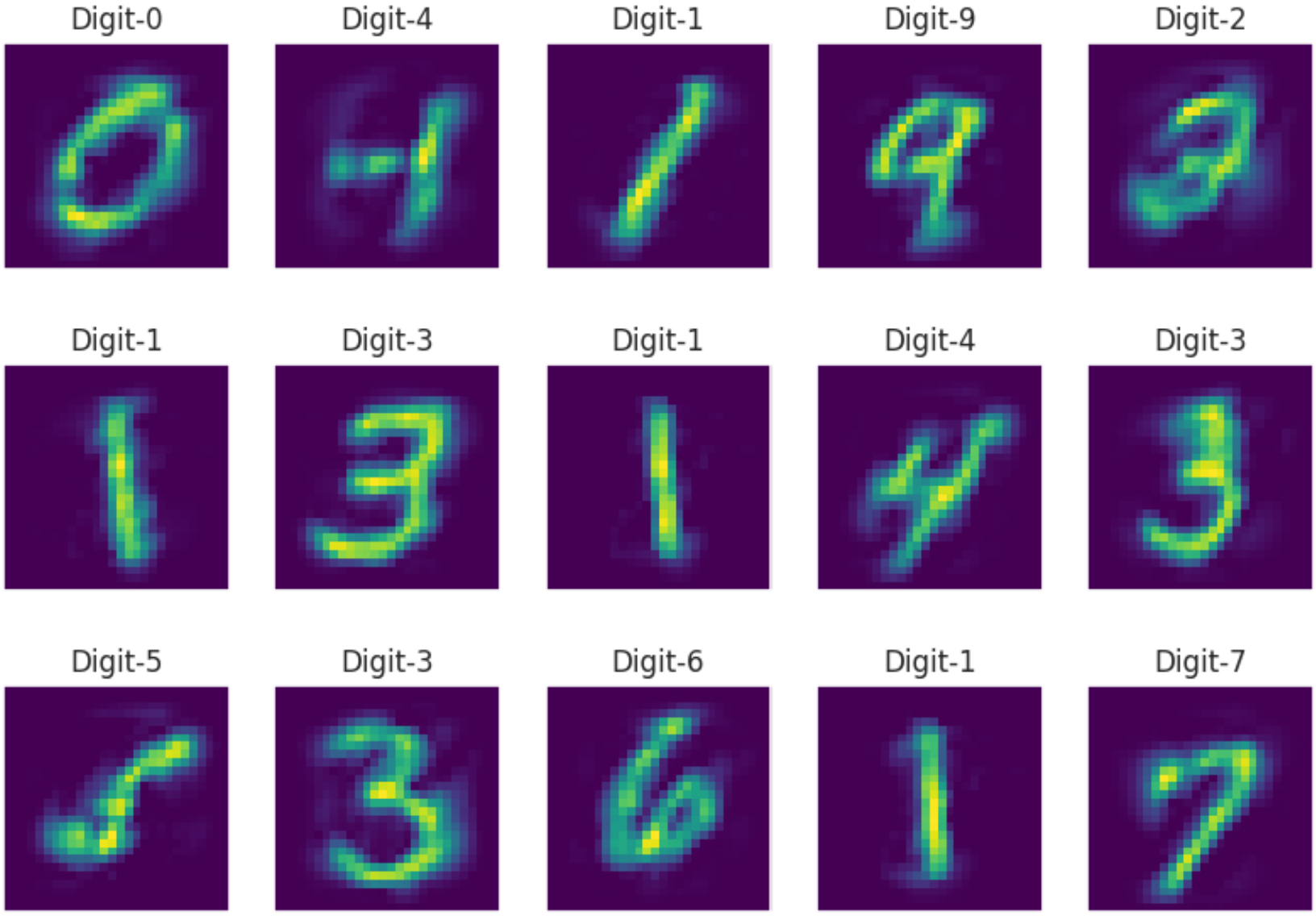


Figure 9: NMF with 40 components

MNIST: kernel-PCA compression

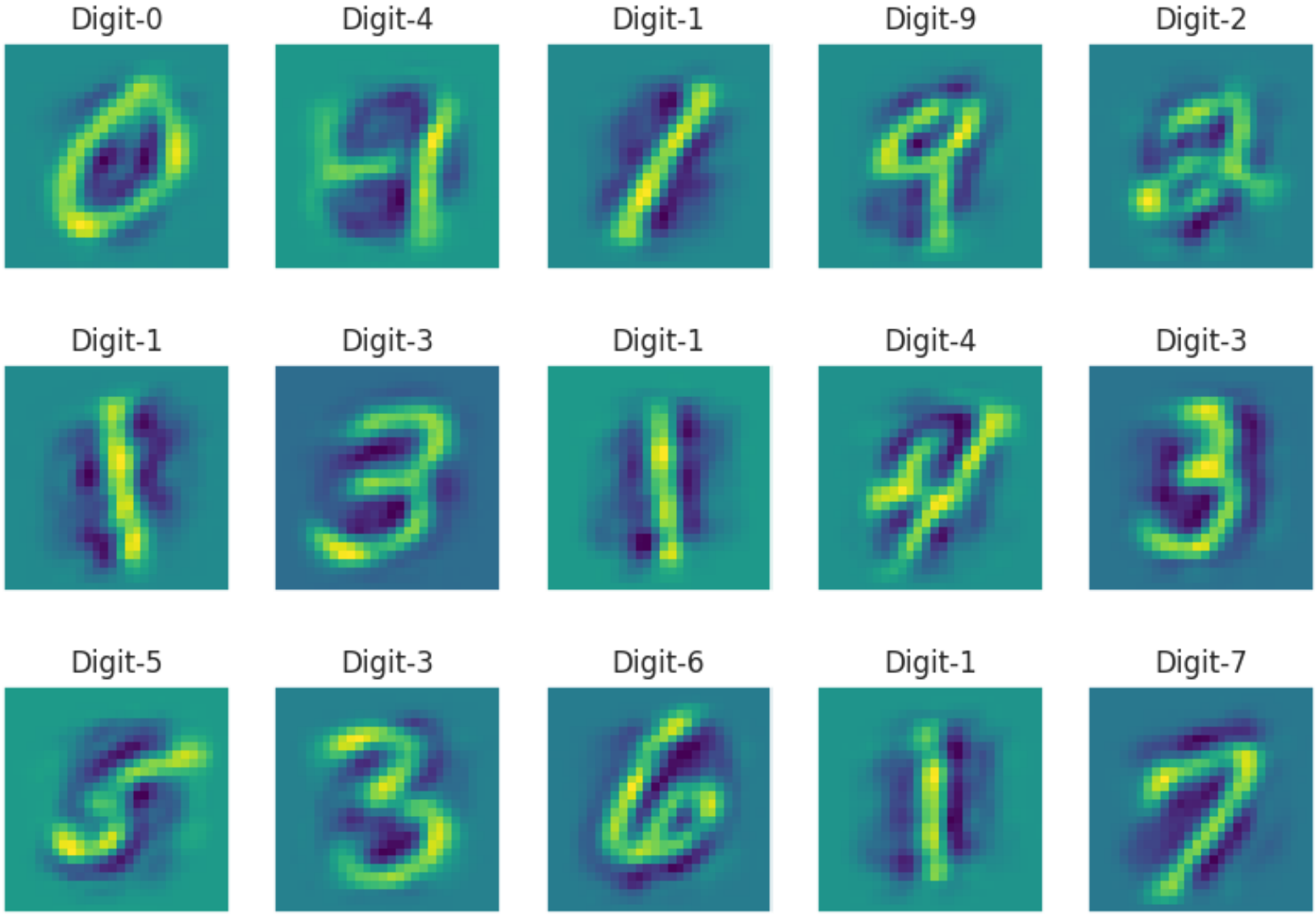


Figure 10: Kernel-PCA with linear kernel and 40 components

MNIST: ICA compression

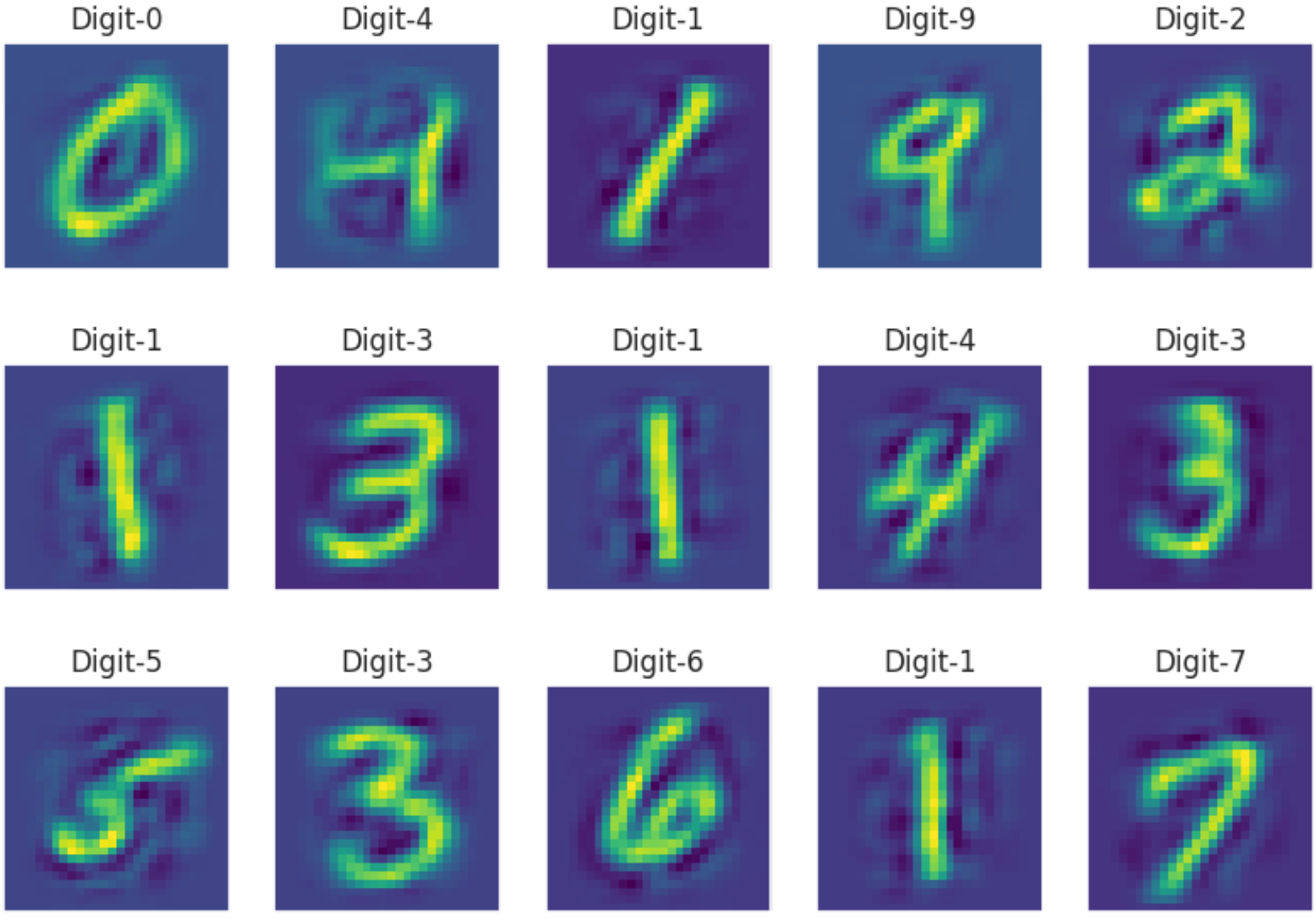


Figure 11: ICA

MNIST: dictionnary learning

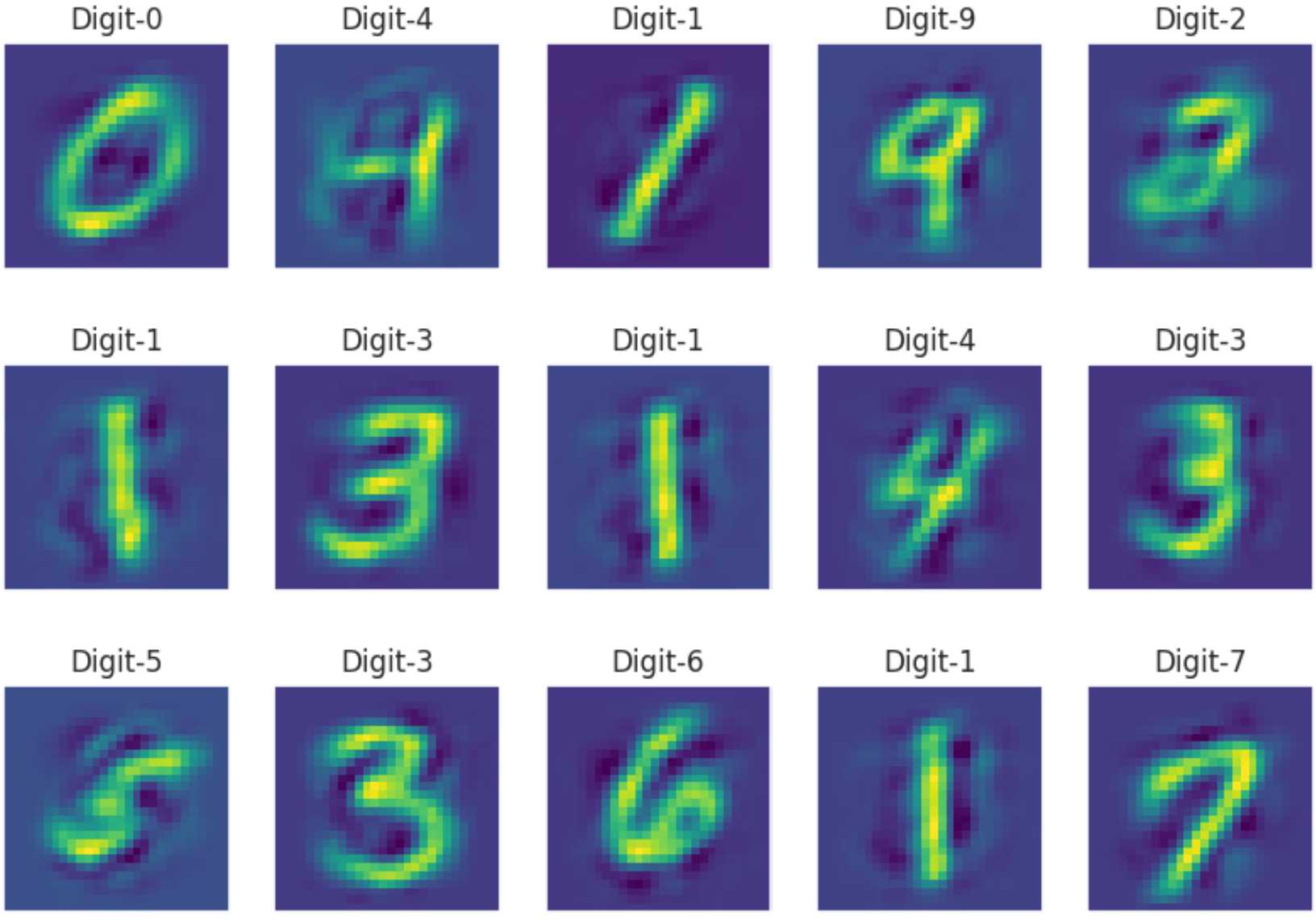


Figure 12: Dictionnary Learning with 25 components

Code using sklearn

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA, NMF, KernelPCA, FastICA, MiniBatchDictionaryLearnin

mnist = pd.read_csv('data/mnist_sample.csv')
labels = mnist.iloc[:,0]
digits = mnist.iloc[:,1:mnist.shape[1]]

## PCA
pca = PCA(n_components=40, random_state=0, whiten=True)
pca.fit(digits)
digits_PCA = pca.inverse_transform(pca.transform(digits))

## NMF
nmf = NMF(n_components=40, random_state=0)
nmf.fit(digits)
digits_NMF = nmf.inverse_transform(nmf.transform(digits))

## Kernel-PCA
kpca = KernelPCA(n_components=40, kernel='linear', random_state=0, fit_inverse_transform=T
kpca.fit(digits)
digits_kPCA = kpca.inverse_transform(kpca.transform(digits))

# [...]
```

Outline

Probabilistic Gaussian PCA (Tipping and Bishop 1999)

Generative model

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

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

By direct integration[2], the marginal distribution of the observation is

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

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

[2] easy since everything is Gaussian

Estimation

Criterion: negative log-likelihood

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

Maximum likelihood estimator

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

Latent position: posterior distribution

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

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

Estimation: alternative

Expectation-Maximization

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

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

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

↝ can be faster that MLE when $p \gg q$

PCA for counts: poisson lognormal PCA

Generative Model (Chiquet, Mariadassou, and Robin 2018)

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

Estimation: Issues

- The marginal distribution is hard to compute, even numerically

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

⇝ 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_ψ of $p_\theta(\mathbf{Z} \,|\, \mathbf{X})$ minimizing a divergence in a class $\mathcal{Q}$

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

- maximize the ELBO (Evidence Lower BOund)

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

Variational EM for Poisson-lognormal

Consider $\mathcal{Q}$ the class of diagonal multivariate Gaussian distributions.
The ELBO $J(\theta, \psi)$ hat 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 for PLN-PCA

Selection of number of components (rank k)

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

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

Visualization: non-nested subspaces (≠ Gaussian PCA)

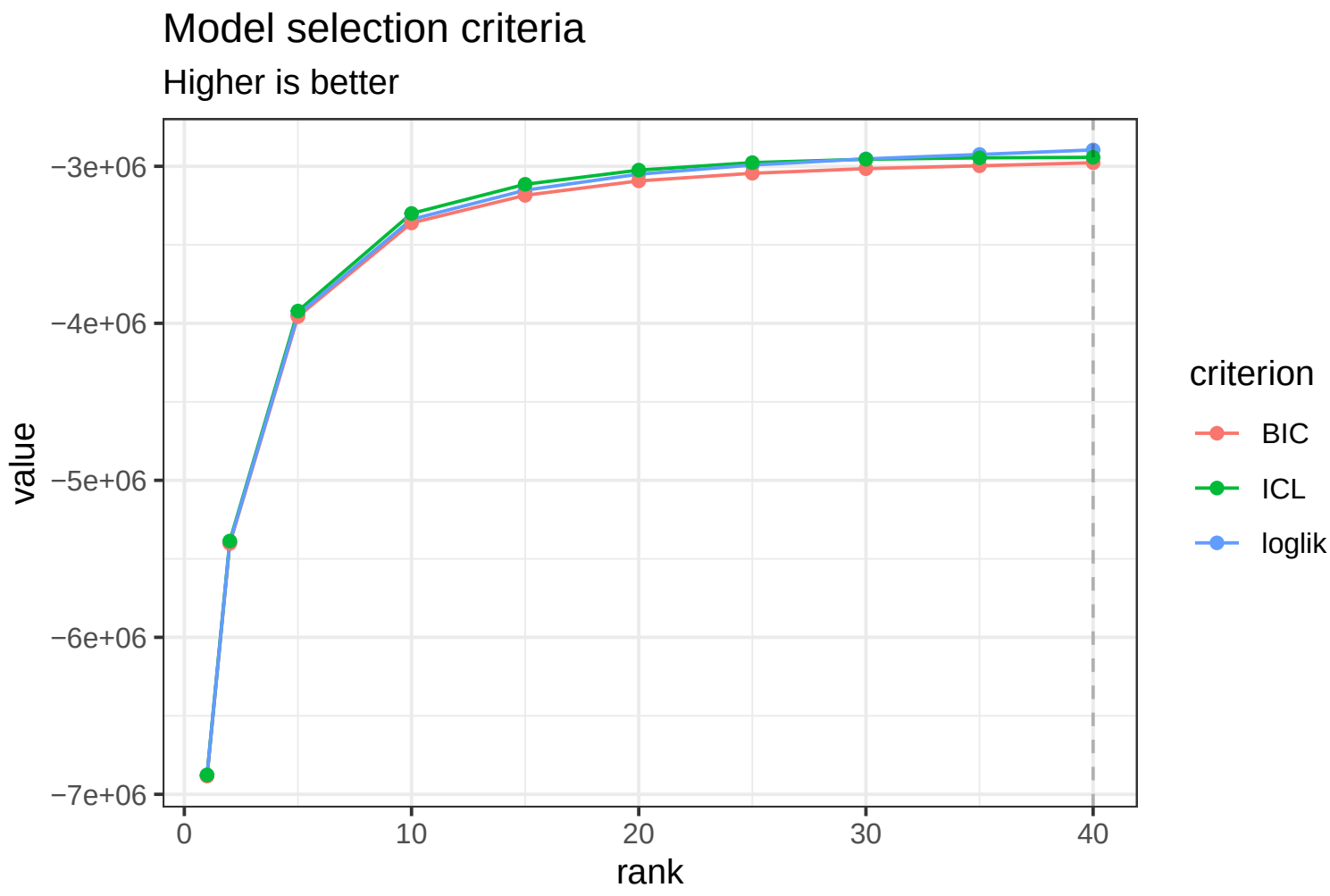
For the selected dimension $\hat{k}$, compute the estimated latent positions $\mathbb{E}_q(\mathbf{Z}_i)$ and perform PCA

Goodness of fit: deviance based criterion

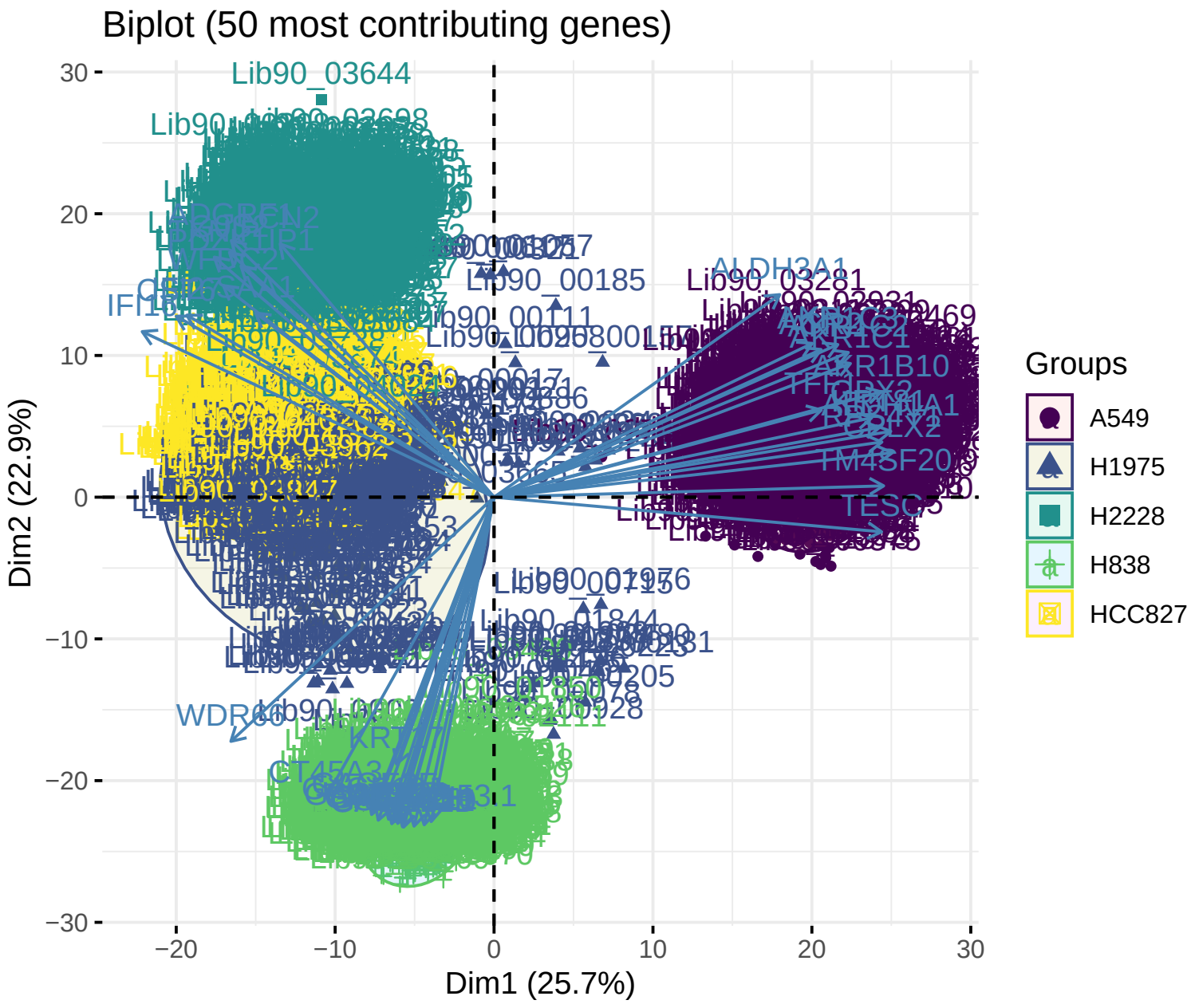
For $\ell_k = \log \mathbb{P}(\mathbf{X}; \lambda^{(k)})$ the Poisson likelihood,

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

Poisson-lognormal PCA for the scRNA data set



Poisson-lognormal PCA for the scRNA data set



Variational Auto-Encoders (Kingma and Welling 2013)

Highly non-linear model

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

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

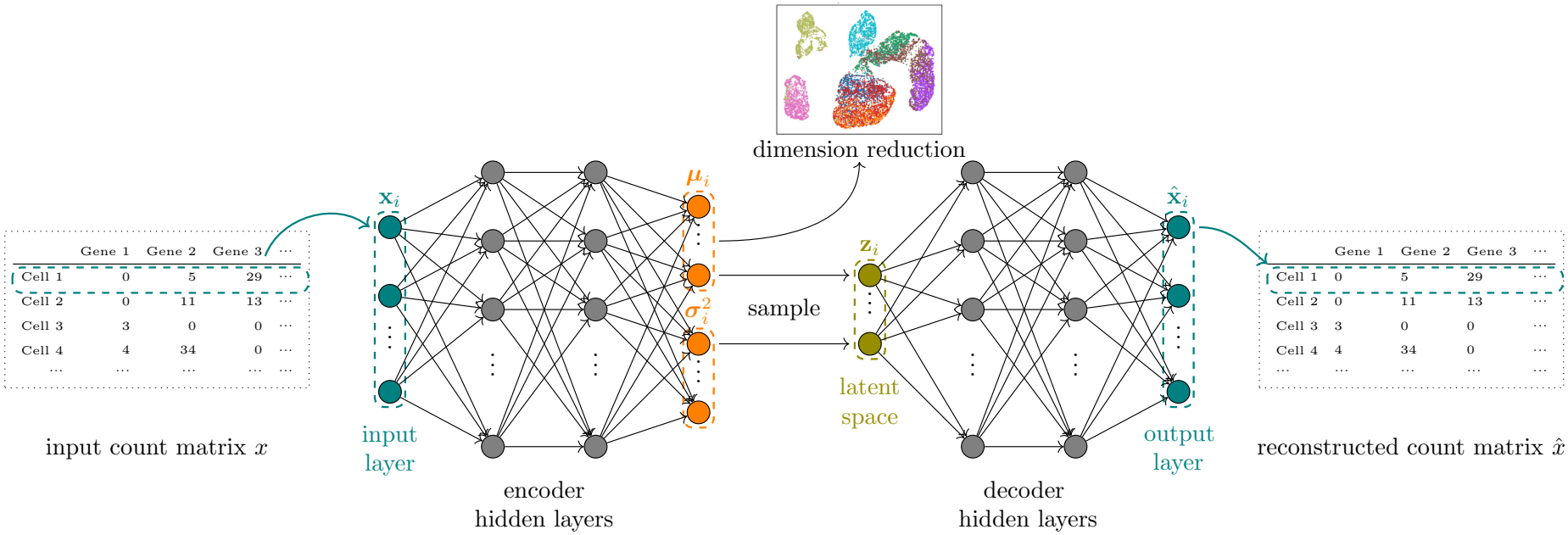


Figure 13: Figure by Hugo Gangloff

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_ψ, $\psi = \{\boldsymbol{\mu}_i, \boldsymbol{\sigma}^2\}$:

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

Optimization/training

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

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

Variational Auto-Encoders

Likelihoods relevent 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,

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

Promising works and questions

- 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 approches

Variational Auto-Encoders on scRNA data[3]

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

Negative-Binomial distribution

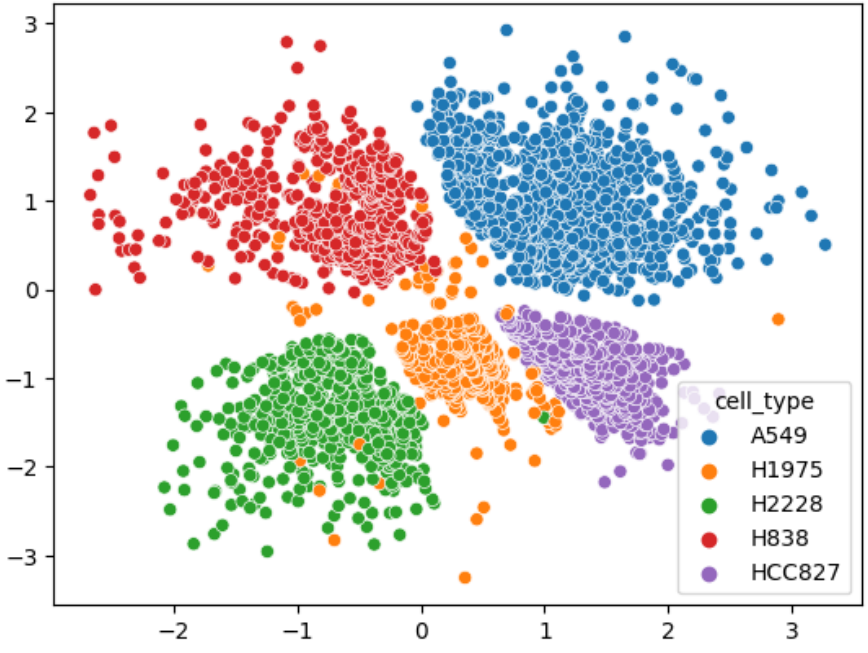


Figure 14: Negative Binomial

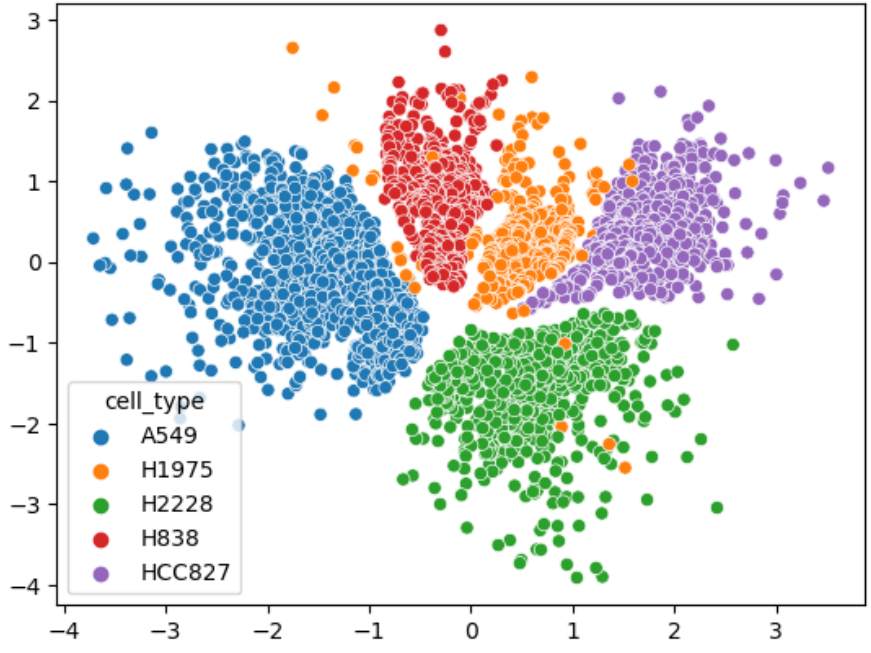


Figure 15: Zero-Inflated Negative Binomial

[3] based on code by Hugo Gangloff

Variational Auto-Encoders on scRNA data[4]

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

Poisson distribution

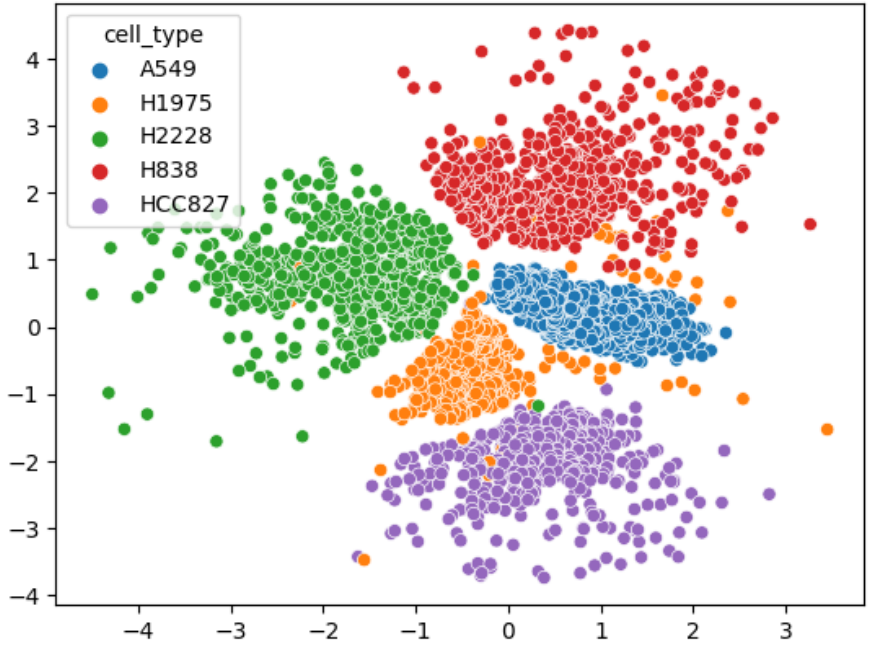


Figure 16: Poisson

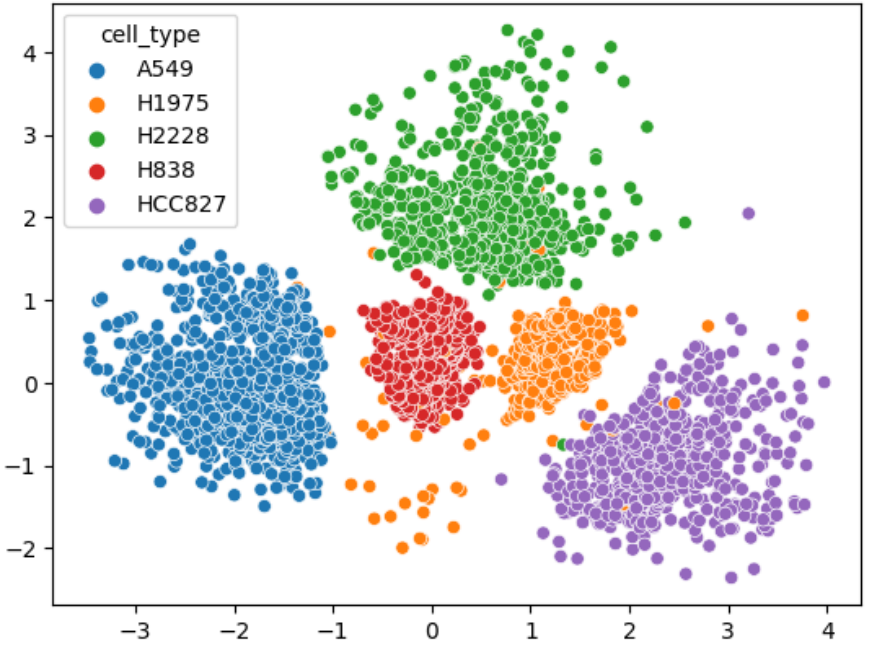


Figure 17: Zero-Inflated Poisson

[4] based on code by Hugo Gangloff

Outline

Pairwise Relation

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

Distance Preservation

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

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

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

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:

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

- 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

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

Isomap

Basic idea

- Metric MDS performs embedding based on pairwise Euclidean-based distance
- Isomap embeds a distance induced by a neighborhood graph

Formally, consider a neighborhood $\mathcal{N}_i$ for each point, then

$$d_{ii'} = \begin{cases} +\infty & \text{if } j \notin \mathcal{N}_i \\ \|\mathbf{x}_i - \mathbf{x}_{i'}\| & \end{cases},$$

and compute the shortest path distance for each pair prior to MDS.

Laplacian Eigenmaps

TODO

Classical embeddings on scRNA data set[5]

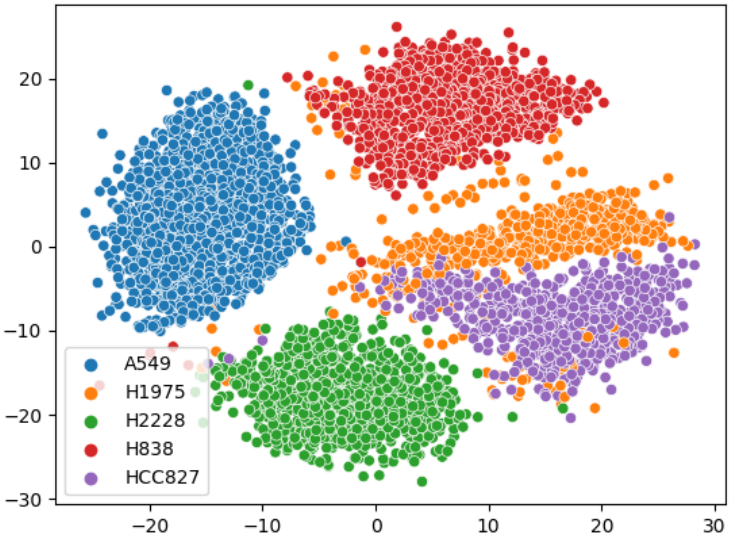


Figure 18: Multidimensional Scaling

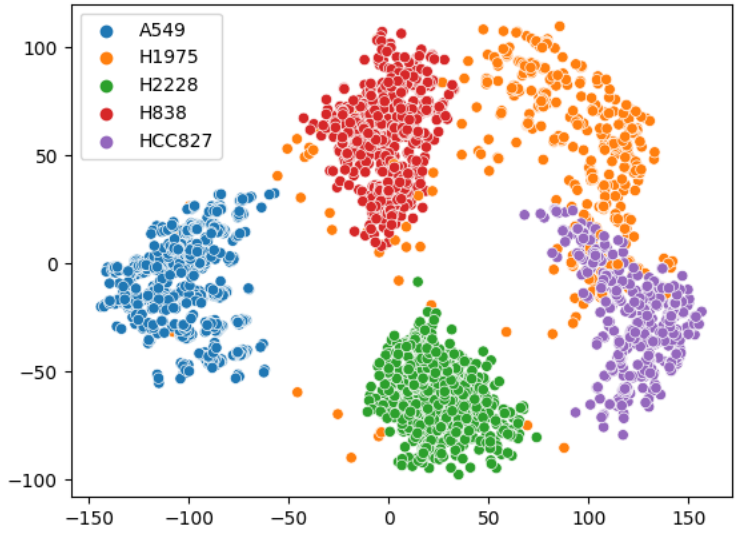


Figure 19: Isomap

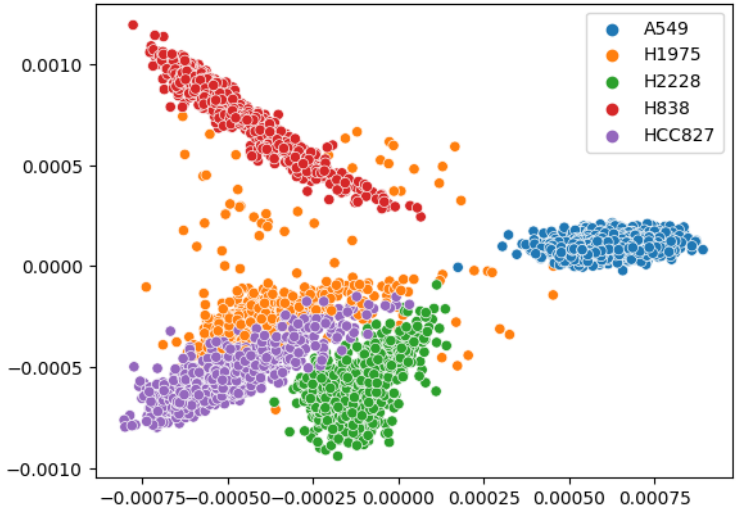


Figure 20: Laplacian Eigenmap

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**
- σ_i adjusts to local densities (neighborhood of i)

Perplexity

A smoothed effective number of neighbors:

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

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

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

Similarities in the low dimension space

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

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

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

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

Optimization

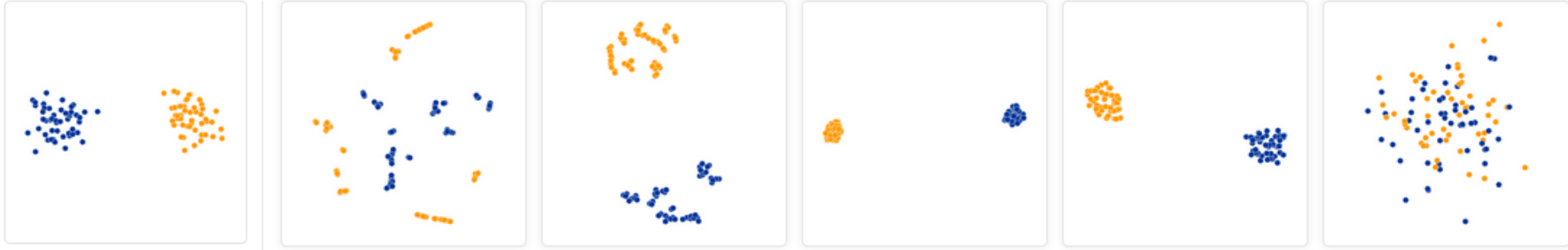
Criterion – Kullback-Leibler between p and q : $C(\mathbf{z}) = \sum_{ij} KL(p_{ij}, q_{ij})$

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

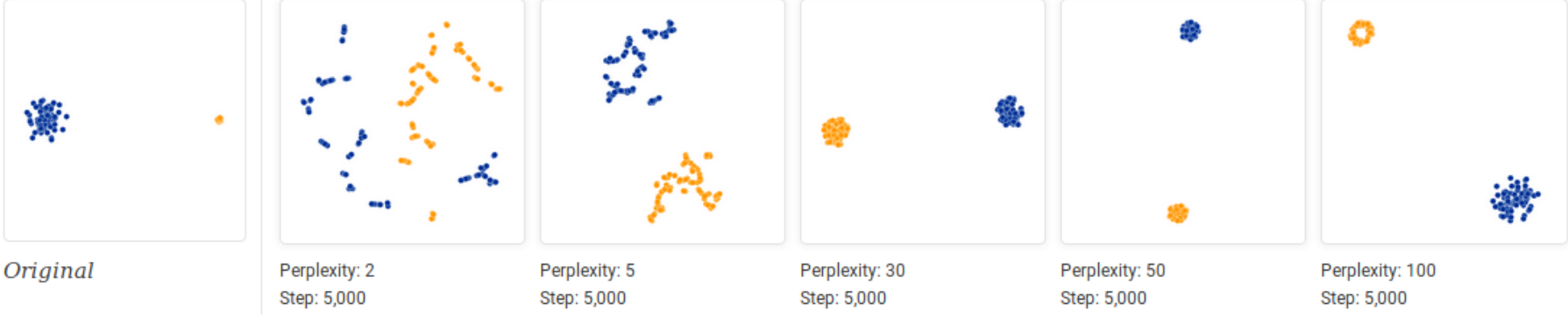
Initiatization – reduce original data with PCA then initialized by $\mathcal{N}(0, \epsilon I_q)$

Empirical properties of tSNE (1)

Effect of Hyperparameters : Perplexity



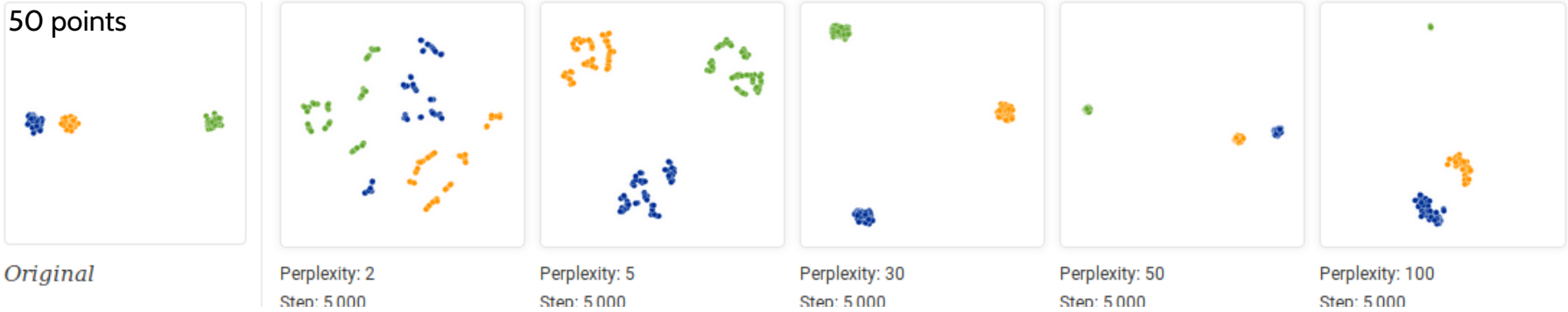
tSNE does not account for heteroscedasticity

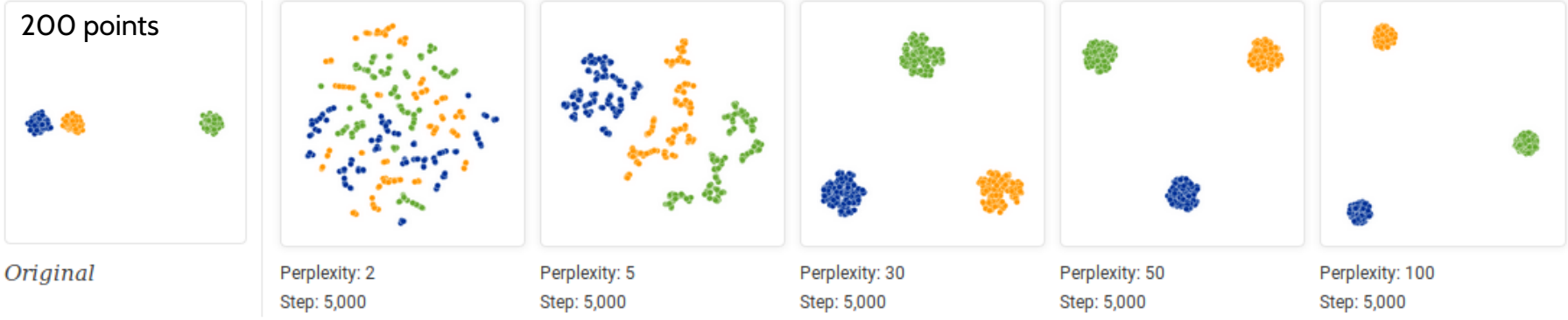


https://distill.pub/2016/misread-tsne/

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

McInnes, Healy, and Melville (2018)

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 data[6]

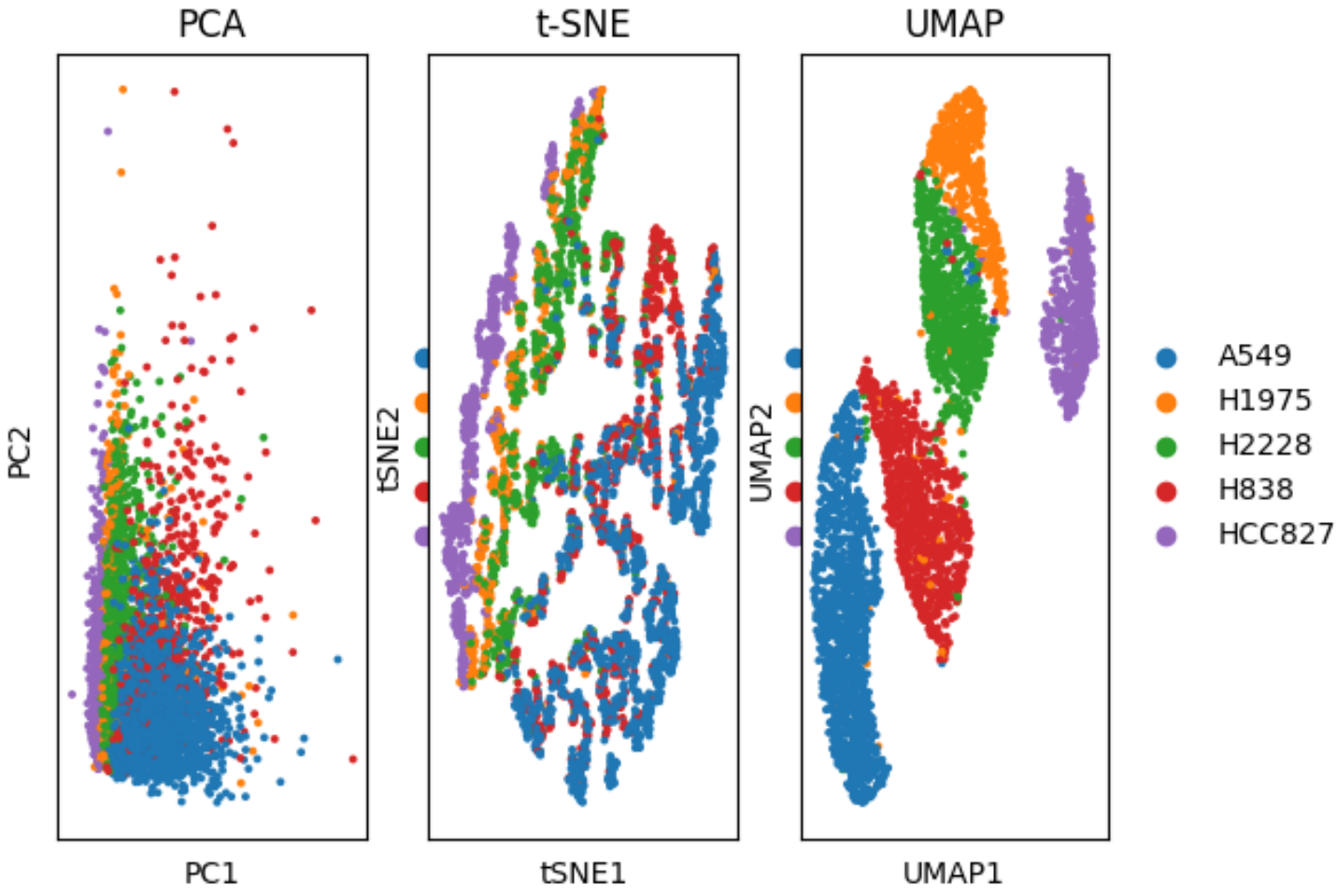


Figure 21: tSNE + UMAP on raw data

[6] using the Python module `scanpy`

tSNE and UMAP scRNA data[7]

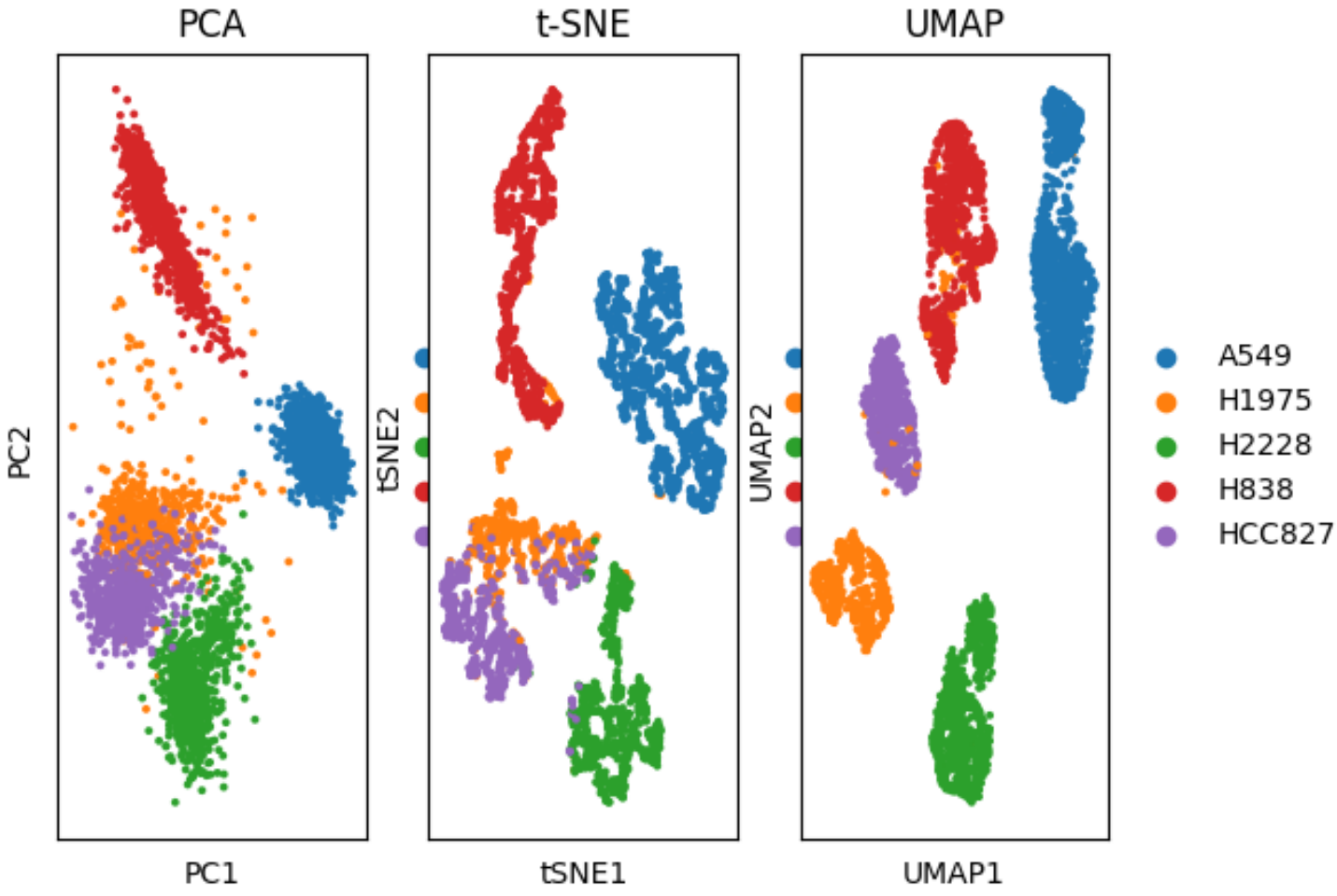
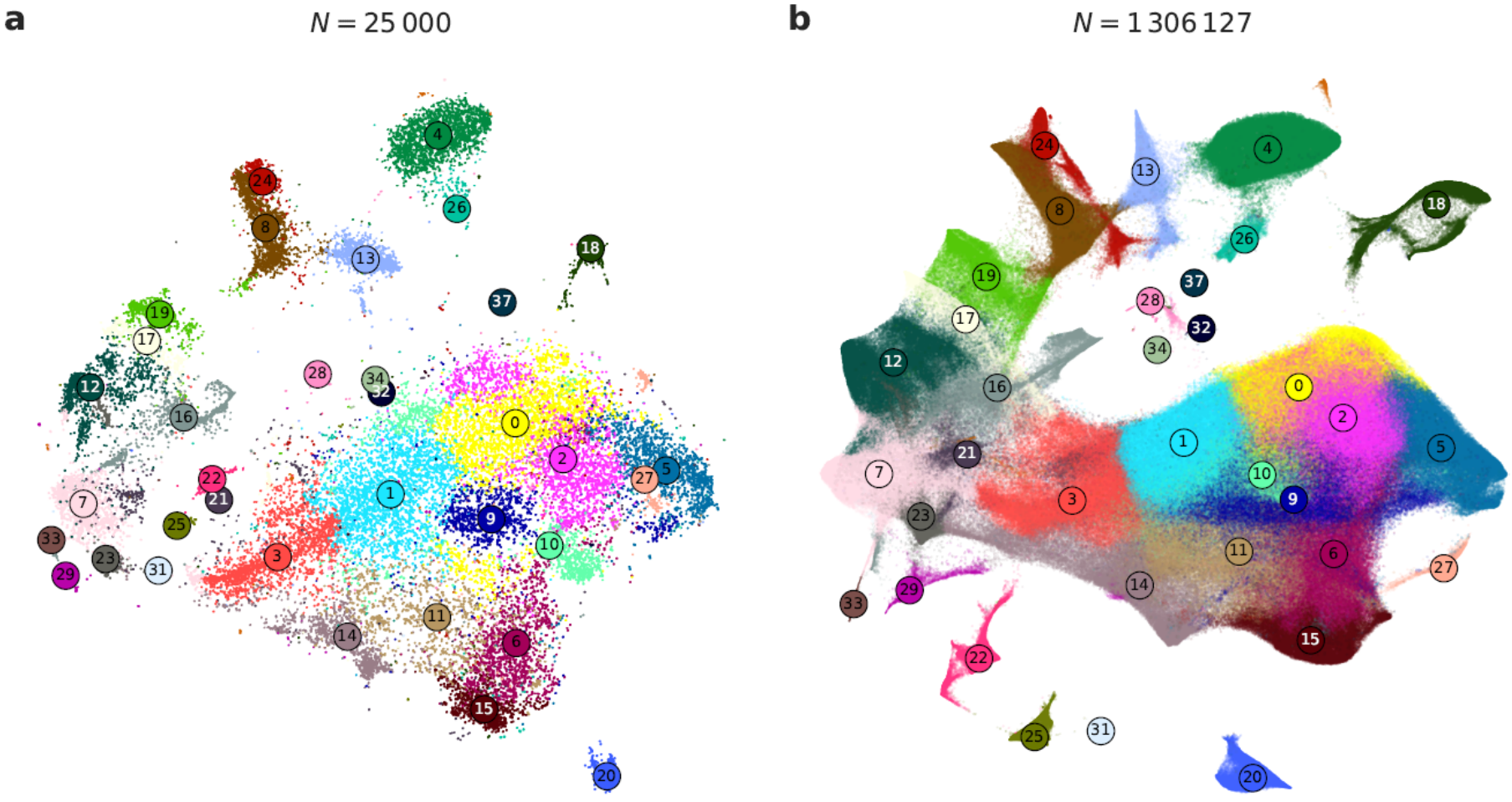


Figure 22: tSNE + UMAP on log-transformed data

[7] using the Python module `scanpy`

tSNE on large scRNA Gene Expression (Kobak and Berens 2018)



Outline

Hidden Graph to structure observations

Consider W the adjacency matrix of a hidden random graph[8]

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:

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

[8] we start with one connected component

Conditional distribution of X on a graph W_X

Consider a Matrix Normal model with row and column dependencies

$$X \mid W_X \sim \mathscr{M}\mathscr{N}\left(0, L_X^{-1}, \Sigma^{-1}\right),$$

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:

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

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

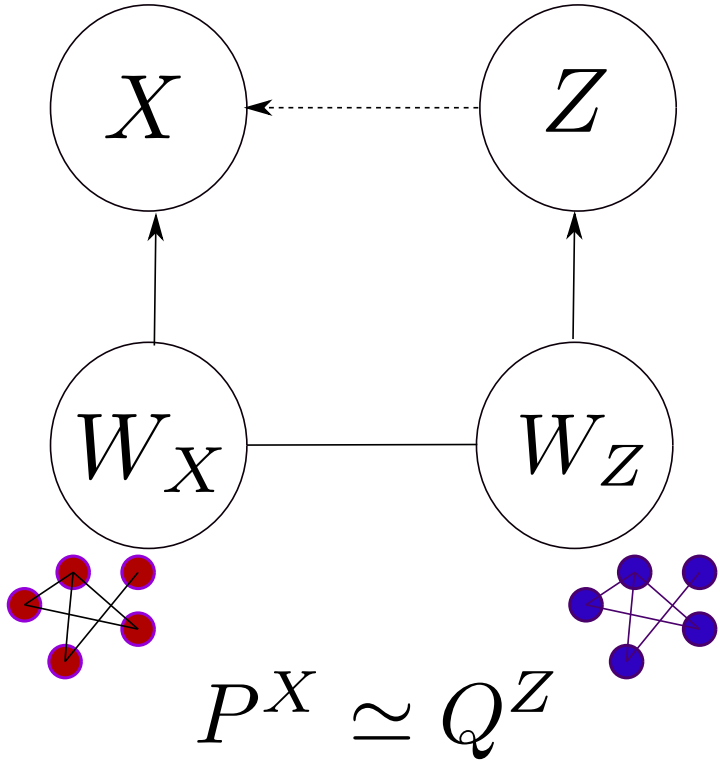
$$\mathbb{P}(Z \mid W_Z) \propto \prod_{(i,j)\in[n]^2} k(Z_i - Z_j)^{W_{Z,ij}}$$

Embedding with Graph Coupling

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

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

$\rightsquigarrow$ Z becomes a parameter to be estimated



Probabilistic Coupling

Graph Coupling with Z as a parameter

Consider the cross entropy between posteriors

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

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

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

Connection with the KL between posteriors

$$\mathrm{KL}\left(\mathbf{P}^X, \mathbf{Q}^Z\right) = \mathscr{H}\left(\mathbf{P}^X, \mathbf{Q}^Z\right) - \mathscr{H}\left(\mathbf{P}^X, \mathbf{P}^X\right)$$

Conjugate priors and posteriors for hidden graphs

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

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

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

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

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

Mixing Prior distributions for coupling

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

Model-based Neighbor Embedding

Choosing $\mathscr{P}_X = \mathscr{P}_Z = \mathscr{D}$ lead us to $\mathscr{H}_{D,D} = -\sum_{i \neq j} P_{ij}^D \log Q_{ij}^D$ 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	$\overline{P}_{ij}^D = P_{ij}^D + P_{ji}^D$	$Q_{ij}^E = \frac{k_z(\mathbf{z}_i - \mathbf{z}_j)}{\sum_{\ell,t} k_z(\mathbf{z}_\ell - \mathbf{z}_t)}$	$-\sum_{i<j} \overline{P}_{ij}^D \log Q_{ij}^E$
LargeVis	$\overline{P}_{ij}^D = P_{ij}^D + P_{ji}^D$	$Q_{ij}^B = \frac{k_z(\mathbf{z}_i - \mathbf{z}_j)}{1 + k_z(\mathbf{z}_i - \mathbf{z}_j)}$	$-\sum_{i<j} \overline{P}_{ij}^D \log Q_{ij}^B + \left(2 - \overline{P}_{ij}^D\right) \log(1 - Q_{ij}^B)$
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

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.

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.

References II

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.

Hastie, Trevor, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Vol. 2. Springer.

Hinton, Geoffrey E, and Sam Roweis. 2002. "Stochastic Neighbor Embedding." *Advances in Neural Information Processing Systems* 15.

Hotelling, Harold. 1933. "Analysis of a Complex of Statistical Variables into Principal Components." *Journal of Educational Psychology* 24 (6): 417.

Kingma, Diederik P, and Max Welling. 2013. "Auto-Encoding Variational Bayes." *arXiv Preprint arXiv:1312.6114*.

Kobak, Dmitry, and Philipp Berens. 2018. "The Art of Using t-SNE for Single-Cell Transcriptomics." *bioRxiv*. https://doi.org/10.1101/453449.

References III

Maaten, Laurens van der, and Geoffrey Hinton. 2008. "Visualizing Data Using t-SNE." *Journal of Machine Learning Research* 9: 2579–2605.

McInnes, L., J. Healy, and J. Melville. 2018. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." *Arxiv*, no. 1802.03426: 1–63. https://arxiv.org/abs/1802.03426.

Murphy, Kevin P. 2022. *Probabilistic Machine Learning: An Introduction*. MIT Press. probml.ai.

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. 1998. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem." *Neural Computation* 10 (5): 1299–319.

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.

References IV

Sra, Suvrit, and Inderjit Dhillon. 2005. “Generalized Nonnegative Matrix Approximations with Bregman Divergences.” *Advances in Neural Information Processing Systems* 18.

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.

Wainwright, M. J., and M. I. Jordan. 2008. “Graphical Models, Exponential Families, and Variational Inference.” *Found. Trends Mach. Learn.* 1 (1–2): 1–305.

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.