Data anaysis and Unsupervised Learning

Dimensionality Reduction: Beyond PCA and Non Linear Methods

MAP 573, 2020 – Julien Chiquet

École Polytechnique, Autumn semester, 2020

https://jchiquet.github.io/MAP573

Outline

Introduction

Motivations

Part I

Introduction

Packages required for reproducing the slides

```r
library(tidyverse)  # opinionated collection of packages for data manipulation
library(GGally)     # extension to ggplot vizualization system
library(FactoMineR) # PCA and oter linear method for dimension reduction
library(factoextra) # fancy plotting for FactoMineR output
# color and plots themes
library(RColorBrewer)
pal <- brewer.pal(10, "Set3")
theme_set(theme_bw())
```

Companion data set: 'scRNA'

Subsamples of normalized Single-Cell RNAseq

Description: *subsample of a large data set*

Gene-level expression of 100 representative genes for a collection of 301 cells spreaded in 11 cell-lines. Original transcription data are measured by counts obtained by *RNAseq* and normalized to be close to Gaussian.

Pollen, Alex A., et al. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex.
Nature biotechnology 32.10 (2014): 1053.

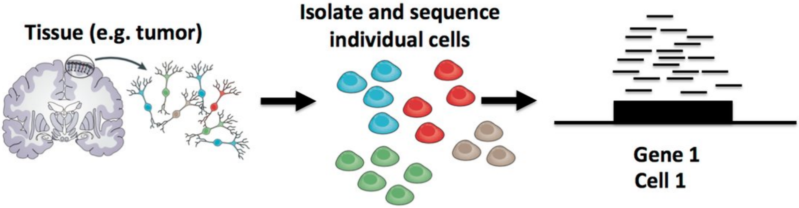


Figure: Single Cell RNA sequencing data: general principle – source: Stephanie Hicks

Companion data set: 'scRNA'

Brief data summary I

Data manipulation

```r
load("../../data/scRNA.RData")
scRNA <-  pollen$data %>% t() %>% as_tibble() %>%
  add_column(cell_type = pollen$celltypes)
```

Cell types

```r
scRNA %>% dplyr::select(cell_type) %>% summary()  %>% knitr::kable()
```

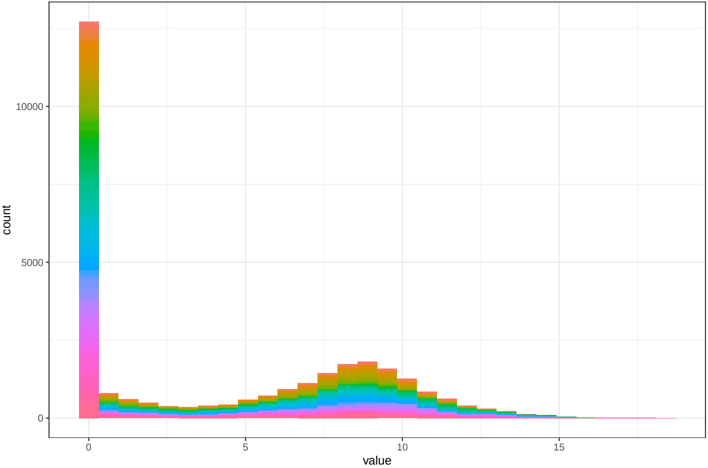
	cell_type
	HL60 :54
	K562 :42
	Kera :40
	BJ :37
	GW16 :26
	hiPSC :24
	(Other):78

Companion data set II: 'scRNA'

Brief data summary II

Histogram of normalized expression

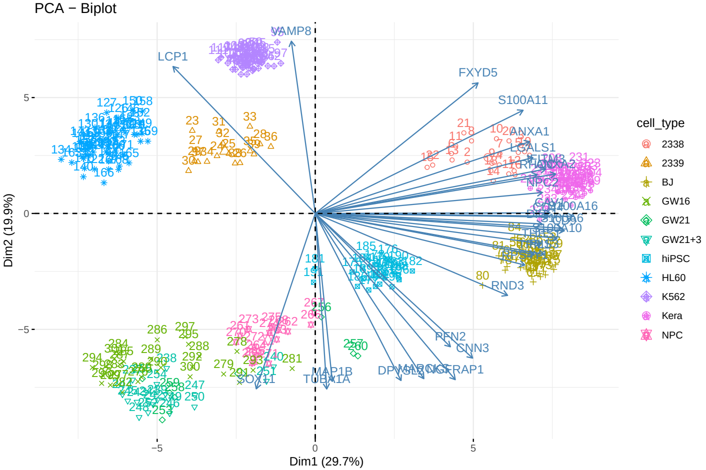
```
scRNA %>% dplyr::select(-cell_type) %>% pivot_longer(everything()) %>%
  ggplot() + aes(x = value, fill = name) + geom_histogram(show.legend = FALSE)
```



Companion data set: 'scRNA'

PCA

```
scRNA %>% FactoMineR::PCA(graph = FALSE, quali.sup = which(colnames(scRNA) == "cell
  factoextra::fviz_pca_biplot(select.var = list(contrib = 30), habillage = "cell_ty
```



PCA (and linear methods) limitations

Account for complex pattern

- Linear methods are powerful for **planar structures**
- May fail at describing **manifolds**

Preserve local geometry

- High dimensional data are characterized by **multiscale properties** (local / global structures)
- Non Linear projection helps at preserving **local characteristics** of distances

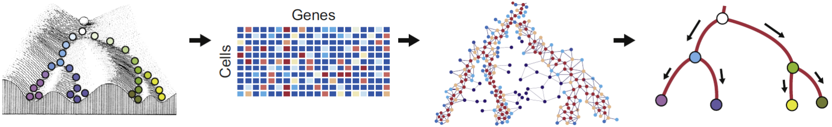


Figure: Intuition of manifolds and geometry underlying sc-data – source: F. Picard

Companion data set II: 'mollusk'

Abundance table (Species counts spread in various sites)

Description: *small size count data*

Abundance of 32 mollusk species in 163 samples. For each sample, 4 additional covariates are known.

Richardot-Coulet, M., Chessel D. and Bournaud M. Typological value of the benthos of old beds of a large river. Methodological approach. Archiv fùr Hydrobiologie, 107.

```
mollusk <- PLNmodels::mollusk$Abundance %>% as_tibble() %>%
   add_column(site   = PLNmodels::mollusk$Covariate$site,
              season = PLNmodels::mollusk$Covariate$season)
```

External Covariates

```
mollusk %>% dplyr::select(site, season) %>% summary() %>% t() %>% knitr::kable()
```

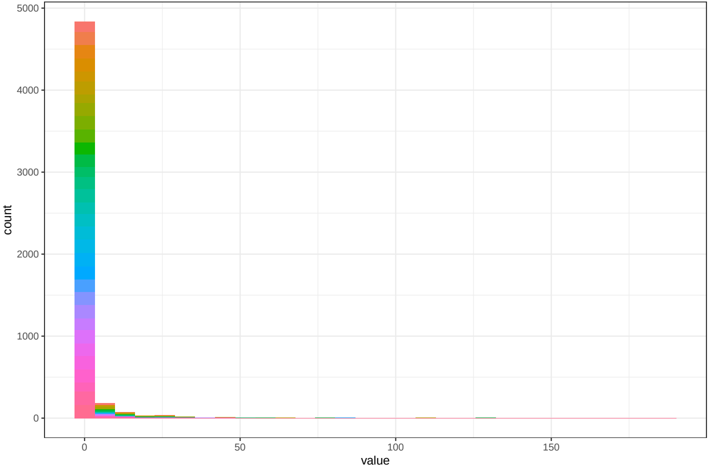
site	Negria1 :24	Negria2 :24	Pecheurs1:24	Pecheurs2:24	GGravier3:22	GGravie
season	automn:41	spring:44	summer:44	winter:34	NA	NA

Companion data set: 'mollusk'

Brief data summary II

Histogram of raw counts

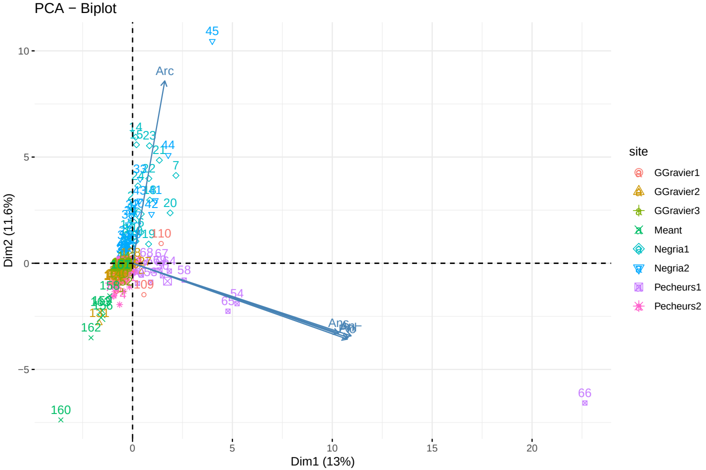
```
mollusk %>% dplyr::select(-site, -season) %>% pivot_longer(everything()) %>%
  ggplot() + aes(x = value, fill = name) + geom_histogram(show.legend = FALSE)
```



Companion data set: 'mollusk'

PCA

```
mollusk %>% PCA(graph = FALSE, quali.sup = which(map_lgl(mollusk, is.factor))) %>%
  fviz_pca_biplot(select.var = list(contrib = 5), habillage = "site")
```



PCA (and linear methods) limitations

Account for complex data distribution

- Linear methods /PCA are tied to an hidden **Gaussian assumption**
- Fail with **Count data**
- Fail with **Skew data**

Possible solutions

- Probabilistic (non Gaussian) models
- Need transformed (non-linear) input space

Dimension reduction: revisiting the problem setup

Settings

- Training data : $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^p$, (i.i.d.)
- Space $\mathbb{R}^p$ of possibly high dimension $(n \ll p)$

Dimension Reduction Map

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

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

How should we design/construct Φ?

Criterion

- Geometrical approach (**see slides on PCA**)
- Reconstruction error
- Relationship preservation

Form of the map Φ

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

Part II

Non-linear methods

Outline

Non-linear methods

1. Motivated by reconstruction error
    - General goal
    - Non-negative matrix factorization
    - Kernel-PCA
    - Auto-Encoder

2. Relation preservation

Outline

Non-linear methods

1. Motivated by reconstruction error
   - General goal
   - Non-negative matrix factorization
   - Kernel-PCA
   - Auto-Encoder

2. Relation preservation

Reconstruction error approach

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

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

2. Construct $\widetilde{\Phi}$ from $\mathbb{R}^q$ to $\mathbb{R}^p$ (reconstruction formula)
3. Control an error between $\mathbf{x}$ and its reconstruction $\tilde{\Phi}(\Phi(\mathbf{x}))$, e.g

$$\sum_{i=1}^{n} \left\| \mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i)) \right\|^2$$

Reinterpretation of PCA

PCA model

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

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

$$\mathbf{x} \simeq \tilde{\Phi}(\tilde{\mathbf{x}}) = \boldsymbol{\mu} + \mathbf{V}\tilde{\mathbf{x}}$$

$\rightsquigarrow$ Model with **Linear assumption + ortho-normality constraints**

PCA reconstruction error

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

Solution (explicit)

- $\boldsymbol{\mu} = \bar{\mathbf{x}}$ the empirical mean
- $\mathbf{V}$ an orthonormal basis of the space spanned by the q first eigenvectors of the empirical covariance matrix

Reinterpretation of PCA

PCA model

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

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

$$\mathbf{x} \simeq \tilde{\Phi}(\tilde{\mathbf{x}}) = \boldsymbol{\mu} + \mathbf{V}\tilde{\mathbf{x}}$$

$\rightsquigarrow$ Model with **Linear assumption + ortho-normality constraints**

PCA reconstruction error

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

Solution (explicit)

- $\boldsymbol{\mu} = \bar{\mathbf{x}}$ the empirical mean
- $\mathbf{V}$ an orthonormal basis of the space spanned by the q first eigenvectors of the empirical covariance matrix

Important digression: SVD

Singular Value Decomposition (SVD)

The SVD of $\mathbf{M}$ a $n \times p$ matrix is the factorization given by

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top,$$

where $r = \min(n, p)$ and

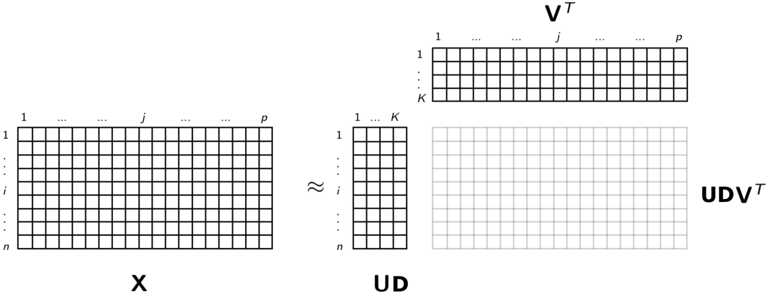
- $\mathbf{D}_{r\times r} = \text{diag}(\delta_1, ...\delta_r)$ is the diagonal matrix of singular values.
- $\mathbf{U}$ is orthonormal, whose columns are eigen vectors of $(\mathbf{M}\mathbf{M}^T)$
- $\mathbf{V}$ is orthonormal whose columns are eigen vectors of $(\mathbf{M}^T\mathbf{M})$

$\rightsquigarrow$ Time complexity in $\mathcal{O}(npqr)$ (less when $k \ll r$ components are required)

Connection with eigen decomposition of the covariance matrix

$$\begin{aligned}\mathbf{M}^\top\mathbf{M} &= \mathbf{V}\mathbf{D}\mathbf{U}^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top \\ &= \mathbf{V}\mathbf{D}^2\mathbf{V}^\top = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top\end{aligned}$$

PCA solution is given by SVD of the centered data matrix



Since $\tilde{\mathbf{X}} = \mathbf{X}^c\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{V} = \mathbf{U}\mathbf{D}$, PCA can be rephrased as

$$\hat{\mathbf{X}}^c = \mathbf{F}\mathbf{V}^\top = \underset{\mathbf{F}\in\mathcal{M}_{p,q},\mathbf{V}\in\mathcal{O}_{p,q}}{\arg\min} \left\|\mathbf{X}^c - \mathbf{F}\mathbf{V}^\top\right\|_F^2 \text{ with } \|\mathbf{A}\|_F^2 = \sum_{ij} a_{ij}^2,$$

$$\left.\begin{matrix}\tilde{\mathbf{X}} \in \mathbb{R}^{n\times q}, \mathbf{V} \in \mathbb{R}^{p\times q}\end{matrix}\right\} \text{ Best linear low-rank representation of } \mathbf{X}$$

Outline

Non-linear methods

1. Motivated by reconstruction error
    - General goal
    - Non-negative matrix factorization
    - Kernel-PCA
    - Auto-Encoder

2. Relation preservation

Outline

Non-linear methods

1. Motivated by reconstruction error
   - General goal
   - Non-negative matrix factorization
   - Kernel-PCA
   - Auto-Encoder
2. Relation preservation

Kernel-PCA

Principle: non linear transformation of $\mathbf{x}$ before PCA

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

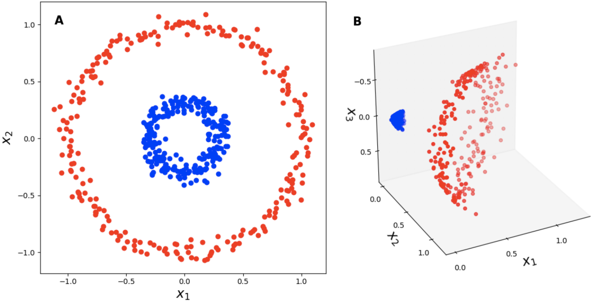


Figure: $\Psi : \mathbf{x} \rightarrow \Psi(\mathbf{x})$

Kernel-PCA

Kernel PCA Model

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

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

Kernel trick

Never calculate $\Psi(\mathbf{x}_i)$ thanks to the kernel trick:

$$K = k(\mathbf{x}, \mathbf{y}) = (\Psi(\mathbf{x}), \Psi(\mathbf{y})) = \Psi(\mathbf{x})^T \Psi(\mathbf{y})$$

Solution

Eigen-decomposition of the doubly centered kernel matrix $\mathbf{K} = k(\mathbf{x}_i, \mathbf{x}_{i'})$

$$\tilde{\mathbf{K}} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top / n)\mathbf{K}(\mathbf{I} - \mathbf{1}\mathbf{1}^\top / n) = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$$

Choice of a kernel

A symmetric positive definite function $k(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$, which depends on the kind of similarity assumed

Some common kernels

- **Polynormial Kernel**

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

- **Gaussian (radial) kernel**

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

- **Laplacian kernel**

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

⇝ Kernel PCA suffers from the choice of the Kernel to correctly

Example on scRNA

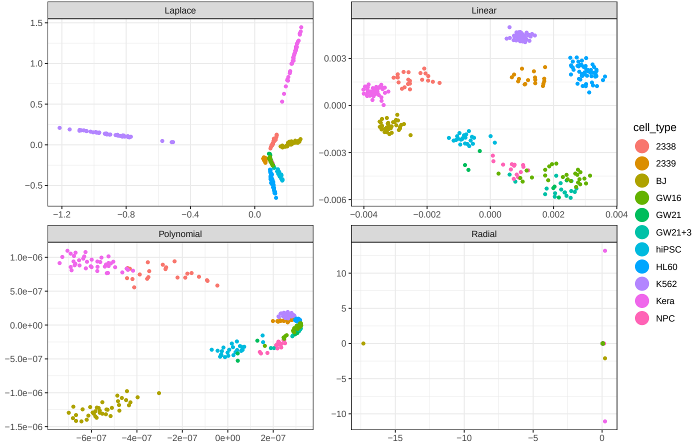
Run the fit

```
scRNA_expr <- scRNA %>% select(-cell_type) %>% as.matrix()

kPCA_radial <-
  kpca(scRNA_expr, kernel = "rbfdot", features = 2, kpar = list(sigma = 0.5)) %>%
  pcv() %>% as.data.frame() %>%
  add_column(kernel = "Radial") %>%
  add_column(cell_type = scRNA$cell_type)

kPCA_linear <-
  kpca(scRNA_expr, kernel = "vanilladot", features = 2, kpar = list()) %>%
  pcv() %>% as.data.frame() %>%
  add_column(kernel = "Linear") %>%
  add_column(cell_type = scRNA$cell_type)

kPCA_polydot <- kpca(scRNA_expr, kernel = "polydot", features = 2, kpar = list(degr
  pcv() %>% as.data.frame() %>%
  add_column(kernel = "Polynomial") %>%
  add_column(cell_type = scRNA$cell_type)

kPCA_laplacedot <- kpca(scRNA_expr, kernel = "laplacedot", features = 2) %>%
  pcv() %>% as.data.frame() %>%
  add_column(kernel = "Laplace") %>%
  add_column(cell_type = scRNA$cell_type)
```

Example on scRNA

Compare the projection

```
rbind(kPCA_linear, kPCA_polydot, kPCA_radial, kPCA_laplacedot) %>%
  ggplot(aes(x = V1, y = V2, color = cell_type)) +
  geom_point(size=1.25) + guides(colour = guide_legend(override.aes = list(size=6))
  facet_wrap(.~kernel, scales = 'free') + labs(x = '', y = '')
```



Outline

Non-linear methods

1. Motivated by reconstruction error
    - General goal
    - Non-negative matrix factorization
    - Kernel-PCA
    - Auto-Encoder
2. Relation preservation

Outline

Non-linear methods

1. Motivated by reconstruction error

2. Relation preservation
   - General goal
   - MDS
   - t-SNE
   - UMAP

Outline

Non-linear methods

1. Motivated by reconstruction error

2. Relation preservation
   - General goal
   - MDS
   - t-SNE
   - UMAP

Pairwise Relation

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

Distance Preservation

- Construct a map Φ from the space $\mathbb{R}^d$ into a space $\mathbb{R}^{d'}$ of smaller dimension:

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

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

Multidimensional scaling

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

Outline

Non-linear methods

1. Motivated by reconstruction error

2. Relation preservation
   - General goal
   - MDS
   - t-SNE
   - UMAP

Outline

Non-linear methods

1. Motivated by reconstruction error

2. Relation preservation
   - General goal
   - MDS
   - t-SNE
   - UMAP

Stochastic Neighbor Embedding [van der Maaten and Hinton, 2008]

- $(x_1, \dots, x_n)$ are the points in the high dimensional space $\mathbb{R}^p$,
- Consider a similarity between points:

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

- σ smooths the data (linked to the regularity of the target manifold)
- σ is chosen such that the entropy of p is fixed to a given value of the so-called perplexity

$$\exp\left(-\sum_{ij} p_{ij} \log(p_{ij})\right)$$

The perplexity parameter

- σ_i Should adjust to local densities (neighborhood of point i)
- Define the Shannon entropy of $p_i = (p_{1|i}, \ldots, p_{n|i})$

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

- The perplexity is defined by:

$$Perp(p_i) = 2^{H(p_i)}$$

- Interpreted as the smoothed effective number of neighbors.
- SNE performs a binary search for the value of si that produces a p_i with a fixed perplexity that is specified by the user.

tSNE and Student / Cauchy kernels

- Consider $(y_1, \ldots, y_n)$ are points in the low dimensional space $\mathbb{R}^2$
- Consider a similarity between points in the new representation:

$$q_{i|j} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_k - y_j\|^2)}$$

- Robustify this kernel by using Student(1) kernels (ie Cauchy)

$$q_{i|j} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_i - y_k\|^2)^{-1}}$$

Optimizing tSNE

- Minimize the KL between p and q so that the data representation minimizes:
$$C(y) = \sum_{ij} KL(p_{ij}, q_{ij})$$
- The cost function is not convex
$$\left[\frac{\partial C(y)}{\partial y}\right]_i = \sum_j (p_{ij} - q_{ij})(y_i - y_j)$$
- Interpreted as the resultant force created by a set of springs between the map point y_i and all other map points $(y_j)_j$. All springs exert a force along the direction $(y_i - y_j)$.
- $(p_{ij} - q_{ij})$ is viewed as a stiffness of the force exerted by the spring between y_i and y_j.

Customed Gradient descent

- Gradient descent initialized by sampling map points randomly from an isotropic Gaussian with small variance centered around the origin
- Gradient update using

$$y^{(t)} = y^{(t-1)} + \eta \frac{\partial C(y)}{\partial y} + \alpha(t)(y^{(t-1)} - y^{(t-2)})$$

- η learning rate, $\alpha(t)$ momentum at iteration t.
- Gaussian noise is added to the map points to perform simulated annealing.

Properties of t-SNE

- good at preserving local distances (intra-cluster variance)
- not so good for global representation (inter-cluster variance)
- hence good at creating clusters of points that are close, but bad at positionning clusters wrt each other
- preprocessing very important : initialize with PCA and feature selection plus log transform (non linear transform)
- percent of explained variance ? interpretation of the q distribution ?

Example on scRNA I

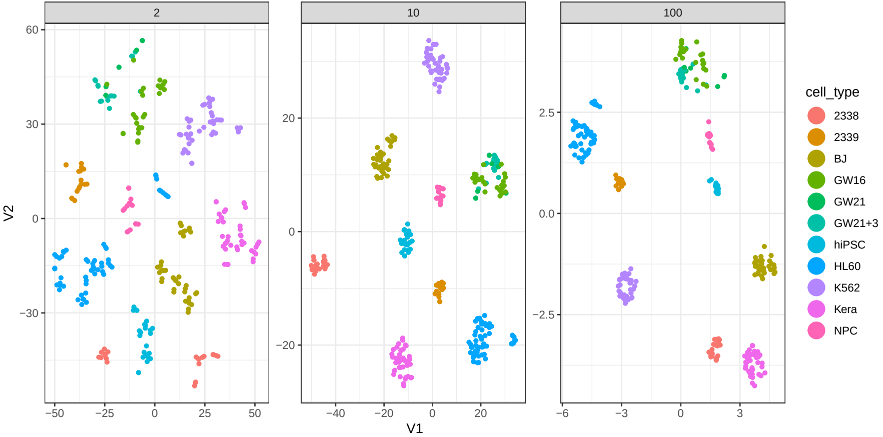
Run the fit

```
scRNA_expr <- scRNA %>% select(-cell_type) %>% as.matrix()

tSNE_perp2   <- Rtsne(scRNA_expr, perplexity =   2)$Y %>%
  as.data.frame() %>% add_column(perplexity = 2) %>% add_column(cell_type = scRNA$c

tSNE_perp10  <- Rtsne(scRNA_expr, perplexity =  10)$Y %>%
  as.data.frame() %>% add_column(perplexity = 10) %>% add_column(cell_type = scRNA$

tSNE_perp100 <- Rtsne(scRNA_expr, perplexity = 100)$Y %>%
  as.data.frame() %>% add_column(perplexity = 100) %>% add_column(cell_type = scRNA
```

Compare perplexity

```
rbind(tSNE_perp2,tSNE_perp10,tSNE_perp100) %>%
  ggplot(aes(x = V1, y = V2, color = cell_type)) +
     geom_point(size=1.25) +
     guides(colour = guide_legend(override.aes = list(size=6))) +
  facet_wrap(.~perplexity, scales = 'free')
```

Example on scRNA II



Example on 'mollusk' I

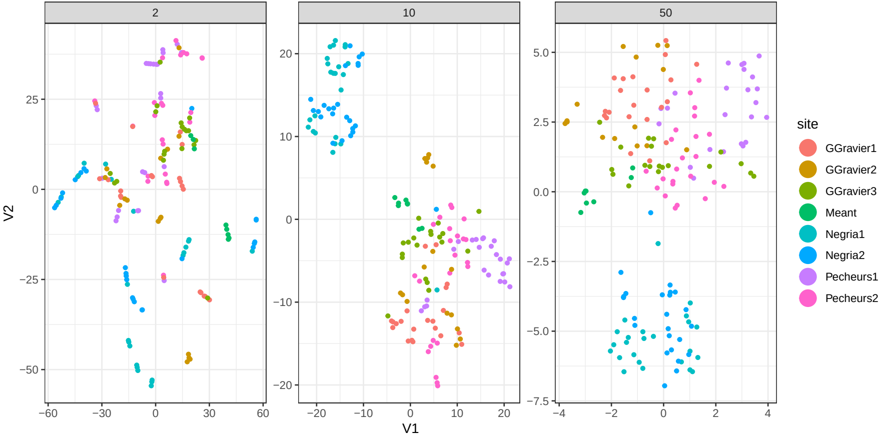
Run the fit

```
duplicated <- duplicated(mollusk %>% select(-site, -season))
mollusk_ab <- mollusk %>% select(-site, -season) %>% filter(!duplicated) %>%  as.ma

tSNE_perp2   <- Rtsne(mollusk_ab, perplexity =   2)$Y %>%
  as.data.frame() %>% add_column(perplexity = 2) %>% add_column(site = mollusk$site

tSNE_perp10  <- Rtsne(log(1 + mollusk_ab), perplexity =  10)$Y %>%
  as.data.frame() %>% add_column(perplexity = 10) %>% add_column(site = mollusk$sit

tSNE_perp50 <- Rtsne(log(1 + mollusk_ab), perplexity = 50)$Y %>%
  as.data.frame() %>% add_column(perplexity = 50) %>% add_column(site = mollusk$sit
```

Compare perplexity

```
rbind(tSNE_perp2,tSNE_perp10,tSNE_perp50) %>%
  ggplot(aes(x = V1, y = V2, color = site)) +
     geom_point(size=1.25) +
     guides(colour = guide_legend(override.aes = list(size=6))) +
  facet_wrap(.~perplexity, scales = 'free')
```

Example on 'mollusk' II



Outline

Non-linear methods

1. Motivated by reconstruction error

2. Relation preservation

   General goal

   MDS

   t-SNE

   UMAP

Uniform Manifold Approximation and Projection [McInnes et al., 2018]

Properties of UMAP

-

Example on scRNA I

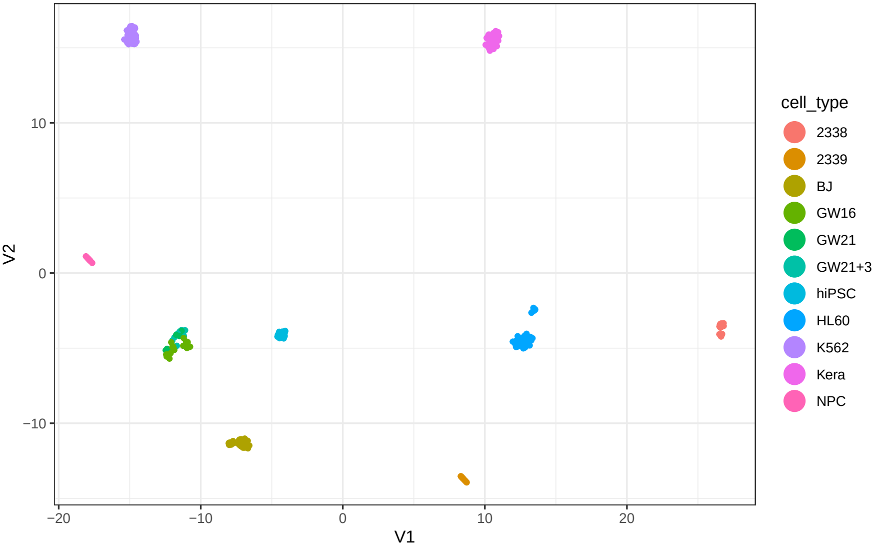
Run the fit

```r
scRNA_expr <- scRNA %>% select(-cell_type) %>% as.matrix()
umap_fit   <- umap(scRNA_expr)$layout %>%
  as.data.frame() %>% add_column(cell_type = scRNA$cell_type)
```

Visualization

```r
umap_fit %>%
  ggplot(aes(x = V1, y = V2, color = cell_type)) +
     geom_point(size=1.25) +
     guides(colour = guide_legend(override.aes = list(size=6)))
```

Example on scRNA II



Example on 'mollusk' I

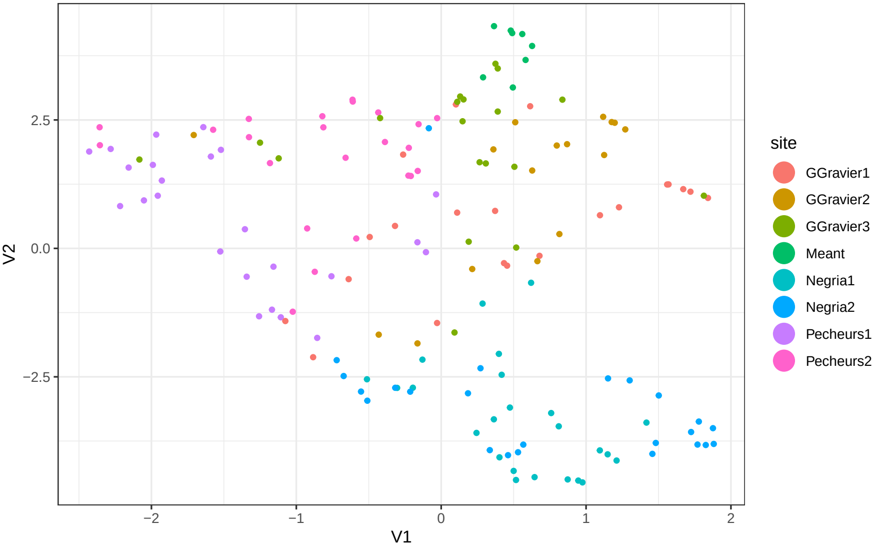
Run the fit

```
duplicated <- duplicated(mollusk %>% select(-site, -season))
mollusk_ab <- mollusk %>% select(-site, -season) %>% filter(!duplicated) %>%  as.ma
umap_fit   <- umap(mollusk_ab)$layout %>%
  as.data.frame() %>% add_column(site = mollusk$site[!duplicated])
```

Visualization

```
umap_fit %>%
  ggplot(aes(x = V1, y = V2, color = site)) +
     geom_point(size=1.25) +
     guides(colour = guide_legend(override.aes = list(size=6)))
```

Example on 'mollusk' II



References I

McInnes, L., Healy, J., and Melville, J. (2018).

Umap: Uniform manifold approximation and projection for dimension reduction.

arXiv preprint arXiv:1802.03426.

van der Maaten, L. and Hinton, G. (2008).

Visualizing Data using t-SNE.

Journal of Machine Learning Research, 9(Nov):2579–2605.