Machine Learning

Unsupervised Methods
Part 2

Sepp Hochreiter

Institute of Bioinformatics
Johannes Kepler University, Linz, Austria

Outline

3 Principal Component Analysis

3.1 The Method

3.2 Variance Maximization

3.3 Uniqueness

3.4 Properties of PCA

3.5 Examples

3.6 Kernel Principal Component Analysis

4 Independent Component Analysis

4.1 Identifiability and Uniqueness

4.2 Measuring Independence

4.3 Whitening and Rotation Algorithms

4.4 INFOMAX Algorithm

4.5 EASI Algorithm

4.6 FastICA Algorithm

4.7 ICA Extensions

4.8 ICA vs. PCA

4.9 Artificial ICA Examples

4.9.1 Whitening and Rotation

4.10 Real World ICA Examples

4.11 Kurtosis Maximization Results in Independent Components

Outline

5 Factor Analysis

5.1 The Factor Analysis Model

5.2 Maximum Likelihood Factor Analysis

5.3 Factor Analysis vs. PCA and ICA

5.4 Artificial Factor Analysis Examples

5.5 Real World Factor Analysis Examples

6 Scaling and Projection Methods

6.1 Projection Pursuit

6.2 Multidimensional Scaling

6.3 Non-negative Matrix Factorization

6.4 Locally Linear Embedding

6.5 Isomap

6.6 The Generative Topographic Mapping

6.7 t-Distributed Stochastic Neighbor Embedding

6.8 Self-Organizing Maps

7 Clustering

7.1 Mixture Models

7.2 k-Means Clustering

7.3 Hierarchical Clustering

7.4 Similarity-Based Clustering

Outline

8 Biclustering

8.1 Types of Biclusters

8.2 Overview of Biclustering Methods

8.3 FABIA Biclustering

8.4 Examples

9 Hidden Markov Models

9.1 Hidden Markov Models in Bioinformatics

9.2 Hidden Markov Model Basics

9.3 Expectation Maximization for HMM: Baum-Welch Algorithm

9.4 Viterby Algorithm

9.5 Input Output Hidden Markov Models

9.6 Factorial Hidden Markov Models

9.7 Memory Input Output Factorial Hidden Markov Models

9.8 Tricks of the Trade

9.9 Profile Hidden Markov Models

10 Boltzmann Machines

10.1 The Boltzmann Machine

10.2 Learning in the Boltzmann Machine

10.3 The Restricted Boltzmann Machine

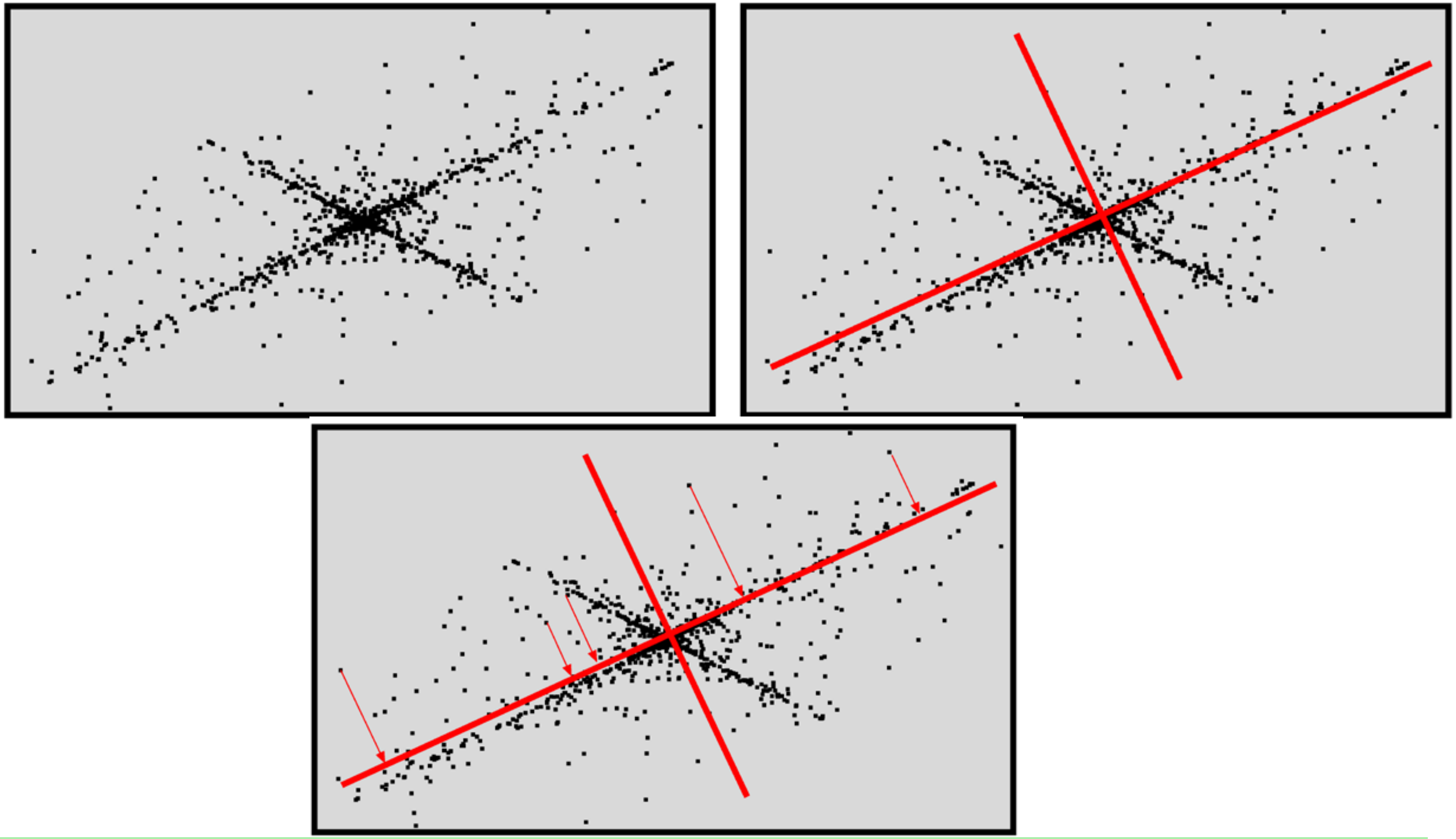
Chapter 3

Principal Component Analysis

Principal Component Analysis

Principal Component Analysis (PCA), Karhunen-Loéve transform (KTL), Hotelling transform makes a transformation of the coordinate system:

- data has largest variance along the first coordinate
- second largest data variance is along the second coordinate



Principal Component Analysis

summarize multivariate data by PCA via projecting observations onto the first principal components: for visualization the first two

data $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ summarized by $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$
data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$

rows of the data matrix contain the observations
columns contain the features

We assume that the features have zero sample mean (otherwise, the feature mean must be subtracted)

Principal Component Analysis

sample covariance matrix $\boldsymbol{C} \in \mathbb{R}^{m \times m}$ of features across observations is

$$C_{st} = \frac{1}{n} \sum_{i=1}^{n} x_{is} \, x_{it} \text{ , where } x_{is} = (\boldsymbol{x}_i)_s \text{ and } x_{it} = (\boldsymbol{x}_i)_t$$

$$\boldsymbol{C} = \frac{1}{n} \boldsymbol{X}^T \boldsymbol{X} = \frac{1}{n} \boldsymbol{U} \boldsymbol{D}_m \boldsymbol{U}^T$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ is orthogonal and $\boldsymbol{D}_m \in \mathbb{R}^{m \times m}$ diagonal

This is the eigendecomposition or spectral decomposition of $\boldsymbol{C}$, which is a symmetric positive definite matrix

diagonal entries of $\boldsymbol{D}_m$: eigenvalues (positive, sorted decreasingly)
column vectors $\boldsymbol{u}_i = [\boldsymbol{U}]_i$: eigenvectors (principal components)

first principal component corresponds to the largest eigenvalue

assume that $n \geq m$ and at least m linear independent observations
→ $\boldsymbol{C}$ has full rang (often ensured by unsupervised feature selection)

Principal Component Analysis

singular value decomposition (SVD) $\boldsymbol{X} \ = \ \boldsymbol{V} \ \boldsymbol{D} \ \boldsymbol{U}^T$

where $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ are orthogonal, $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ is diagonal with positive entries, the singular values, sorted decreasingly

Computing $\boldsymbol{X}^T \boldsymbol{X}$ we see that $\boldsymbol{D}_m = \boldsymbol{D}^T \ \boldsymbol{D}$ (the eigenvalues are the singular values squared) and $\boldsymbol{U}$ is the orthogonal matrix from PCA.

PCA projection: $\boldsymbol{Y} \ = \ \boldsymbol{X} \ \boldsymbol{U} \ = \ \boldsymbol{V} \ \boldsymbol{D}$

SVD automatically provides the PCA projections via $\boldsymbol{V} \boldsymbol{D}$
For single observations $\boldsymbol{x}$ the projection is $\boldsymbol{y} \ = \ \boldsymbol{U}^T \boldsymbol{x}$

PCA is a matrix decomposition problem: $\boldsymbol{X} \ = \ \boldsymbol{Y} \boldsymbol{U}^T$

where $\boldsymbol{U}$ is orthogonal, $\boldsymbol{Y}^T \ \boldsymbol{Y} \ = \ \boldsymbol{D}_m$ (the $\boldsymbol{y}$ are orthogonal, decorrelated), and the eigenvalues $\boldsymbol{D}_m$ are sorted decreasing; for single observations that is $\boldsymbol{x} \ = \ \boldsymbol{U} \ \boldsymbol{y}$

Principal Component Analysis

outer product representation:

$$\boldsymbol{X} \;=\; \sum_{i=1}^{m} D_{ii}\; \boldsymbol{v}_i \boldsymbol{u}_i^T \;=\; \sum_{i=1}^{m} \boldsymbol{y}_i \boldsymbol{u}_i^T$$

$\boldsymbol{u}_i$ is the i-th orthogonal column vector of $\boldsymbol{U}$

$\boldsymbol{v}_i$ is the i-th orthogonal column vector of $\boldsymbol{V}$

$\boldsymbol{y}_i \;=\; D_{ii}\; \boldsymbol{v}_i$

Principal Component Analysis

Iterative methods for PCA:
current projection is $t = \boldsymbol{u}^T \boldsymbol{x}$ then Oja's rule is

$$\boldsymbol{u}^{\mathrm{new}} = \boldsymbol{u} + \eta \left(t\, \boldsymbol{x} - t^2\, \boldsymbol{u} \right)$$

where η is the learning rate

The eigenvectors of $\boldsymbol{C}$ are the fixed points of Oja's rule; only the eigenvector with largest eigenvalue is a stable fixed point

$$\begin{aligned} \mathrm{E}_{\boldsymbol{x}}(\boldsymbol{u}^{\mathrm{new}}) &= \boldsymbol{u} + \eta\, \mathrm{E}_{\boldsymbol{x}} \left(\boldsymbol{x}(\boldsymbol{x}^T \boldsymbol{u}) - (\boldsymbol{u}^T \boldsymbol{x})(\boldsymbol{x}^T \boldsymbol{u})\, \boldsymbol{u} \right) = \\ & \boldsymbol{u} + \eta \left(\mathrm{E}_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{u} - (\boldsymbol{u}^T E_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{u})\, \boldsymbol{u} \right) = \\ & \boldsymbol{u} + \eta \left(\boldsymbol{C}\boldsymbol{u} - (\boldsymbol{u}^T \boldsymbol{C}\boldsymbol{u})\, \boldsymbol{u} \right) \end{aligned}$$

If $\boldsymbol{u}$ is an eigenvector of $\boldsymbol{C}$ with eigenvalue λ then

$$\mathrm{E}_{\boldsymbol{x}}(\boldsymbol{u}^{\mathrm{new}}) = \boldsymbol{u} + \eta \left(\lambda \boldsymbol{u} - \lambda\, \boldsymbol{u} \right) = \boldsymbol{u}$$

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

The **first principal component** $\boldsymbol{u}_1$ is the direction of **maximum variance**:

$$\boldsymbol{u}_1 \;=\; \arg\max_{\|\boldsymbol{u}\|=1} \sum_{i=1}^{n} \left(\boldsymbol{u}^T \; \boldsymbol{x}_i\right)^2$$

$$\sum_{i=1}^{n} \left(\boldsymbol{u}^T \boldsymbol{x}_i\right)^2 \;=\; \sum_{i=1}^{n} \left(\boldsymbol{u}^T \boldsymbol{x}_i\right) \left(\boldsymbol{x}_i^T \boldsymbol{u}\right) \;=\; \boldsymbol{u}^T \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T \boldsymbol{u} \;=\; n \; \boldsymbol{u}^T \boldsymbol{C} \boldsymbol{u}$$

$$\boldsymbol{u}^T \boldsymbol{C} \boldsymbol{u} \;=\; \sum_{i=1}^{m} \lambda_i a_i^2$$

$$\boldsymbol{C} = \sum\nolimits_{i=1}^{m} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$$

$$\boldsymbol{u} = \sum\nolimits_{i=1}^{m} a_i \boldsymbol{u}_i \qquad \sum\nolimits_{i=1}^{m} a_i^2 = 1$$

This sum is maximal for $a_1 = 1, a_i = 0, i \neq 1$ because $\lambda_1 > \lambda_i > 0$

principal components are the direction of maximal variance orthogonal to all previous components:

$$\boldsymbol{x}_i^k \;=\; \boldsymbol{x}_i \;-\; \sum_{t=1}^{k-1} \left(\boldsymbol{u}_t^T \; \boldsymbol{x}_i\right) \boldsymbol{u}_t \qquad\qquad \boldsymbol{u}_k \;=\; \arg\max_{\|\boldsymbol{u}\|=1} \sum_{i=1}^{n} \left(\boldsymbol{u}^T \; \boldsymbol{x}_i^k\right)^2$$

inductively been proved analog to the first principal component
first l components span l-dimensional space of maximal variance

Principal Component Analysis

Is there only one PCA solution? $\boldsymbol{X} \; = \; \boldsymbol{Y}\boldsymbol{U}^T$

$\boldsymbol{U}$ is orthogonal, $\boldsymbol{Y}^T \boldsymbol{Y} \; = \; \boldsymbol{D}_m$, $\boldsymbol{D}_m$ is diagonal with sorted values

PCA is unique up to signs, if the eigenvalues of the covariance matrix are different from each other (proof: see manuscript).

At most one eigenvalue can be zero, which can be removed.

Principal Component Analysis

- first l principal components span l-dim. space of **maximal variance**

$$\sum_{i=1}^{l} \boldsymbol{u}_i^T \, \boldsymbol{C} \, \boldsymbol{u}_i \quad \text{s.t.} \quad \boldsymbol{u}_i^T \, \boldsymbol{u}_j = \delta_{ij}$$

- **projections** onto PCs have **zero means**:

$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{u}_k^T \, \boldsymbol{x}_i \; = \; \boldsymbol{u}_k^T \left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \right) \; = \; \boldsymbol{u}_k^T \boldsymbol{0} \; = \; 0$$

- projections onto PCs are mutually **uncorrelated** (**orthogonal**):

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{u}_t^T \, \boldsymbol{x}_i) \, (\boldsymbol{u}_s^T \, \boldsymbol{x}_i) \; &= \; \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{u}_t^T \, \boldsymbol{x}_i) \, (\boldsymbol{x}_i^T \, \boldsymbol{u}_s) \\ &= \; \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{u}_t^T \, (\boldsymbol{x}_i \, \boldsymbol{x}_i^T) \, \boldsymbol{u}_s \\ &= \; \boldsymbol{u}_t^T \left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \, \boldsymbol{x}_i^T \right) \boldsymbol{u}_s \\ &= \; \boldsymbol{u}_t^T \, \boldsymbol{C} \, \boldsymbol{u}_s \; = \; \lambda_s \, \boldsymbol{u}_t^T \, \boldsymbol{u}_s \; = \; 0 \end{aligned}$$

Principal Component Analysis

- the **sample variance** of the k-th projection is equal to the k-th **eigenvalue** of the sample covariance matrix:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{u}_k^T\boldsymbol{x}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}_k^T\left(\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{u}_k = \boldsymbol{u}_k^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\,\boldsymbol{x}_i^T\right)\boldsymbol{u}_k = \boldsymbol{u}_k^T\boldsymbol{C}\boldsymbol{u}_k = \lambda_k\,\boldsymbol{u}_k^T\boldsymbol{u}_k = \lambda_k$$

- **PCs** are ranked **decreasingly** according to their eigenvalues

- The first l PCs **minimize the mean-squared error**: $\hat{\boldsymbol{x}} = \sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\boldsymbol{x}$ mean-squared error is

$$\begin{aligned}
&\mathrm{E}\left(\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2\right) = \mathrm{E}\left(\boldsymbol{x}^T\boldsymbol{x} - 2\,\boldsymbol{x}^T\,\hat{\boldsymbol{x}} + \hat{\boldsymbol{x}}^T\hat{\boldsymbol{x}}\right)\\
&= \mathrm{E}\left(\mathrm{Tr}\left(\boldsymbol{x}\boldsymbol{x}^T\right) - 2\,\mathrm{Tr}\left(\sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\boldsymbol{x}\boldsymbol{x}^T\right) + \mathrm{Tr}\left(\sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\boldsymbol{x}\boldsymbol{x}^T\right)\right)\\
&= \mathrm{Tr}\left(\mathrm{E}\left(\boldsymbol{x}\boldsymbol{x}^T\right) - 2\sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\mathrm{E}\left(\boldsymbol{x}\boldsymbol{x}^T\right) + \sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\mathrm{E}\left(\boldsymbol{x}\boldsymbol{x}^T\right)\right) = \mathrm{Tr}\left(\boldsymbol{C} - \sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\boldsymbol{C}\right)\\
&= \mathrm{Tr}\left(\boldsymbol{C} - \sum_{k=1}^{l}\boldsymbol{u}_k\,\boldsymbol{u}_k^T\sum_{k=1}^{m}\lambda_k\boldsymbol{u}_k\,\boldsymbol{u}_k^T\right) = \mathrm{Tr}\left(\sum_{k=1}^{m}\lambda_k\,\boldsymbol{u}_k\,\boldsymbol{u}_k^T - \sum_{k=1}^{l}\lambda_k\boldsymbol{u}_k\,\boldsymbol{u}_k^T\right)\\
&= \mathrm{Tr}\left(\sum_{k=l+1}^{m}\lambda_k\,\boldsymbol{u}_k\,\boldsymbol{u}_k^T\right) = \sum_{k=l+1}^{m}\lambda_k\,\mathrm{Tr}\left(\boldsymbol{u}_k\,\boldsymbol{u}_k^T\right) = \sum_{k=l+1}^{m}\lambda_k\,\mathrm{Tr}\left(\boldsymbol{u}_k^T\,\boldsymbol{u}_k\right) = \sum_{k=l+1}^{m}\lambda_k
\end{aligned}$$

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

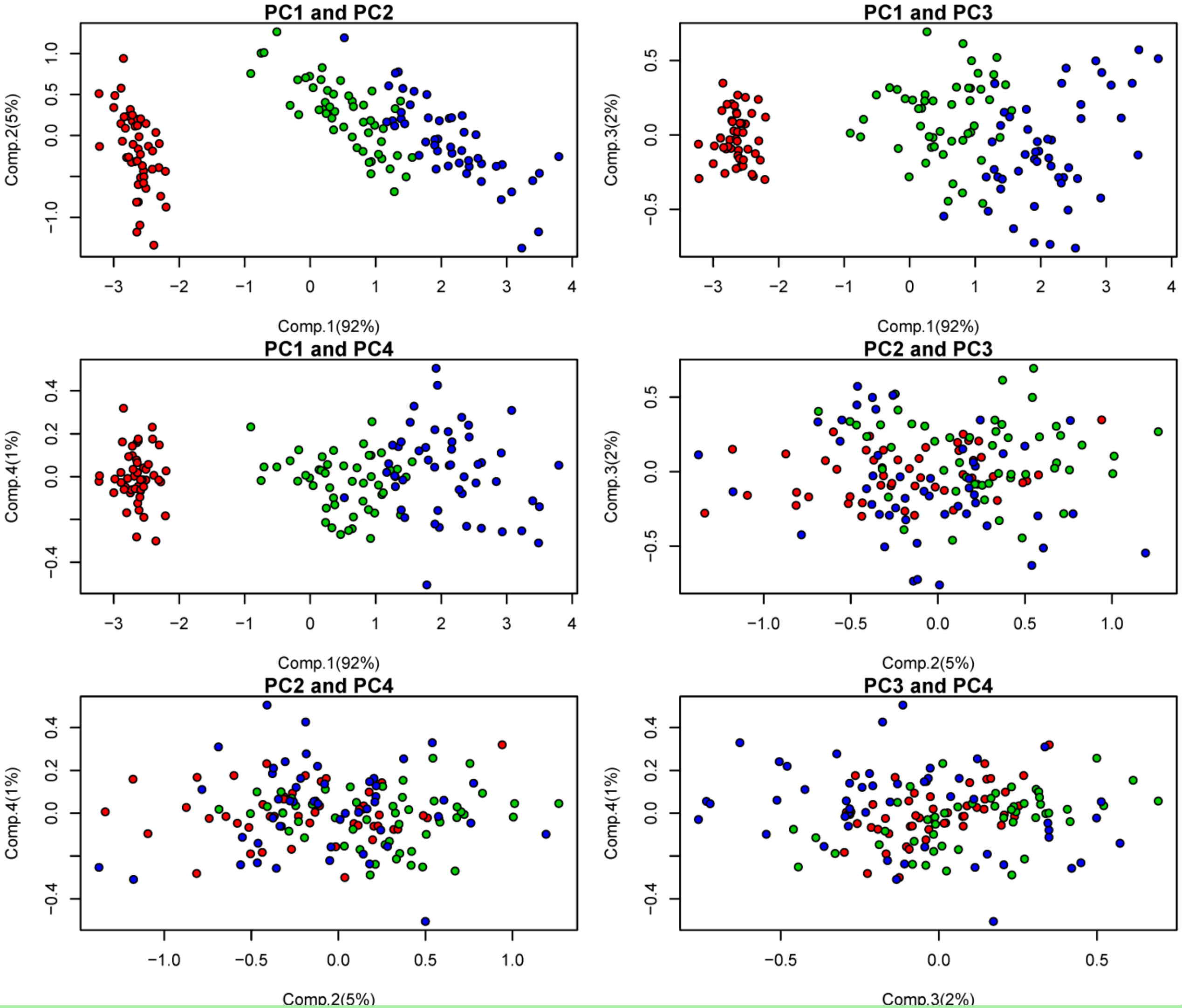
Iris Data Set

```
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000
```

the first principal component explains 92% of the variance in the data
→ features are correlated which is captured by PC1

Principal Component Analysis

Only PC1 helps to separate the iris species:



Principal Component Analysis

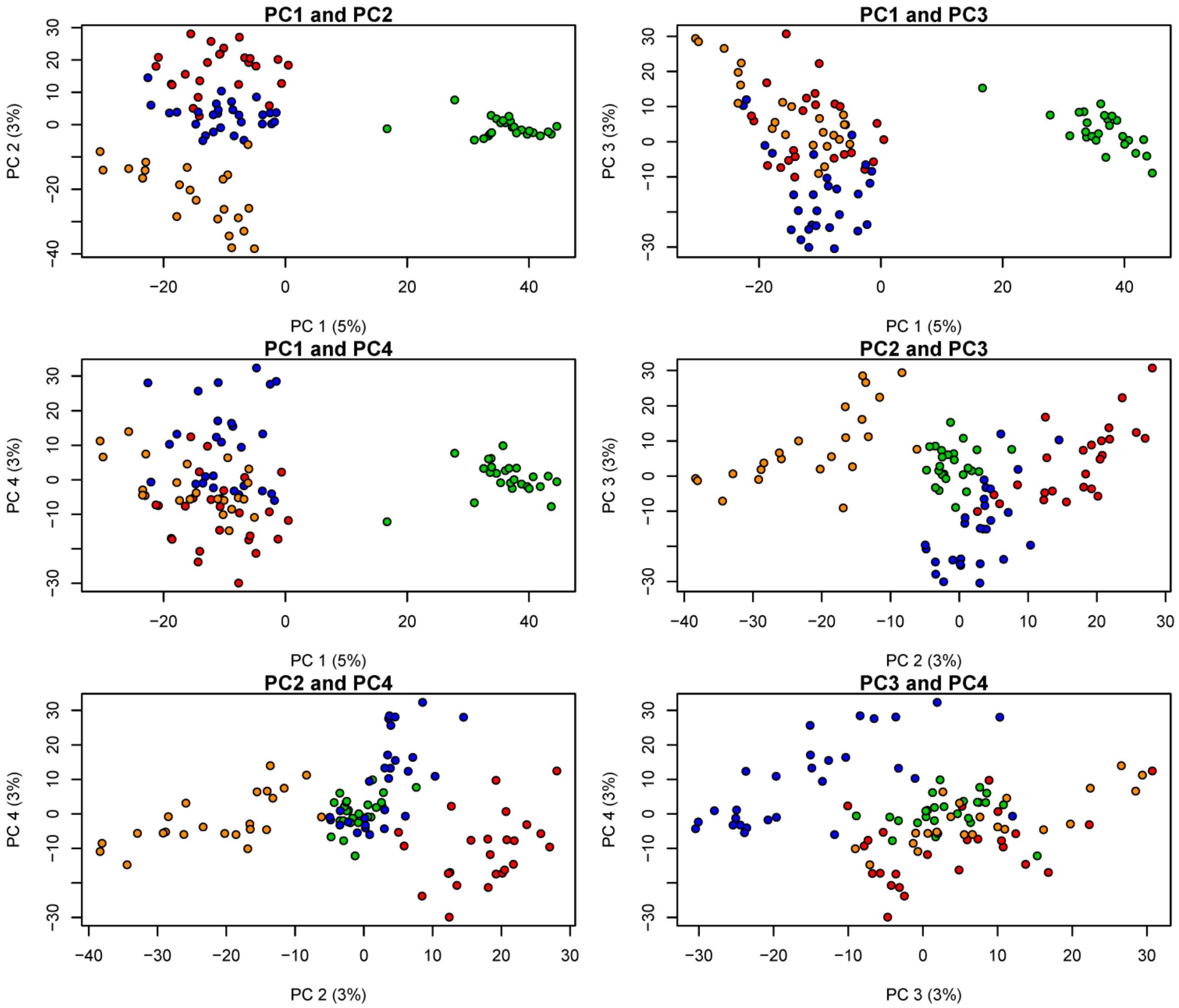
Multiple Tissue Data Set

- gene expression values microarray
- human and mouse
- 102 samples
- 5,565 genes
- different tissue types
  - breast (Br)
  - prostate (Pr)
  - lung (Lu)
  - colon (Co)

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization



PC1 separates the prostate samples (green) from the rest.

PC2 separates the colon samples (orange) but also breast samples (red).

PC3 separates some lung samples (blue).

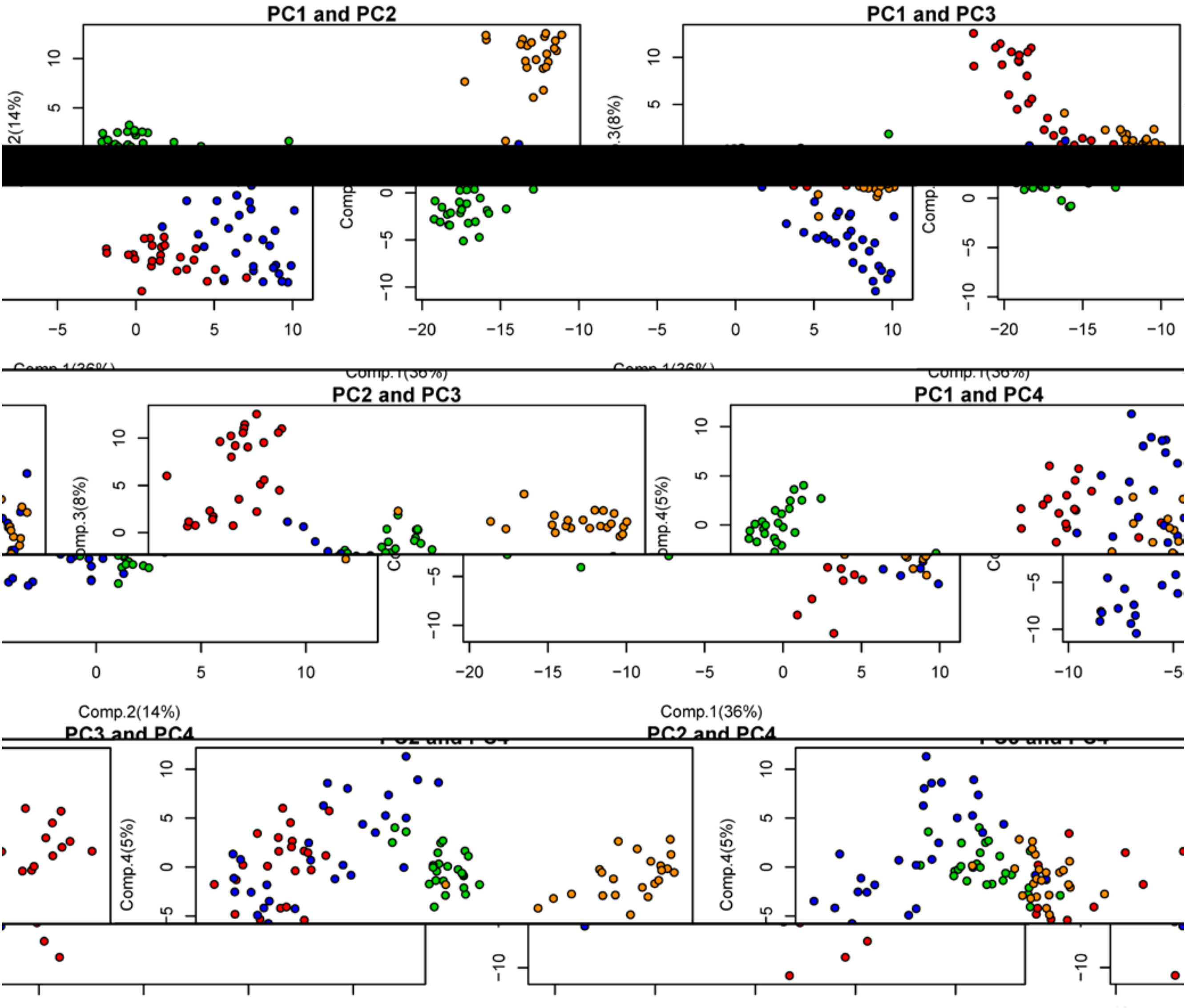
variance filtering before PCA is justified for microarray data

```
XMultiF1:   101 features
XMultiF2:   13  features
XMultiF3:   5   features
```

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization



[redacted]

101 genes with the highest variance

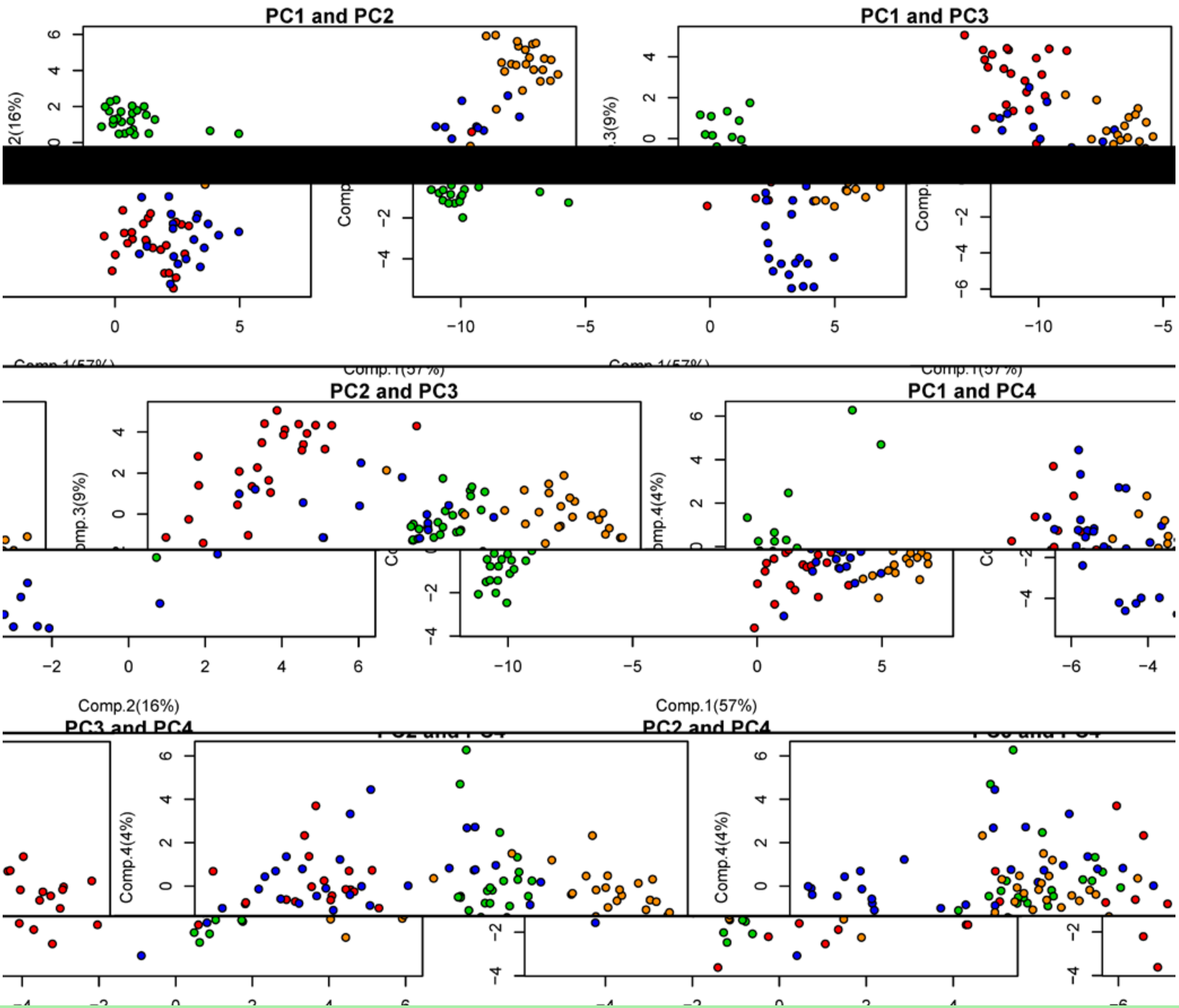
PC1 separates the prostate samples (green)

PC2 separates the colon samp. (orange)

PC3 separates the breast samples (red) and lung samp. (blue)

PCA on filtered genes performs better than PCA on all genes for tissue separat.

Principal Component Analysis



[redacted]

13 genes with largest variance

PC1 separates the prostate samples (green)

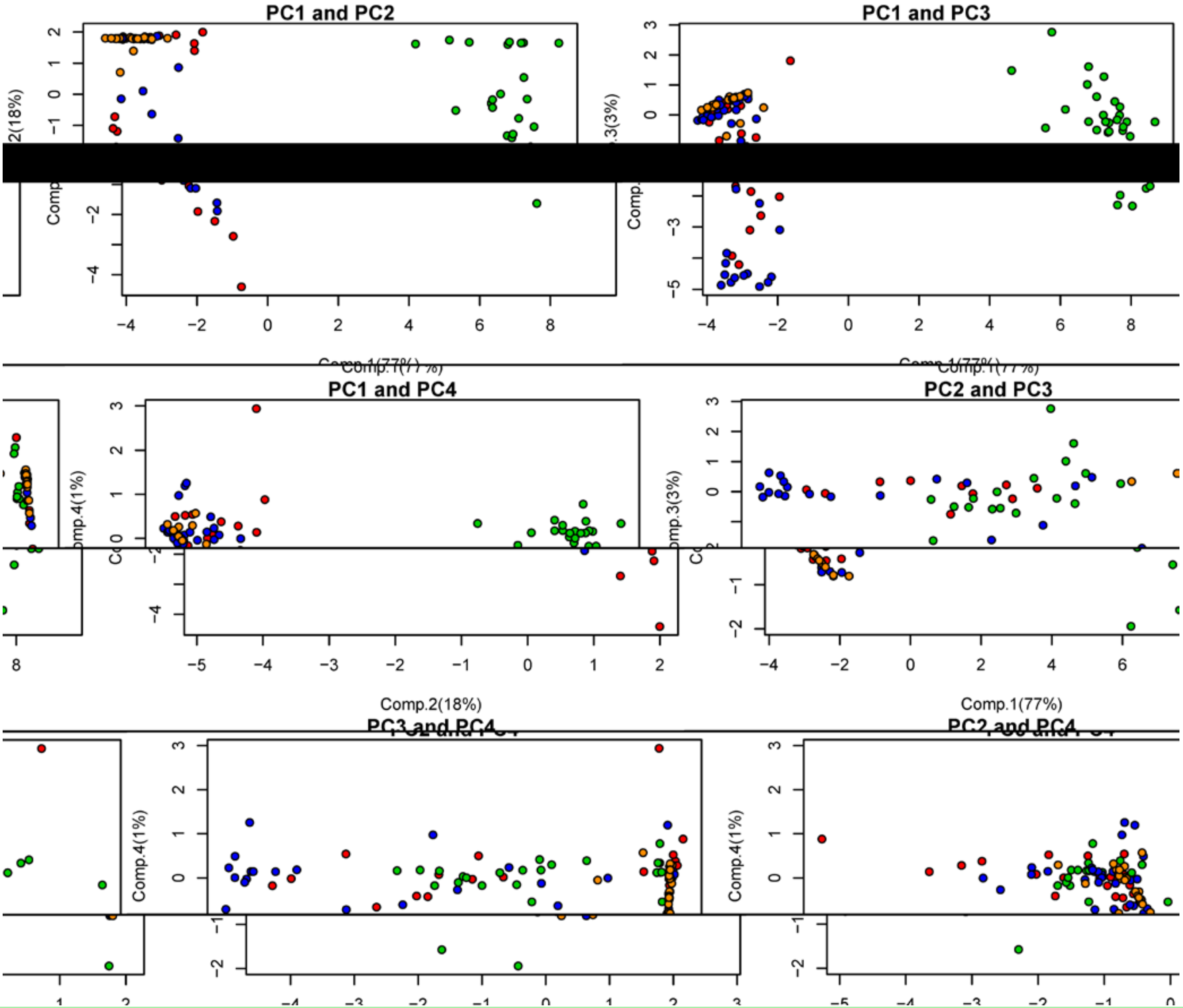
PC2 separates the colon samples (orange)

PC3 separates the breast (red) and lung (blue) samples

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization


[redacted]

5 genes with largest variance

Still PC1 separates the prostate samples (green)

However other tissues are difficult to separate

Principal Component Analysis

4 out of 5 genes are highly correlated:

```
             ACPP        KLK2         KRT5        MSMB       TRGC2
ACPP   1.000000000  0.97567890 -0.004106762  0.90707887 0.947433227
KLK2   0.975678903  1.00000000 -0.029900946  0.89265825 0.951841913
KRT5  -0.004106762 -0.02990095  1.000000000 -0.05565599 0.008877815
MSMB   0.907078869  0.89265825 -0.055655985  1.00000000 0.870922667
TRGC2  0.947433227  0.95184191  0.008877815  0.87092267 1.000000000
```

GeneCards database:

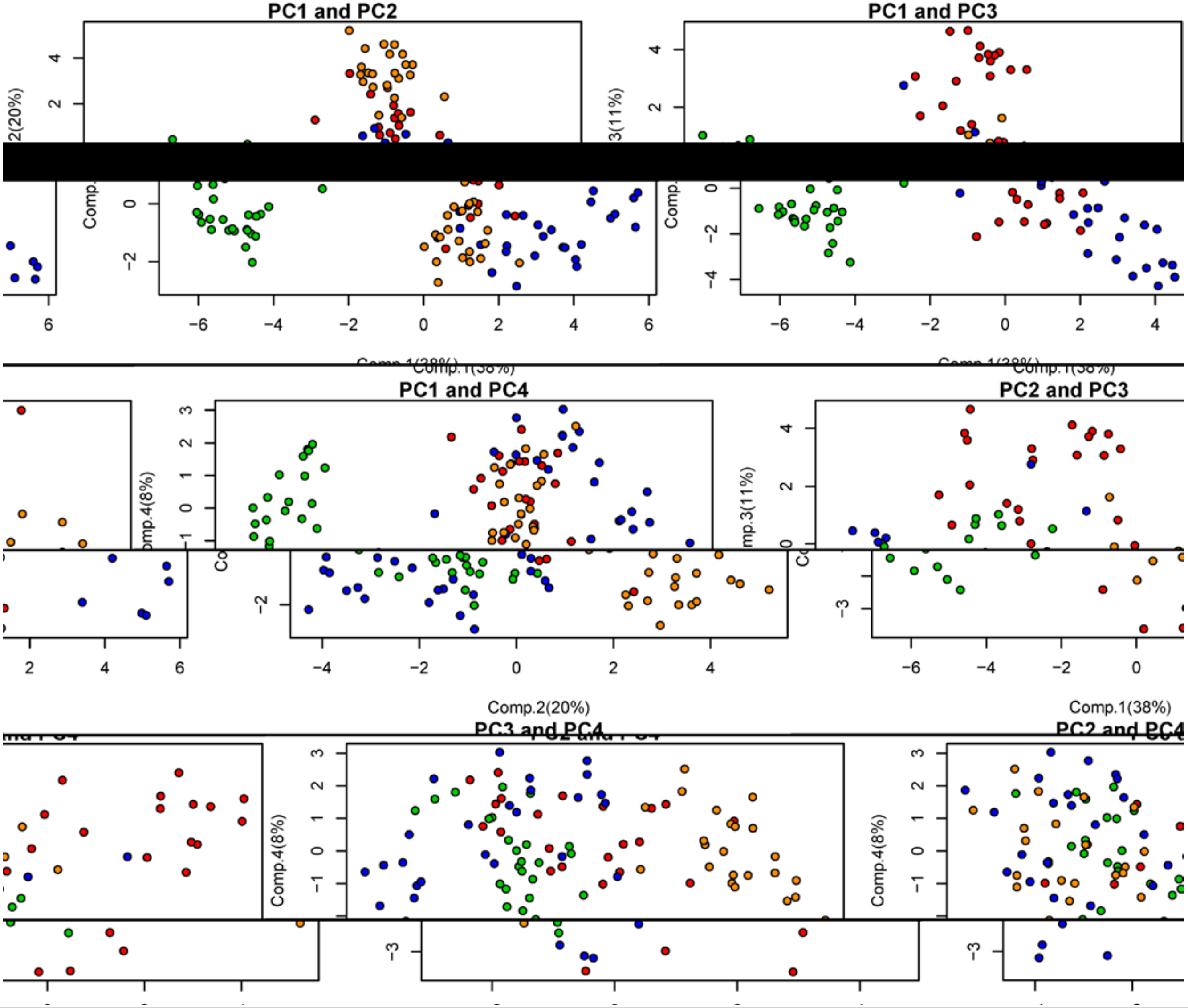
- ACPP “is synthesized under androgen regulation and is secreted by the epithelial cells of the prostate gland”
- KLK2 “is primarily expressed in prostatic tissue and is responsible for cleaving pro-prostate-specific antigen into its enzymatically active form” (KLK3 is the PSA gene)
- MSMB “is synthesized by the epithelial cells of the prostate gland and secreted into the seminal plasma”

Principal Component Analysis

genes which are not correlated to each other → clustering & prototype

```
Correlation of XMultiF4
               ABP1       ACPP      AKR1C1     ALDH1A3      ANXA8          APOD
ABP1     1.00000000 -0.1947766 -0.04224634 -0.21577195 -0.2618053 -0.3791812658
ACPP    -0.19477662  1.0000000 -0.22929893  0.88190657 -0.2978638  0.4964638048
AKR1C1  -0.04224634 -0.2292989  1.00000000 -0.07536066  0.4697886 -0.1793466620
ALDH1A3 -0.21577195  0.8819066 -0.07536066  1.00000000 -0.1727669  0.4113925823
ANXA8   -0.26180526 -0.2978638  0.46978864 -0.17276688  1.0000000 -0.1863923785
APOD    -0.37918127  0.4964638 -0.17934666  0.41139258 -0.1863924  1.0000000000
```

Principal Component Analysis


[redacted]

10 uncor-related genes

tissues are not as well separated as with maximal variance

→ highly variable genes missed

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

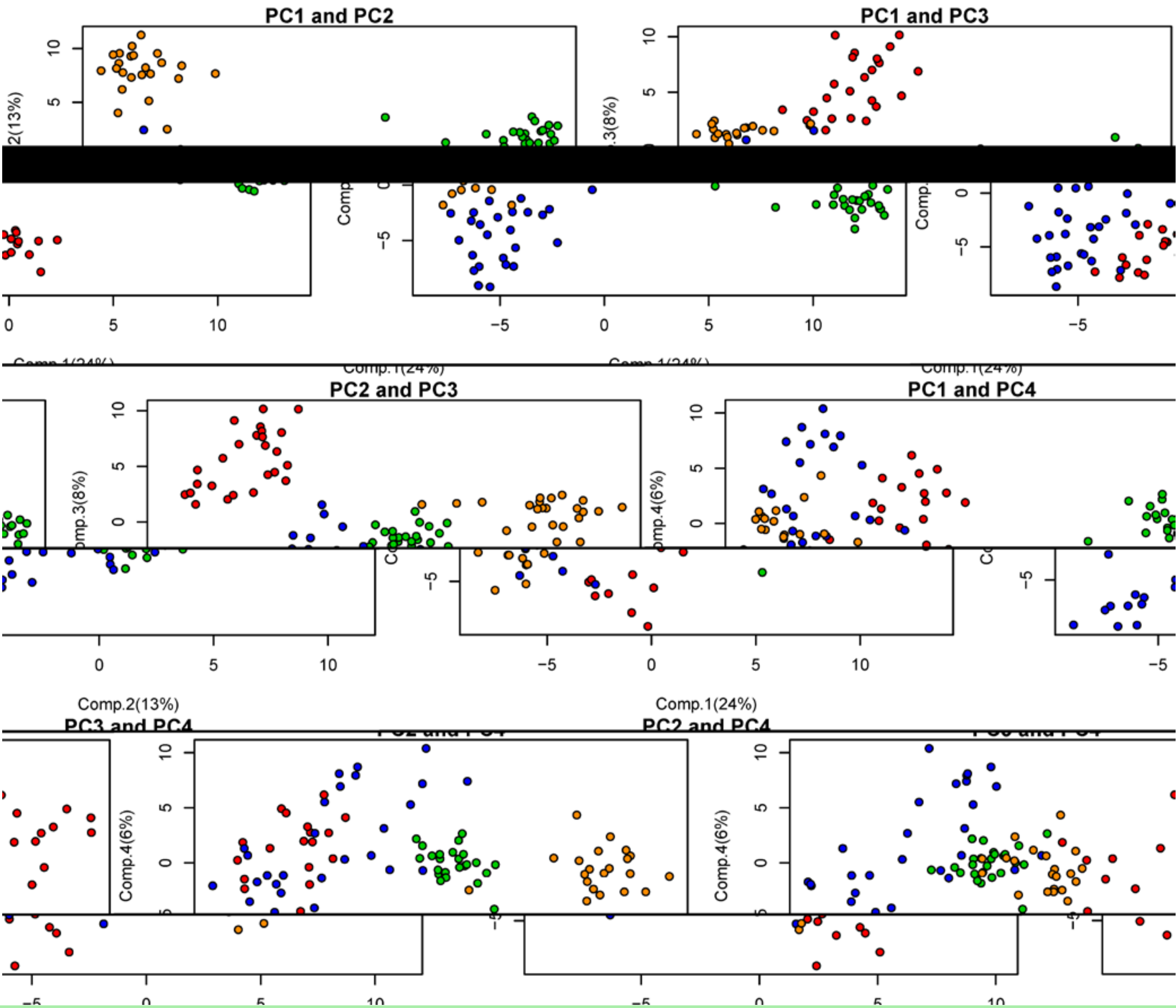
hierarchical clustering and variance maximization within one cluster:

```
1   2    3   4   5   6   7   8  9 10 11 12 13 14 15 16 17 18
682 126 1631 742 347 797 196 104 44 35  5  8 12  5 12 14  5 71
 19  20   21  22  23  24  25  26 27 28 29 30 31 32 33 34 35 36
 22   8   16  32  48  72   2  93 22 22 56  9 54  7  4  2 16 26
 37  38   39  40  41  42  43  44 45 46 47 48 49 50 51 52 53 54
  3   8   42   1   9   1   7  14  1  2  8  3  2 20  3  2  9  7
 55  56   57  58  59  60  61  62 63 64 65 66 67 68 69 70 71 72
  3   2    1   5   2   2   1   1  1  3  9  3  3  3  3  1  2  3
 73  74   75  76  77  78  79  80 81 82 83 84 85 86 87 88 89 90
  1   1    1   1   2   2   1   3  1  2  1  1  2  1  2  2  1  1
 91  92
  1   1
```

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization



[redacted]

92 genes uncorrelated but maximal variance

very similar to variance based feature selection

PC3 separates breast (red) from lung (blue) samp.

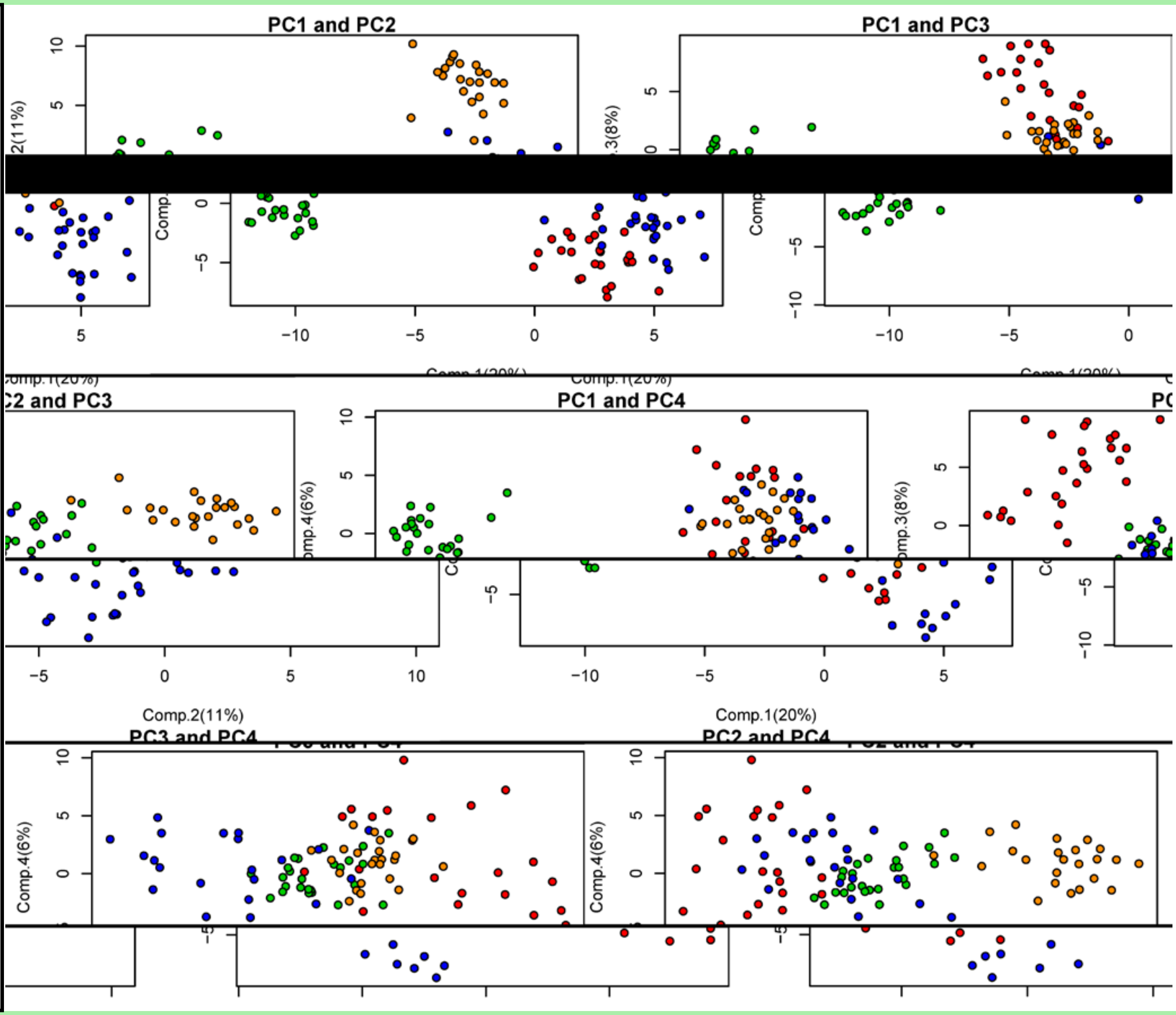
Correlation as distance measure for clustering

```
Genes 2964 and 4663 are constant!
First remove these genes
```

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization



[redacted]

clustering with correlation coefficient.

95 clusters: gene with maximal variance

very similar to variance based feature selection

PC3 separates breast (red) from lung (blue) samp.

Principal Component Analysis

Kernel Principal Component Analysis or **kernel PCA** (KPCA) extends PCA to nonlinear projections using kernel techniques

linear operations of PCA are performed in a reproducing kernel Hilbert space to which the vectors are non-linearly mapped

$$\boldsymbol{x} \ \mapsto \ \boldsymbol{\Phi}(\boldsymbol{x})$$

Assume data is centered in the feature space:

$$\sum_{i=1}^{n} \boldsymbol{\Phi}(\boldsymbol{x}_i) \ = \ \boldsymbol{0}$$

covariance matrix in feature space is given by

$$\boldsymbol{C} \ = \ \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\Phi}(\boldsymbol{x}_i) \ \boldsymbol{\Phi}^T(\boldsymbol{x}_i)$$

gram matrix:

$$\boldsymbol{K} \ = \ \sum_{i=1}^{n} \boldsymbol{\Phi}^T(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}_i)$$

We search for

$$\boldsymbol{C} \ \boldsymbol{w} \ = \ \lambda \ \boldsymbol{w}$$

Problem: we only have

$$K_{ij} \ = \ k(\boldsymbol{x}_i, \boldsymbol{x}_j) \ = \ \boldsymbol{\Phi}^T(\boldsymbol{x}_j)\boldsymbol{\Phi}(\boldsymbol{x}_i)$$

Principal Component Analysis

Principal components can be only in directions, where the data has variance (PCA maximizes the variance).

We restrict the solutions to the span of $\{\boldsymbol{\Phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\Phi}(\boldsymbol{x}_n)\}$

$$\forall_{1 \leq s \leq n} : \quad (\lambda \, \boldsymbol{w})^T \boldsymbol{\Phi}(\boldsymbol{x}_s) = \lambda \, \boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}_s) = (\boldsymbol{C} \, \boldsymbol{w})^T \boldsymbol{\Phi}(\boldsymbol{x}_s) = \boldsymbol{w}^T \boldsymbol{C} \, \boldsymbol{\Phi}(\boldsymbol{x}_s)$$

$$\boxed{\boldsymbol{C} \, \boldsymbol{w} = \lambda \, \boldsymbol{w}}$$

The solutions of these equations are unique in the span of the mapped data vectors and correspond to eigenvectors of $\boldsymbol{C}$ in the span.

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \, \boldsymbol{\Phi}(\boldsymbol{x}_i)$$

Inserting this equation gives

$$\lambda \sum_{i=1}^{n} \alpha_i \, \boldsymbol{\Phi}^T(\boldsymbol{x}_i) \boldsymbol{\Phi}(\boldsymbol{x}_s) =$$

$$\frac{1}{n} \left(\sum_{i=1}^{n} \alpha_i \sum_{j=1}^{n} \boldsymbol{\Phi}^T(\boldsymbol{x}_i) \left(\boldsymbol{\Phi}(\boldsymbol{x}_j) \, \boldsymbol{\Phi}^T(\boldsymbol{x}_j) \right) \right) \boldsymbol{\Phi}(\boldsymbol{x}_s)$$

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

Gram matrix $\boldsymbol{K}$ with $K_{ij} \;=\; \boldsymbol{\Phi}^T(\boldsymbol{x}_j)\boldsymbol{\Phi}(\boldsymbol{x}_i)$

We obtain from the last equation: $n\,\lambda\,\boldsymbol{K}\,\boldsymbol{\alpha} \;=\; \boldsymbol{K}^2\,\boldsymbol{\alpha}$

solve the eigenvalue problem $n\,\lambda\,\boldsymbol{\alpha} \;=\; \boldsymbol{K}\,\boldsymbol{\alpha}$

The eigenvectors have to have length 1:

$$1 \;=\; \boldsymbol{w}^T\boldsymbol{w} \;=\; \sum_{ij=(1,1)}^{(n,n)} \alpha_i\,\alpha_j\,\boldsymbol{\Phi}^T(\boldsymbol{x}_j)\boldsymbol{\Phi}(\boldsymbol{x}_i) \;=$$

$$\sum_{ij=(1,1)}^{(n,n)} \alpha_i\,\alpha_j\,K_{ij} \;=\; \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} \;=\; n\,\lambda\,\boldsymbol{\alpha}^T\boldsymbol{\alpha}$$

$$n\,\lambda\,\|\boldsymbol{\alpha}\|^2 \;=\; 1$$

$$\|\boldsymbol{\alpha}\| \;=\; \frac{1}{\sqrt{n\,\lambda}}$$

$$\alpha_i^{\mathrm{new}} \;=\; \frac{\alpha_i}{\|\boldsymbol{\alpha}\|\,\sqrt{n\,\lambda}}$$

Principal Component Analysis

The projection onto the eigenvectors can be computed as

$$\boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}) \; = \; \sum_{i=1}^{n} \alpha_i \; \boldsymbol{\Phi}^T(\boldsymbol{x}_i) \boldsymbol{\Phi}(\boldsymbol{x}) \; = \; \sum_{i=1}^{n} \alpha_i \; k(\boldsymbol{x}_i, \boldsymbol{x})$$

data centering in feature space:

$$\left(\boldsymbol{\Phi}(\boldsymbol{x}_i) \; - \; \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{\Phi}(\boldsymbol{x}_t) \right)^T \left(\boldsymbol{\Phi}(\boldsymbol{x}_j) \; - \; \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{\Phi}(\boldsymbol{x}_t) \right) \; =$$

$$\boldsymbol{\Phi}^T(\boldsymbol{x}_i) \boldsymbol{\Phi}(\boldsymbol{x}_j) \; - \; \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{\Phi}^T(\boldsymbol{x}_t) \boldsymbol{\Phi}(\boldsymbol{x}_j) \; - \; \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{\Phi}^T(\boldsymbol{x}_i) \boldsymbol{\Phi}(\boldsymbol{x}_t) \; +$$

$$\frac{1}{n^2} \sum_{(s,t)=(1,1)}^{(n,n)} \boldsymbol{\Phi}^T(\boldsymbol{x}_s) \boldsymbol{\Phi}(\boldsymbol{x}_t)$$

Principal Component Analysis

$$\frac{1}{n} \sum_{t=1}^{n} \mathbf{\Phi}^T(\boldsymbol{x}_t) \mathbf{\Phi}(\boldsymbol{x}_i) \;=\; \left[\frac{1}{n} \boldsymbol{K} \; \mathbf{1} \right]_i$$

$$\frac{1}{n} \sum_{t=1}^{n} \mathbf{\Phi}^T(\boldsymbol{x}_i) \mathbf{\Phi}(\boldsymbol{x}_t) \;=\; \left[\frac{1}{n} \mathbf{1}^T \boldsymbol{K} \right]_i$$

$$\frac{1}{n^2} \sum_{(s,t)=(1,1)}^{(n,n)} \mathbf{\Phi}^T(\boldsymbol{x}_s) \mathbf{\Phi}(\boldsymbol{x}_t) \;=\; \frac{1}{n^2} \; \mathbf{1}^T \boldsymbol{K} \; \mathbf{1}$$

centered kernel matrix:

$$\boldsymbol{K} \;-\; \frac{1}{n} \; \boldsymbol{K} \; \mathbf{1} \; \mathbf{1}^T \;-\; \frac{1}{n} \; \mathbf{1} \; \mathbf{1}^T \boldsymbol{K} \;+\; \frac{1}{n^2} \; \left(\mathbf{1}^T \boldsymbol{K} \; \mathbf{1} \right) \mathbf{1} \; \mathbf{1}^T$$

new data point:

$$\boldsymbol{k}(\boldsymbol{x}, .) \;=\; \left(k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_l) \right)^T$$

$$\boldsymbol{k}(\boldsymbol{x}, .) \;-\; \frac{1}{n} \; \boldsymbol{K} \; \mathbf{1} \;-\; \frac{1}{n} \; \mathbf{1}^T \boldsymbol{k}(\boldsymbol{x}, .) \; \mathbf{1} \;+\; \frac{1}{n^2} \; \left(\mathbf{1}^T \boldsymbol{K} \; \mathbf{1} \right) \mathbf{1}$$

Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

Given: gram matrix $\boldsymbol{K}$ with $K_{ij} \; = \; k(\boldsymbol{x}_i, \boldsymbol{x}_j)$

Centering

center the Gram matrix $\boldsymbol{K}$ $\quad \boldsymbol{K} \, - \, \frac{1}{n} \, \boldsymbol{K} \, \boldsymbol{1} \, \boldsymbol{1}^T \, - \, \frac{1}{n} \, \boldsymbol{1} \, \boldsymbol{1}^T \boldsymbol{K} \, + \, \frac{1}{n^2} \, \left(\boldsymbol{1}^T \boldsymbol{K} \, \boldsymbol{1}\right) \boldsymbol{1} \, \boldsymbol{1}^T$

Eigenvalues

compute eigenvectors $\boldsymbol{\alpha}$ and eigenvalues λ of the Gram matrix $\boldsymbol{K}$

Normalization

normalize eigenvectors $\boldsymbol{\alpha}$ $\quad \alpha_i^{\mathrm{new}} \; = \; \frac{\alpha_i}{\|\boldsymbol{\alpha}\| \, \sqrt{n \, \lambda}}$

Projection of a new vector

project a new vector $\boldsymbol{x}$ onto eigenvectors by center and project it

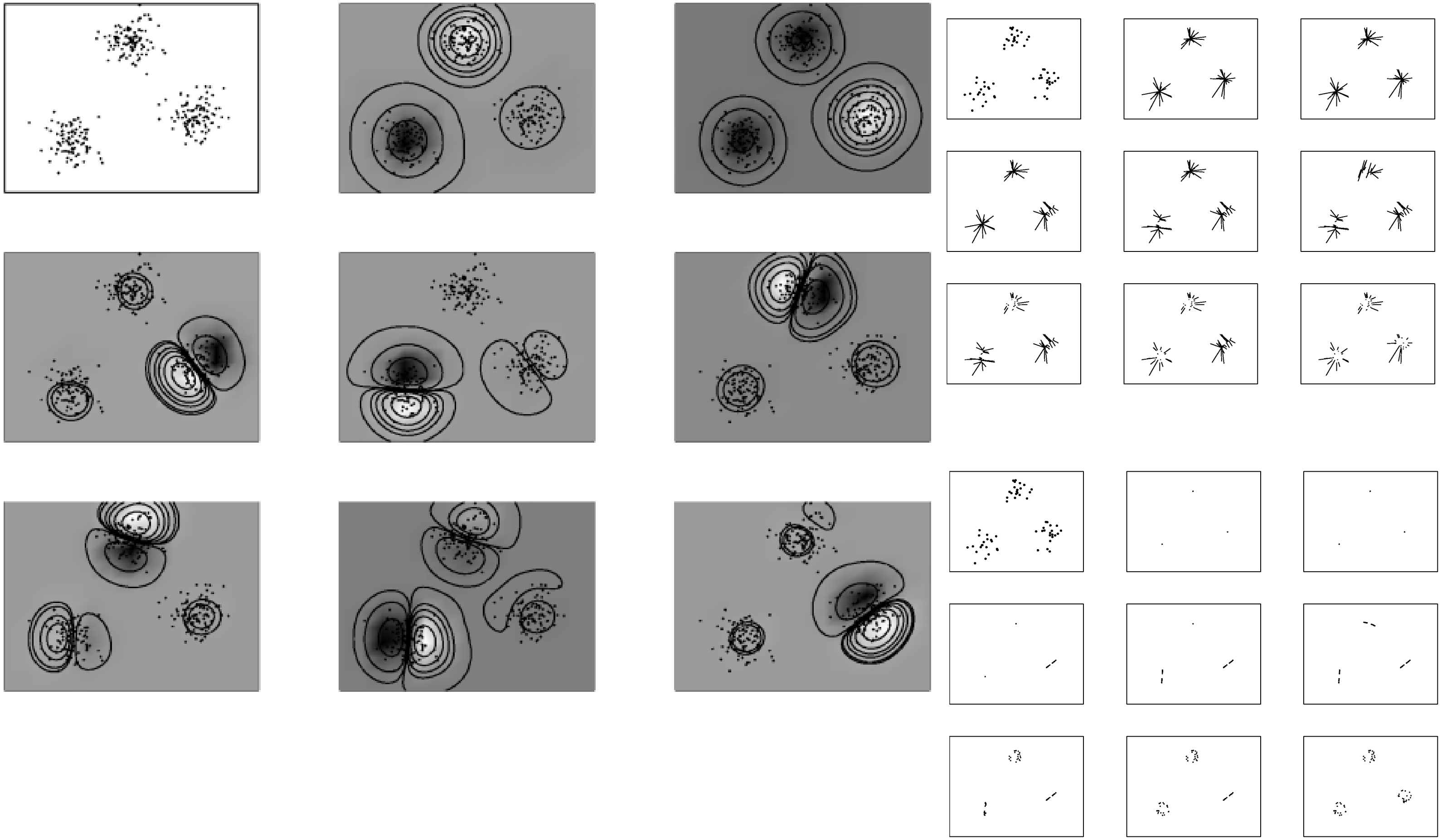
$$\boldsymbol{k}(\boldsymbol{x}, .) \; - \; \frac{1}{n} \; \boldsymbol{K} \; \boldsymbol{1} \; - \; \frac{1}{n} \; \boldsymbol{1}^T \boldsymbol{k}(\boldsymbol{x}, .) \; \boldsymbol{1} \; + \; \frac{1}{n^2} \; \left(\boldsymbol{1}^T \boldsymbol{K} \; \boldsymbol{1}\right) \boldsymbol{1}$$

$$\boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}) \; = \; \sum_{i=1}^{n} \alpha_i \; \boldsymbol{\Phi}^T(\boldsymbol{x}_i) \boldsymbol{\Phi}(\boldsymbol{x}) \; = \; \sum_{i=1}^{n} \alpha_i \; k(\boldsymbol{x}_i, \boldsymbol{x})$$

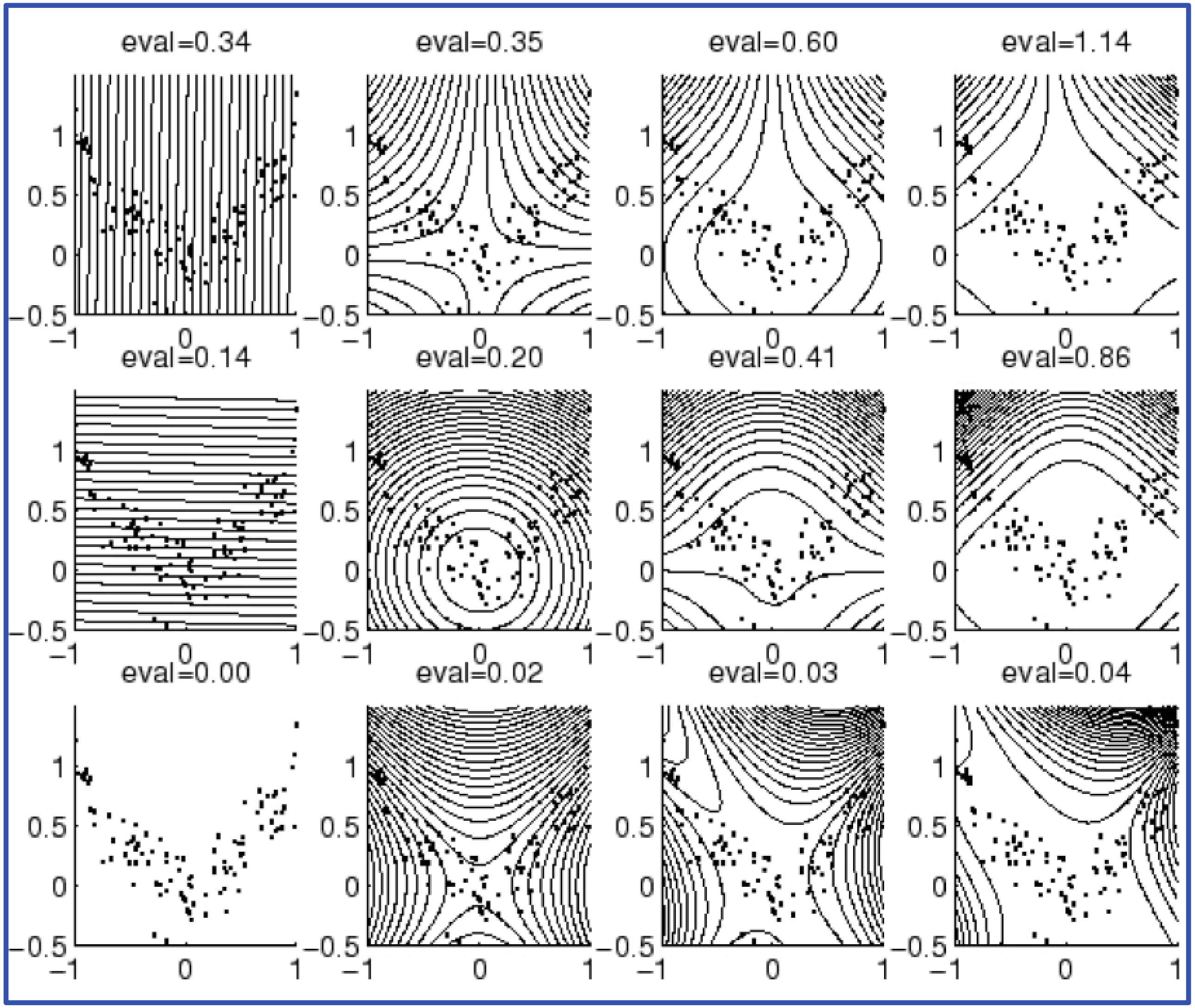
Principal Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization



Principal Component Analysis



Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

Independent component analysis (ICA): statistically independent components

ICA differs from PCA:

- ICA does not maximize the variance,
- ICA does not enforce orthogonal projection or demixing matrices,
- ICA aims at statistically independent components,
- ICA components are not ranked.

Generative: $\boldsymbol{x} \ = \ \boldsymbol{U} \ \boldsymbol{y}$

independent sources:

Descriptive: $\boldsymbol{y} \ = \ \boldsymbol{W} \ \boldsymbol{x}$

$$p(\boldsymbol{y}) \ = \ \prod_{j=1}^{l} p(y_j) \qquad l \leq m$$

$\boldsymbol{W} = \boldsymbol{U}^{-1}$

matrix decomposition: $\boldsymbol{X} \ = \ \boldsymbol{Y} \ \boldsymbol{U}^T$

$\boldsymbol{Y}^T \ \boldsymbol{Y} \ = \ \boldsymbol{D}_m$ → decorrelated but even statistically independent

$\boldsymbol{U}$ is not required to be orthogonal

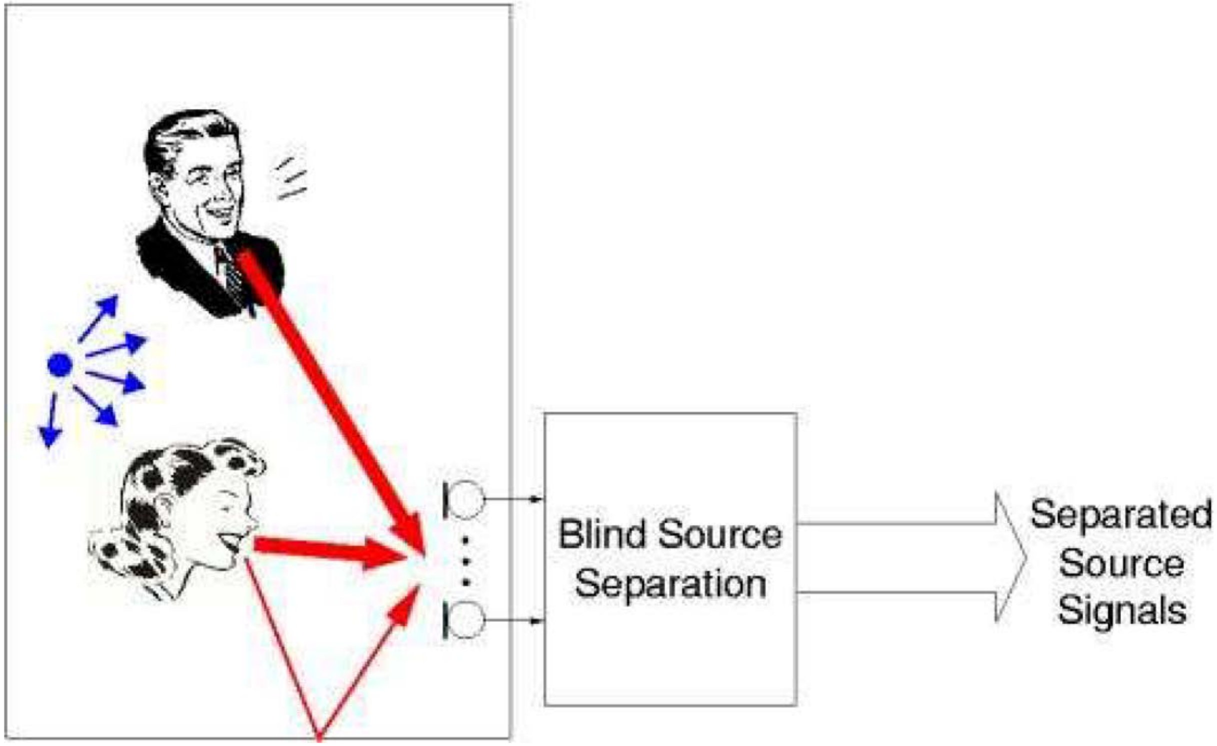
Independent Component Analysis

The outer product representation is $\boldsymbol{X} = \sum_{j=1}^{l} \boldsymbol{y}_j \boldsymbol{u}_j^T$

$\boldsymbol{u}_j$ j-th column vector of $\boldsymbol{U}$

$\boldsymbol{y}_j$ j-th column vector of $\boldsymbol{Y}$

two speakers speak independently; microphones record acoustic signals



Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

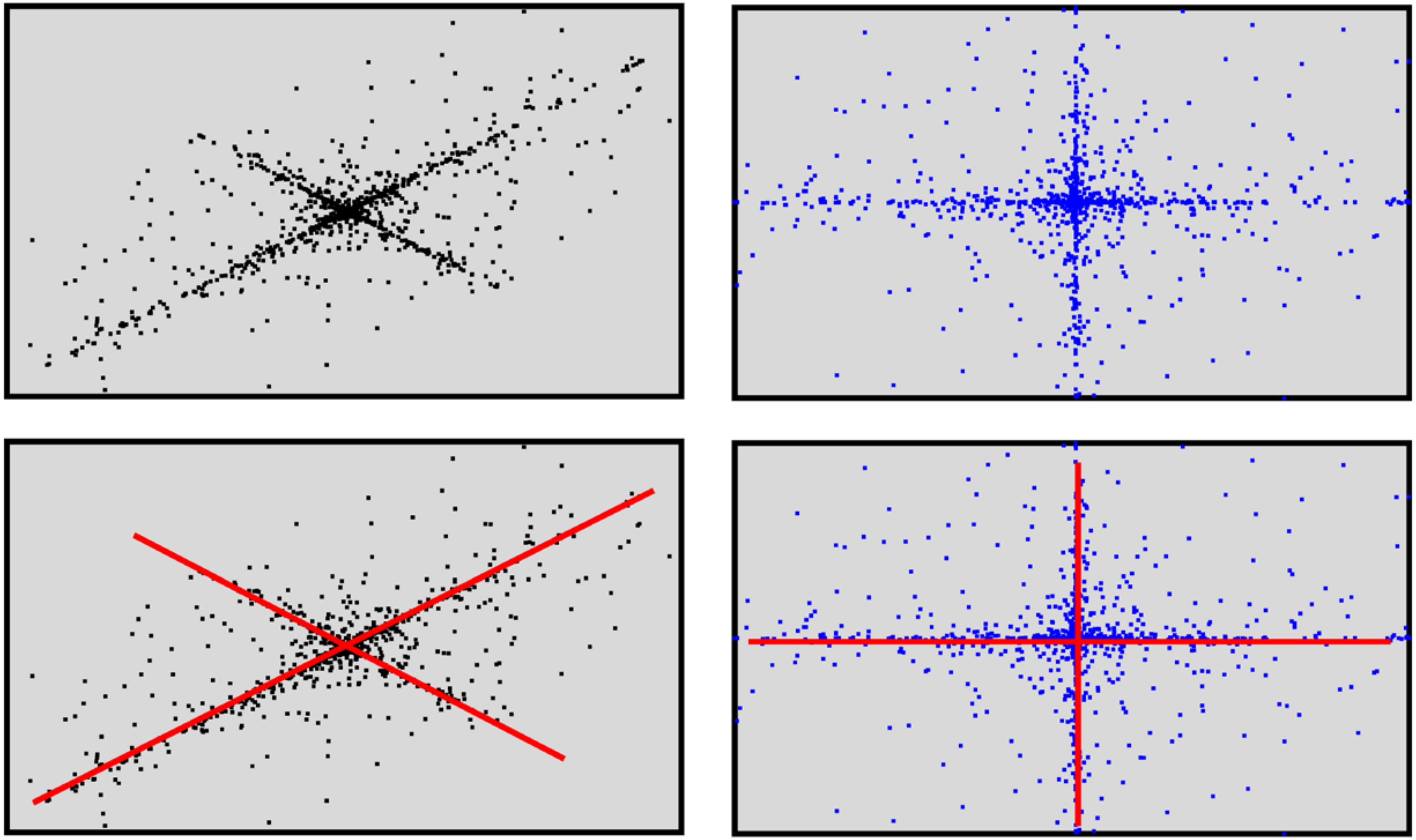
4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

Original: 🔊 🔊 🔊 🔊 🔊

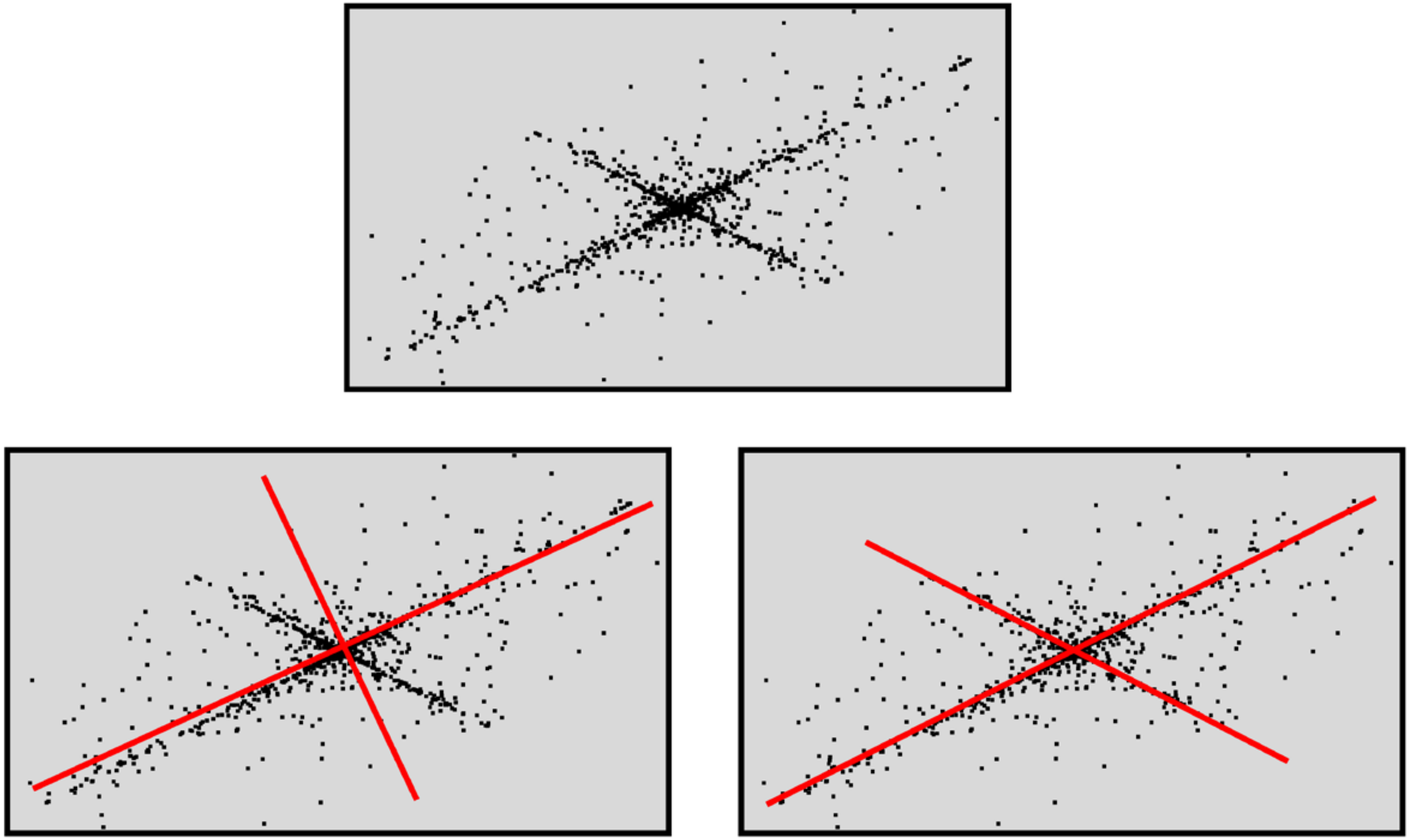
Mixtures: 🔊 🔊 🔊 🔊 🔊

Demixed by ICA: 🔊 🔊 🔊 🔊 🔊

Independent Component Analysis



ICA vs. PCA



Independent Component Analysis

ICA solution is not unique: $\boldsymbol{x} \ = \ \boldsymbol{U} \, \boldsymbol{P}^{-1} \, \boldsymbol{P} \, \boldsymbol{y}$

another solution $\boldsymbol{Y}'$ we have $\boldsymbol{Y}' \ = \ \boldsymbol{P} \, \boldsymbol{Y}$
with $\boldsymbol{Y}'^T \boldsymbol{Y}' = \boldsymbol{D}'_m$

Theorem 1 (Darmois' theorem (1953))
Define the two random variables x_1 and x_2 as

$$x_1 \ = \ \sum_{j=1}^{m} a_j y_j \quad \textit{and} \quad x_2 \ = \ \sum_{j=1}^{m} b_j y_j$$

where y_i are independent random variables. Then if x_1 and x_2 are independent, all variables y_j for which $a_j b_j \neq 0$ are Gaussian.

if two variables are independent from each other and they are a weighted sum of independent variables, then they are constructed by mutually different variables.

The exception in the theorem are Gaussian distributions.

Independent Component Analysis

$$\boldsymbol{Y}' \;=\; \boldsymbol{P}\,\boldsymbol{Y}$$

$\boldsymbol{P}$ cannot mix the statistically independent components of $\boldsymbol{y}$

⇒ $\boldsymbol{P}$ is a product of a permutation and a scaling matrix

The ICA solution is for non-Gaussian sources unique up to permutation and scaling

ICA assumptions:

- non-Gaussian sources
- $l \leq m$ at least as many observation as sources
- $\boldsymbol{U}$ has full rank l

Let $l = m$ and $\boldsymbol{U}^{-1} \in \mathbb{R}^{m \times m}$ exists → generative framework

generative framework: assumptions on the densities $p(y_i)$

approximated by super-Gaussians or unimodal distributions

Independent Component Analysis

objective for measuring independence

$$p(y_i \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_l) \; = \; p(y_i)$$

more than pairwise independence

Two criteria for independence:

- mutual information between components
- non-Gaussianity of components

Independent Component Analysis

Mutual information: entropy of a factorial code is larger than the entropy of the joint distribution

$$I(y_1, \ldots, y_l) \; = \; \sum_{j=1}^{l} H(y_j) \; - \; H(\boldsymbol{y})$$

where H denotes the entropy $H(\boldsymbol{a}) \; = \; - \int p(\boldsymbol{a}) \; \ln p(\boldsymbol{a}) \; d\boldsymbol{a}$

$$\boldsymbol{y} \; = \; \boldsymbol{W} \; \boldsymbol{x}$$

$$I(y_1, \ldots, y_m) \; = \; \sum_{j=1}^{m} H(y_j) \; - \; H(\boldsymbol{x}) \; - \; \ln |\boldsymbol{W}|$$

where $|\boldsymbol{W}|$ is the absolute value of the determinant $p(\boldsymbol{y}) \; = \; \frac{p(\boldsymbol{x})}{|\boldsymbol{W}|}$

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

Non-Gaussianity

Negentropy: $J(\boldsymbol{y}) \; = \; H(\boldsymbol{y}_{\mathrm{gauss}}) \; - \; H(\boldsymbol{y})$

where $\boldsymbol{y}_{\mathrm{gauss}}$ is a Gaussian random vector with the same covariance matrix as $\boldsymbol{y}$

maximizing the negentropy → minimizes mutual information

Gaussian: distribution with max. entropy given mean and variance
→ negentropy is closely related to entropy maximization

estimation of the negentropy is difficult

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

non-Gaussianity is measured by other parameters, e.g. fourth cummulant, the kurtosis

$$\begin{aligned}\kappa_1 &= \mathrm{E}(x) = 0 \\ \kappa_2 &= \mathrm{E}(x^2) \\ \kappa_3 &= \mathrm{E}(x^3) \\ \kappa_4 &= \mathrm{E}(x^4) - 3\left(\mathrm{E}(x^2)\right)^2\end{aligned}$$

Gaussians:
$$\kappa_3 = \kappa_4 = 0$$

positive kurtosis: super-Gaussians (smaller tails than Gaussians)
negative kurtosis: sub-Gaussians (larger tails than Gaussians)

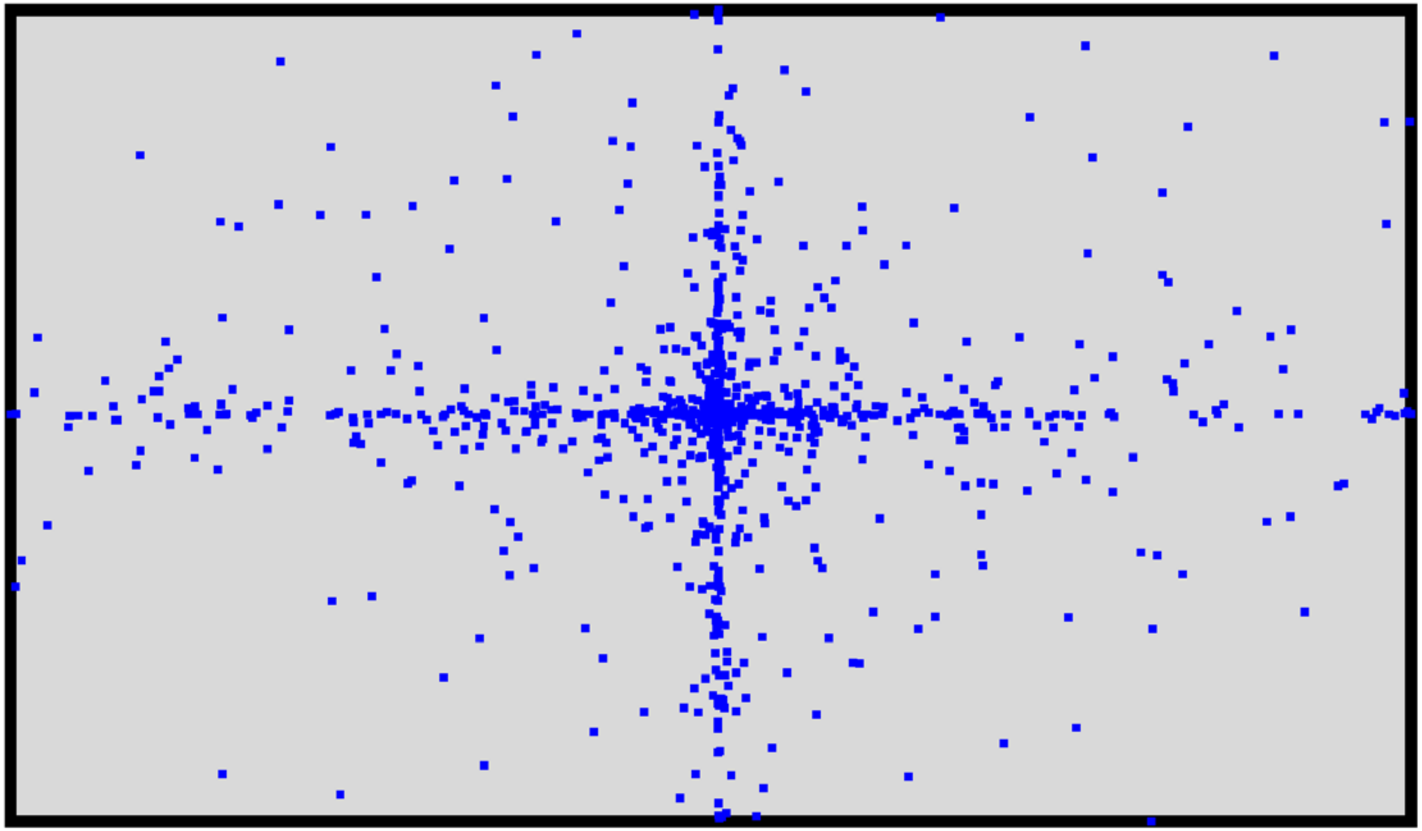
For x_1 and x_2 independent:

$$\begin{aligned}\kappa_4(x_1 + x_2) &= \kappa_4(x_1) + \kappa_4(x_2) \\ \kappa_4(\alpha\, x) &= \alpha^4\, \kappa_4(x)\end{aligned}$$

For super-Gaussians the kurtosis should be maximized because mixtures have a smaller kurtosis than the original sources.

maximizing the kurtosis = maximizing the sparseness

Sparseness: variable rarely deviates from zero; it deviates the values are relatively large compared to Gaussian with the same variance.
→ sparseness does not mean small variance



Independent Component Analysis

kurtosis: fourth moments → not robust; affected by outliers

contrast functions: measure independence of the variables:

- kurtosis: $\kappa_4(y)$
- $\frac{1}{12}\ \kappa_3^2(y)\ +\ \frac{1}{48}\ \kappa_4^2(y)$, where the variable y is normalized to zero mean and unit variance
- $|\mathrm{E}_y(G(y))\ -\ \mathrm{E}_\nu(G(\nu))|^p$, where ν is a standardized Gaussian, p=1,2, and y is normalized to zero mean and unit variance. Here G can be the kurtosis for which $G(\nu)$=0 would hold. Other choices for G are $G(x) = \log\cosh(ax)$ and $G(x) = \exp(-ax^2/2)$ with $a \geq 1$

Independent Component Analysis

whitening and rotation

independence measured by non-Gaussianity, e.g. FastICA

whitened or sphered data: $\boldsymbol{Y}^T \boldsymbol{Y} = \boldsymbol{I}$ $\quad \boldsymbol{X} = \boldsymbol{Y} \boldsymbol{U}^T$ $\quad \boldsymbol{Y} = \boldsymbol{X} \boldsymbol{U}^{-T}$

first step in ICA: sphere data because ICA is not unique up to scaling

$$\boldsymbol{I} = \boldsymbol{C}^{-1/2} \underbrace{\frac{1}{n} \boldsymbol{X}^T \boldsymbol{X}}_{\boldsymbol{C}} \boldsymbol{C}^{-1/2} = \frac{1}{n} \boldsymbol{C}^{-1/2} \boldsymbol{U} \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{U}^T \boldsymbol{C}^{-1/2} = \frac{1}{n} \boldsymbol{C}^{-1/2} \boldsymbol{U} \boldsymbol{U}^T \boldsymbol{C}^{-1/2}$$

$\hat{\boldsymbol{U}} = \frac{1}{\sqrt{n}} \boldsymbol{C}^{-1/2} \boldsymbol{U}$ is orthogonal $\quad \boldsymbol{C}^{-1/2}$ is symmetric $\quad \hat{\boldsymbol{U}}^T$ is orthogonal

$$\boldsymbol{Y} = \frac{\sqrt{n}}{\sqrt{n}} \boldsymbol{X} \boldsymbol{C}^{-1/2} \boldsymbol{C}^{1/2} \boldsymbol{U}^{-T} = \frac{1}{\sqrt{n}} \boldsymbol{X} \boldsymbol{C}^{-1/2} \hat{\boldsymbol{U}}^{-T} = \frac{1}{\sqrt{n}} \boldsymbol{X} \boldsymbol{C}^{-1/2} \hat{\boldsymbol{U}}$$

First whitening $\boldsymbol{X} \boldsymbol{C}^{-1/2}$ then determine orthogonal $\hat{\boldsymbol{U}}$ (rotation)

Objective of rotation is super-Gaussian (kurtosis) or sparseness

Independent Component Analysis

INFOMAX minimizes the mutual information between components

the entropy $H(\boldsymbol{g}(\boldsymbol{y}))$ is maximized, where $\boldsymbol{g}(\boldsymbol{y}) = (g(y_1), g(y_2), \ldots, g(y_l))$

$$\boldsymbol{y} \ = \ \boldsymbol{W} \ \boldsymbol{x}$$

Maximal entropy: $I(g(y_1), \ldots, g(y_l)) \ = \ \sum_{j=1}^{l} H(g(y_j)) \ - \ H(\boldsymbol{g}(\boldsymbol{y})) \ = \ 0$

and the components $(g(y_1), \ldots, g(y_l))$ are statistically independent

common choice: $g(y_i) \ = \ \tanh(y_i)$

$$p(\boldsymbol{g}(\boldsymbol{y})) \ = \ p(\boldsymbol{x}) \left| \frac{\partial \boldsymbol{g}(\boldsymbol{y})}{\partial \boldsymbol{y}} \ \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \right|^{-1} \ = \ p(\boldsymbol{x}) \left| \frac{\partial \boldsymbol{g}(\boldsymbol{y})}{\partial \boldsymbol{y}} \boldsymbol{W} \right|^{-1}$$

$$\left| \frac{\partial \boldsymbol{g}(\boldsymbol{y})}{\partial \boldsymbol{y}} \boldsymbol{W} \right| \ = \ \left| \prod_{j=1}^{l} g'(y_j) \right| \ |\boldsymbol{W}|$$

generative framework: $g'(y_i) \ = \ p(y_i)$

g represents a (transformed) probability function

Independent Component Analysis

Entropy: $H(\boldsymbol{g}(\boldsymbol{y})) \; = \; \mathrm{E}\left(- \; \ln p(\boldsymbol{g}(\boldsymbol{y}))\right) \; = \; H(\boldsymbol{x}) \; + \; \mathrm{E}\left(\sum_{j=1}^{l} |\ln g'(y_j)| \right) \; + \; \ln |\boldsymbol{W}|$

$$\boldsymbol{y}_i \; = \; \boldsymbol{W} \; \boldsymbol{x}_i \qquad \approx \; H(\boldsymbol{x}) \; + \; \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{l} |\ln g'(y_{ij})| \; + \; \ln |\boldsymbol{W}|$$

`tanh`:

$g(y_j) \; = \; \tanh(y_j)$ gives $\dfrac{\partial}{\partial \boldsymbol{w}_j} \ln g'(y_j) \; = \; \dfrac{g''(y_j)}{g'(y_j)} \; \boldsymbol{x}^T \; = \; - \; 2 \; g(y_j) \; \boldsymbol{x}^T$

sigmoid:

$g(y_j) \; = \; \dfrac{1}{1 \; + \; e^{-y_j}}$ gives $\dfrac{\partial}{\partial \boldsymbol{w}_j} \ln g'(y_j) \; = \; (1 \; - \; 2 \; g(y_j)) \; \boldsymbol{x}^T$

$$\frac{\partial}{\partial \boldsymbol{W}} \ln |\boldsymbol{W}| \; = \; \left(\boldsymbol{W}^T\right)^{-1}$$

Independent Component Analysis

`tanh`:

$$\frac{\partial}{\partial \boldsymbol{W}} H(\boldsymbol{g}(\boldsymbol{y})) = \left(\boldsymbol{W}^T\right)^{-1} - 2\, \boldsymbol{g}(\boldsymbol{y})\, \boldsymbol{x}^T$$

sigmoid:

$$\frac{\partial}{\partial \boldsymbol{W}} H(\boldsymbol{g}(\boldsymbol{y})) = \left(\boldsymbol{W}^T\right)^{-1} + (\boldsymbol{1} - 2\, \boldsymbol{g}(\boldsymbol{y}))\, \boldsymbol{x}^T$$

Update rules:

`tanh`:

$$\Delta \boldsymbol{W} \propto \left(\boldsymbol{W}^T\right)^{-1} - 2\, \boldsymbol{g}(\boldsymbol{y})\, \boldsymbol{x}^T$$

sigmoid:

$$\Delta \boldsymbol{W} \propto \left(\boldsymbol{W}^T\right)^{-1} + (\boldsymbol{1} - 2\, \boldsymbol{g}(\boldsymbol{y}))\, \boldsymbol{x}^T$$

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

Natural Gradient multiplied with $\boldsymbol{W}^T\boldsymbol{W}$

`tanh`:

$$\Delta \boldsymbol{W} \;\propto\; \left(\boldsymbol{I} \;-\; 2\, \boldsymbol{g}(\boldsymbol{y})\, \boldsymbol{y}^T\right)\; \boldsymbol{W}$$

sigmoid:

$$\Delta \boldsymbol{W} \;\propto\; \left(\boldsymbol{I} \;+\; (1 \;-\; 2\, \boldsymbol{g}(\boldsymbol{y}))\; \boldsymbol{y}^T\right)\; \boldsymbol{W}$$

INFOMAX is equivalent to a generative approach using maximum likelihood

EASI Equivariant Adaptive Separation via Independence (EASI)

Update rule:

$$\Delta \boldsymbol{W} \ \propto \ \left(\boldsymbol{I} \ - \ \boldsymbol{y} \ \boldsymbol{y}^T \ - \ \boldsymbol{g}(\boldsymbol{y})\boldsymbol{y}^T \ + \ \boldsymbol{y} \ \boldsymbol{g}^T(\boldsymbol{y}) \right) \ \boldsymbol{W}$$

nonlinear functions g are the same contrast functions as for INFOMAX

FastICA: probably the most popular ICA algorithm

- whitening and rotation algorithm
- FastICA is a fixed point algorithm (like Oja's rule for PCA)
- kurtosis maximization but extended to other contrast functions

$$\boldsymbol{w}^{\mathrm{new}} = \mathrm{E}\left(\boldsymbol{x}\ g(\boldsymbol{w}^T\ \boldsymbol{x})\right) - \mathrm{E}\left(g'(\boldsymbol{w}^T\ \boldsymbol{x})\right)\ \boldsymbol{w}$$

contrast function: g with derivative g'

FastICA has been extended to extract multiple components

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

ICA extensions:

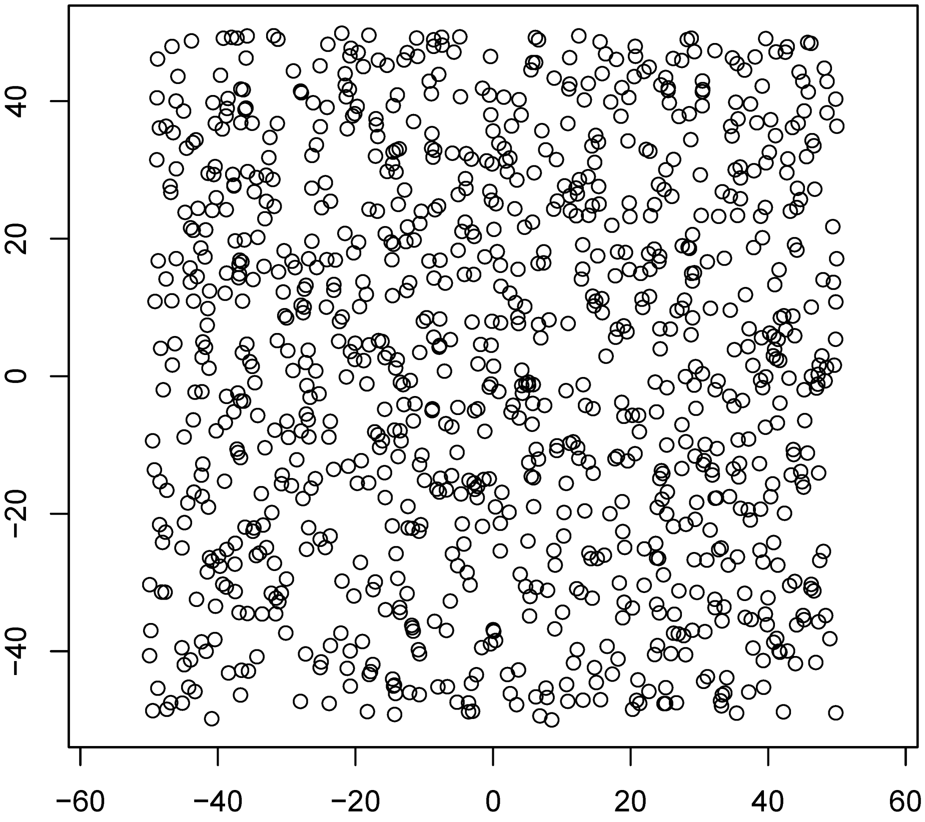
- generative approach
- sub-Gaussian distributions with specific assumptions
- non-linear extensions which are often not unique
- overcomplete basis more sources than observations $l > m$
- fewer sources than observations $l < m$

ICA vs. PCA:

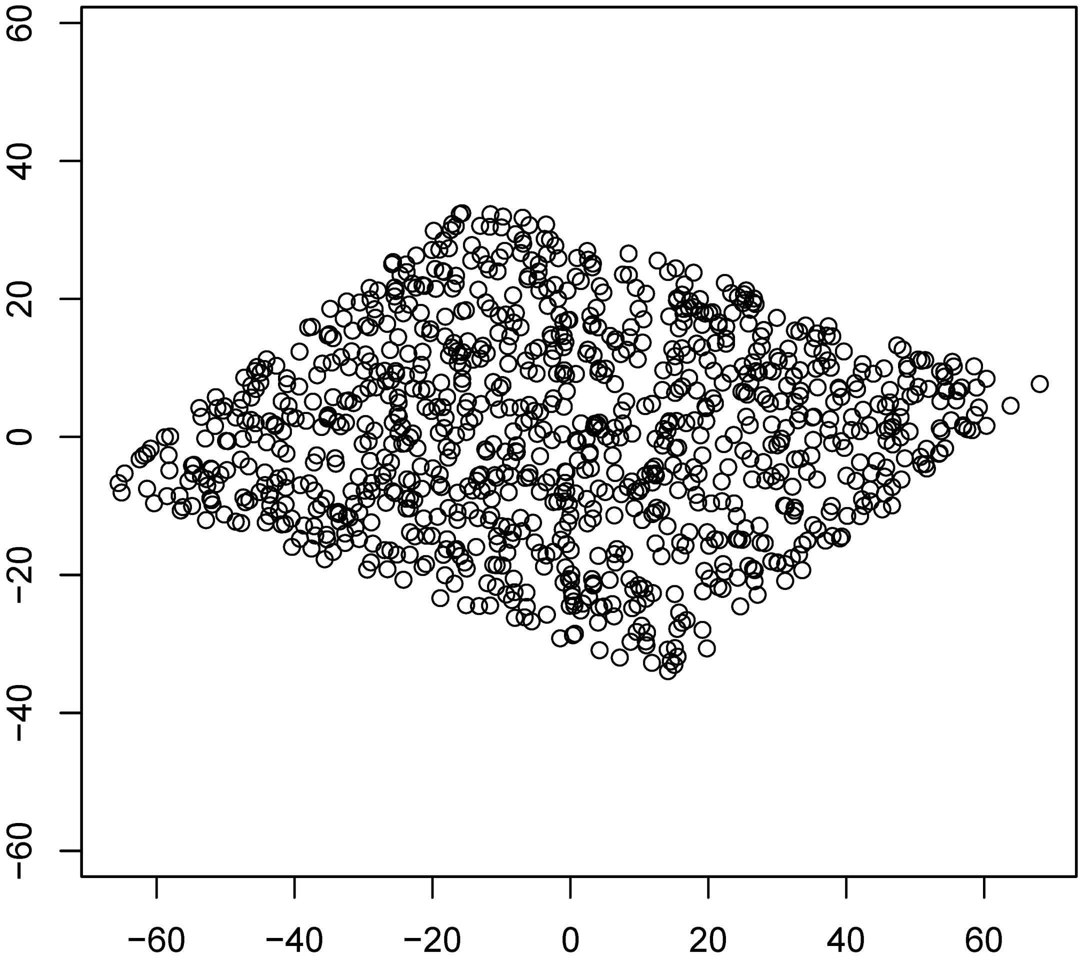
independent component analysis	principal component analysis
causes of the data	geometrical abstractions
statistical independent	decorrelated (orthogonal)
explain super-Gaussians	explain all variance
scale invariant	not scale invariant
unique up to scale and permutation	unique
assume super-Gauss	no assumptions
no ranking	ranked by eigenvalues

whitening and rotation for artificial data

1,000 data points drawn from uniform distributions:

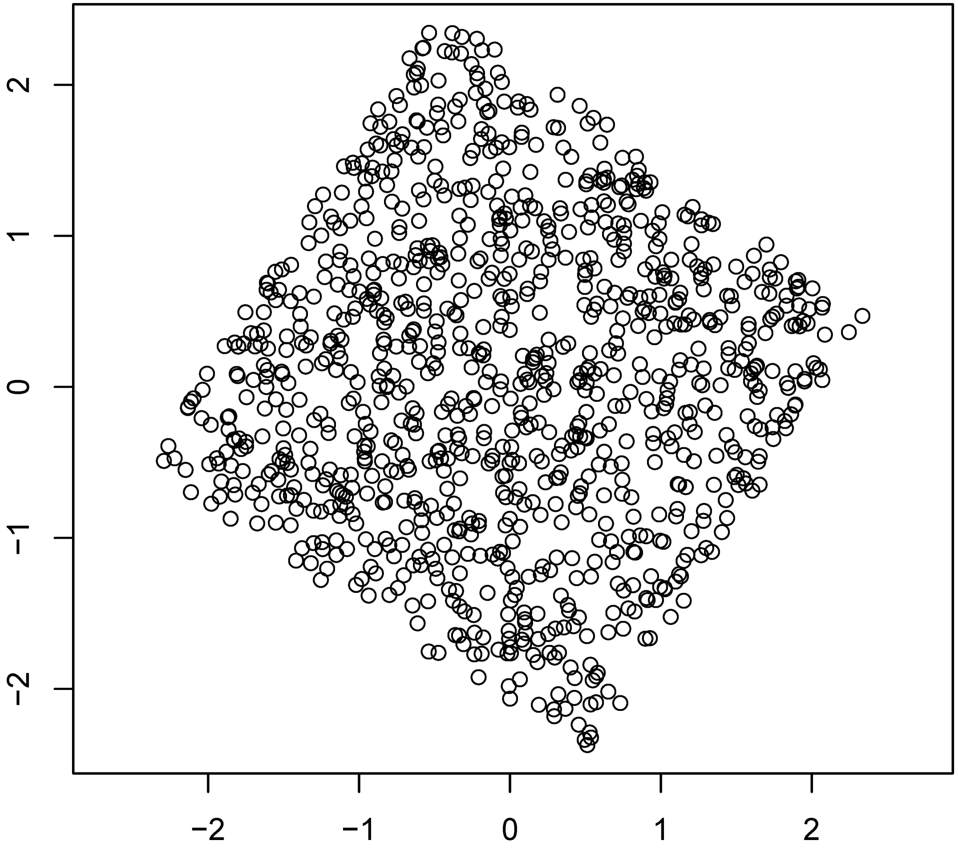


Mixing gives dependent components: observations



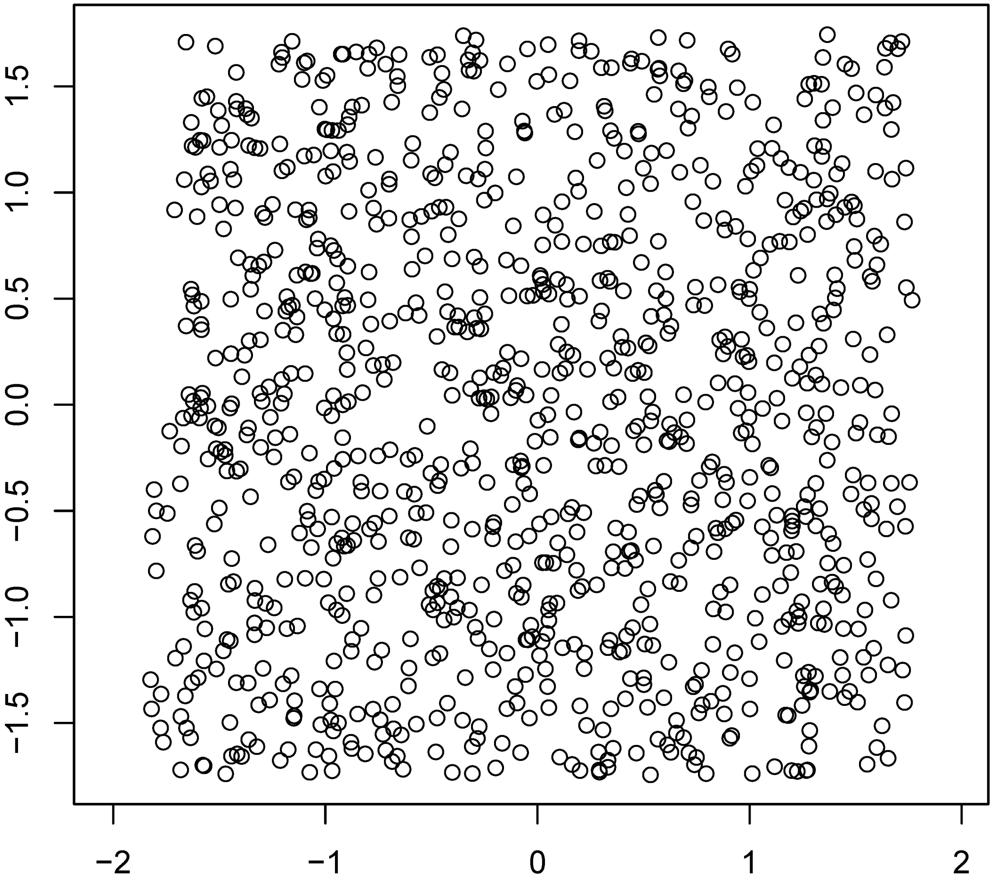
Independent Component Analysis

Whitening of the mixed data:

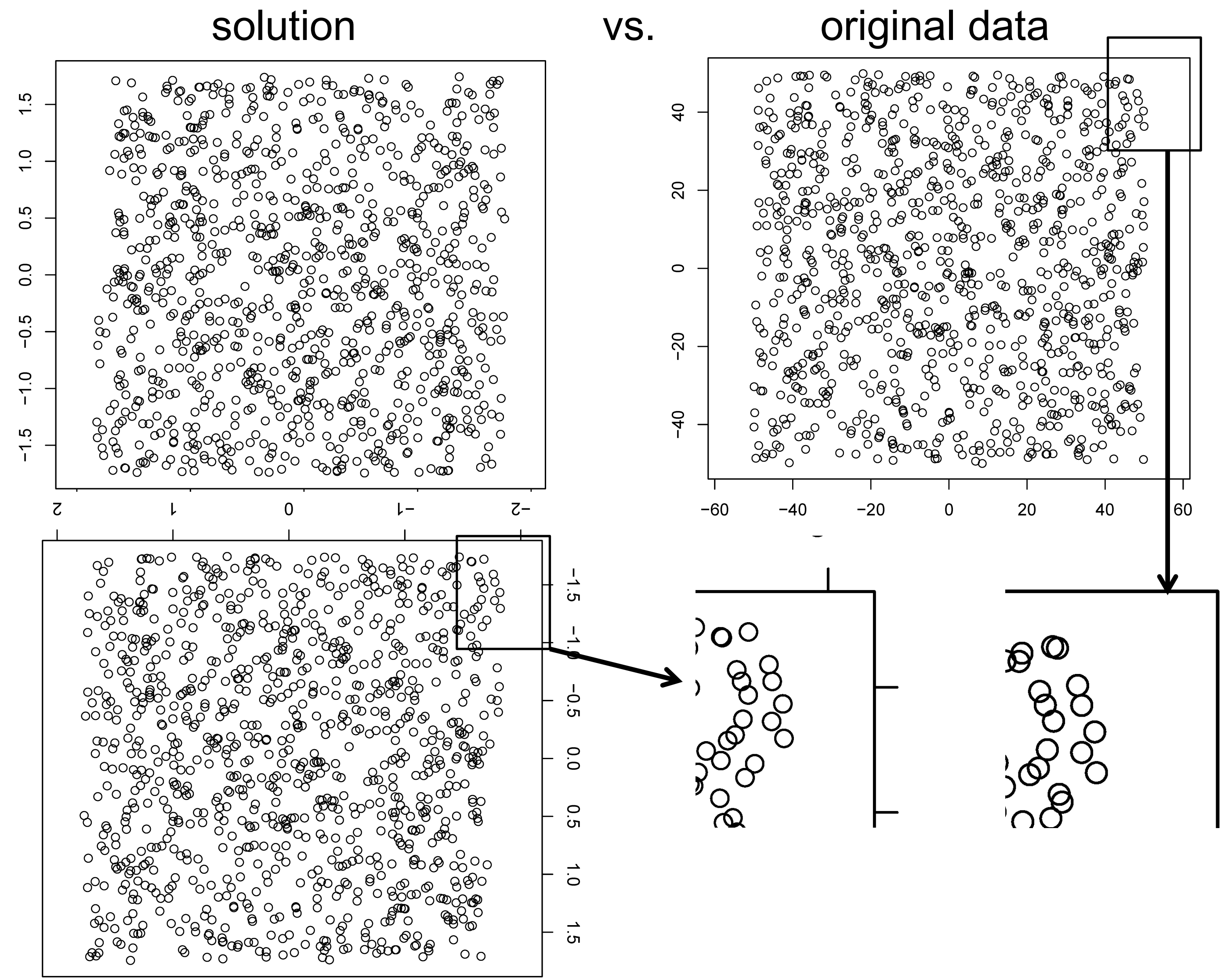


Independent Component Analysis

Rotation of the whitened data:

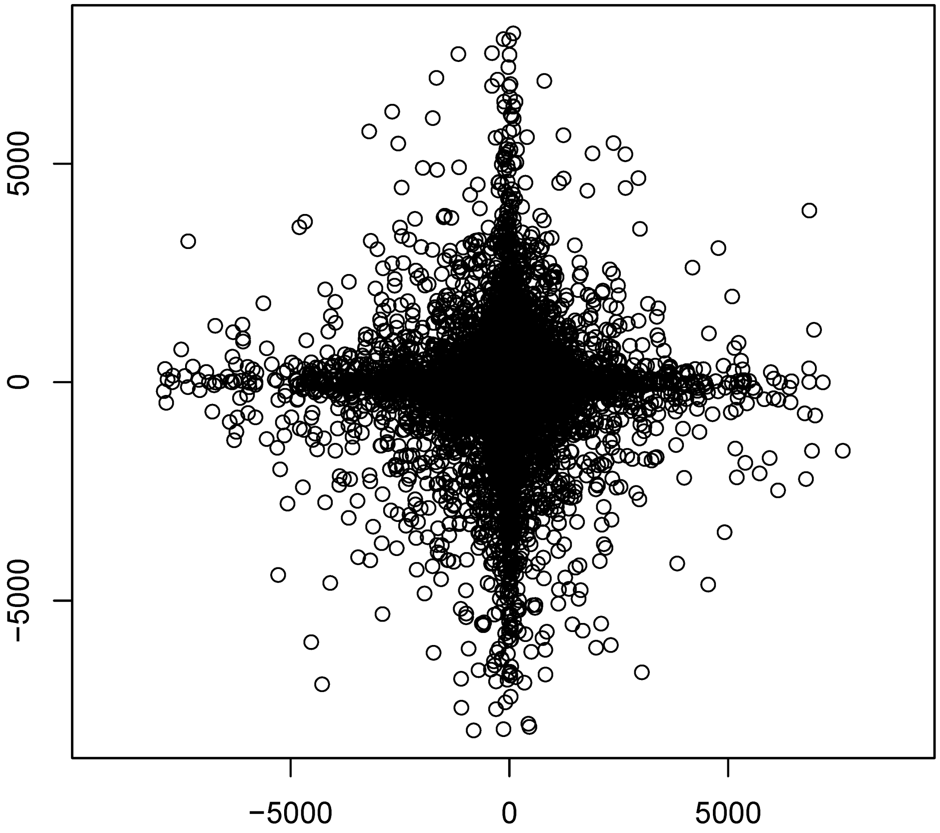


Independent Component Analysis



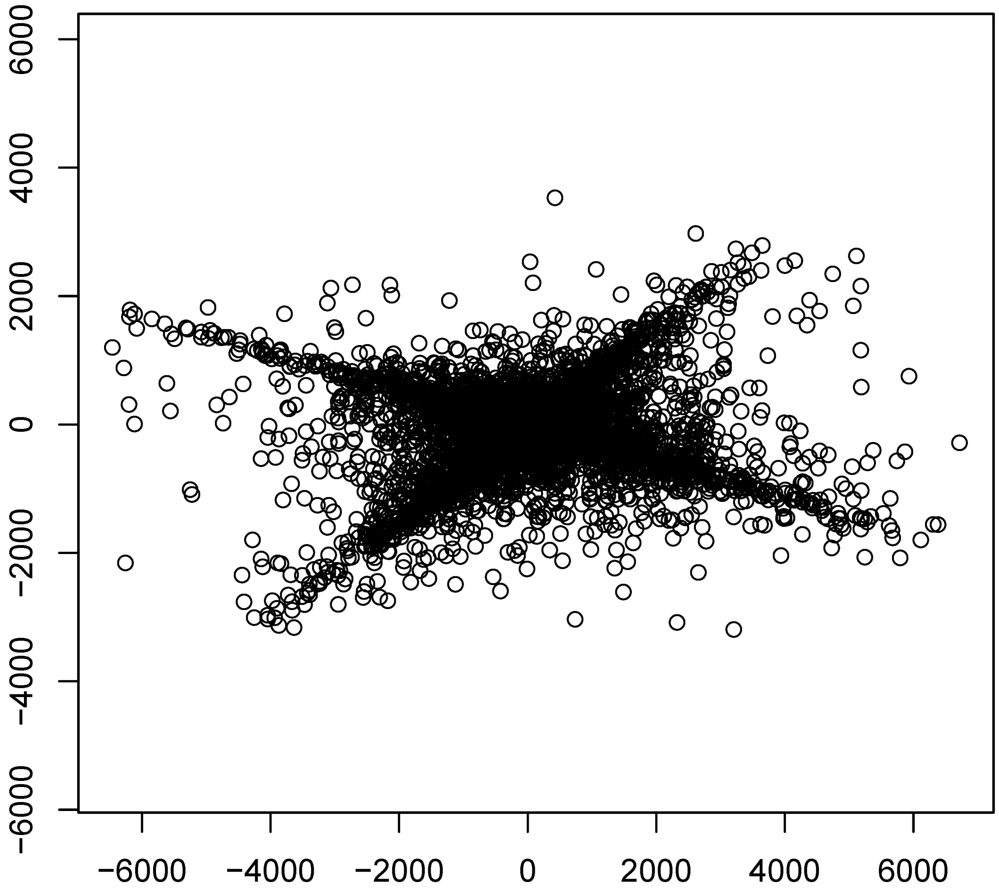
Independent Component Analysis

Toy example with super-Gaussians:



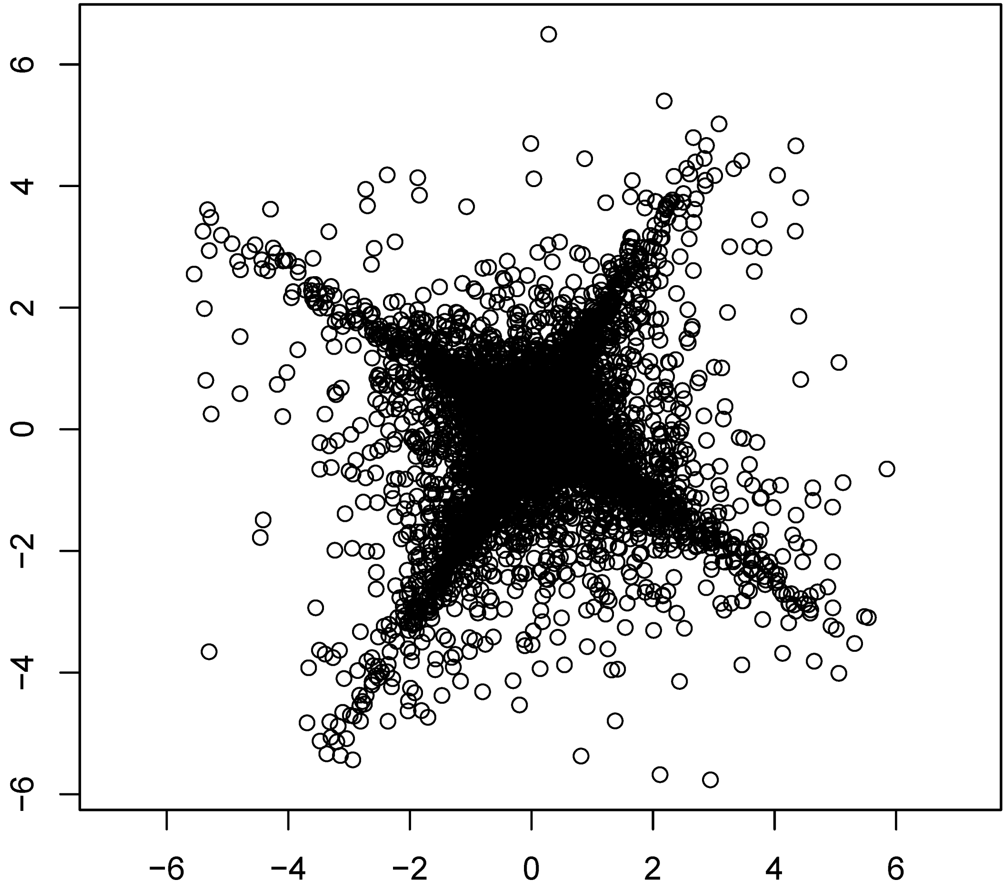
Independent Component Analysis

Mixing:



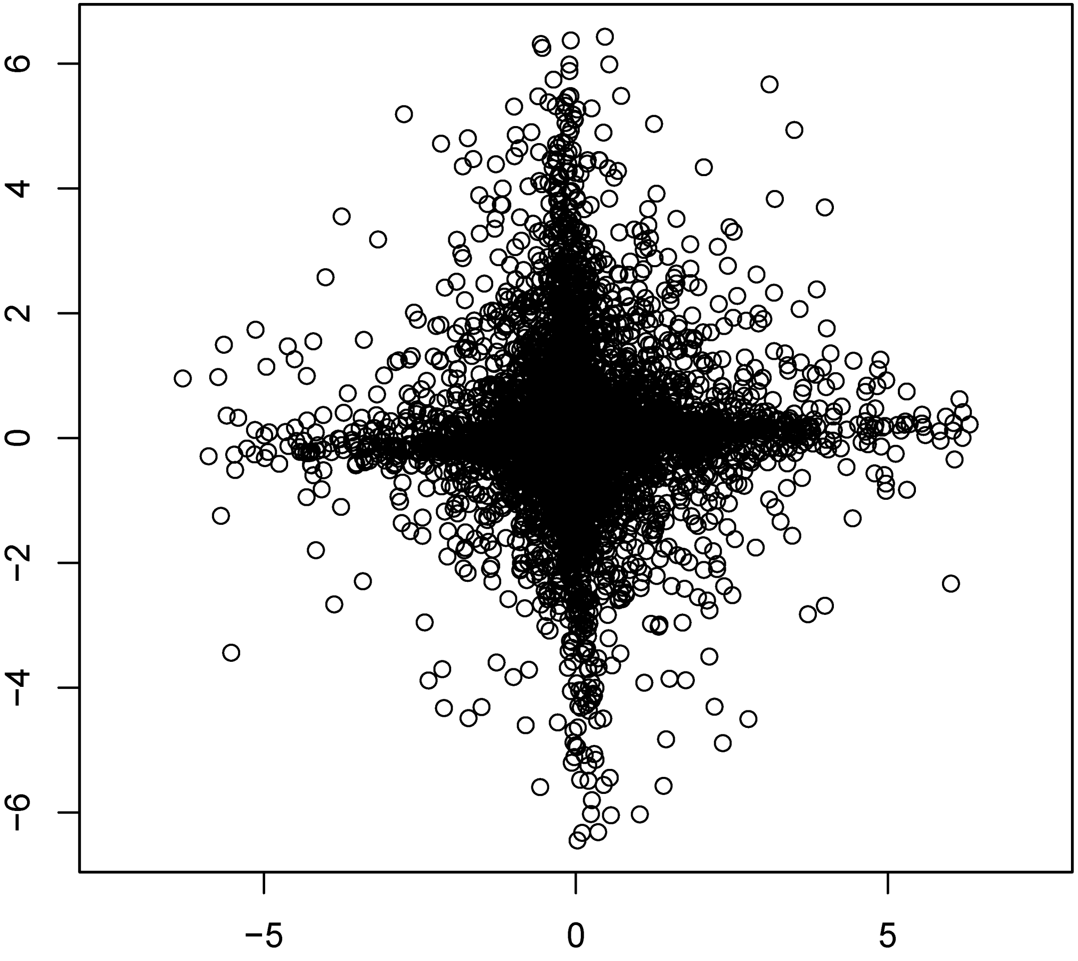
Independent Component Analysis

Whitening:



Independent Component Analysis

Rotation:

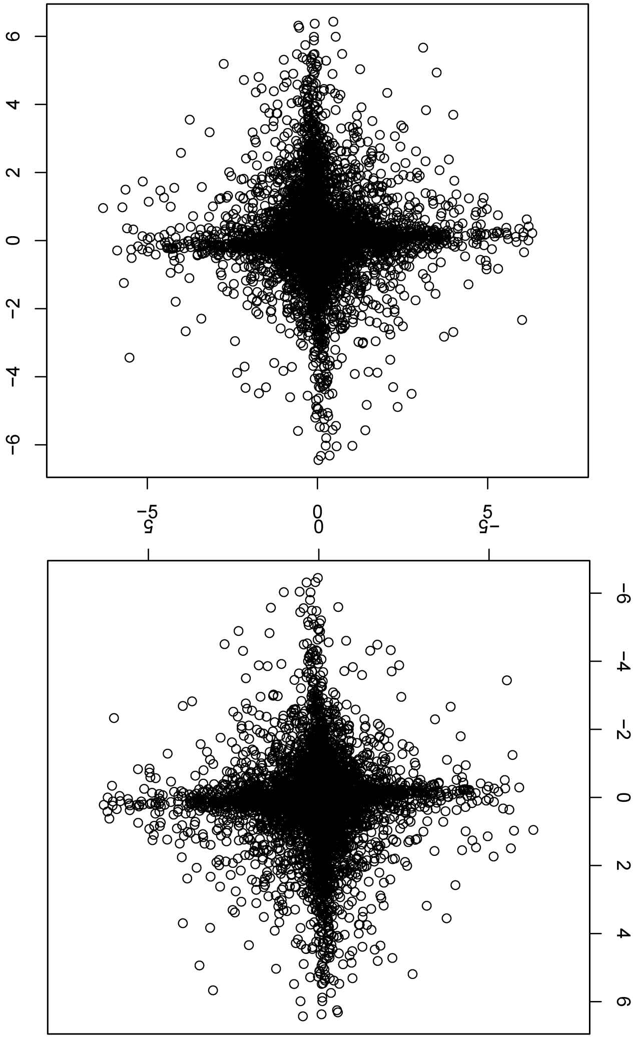


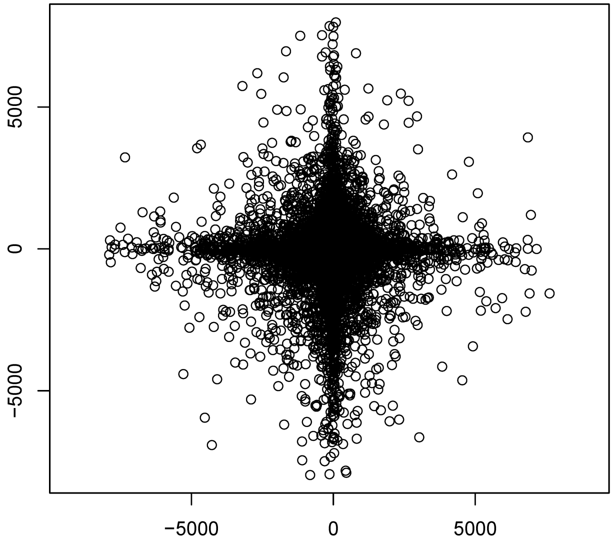
Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

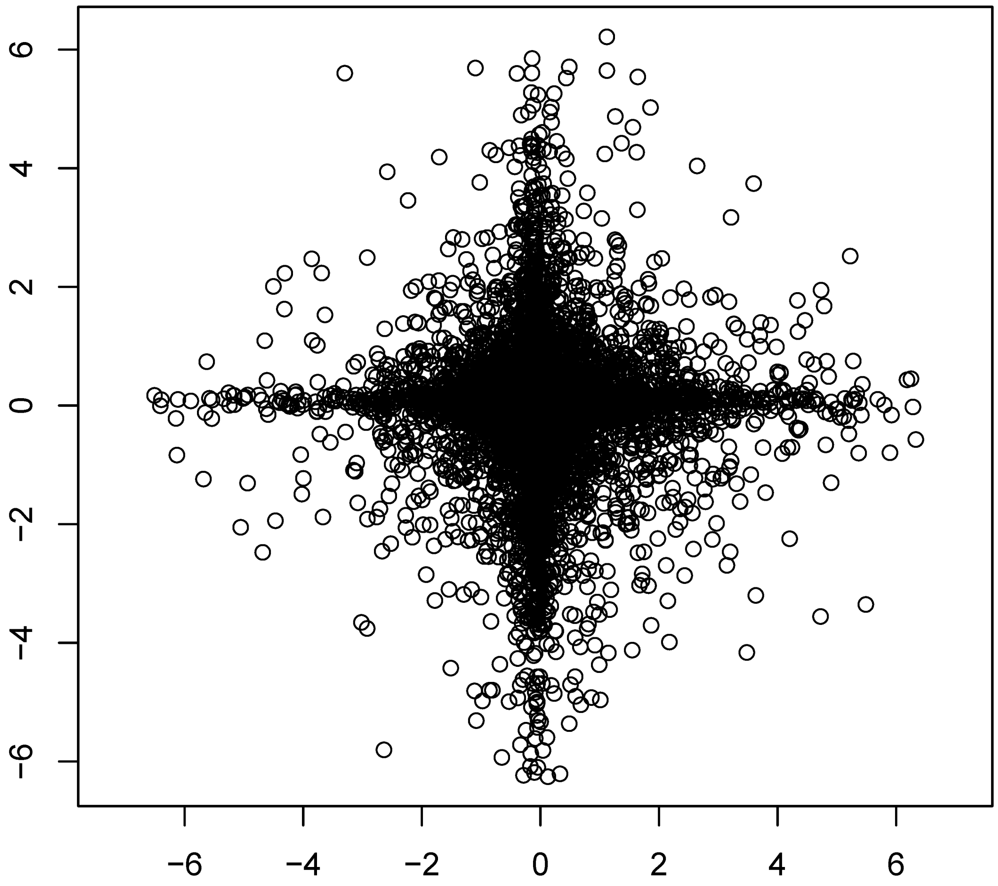
solution vs. original data





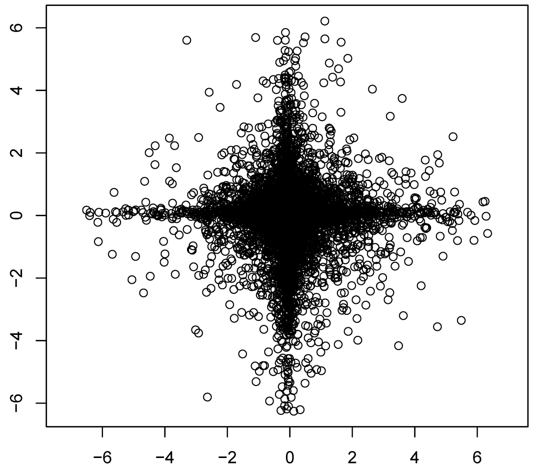
Independent Component Analysis

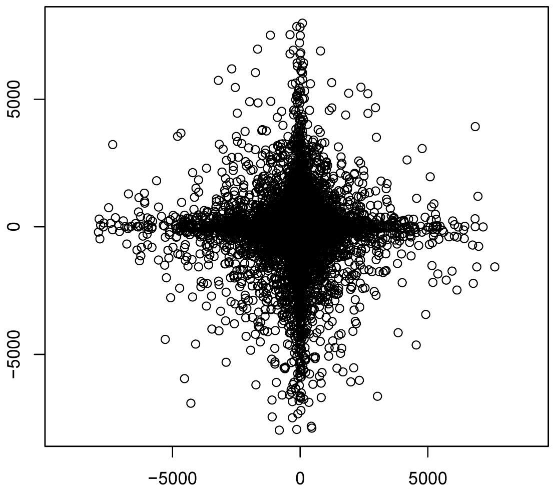
`fastICA:`

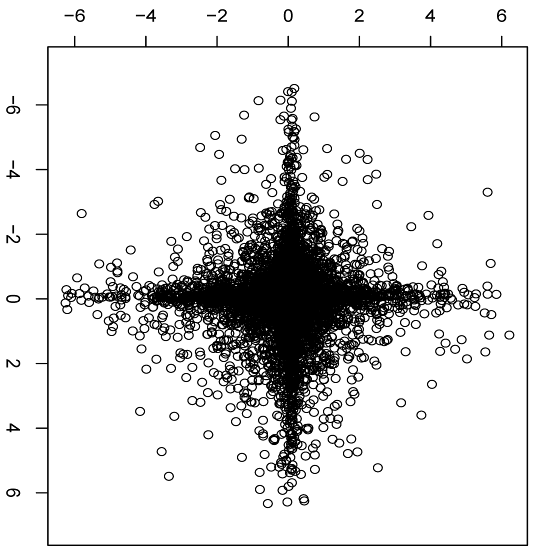


Independent Component Analysis

fastICA solution vs. original data







Iris Data

ICs ordered according to their impact on the observations given by the mixing matrix

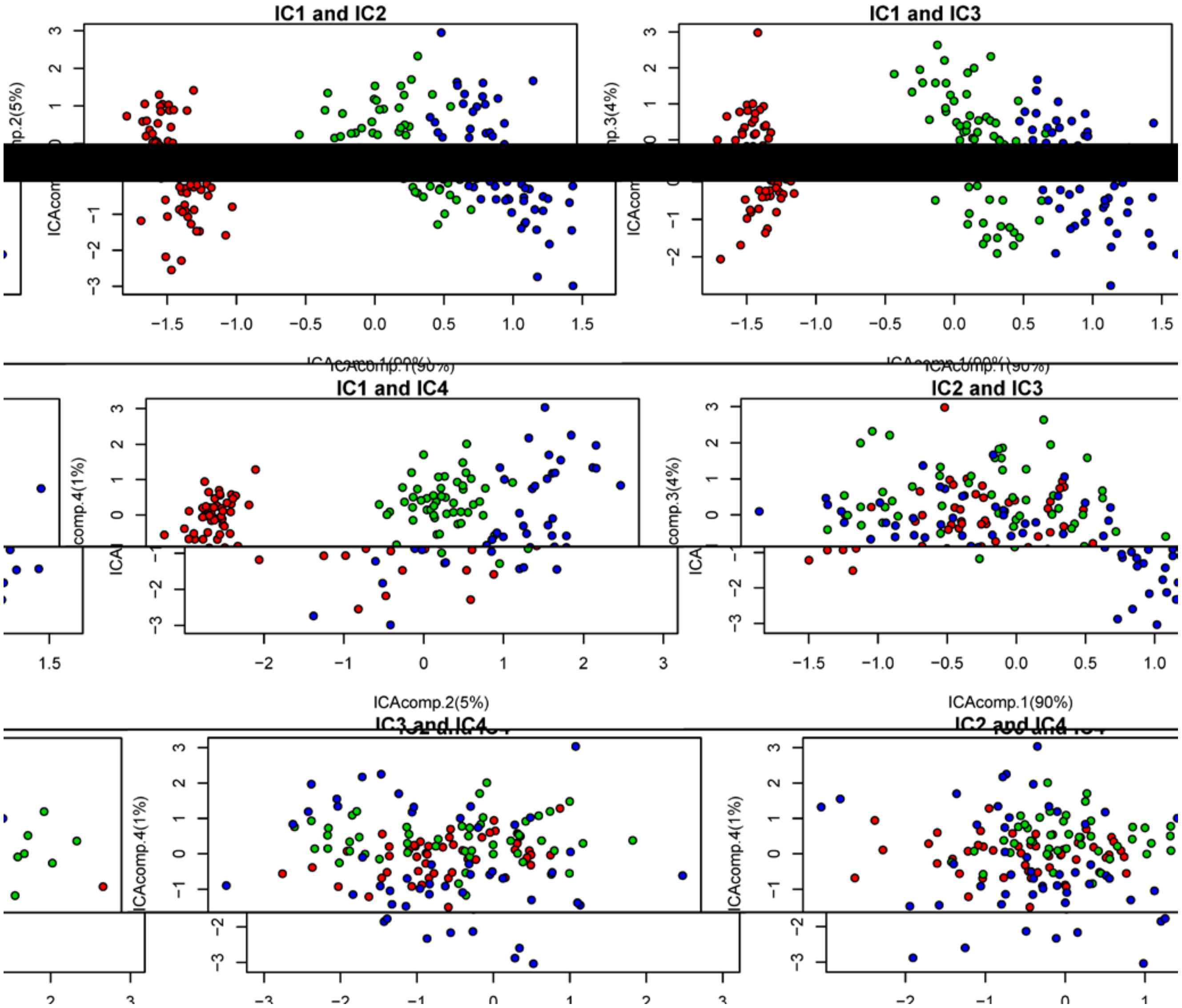
First independent component explains 90% of the variance in the data

Probably IC1 expresses the size of the blossom

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization



[redacted]

Multiple Tissues Data

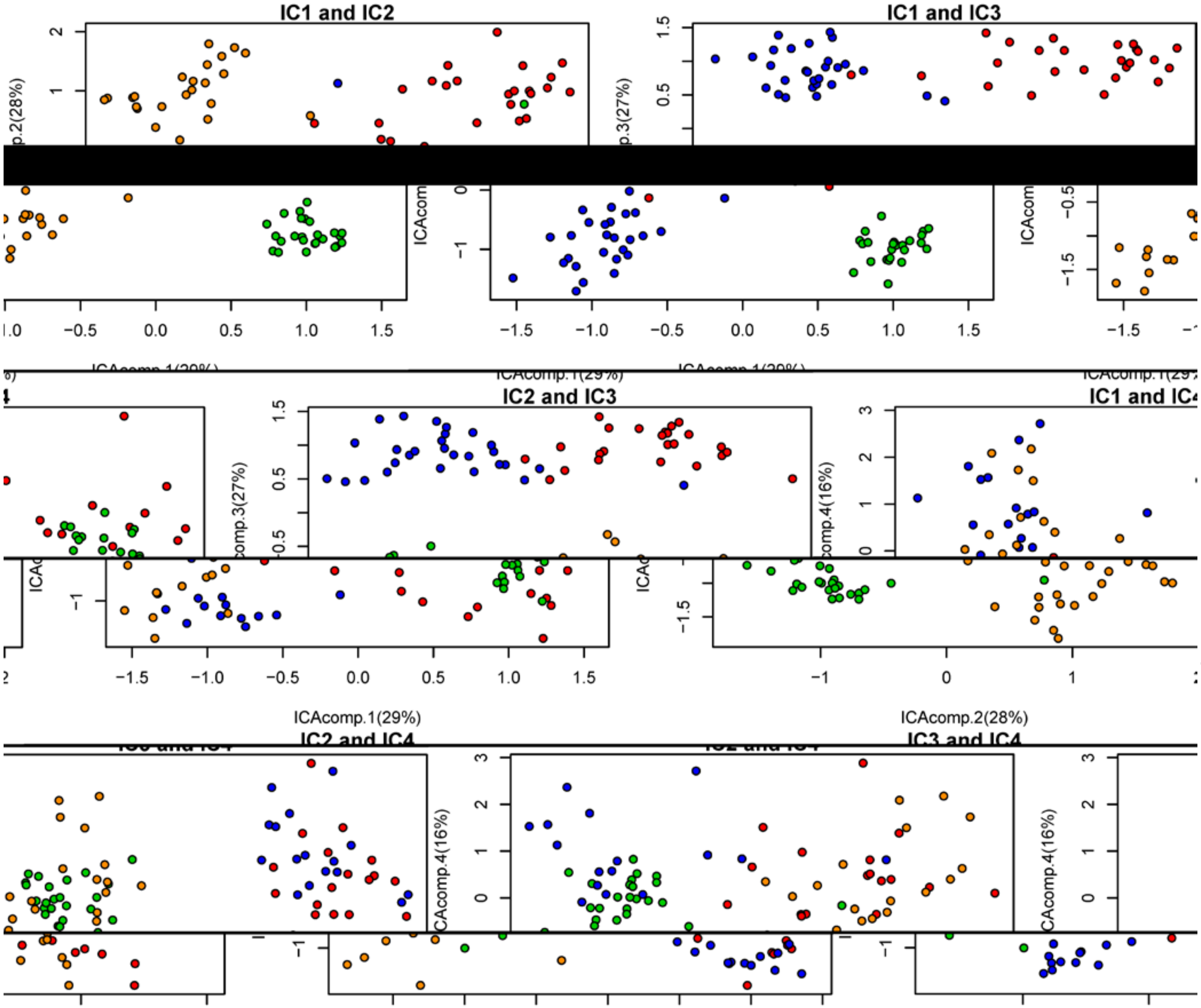
- IC1 separates the prostate samples (green) and the breast samples (red) from the colon samples (orange) and the lung samples (blue). Thus, IC1 separates internal organ tissues (colon and lung) from secretory or reproductive organ samples.
- IC2 separates the prostate samples and the lung samples from the breast samples and the colon samples.
- IC3 separates the prostate samples and the colon samples from the breast samples and the lung samples.

All combinations of the first 3 ICs lead to nice separations except for breast samples and lung samples for which some samples cannot be clearly assigned to one of these two classes.

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization


[redacted]

IC1: prostate (green) and breast (red) vs. colon (orange) and lung (blue)

IC2: prostate and lung vs. breast and colon

IC3: prostate and the colon vs. breast and lung

first 3 ICs lead to nice separations

breast and lung cannot be clearly separated

Independent Component Analysis

genes are correlated to IC1:

```
"SERPINA7" "LAMB3"    "AR"        "CCNG2"    "KLF5"     "CCL20"
"SLC39A14" "ATP1B1"   "GSTP1"     "LAD1"
```

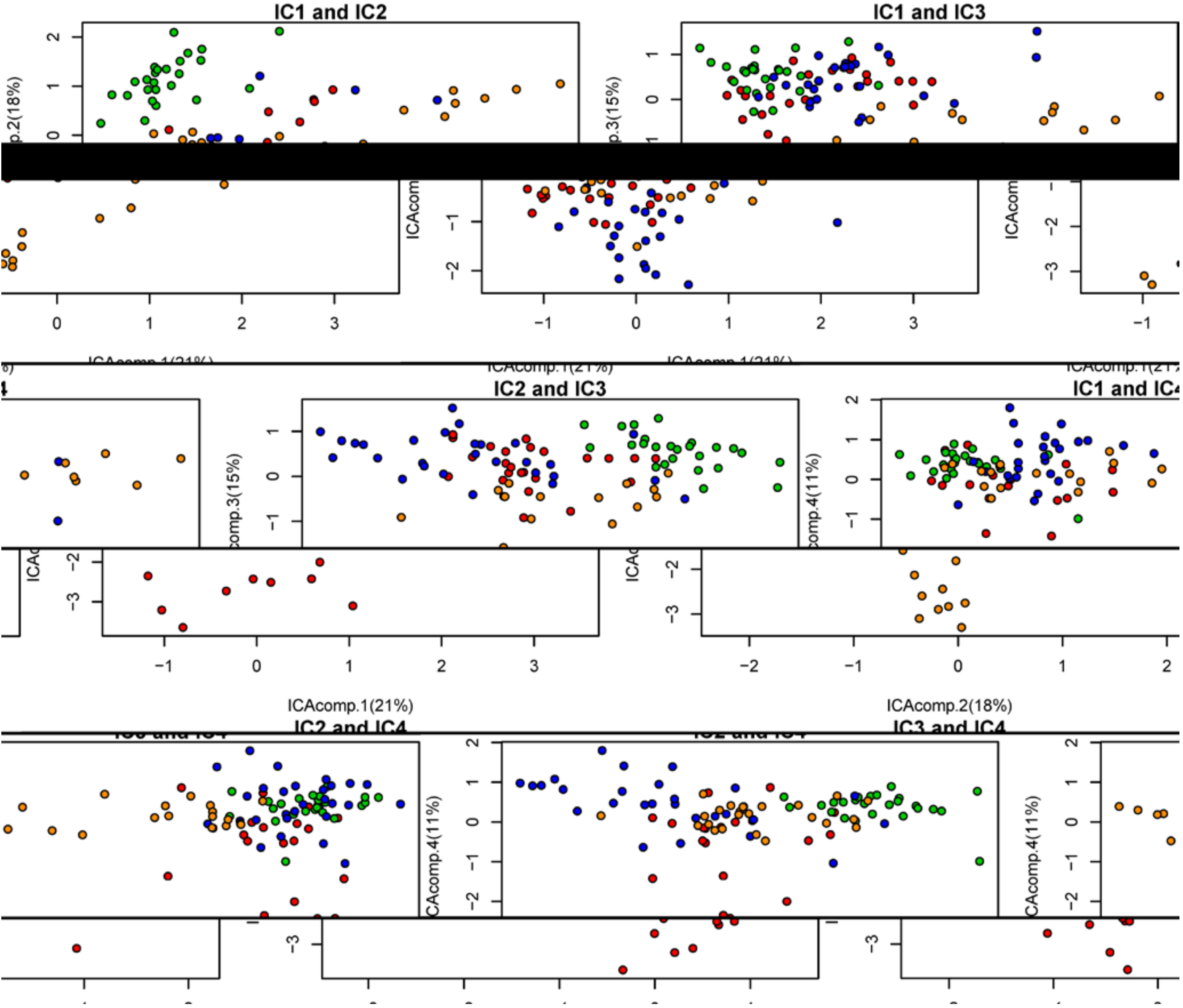
androgen receptor (AR) 3rd most related gene to IC1 which separates prostate and breast samples:

- growth / differentiation of prostate gland is regulated by androgens
- androgens play a role in normal breast physiology

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization


[redacted]

ICA with 8 components

Separation worse than with 4 comp.

prostate (green) is separated by IC1 and IC2

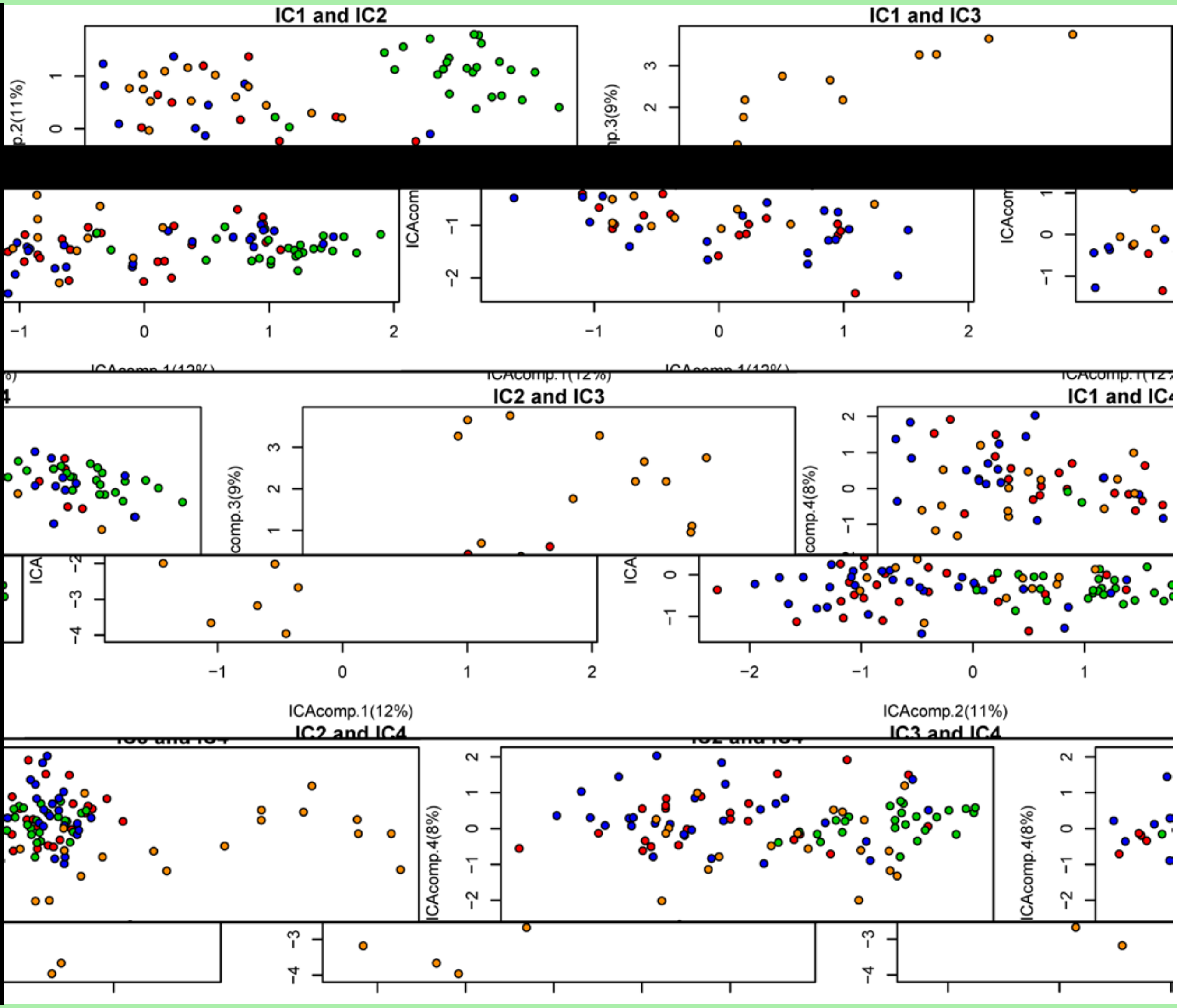
IC3 separates some colon sam. (orange)

ICs focus on smaller subgroups

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization



[redacted]

ICA with 20 components

result is very similar to 8 components

prostate (green) are separated by IC1 and IC2

IC3 and IC4 separate some colon (orange)

Again smaller subgoups found

Independent Component Analysis

y_1 and y_2 are assumed to be independent from each other and to be super Gaussian.
We show that a linear combination of these signal recovers one of both signals by maximizing the kurtosis.

We assume that y_1 and y_2 are zero centered, symmetric (not skewed), and independent of each other.
We obtain following moments and the reconstruction

$$\begin{aligned}
\mathrm{E}(y_1) &= 0 \,, \\
\mathrm{E}(y_2) &= 0 \,, \\
\mathrm{E}(y_1^2) &= v_1 \,, \\
\mathrm{E}(y_2^2) &= v_2 \,, \\
\mathrm{E}(y_1 \; y_2) &= 0 \,, \\
\mathrm{E}(y_1^3) &= 0 \,, \\
\mathrm{E}(y_2^3) &= 0 \,, \\
\mathrm{E}(y_1 \; y_2^2) &= 0 \,, \\
\mathrm{E}(y_2 \; y_1^2) &= 0 \,, \\
\mathrm{E}(y_1^4) &= m_1 \,, \\
\mathrm{E}(y_2^4) &= m_2 \,, \\
\mathrm{E}(y_1 \; y_2^3) &= 0 \,, \\
\mathrm{E}(y_2 \; y_1^3) &= 0 \,, \\
\mathrm{E}(y_1^2 \; y_2^2) &= v_1 \; v_2
\end{aligned}$$

$$y \;=\; a\; y_1 + b\; y_2$$

If y_1 and y_2 are super-Gaussian (have heavy tails) then the maximal kurtosis of $\boldsymbol{y}$ is obtained for $\boldsymbol{a}$=0 or $\boldsymbol{b}$=0 that is $\boldsymbol{y}$ is proportional to one y_i

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

For $\boldsymbol{y}$ we have the moments and the kurtosis:

$$\begin{aligned}
\mathrm{E}(y) &= 0 \ , \\
\mathrm{E}(y^2) &= a^2 \ v_1 \ + \ b^2 \ v_2 \ , \\
\mathrm{E}(y^3) &= 0 \ , \\
\mathrm{E}(y^4) &= a^4 \ m_1 \ + \ 6 \ a^2 \ b^2 \ v_1 \ v_2 \ + \ b^4 \ m_2 \\
k &= \frac{a^4 \ m1 \ + \ 6 \ a^2 \ b^2 \ v_1 \ v_2 \ + \ b^4 \ m_2}{\left(a^2 \ v_1 \ + \ b^2 \ v_2\right)^2}
\end{aligned}$$

The derivatives of the kurtosis with respect to $\boldsymbol{a}$ and $\boldsymbol{b}$ are:

$$\begin{aligned}
\frac{\partial k}{\partial a} &= \frac{4ab^2 \left(a^2 \left(m_1 - 3v_1^2\right) v_2 - b^2 v_1 \left(m_2 - 3v_2^2\right)\right)}{\left(a^2 v_1 + b^2 v_2\right)^3} \\
\frac{\partial k}{\partial b} &= \frac{4a^2 b \left(-a^2 \left(m_1 - 3v_1^2\right) v_2 + b^2 v_1 \left(m_2 - 3v_2^2\right)\right)}{\left(a^2 v_1 + b^2 v_2\right)^3}
\end{aligned}$$

The solution is:

$$\begin{aligned}
a &= 0 \qquad \text{or} \\
b &= 0 \qquad \text{or} \\
a^2 v_2 \left(m_1 - 3v_1^2\right) &= b^2 v_1 \left(m_2 - 3v_2^2\right)
\end{aligned}$$

Independent Component Analysis

3 Principal Component Analysis
3.1 The Method
3.2 Variance Maximization
3.3 Uniqueness
3.4 Properties PCA
3.5 Examples
3.6 Kernel PCA

4 Independent Component Analysis
4.1 Identifiability and Uniqueness
4.2 Measuring Independence
4.3 Whitening and Rotation Algorithms
4.4 INFOMAX
4.5 EASI
4.6 FastICA
4.7 ICA Extensions
4.8 ICA vs. PCA
4.9 Artificial ICA Examples
4.10 Real World ICA Examples
4.11 Kurtosis Maximization

The second order derivatives are:

$$
\begin{aligned}
\frac{\partial^2 k}{\partial a^2} &= \frac{1}{\left(a^2 v_1 + b^2 v_2\right)^4} 4b^2 \left(-3a^4 v_1 \left(m_1 - 3v_1^2\right) v_2 + \right. \\
& \left. b^4 v_1 v_2 \left(-m_2 + 3v_2^2\right) + a^2 b^2 \left(5 m_2 v_1^2 + 3 \left(m_1 - 8v_1^2\right) v_2^2\right)\right) \\
\frac{\partial^2 k}{\partial a^2} &= \frac{1}{\left(a^2 v_1 + b^2 v_2\right)^4} 4a^2 \left(a^4 v_1 \left(-m_1 + 3v_1^2\right) v_2 + \right. \\
& \left. 3b^4 v_1 v_2 \left(-m_2 + 3v_2^2\right) + a^2 b^2 \left(3 m_2 v_1^2 + \left(5m_1 - 24v_1^2\right) v_2^2\right)\right) \\
\frac{\partial^2 k}{\partial a \; \partial b} &= \frac{1}{\left(a^2 v_1 + b^2 v_2\right)^4} 8ab \left(a^4 v_1 \left(m_1 - 3v_1^2\right) v_2 + \right. \\
& \left. b^4 v_1 v_2 \left(m_2 - 3v_2^2\right) - 2a^2 b^2 \left(m_2 v_1^2 + \left(m_1 - 6v_1^2\right) v_2^2\right)\right) \\
\frac{\partial^2 k}{\partial a^2}(a, 0) &= 0 \\
\frac{\partial^2 k}{\partial a^2}(0, b) &= \frac{4 v_1 \left(-m_2 + 3v_2^2\right)}{b^2 v_2^3} \\
\frac{\partial^2 k}{\partial b^2}(0, b) &= 0 \\
\frac{\partial^2 k}{\partial b^2}(a, 0) &= \frac{4 \left(-m_1 + 3v_1^2\right) v_2}{a^2 v_1^3} \\
\frac{\partial^2 k}{\partial a \; \partial b}(0, b) &= 0 \\
\frac{\partial^2 k}{\partial a \; \partial b}(a, 0) &= 0
\end{aligned}
$$

Smaller zero!

Smaller zero!

Independent Component Analysis

For the last root of the derivatives we get

$$\hat{a} \;=\; b \sqrt{\frac{v_1 (m_2 - 3 v_2^2)}{v_2 (m_1 - 3 v_1^2)}}$$

$$\frac{\partial^2 k}{\partial a^2}(\hat{a}, b) \;=\; \frac{\partial^2 k}{\partial b^2}(\hat{a}, b) \;=\; \frac{8 v_1 \left(m_1 - 3 v_1^2\right)^3 v_2^3 \left(m_2 - 3 v_2^2\right)}{b^2 \left(m_2 v_1^2 + \left(m_1 - 6 v_1^2\right) v_2^2\right)^3} \;=$$

$$\frac{8 v_1 \left(m_1 - 3 v_1^2\right)^3 v_2^3 \left(m_2 - 3 v_2^2\right)}{b^2 \left(v_1^2 \left(m_2 - 3 v_2^2\right) + v_2^2 \left(m_1 - 3 v_1^2\right)\right)^3}$$

Larger zero!

$$\frac{\partial^2 k}{\partial a \; \partial b}(\hat{a}, b) \;=\; - \frac{8 \left(m_1 - 3 v_1^2\right)^4 v_2^4 \left(\frac{v_1 \left(m_2 - 3 v_2^2\right)}{\left(m_1 - 3 v_1^2\right) v_2}\right)^{3/2}}{b^2 \left(m_2 v_1^2 + \left(m_1 - 6 v_1^2\right) v_2^2\right)^3} \;=$$

$$- \left(\frac{\left(m_1 - 3 v_1^2\right) v_2}{v_1 \left(m_2 - 3 v_2^2\right)}\right)^{1/2} \frac{\partial^2 k}{\partial a^2}(\hat{a}, b)$$

Smaller zero!

Independent Component Analysis

The eigenvalues of the Hessian are proportional to

$$e_1 \;\propto\; 1 \;-\; \left(\frac{\left(m_1 - 3v_1^2\right) v_2}{v_1 \left(m_2 - 3v_2^2\right)} \right)^{1/2}$$

$$e_2 \;\propto\; 1 \;+\; \left(\frac{\left(m_1 - 3v_1^2\right) v_2}{v_1 \left(m_2 - 3v_2^2\right)} \right)^{1/2}$$

It is impossible that both eigenvalues are negative as required by a maximum. Therefore the maxima are either a = 0 or b = 0 for which the Hessian is negative semidefinite.

If the kurtosis $k_2 > k_1$ then a = 0

If the kurtosis $k_1 > k_2$ then b = 0

Factor Analysis

Factor analysis describes the variability of observations in terms of unobserved latent variables, called factors, and noise

- factors explain correlation between the variables
- remaining variance is explained by Gaussian noise

factor analysis is a generative approach and models both the noise of the observations and their correlation

assumptions on the distribution of factors and noise

Factor Analysis

centered data: $\{\boldsymbol{x}\} \; = \; \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

$$\boldsymbol{x} \; = \; \underbrace{\boldsymbol{U}\boldsymbol{y}}_{\text{signal}} + \underbrace{\boldsymbol{\epsilon}}_{\text{noise}} \quad \text{where} \;\; \boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}\right) \quad \text{and} \quad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Psi}\right)$$

- Observations ---------------------------------- $\boldsymbol{x} \in \mathbb{R}^m$
- Noise -- $\boldsymbol{\epsilon} \in \mathbb{R}^m$
- Factors -- $\boldsymbol{y} \in \mathbb{R}^l$
- Factor loading matrix------------------------ $\boldsymbol{U} \in \mathbb{R}^{m \times l}$
- Diagonal noise covariance matrix ------- $\boldsymbol{\Psi} \in \mathbb{R}^{m \times m}$

$\}$ paramete

$$\boldsymbol{x} \mid \boldsymbol{y} \; \sim \; \mathcal{N}\left(\boldsymbol{U}\boldsymbol{y}, \boldsymbol{\Psi}\right)$$

Factor Analysis

matrix decomposition: $\boldsymbol{X} \; = \; \boldsymbol{Y} \, \boldsymbol{U}^T \; + \; \boldsymbol{\Upsilon}$

model assumptions:

$$\begin{aligned} \frac{1}{n} \, \boldsymbol{Y}^T \boldsymbol{Y} \; &= \; \boldsymbol{I} \\ \boldsymbol{Y}^T \boldsymbol{\Upsilon} \; &= \; \boldsymbol{0} \\ \frac{1}{n} \, \boldsymbol{\Upsilon}^T \boldsymbol{\Upsilon} \; &= \; \boldsymbol{\Psi} \end{aligned}$$

we obtain:

$$\begin{aligned} \frac{1}{n} \, \boldsymbol{X}^T \boldsymbol{X} \; &= \; \frac{1}{n} \, (\boldsymbol{Y} \, \boldsymbol{U}^T \; + \; \boldsymbol{\Upsilon})^T \, (\boldsymbol{Y} \, \boldsymbol{U}^T \; + \; \boldsymbol{\Upsilon}) \\ &= \; \boldsymbol{U} \, \left(\frac{1}{n} \, \boldsymbol{Y}^T \boldsymbol{Y} \right) \, \boldsymbol{U}^T \; + \; \frac{1}{n} \, \boldsymbol{U} \, \boldsymbol{Y}^T \, \boldsymbol{\Upsilon} \; + \; \frac{1}{n} \, \boldsymbol{\Upsilon}^T \boldsymbol{Y} \, \boldsymbol{U}^T \; + \; \frac{1}{n} \, \boldsymbol{\Upsilon}^T \boldsymbol{\Upsilon} \\ &= \; \boldsymbol{U} \, \boldsymbol{U}^T \; + \; \boldsymbol{\Psi} \end{aligned}$$

factor analysis is actually a decomposition of the covariance matrix

$$\boldsymbol{C} = \frac{1}{n} \boldsymbol{X}^T \boldsymbol{X}$$

into an expression of the two parameter matrices.

Factor Analysis

fewer factors than features: $m \geq l$

diagonal $\boldsymbol{\Psi}$: noise of the components are is independent

correlations between observations can only be explained by factors

decomposition of the covariance matrix: $\frac{1}{n} \boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{U} \boldsymbol{U}^T + \boldsymbol{\Psi}$

parameter estimation→maximum likelihood: expectation-maximization

both parameters explain the variance in the observations:

$\boldsymbol{U}$ explains the dependent part

$\boldsymbol{\Psi}$ explains the independent part

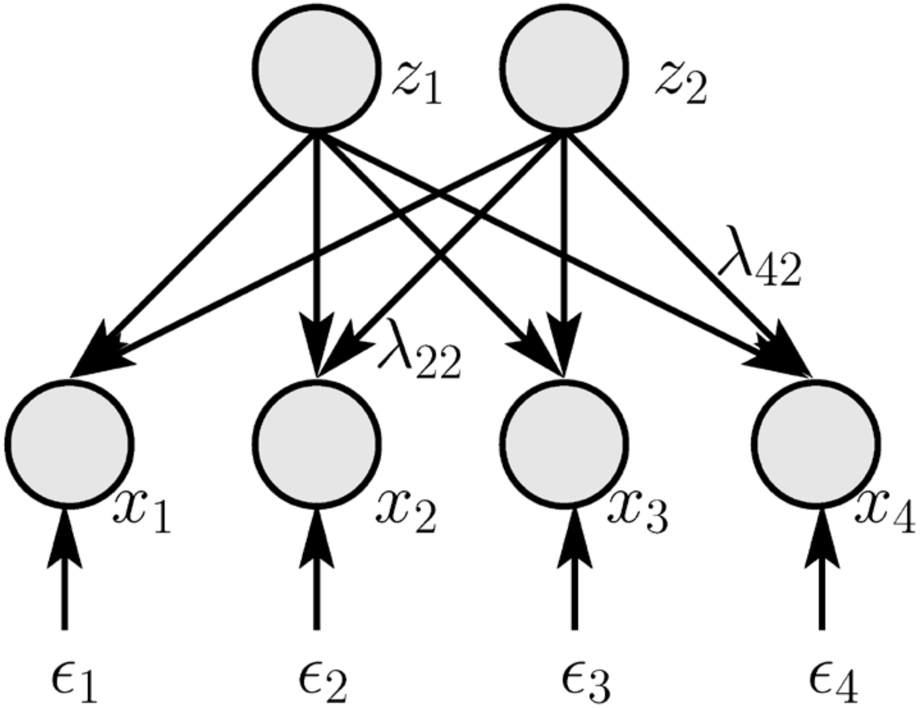
Factor Analysis

factor $\boldsymbol{z}$

loading matrix Λ

observations $\boldsymbol{x}$

noise ϵ



Factor Analysis

Estimation of factors: “projection of the data onto the factors”

regression setting: $\boldsymbol{Y} = \boldsymbol{X} \boldsymbol{A}$ where $\boldsymbol{A}$ is parameter

least squares solution: $\hat{\boldsymbol{A}} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{Y}$

model assumptions and empirical approximations:

$$\boldsymbol{U}\boldsymbol{U}^T + \boldsymbol{\Psi} = \mathrm{Var}(\boldsymbol{x}) \approx \frac{1}{n} \boldsymbol{X}^T \boldsymbol{X}$$

$$\boldsymbol{U} = \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y}) \approx \frac{1}{n} \boldsymbol{X}^T \boldsymbol{Y}$$

estimation for $\boldsymbol{A}$:

$$\hat{\boldsymbol{A}} = \mathrm{E}\left(\left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1}\right) \mathrm{E}\left(\boldsymbol{X}^T \boldsymbol{Y}\right)$$

$$\hat{\boldsymbol{A}} = \left(\boldsymbol{U}\boldsymbol{U}^T + \boldsymbol{\Psi}\right)^{-1} \boldsymbol{U}$$

$$\boldsymbol{Y} = \boldsymbol{X} \left(\boldsymbol{U}\boldsymbol{U}^T + \boldsymbol{\Psi}\right)^{-1} \boldsymbol{U}$$

matrix inversion lemma:

$$\boldsymbol{Y} = \boldsymbol{X} \boldsymbol{\Psi}^{-1} \boldsymbol{U} \left(\boldsymbol{I} + \boldsymbol{U} \boldsymbol{\Psi}^{-1} \boldsymbol{U}^T\right)^{-1}$$

Factor Analysis

outer product representation for $\boldsymbol{l}$ factors: $\boldsymbol{X} \ = \ \sum_{j=1}^{l} \boldsymbol{u}_j \boldsymbol{y}_j^T \ + \ \boldsymbol{\Upsilon}$

$\boldsymbol{u}_j$: $\boldsymbol{j}$-th column vector of $\boldsymbol{U}$
$\boldsymbol{y}_j$: $\boldsymbol{j}$-th row vector of $\boldsymbol{Y}$

communality c_j of an observation variable x_j ($\boldsymbol{j}$-th component of $\boldsymbol{x}$):

$$c_j \ = \ \frac{\mathrm{Var}(x_j) \ - \ \mathrm{Var}(\epsilon_j)}{\mathrm{Var}(x_j)} \ = \ \frac{\sum_{k=1}^{l} \lambda_{jk}^2}{\Psi_{jj} \ + \ \sum_{k=1}^{l} \lambda_{jk}^2}$$

proportion in x_j explained by the factors

Here each factor y_t contributes: $\dfrac{\lambda_{jt}^2}{\Psi_{jj} \ + \ \sum_{k=1}^{l} \lambda_{jk}^2}$

Like with PCA, the projection onto $\boldsymbol{l}$ factors maximizes the variance in the data which can be explained by $\boldsymbol{l}$ factors.

However factor analysis considers only the signal variance (not noise)

Factor Analysis

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

factor projections are orthogonal to each other: $\frac{1}{n}\boldsymbol{Y}^T\boldsymbol{Y} = \boldsymbol{I}$

factors are not unique up to orthogonal transformations (rotations):

$\boldsymbol{Y}\boldsymbol{U}^T = \boldsymbol{Y}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{U}^T = \boldsymbol{Y}'\boldsymbol{U}'^T$ with orthogonal $\boldsymbol{V}$

projections rotated to make the factors more interpretable or to find simpler structures:

- Varimax rotation: maximizes the squared loadings of a factor on all the variables; each factor has either large or small loadings of any particular variable; each variable is assigned to a factor.
- Quartimax rotation: minimizes the number of factors needed to explain each variable; each factor explains many variables; in most cases not interpretable.
- Equimax rotation: compromise between Varimax and Quartimax.

Factor Analysis

$\boldsymbol{x} \mid \boldsymbol{z} \ \sim \ \mathcal{N}\left(\boldsymbol{\Lambda}\boldsymbol{z}, \boldsymbol{\Psi}\right)$ $\boldsymbol{z}$ is given → only noise distribution

First and second moment of the data (factor and noise):

$$\mathrm{E}(\boldsymbol{x}) \ = \ \mathrm{E}(\boldsymbol{\Lambda}\boldsymbol{z} \ + \ \boldsymbol{\epsilon}) \ = \ \boldsymbol{\Lambda}\mathrm{E}(\boldsymbol{z}) \ + \ \mathrm{E}(\boldsymbol{\epsilon}) \ = \ \boldsymbol{0} \ ,$$

$$\mathrm{E}\left(\boldsymbol{x}\ \boldsymbol{x}^T\right) \ = \ \mathrm{E}\left((\boldsymbol{\Lambda}\boldsymbol{z} \ + \ \boldsymbol{\epsilon})(\boldsymbol{\Lambda}\boldsymbol{z} \ + \ \boldsymbol{\epsilon})^T\right) \ =$$

$$\boldsymbol{\Lambda}\mathrm{E}\left(\boldsymbol{z}\ \boldsymbol{z}^T\right)\boldsymbol{\Lambda}^T \ + \ \boldsymbol{\Lambda}\mathrm{E}\left(\boldsymbol{z}\right)\mathrm{E}\left(\boldsymbol{\epsilon}^T\right) \ + \ \mathrm{E}\left(\boldsymbol{z}\right)\mathrm{E}\left(\boldsymbol{\epsilon}\right) \ \boldsymbol{\Lambda}^T \ + \ \mathrm{E}\left(\boldsymbol{\epsilon}\ \boldsymbol{\epsilon}^T\right) \ =$$

$$\boldsymbol{\Lambda}\ \boldsymbol{\Lambda}^T \ + \ \boldsymbol{\Psi}$$

Distribution of the data:

$$\boldsymbol{x} \ \sim \ \mathcal{N}\left(\boldsymbol{0} \ , \ \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T \ + \ \boldsymbol{\Psi}\right)$$

observations are Gaussian distributed

Factor Analysis

log-likelihood:

$$\log \prod_{i=1}^{l} (2\pi)^{-d/2} \left| \boldsymbol{\Lambda} \boldsymbol{\Lambda}^T + \boldsymbol{\Psi} \right|^{-1/2}$$

$$\exp \left(- \frac{1}{2} \left((\boldsymbol{x}^i)^T \left(\boldsymbol{\Lambda} \boldsymbol{\Lambda}^T + \boldsymbol{\Psi} \right)^{-1} \boldsymbol{x}^i \right) \right)$$

maximize the likelihood is difficult: no closed form

Factor Analysis

EM-algorithm: hidden states are the factors

$$Q_i(\boldsymbol{z}^i) \;=\; p\left(\boldsymbol{z}^i \mid \boldsymbol{x}^i; \boldsymbol{\Lambda}^{\mathrm{old}}, \boldsymbol{\Psi}^{\mathrm{old}}\right)$$

"old" is skipped in the following

$$\boldsymbol{z}^i \mid \boldsymbol{x}^i \;\sim\; \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{z}^i|\boldsymbol{x}^i}, \boldsymbol{\Sigma}_{\boldsymbol{z}^i|\boldsymbol{x}^i}\right)$$

$$\boldsymbol{\mu}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \;=\; \left(\boldsymbol{x}^i\right)^T \left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1} \boldsymbol{\Lambda}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \;=\; \boldsymbol{I} - \boldsymbol{\Lambda}^T \left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1} \boldsymbol{\Lambda} \;+$$
$$\left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1} \boldsymbol{x}^i \left(\boldsymbol{x}^i\right)^T \left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1}$$

we used

$$\boldsymbol{v} \sim \mathcal{N}\left(\boldsymbol{\mu}_v, \Sigma_{vv}\right) , \; \boldsymbol{u} \sim \mathcal{N}\left(\boldsymbol{\mu}_u, \Sigma_{uu}\right) ,$$
$$\Sigma_{uv} \;=\; \mathrm{Covar}(\boldsymbol{u}, \boldsymbol{v}) \text{ and } \; \Sigma_{vu} \;=\; \mathrm{Covar}(\boldsymbol{v}, \boldsymbol{u}) :$$
$$\boldsymbol{v} \mid \boldsymbol{u} \;\sim\; \mathcal{N}\left(\boldsymbol{\mu}_v + \Sigma_{vu}\Sigma_{uu}^{-1}\left(\boldsymbol{u} - \boldsymbol{\mu}_u\right) , \; \Sigma_{vv} + \Sigma_{vu}\Sigma_{uu}^{-1}\Sigma_{uv}\right)$$

and

$$\mathrm{E}(\boldsymbol{z}\boldsymbol{x}) \;=\; \boldsymbol{\Lambda}\;\mathrm{E}(\boldsymbol{z}\;\boldsymbol{z}^T) \;=\; \boldsymbol{\Lambda}$$

Factor Analysis

$$Q_i(\boldsymbol{z}^i) \;=\; (2\pi)^{-d/2} \left| \boldsymbol{\Sigma}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \right|^{-1/2}$$

$$\exp\left(-\frac{1}{2} \; (\boldsymbol{z}^i \;-\; \boldsymbol{\mu}_{\boldsymbol{z}^i|\boldsymbol{x}^i})^T \; \boldsymbol{\Sigma}^{-1}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \, (\boldsymbol{z}^i \;-\; \boldsymbol{\mu}_{\boldsymbol{z}^i|\boldsymbol{x}^i}) \right)$$

lower bound for the likelihood:

$$\log\left(p(\boldsymbol{x}^i \mid \boldsymbol{\Lambda}, \boldsymbol{\Psi})\right) \;=$$

$$\log\left(\int_{\mathbb{R}^p} \frac{Q_i(\boldsymbol{z}^i)\; p(\boldsymbol{x}^i, \boldsymbol{z}^i \mid \boldsymbol{\Lambda}, \boldsymbol{\Psi})}{Q_i(\boldsymbol{z}^i)} d\boldsymbol{z}^i \right) \;\geq$$

$$\int_{\mathbb{R}^p} Q_i(\boldsymbol{z}^i)\; \log\left(\frac{p(\boldsymbol{x}^i, \boldsymbol{z}^i \mid \boldsymbol{\Lambda}, \boldsymbol{\Psi})}{Q_i(\boldsymbol{z}^i)} \right) d\boldsymbol{z}^i$$

expectation

$$\mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i}\left(f(\boldsymbol{z}^i)\right) \;=\; \int_{\mathbb{R}^p} Q_i(\boldsymbol{z}^i)\; f(\boldsymbol{z}^i)\; d\boldsymbol{z}^i$$

Factor Analysis

M-step maximizes

$$\log \mathcal{L} \;=\; \frac{d\, l}{2} \log\left(2\pi\right) \;-\; \frac{l}{2} \log \left|\boldsymbol{\Psi}\right| \;-\; \frac{1}{2} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\left(\boldsymbol{x}^i \;-\; \boldsymbol{\Lambda} \boldsymbol{z}^i\right)^T \boldsymbol{\Psi}^{-1} \left(\boldsymbol{x}^i \;-\; \boldsymbol{\Lambda} \boldsymbol{z}^i\right)\right)$$

optimality criteria

$$\frac{1}{l} \nabla_{\boldsymbol{\Lambda}} \log \mathcal{L} \;=\; \frac{1}{l} \sum_{i=1}^{l} \boldsymbol{\Psi}^{-1}\, \boldsymbol{\Lambda}\, \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \,\left(\boldsymbol{z}^i\right)^T\right) \;-\; \frac{1}{l} \sum_{i=1}^{l} \boldsymbol{\Psi}^{-1}\, \boldsymbol{x}^i\, \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i\right) \;=\; \boldsymbol{0}$$

and

$$\nabla_{\boldsymbol{\Psi}} \log \mathcal{L} \;=\; -\frac{l}{2} \boldsymbol{\Psi}^{-1} \;+\; \frac{1}{2} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{\Psi}^{-1} \left(\boldsymbol{x}^i \;-\; \boldsymbol{\Lambda} \boldsymbol{z}^i\right) \,\left(\boldsymbol{x}^i \;-\; \boldsymbol{\Lambda} \boldsymbol{z}^i\right)^T \boldsymbol{\Psi}^{-1} \right) \;=\; \boldsymbol{0}$$

Factor Analysis

Solving above equations gives:

$$\boldsymbol{\Lambda}^{\mathrm{new}} = \left(\frac{1}{l} \sum_{i=1}^{l} \boldsymbol{x}^i \ \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \right) \right) \left(\frac{1}{l} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \ (\boldsymbol{z}^i)^T \right) \right)^{-1}$$

and

$$\boldsymbol{\Psi}^{\mathrm{new}} =$$

$$\mathrm{diag} \left(\frac{1}{l} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\left(\boldsymbol{x}^i - \boldsymbol{\Lambda}^{\mathrm{new}} \boldsymbol{z}^i \right) \left(\boldsymbol{x}^i - \boldsymbol{\Lambda}^{\mathrm{new}} \boldsymbol{z}^i \right)^T \right) \right) =$$

$$\mathrm{diag} \left(\frac{1}{l} \sum_{i=1}^{l} \boldsymbol{x}^i \ (\boldsymbol{x}^i)^T - \frac{1}{l} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \right) \boldsymbol{x}^i \left(\boldsymbol{\Lambda}^{\mathrm{new}} \right)^T - \right.$$

$$\frac{1}{l} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \right) \boldsymbol{\Lambda}^{\mathrm{new}} (\boldsymbol{x}^i)^T +$$

$$\left. \frac{1}{l} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \ (\boldsymbol{z}^i)^T \right) \boldsymbol{\Lambda}^{\mathrm{new}} \left(\boldsymbol{\Lambda}^{\mathrm{new}} \right)^T \right)$$

Factor Analysis

Loading matrix update gives

$$\boldsymbol{\Lambda}^{\text{new}} \left(\frac{1}{l} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \; (\boldsymbol{z}^i)^T \right) \right) \;=\; \left(\frac{1}{l} \sum_{i=1}^{l} \boldsymbol{x}^i \; \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \right) \right)$$

which can be inserted into the update of the noise covariance

→ one term $\frac{1}{l} \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \right) \boldsymbol{\Lambda}^{\text{new}} (\boldsymbol{x}^i)^T$ cancels

$$\boldsymbol{\Psi}^{\text{new}} \;=\; \frac{1}{l} \operatorname{diag} \left(\sum_{i=1}^{l} \boldsymbol{x}^i \; (\boldsymbol{x}^i)^T \;-\; \sum_{i=1}^{l} \mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \left(\boldsymbol{z}^i \right) \boldsymbol{x}^i \; (\boldsymbol{\Lambda}^{\text{new}})^T \right)$$

EM updates:

E-step:

$$\mathrm{E}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}\left(\boldsymbol{z}^i\right) \;=\; \boldsymbol{\mu}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}$$

$$\mathrm{E}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}\left(\boldsymbol{z}^i\;(\boldsymbol{z}^i)^T\right) \;=\; \boldsymbol{\mu}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}\;\boldsymbol{\mu}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}^T \;+\; \boldsymbol{\Sigma}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}$$

M-step:

$$\boldsymbol{\Lambda}^{\mathrm{new}} \;=$$

$$\left(\frac{1}{l}\sum_{i=1}^{l}\boldsymbol{x}^i\;\mathrm{E}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}\left(\boldsymbol{z}^i\right)\right)\;\left(\frac{1}{l}\sum_{i=1}^{l}\mathrm{E}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}\left(\boldsymbol{z}^i\;(\boldsymbol{z}^i)^T\right)\right)^{-1}$$

$$\boldsymbol{\Psi}^{\mathrm{new}} \;=$$

$$\frac{1}{l}\mathrm{diag}\left(\sum_{i=1}^{l}\boldsymbol{x}^i\;(\boldsymbol{x}^i)^T \;-\; \sum_{i=1}^{l}\mathrm{E}_{\boldsymbol{z}^i \mid \boldsymbol{x}^i}\left(\boldsymbol{z}^i\right)\boldsymbol{x}^i\;(\boldsymbol{\Lambda}^{\mathrm{new}})^T\right)$$

Factor Analysis

Speed Ups:

matrix inversion lemma: $\boldsymbol{d}$ > $\boldsymbol{p}$ (compute in $\boldsymbol{p}$-dimensional space)

$$\left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \;=\; \boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\left(\boldsymbol{I} \;+\; \boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\right)^{-1}\boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}$$

$\boldsymbol{\Psi}^{-1}$ can be evaluated very → diagonal matrix

covariance ***C*** *only once computed:*

$$\frac{1}{l}\sum_{i=1}^{l}\boldsymbol{x}^i\;\mathrm{E}_{\boldsymbol{z}^i|\boldsymbol{x}^i}\left(\boldsymbol{z}^i\right) \;=$$

$$\left(\frac{1}{l}\sum_{i=1}^{l}\boldsymbol{x}^i\left(\boldsymbol{x}^i\right)^T\right)\;\left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Lambda} \;=$$

$$\boldsymbol{C}\;\left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Lambda} \;=$$

$$\boldsymbol{C}\;\left(\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda} \;-\; \boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\left(\boldsymbol{I} \;+\; \boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\right)^{-1}\boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\right) \;=$$

$$\boldsymbol{C}\;\left(\boldsymbol{A} \;-\; \boldsymbol{A}\left(\boldsymbol{I} \;+\; \boldsymbol{B}\right)^{-1}\boldsymbol{B}\right)$$

$$\boxed{\begin{aligned}\boldsymbol{A} &= \boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\\ \boldsymbol{B} &= \boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T\boldsymbol{A}\end{aligned}}$$

Factor Analysis

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

$$\frac{1}{l} \sum_{i=1}^{l} \boldsymbol{\Sigma}_{\boldsymbol{z}^i | \boldsymbol{x}^i} \; =$$

$$\boldsymbol{I} \; - \; \boldsymbol{\Lambda}^T \; \left(\boldsymbol{\Lambda} \, \boldsymbol{\Lambda}^T \; + \; \boldsymbol{\Psi}\right)^{-1} \boldsymbol{\Lambda} \; +$$

$$\left(\boldsymbol{\Lambda} \, \boldsymbol{\Lambda}^T \; + \; \boldsymbol{\Psi}\right)^{-1} \left(\frac{1}{l} \sum_{i=1}^{l} \boldsymbol{x}^i \; \left(\boldsymbol{x}^i\right)^T \right) \left(\boldsymbol{\Lambda} \, \boldsymbol{\Lambda}^T \; + \; \boldsymbol{\Psi}\right)^{-1} \; =$$

$$\boldsymbol{I} \; - \; \boldsymbol{\Lambda}^T \; \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda} \; + \; \boldsymbol{\Lambda}^T \; \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda} \left(\boldsymbol{I} \; + \; \boldsymbol{\Lambda}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda}\right)^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda} \; +$$

$$\left(\boldsymbol{\Psi}^{-1} \; - \; \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda} \left(\boldsymbol{I} \; + \; \boldsymbol{\Lambda}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda}\right)^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Psi}^{-1}\right) \; \boldsymbol{C}$$

$$\left(\boldsymbol{\Psi}^{-1} \; - \; \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda} \left(\boldsymbol{I} \; + \; \boldsymbol{\Lambda}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\Lambda}\right)^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Psi}^{-1}\right) \; =$$

$$\boldsymbol{I} \; - \; \boldsymbol{B} \; + \; \boldsymbol{B} \left(\boldsymbol{I} \; + \; \boldsymbol{B}\right)^{-1} \boldsymbol{B} \; +$$

$$\left(\boldsymbol{\Psi}^{-1} \; - \; \boldsymbol{A} \left(\boldsymbol{I} \; + \; \boldsymbol{B}\right)^{-1} \boldsymbol{A}^T\right) \; \boldsymbol{C} \; \left(\boldsymbol{\Psi}^{-1} \; - \; \boldsymbol{A} \left(\boldsymbol{I} \; + \; \boldsymbol{B}\right)^{-1} \boldsymbol{A}^T\right)$$

Factor Analysis

$$\frac{1}{l}\sum_{i=1}^{l} \boldsymbol{\mu}_{\boldsymbol{z}^i|\boldsymbol{x}^i} \; \boldsymbol{\mu}^T_{\boldsymbol{z}^i|\boldsymbol{x}^i} \;=$$

$$\boldsymbol{\Lambda}^T \left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \left(\frac{1}{l}\sum_{i=1}^{l} \boldsymbol{x}^i \left(\boldsymbol{x}^i\right)^T\right) \left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \boldsymbol{\Lambda} \;=$$

$$\boldsymbol{\Lambda}^T \left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \boldsymbol{C} \; \left(\boldsymbol{\Lambda}\,\boldsymbol{\Lambda}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \boldsymbol{\Lambda} \;=$$

$$\left(\boldsymbol{A} \;-\; \boldsymbol{A}\left(\boldsymbol{I} \;+\; \boldsymbol{B}\right)^{-1} \boldsymbol{B}\right)^T \boldsymbol{C} \left(\boldsymbol{A} \;-\; \boldsymbol{A}\left(\boldsymbol{I} \;+\; \boldsymbol{B}\right)^{-1} \boldsymbol{B}\right)$$

sums $\sum_{i=1}^{l}$ are removed and the matrix $\boldsymbol{C}$ can be computed once at the beginning of the iterative procedure

MAP factor analysis $p(\boldsymbol{\Lambda}, \boldsymbol{\Psi} \mid \{\boldsymbol{x}\}) \ \propto \ p(\{\boldsymbol{x}\} \mid \boldsymbol{\Lambda}, \boldsymbol{\Psi}) \ p(\boldsymbol{\Lambda})$

posterior $p(\boldsymbol{\Lambda}, \boldsymbol{\Psi} \mid \{\boldsymbol{x}\})$

likelihood $p(\{\boldsymbol{x}\} \mid \boldsymbol{\Lambda}, \boldsymbol{\Psi})$

prior $p(\boldsymbol{\Lambda})$

log-posterior

$$\log \left(p(\boldsymbol{\Lambda}, \boldsymbol{\Psi} \mid \{\boldsymbol{x}\})\right) \ = \ \log \left(p(\{\boldsymbol{x}\} \mid \boldsymbol{\Lambda}, \boldsymbol{\Psi})\right) \ + \ \log \left(p(\boldsymbol{\Lambda})\right)$$

example for the prior: rectified Gaussian $\mathcal{N}_{\mathrm{rect}} \left(\mu_{\Lambda}, \sigma_{\Lambda}\right)$

→ only positive factor loading values

$$y_j \ \sim \ \mathcal{N} \left(\mu_{\Lambda}, \sigma_{\Lambda}\right)$$
$$\lambda_j \ = \ \max\{y_j , 0\}$$

Factor Analysis

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

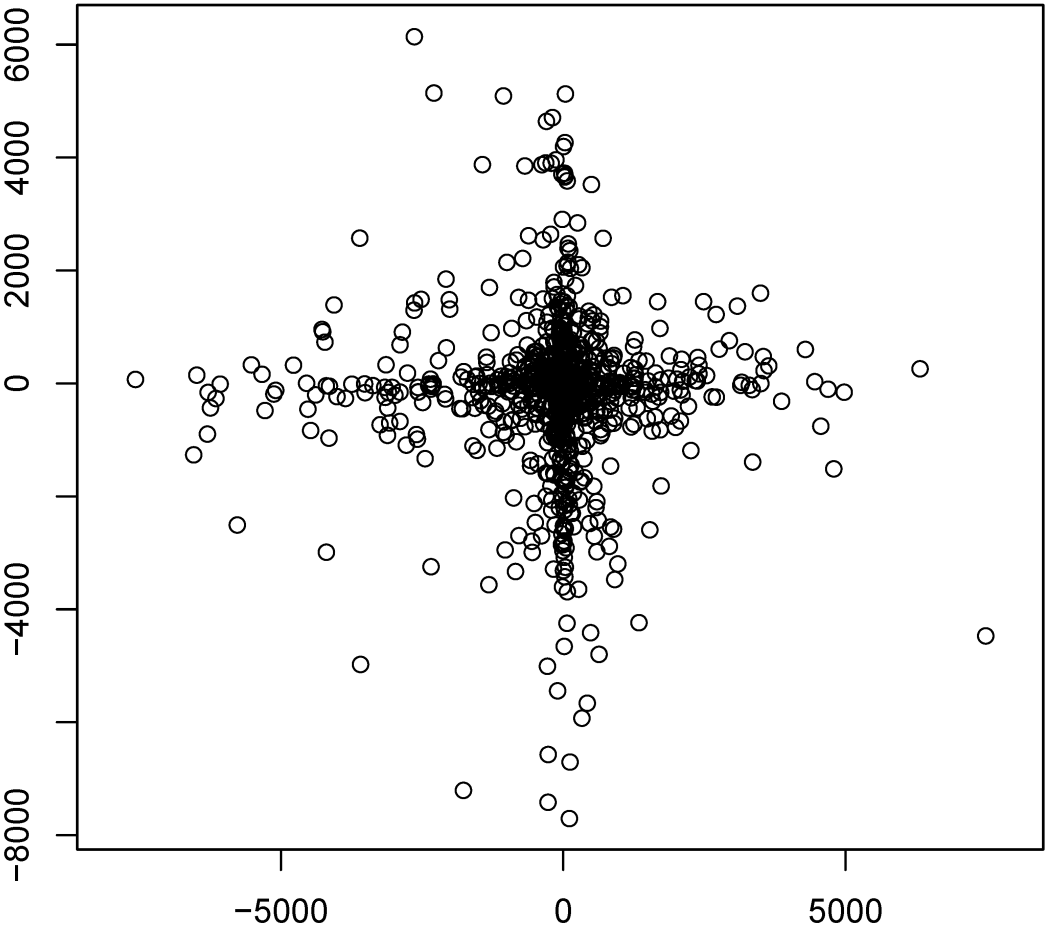
factor analysis	principal component analysis
causes of the data	geometrical abstractions
explain common variances	explain all variance
variance shared	first l with max. variance
scale invariant	not scale invariant
additive noise (variance lost)	no noise
solution not unique	solution unique
model assumptions	no assumptions
solution depends on l	first l unique
projection uses noise	no noise
no ranking	ranked by eigenvalues

Factor Analysis

factor analysis	independent component analysis
additive noise	no noise
solution not unique	unique up to scale & permutation
assumption: Gauss	assumption: super-Gauss
projection averaged over noise	no noise
solution depends on l	does not depend on l

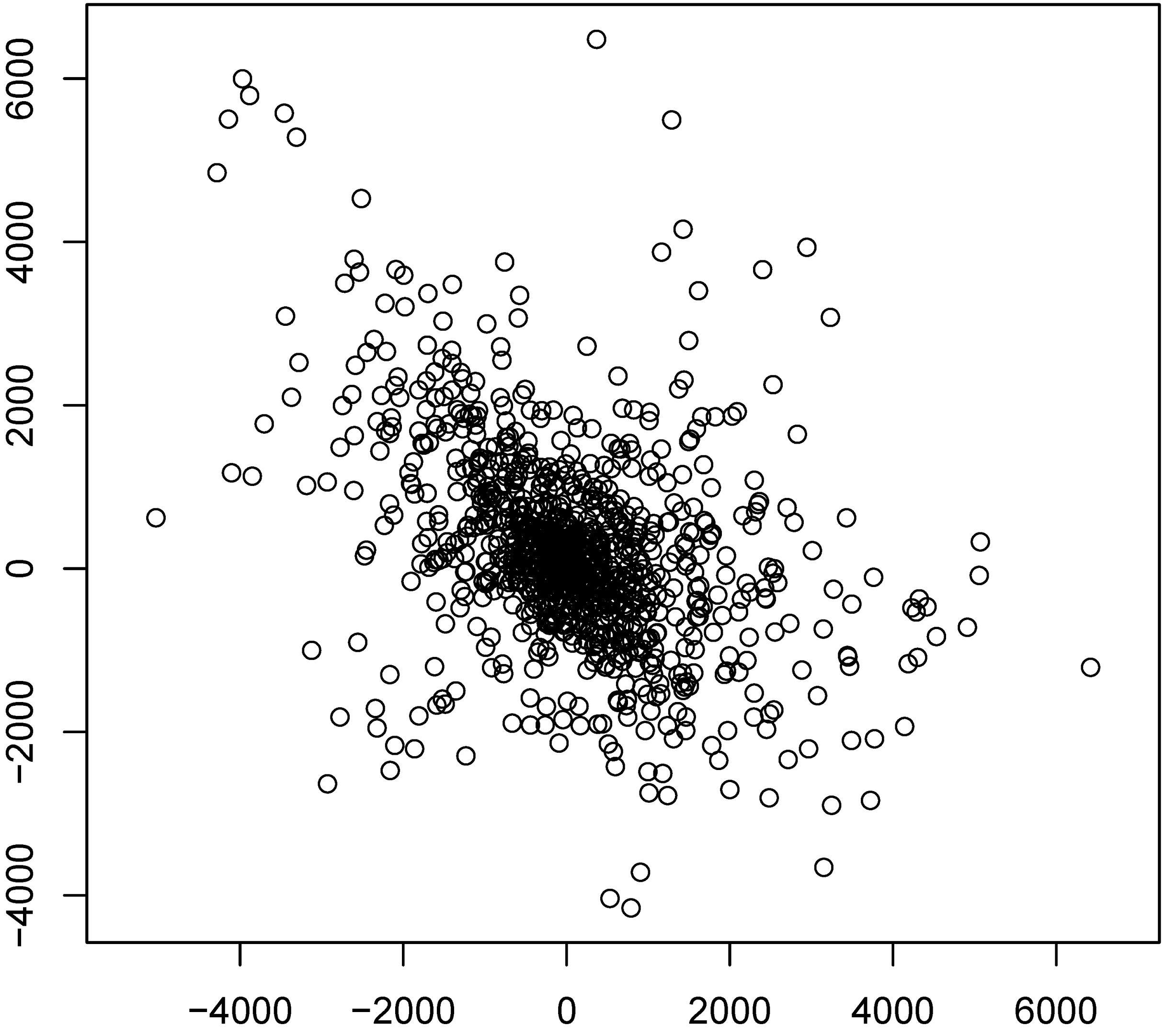
Factor Analysis

50-dimensional data set with linearly mixed super-Gaussians



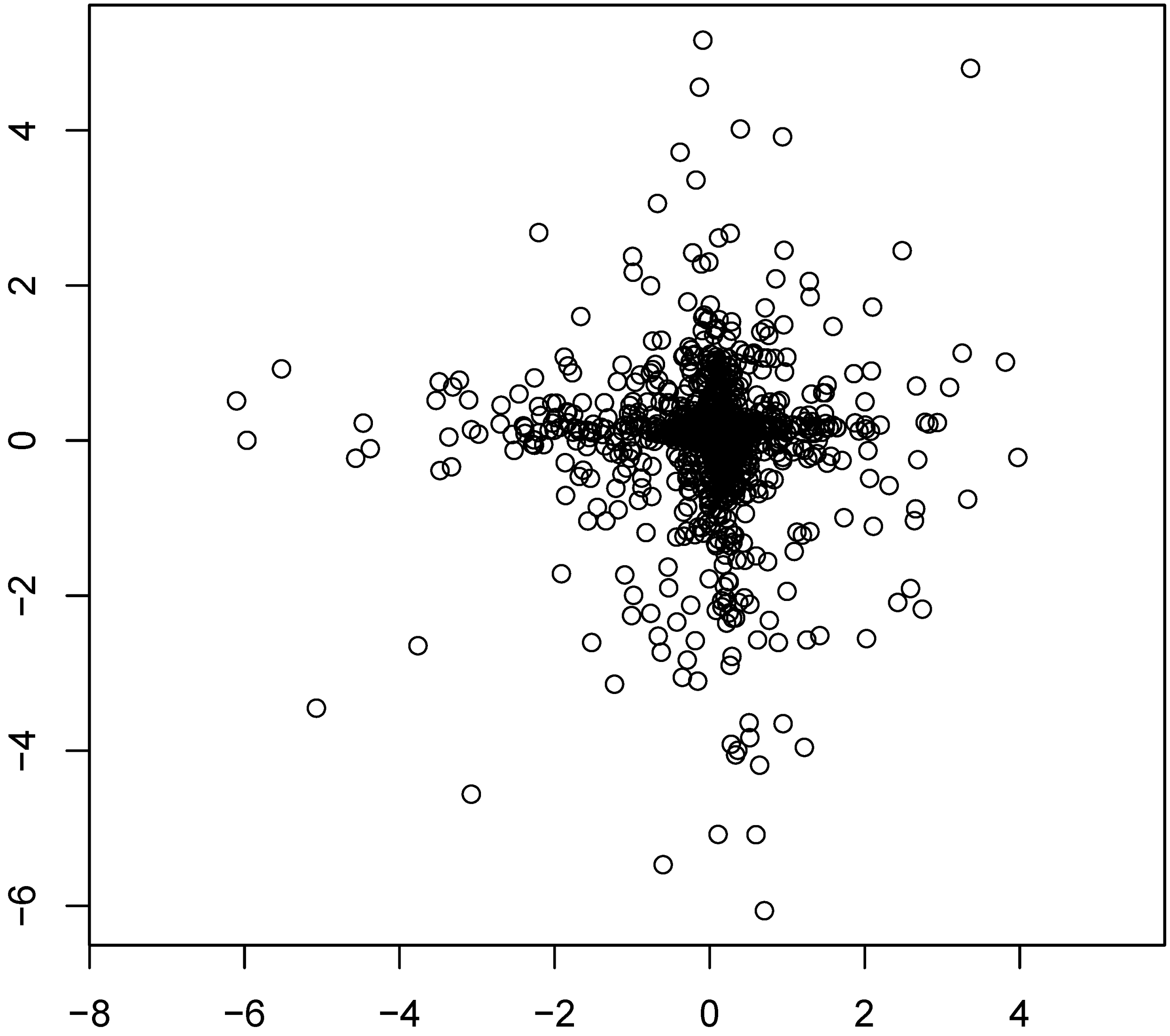
Factor Analysis

Mixing:



Factor Analysis

fastICA:

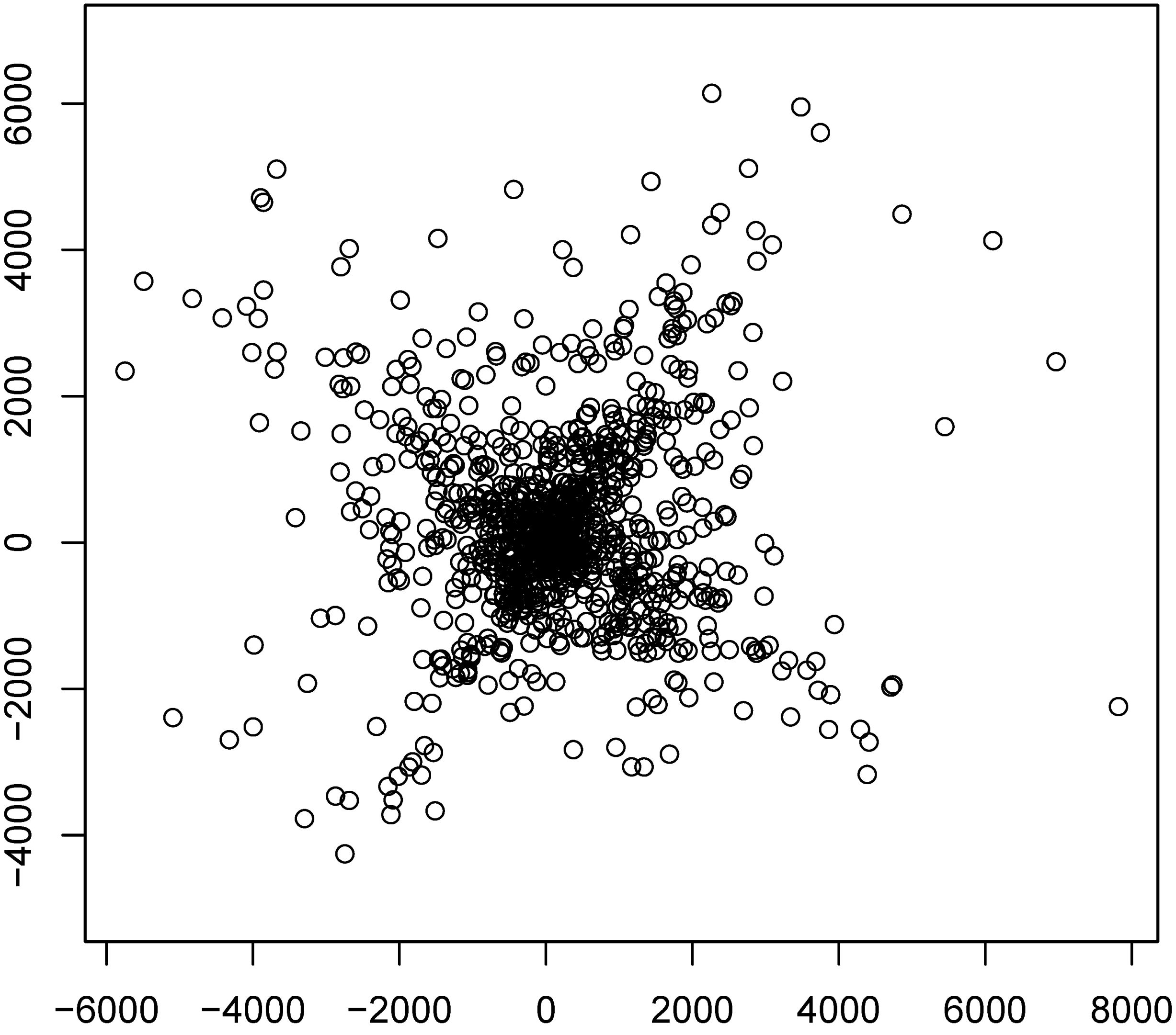


Factor Analysis

factor analysis:

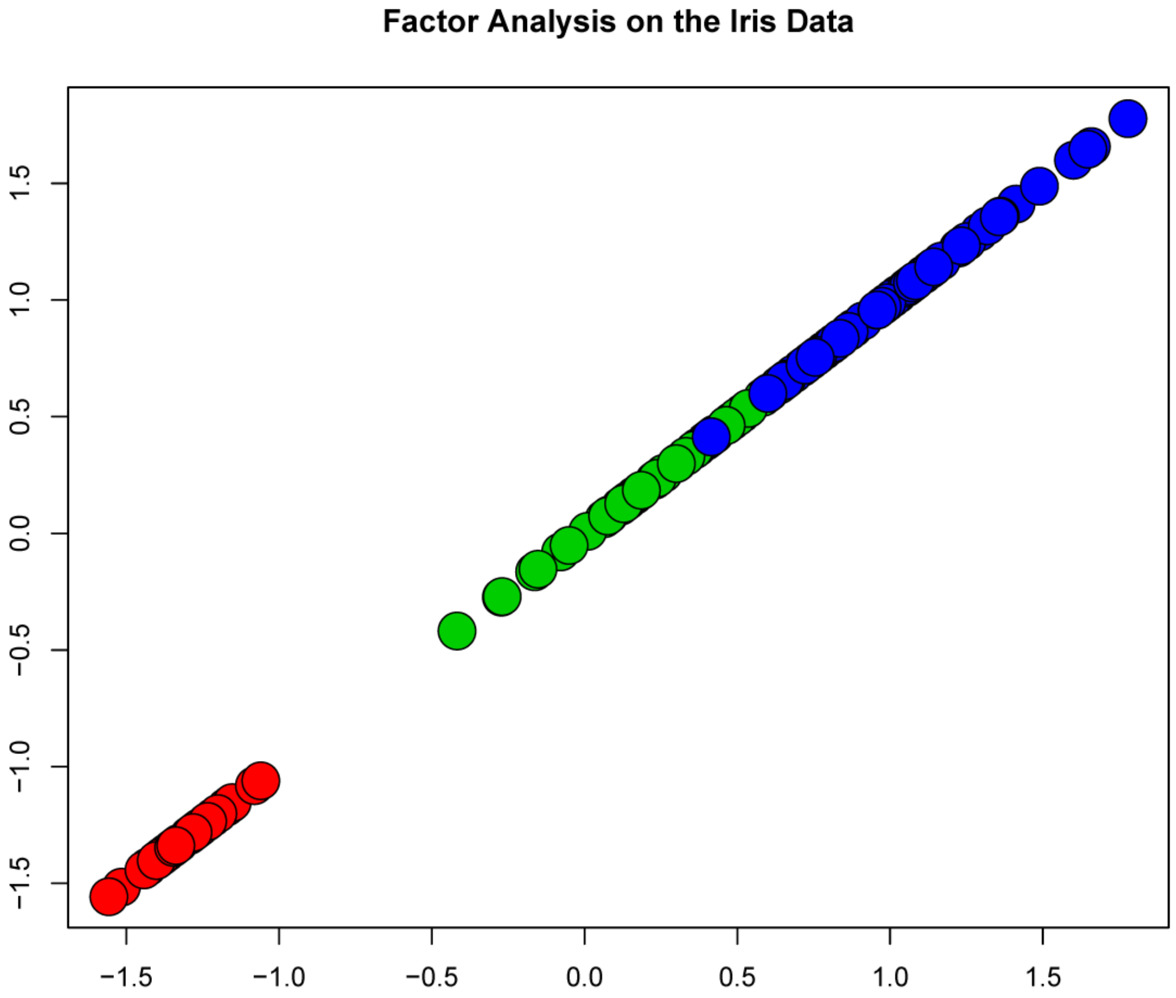
demixing is worse than with ICA

factor analysis assumes normally distributed factors while ICA super-Gaussians → ICA better suited



Factor Analysis

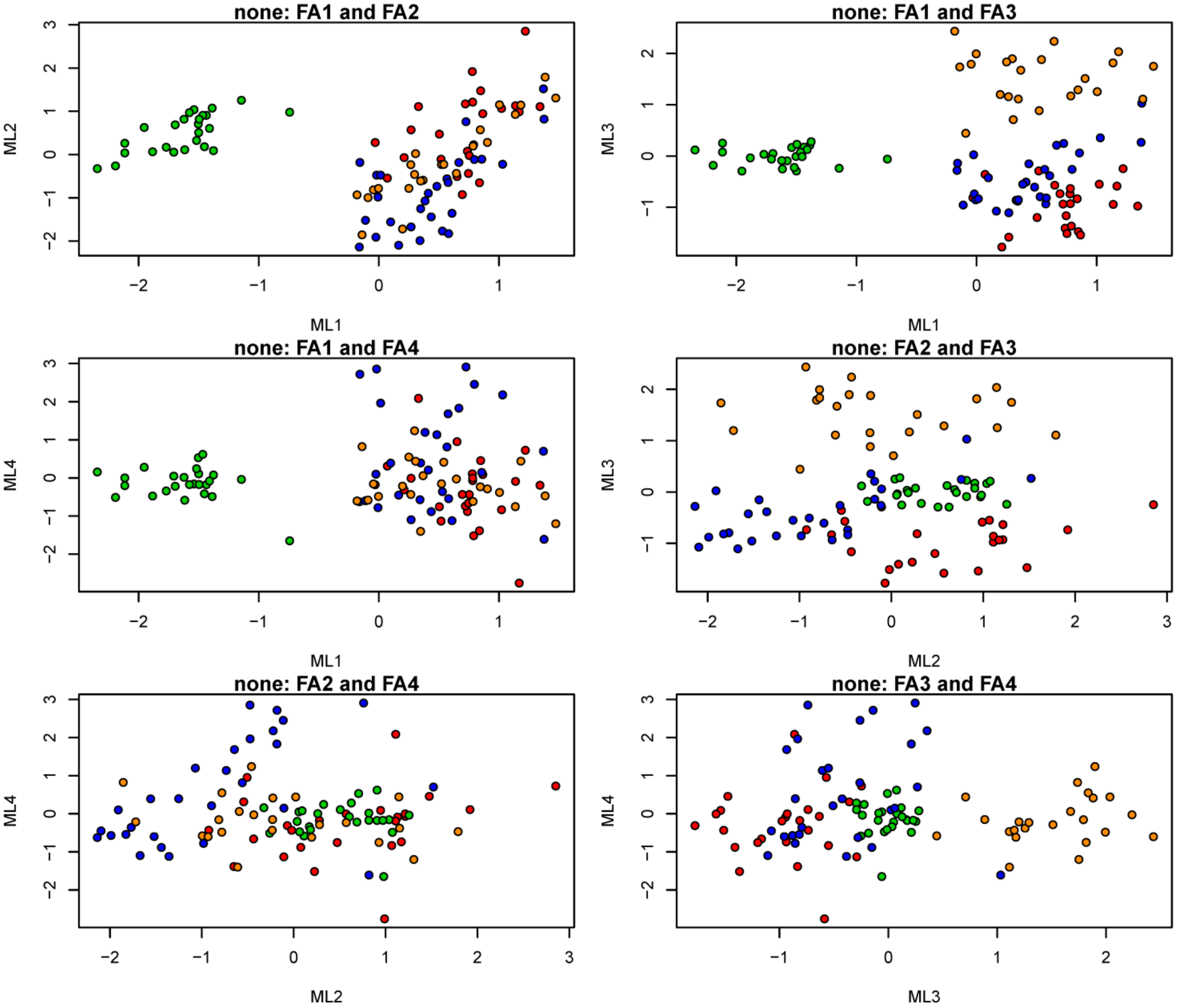
Iris Data Set only one factor:



Multiple Tissue Data Set: 4 factors

n features with largest variance ensure a full rank covariance matrix.

Factor Analysis



FA without rotation

FA1 separates prostate (green)

FA2 separates breast (red) from lung (blue) → not very good

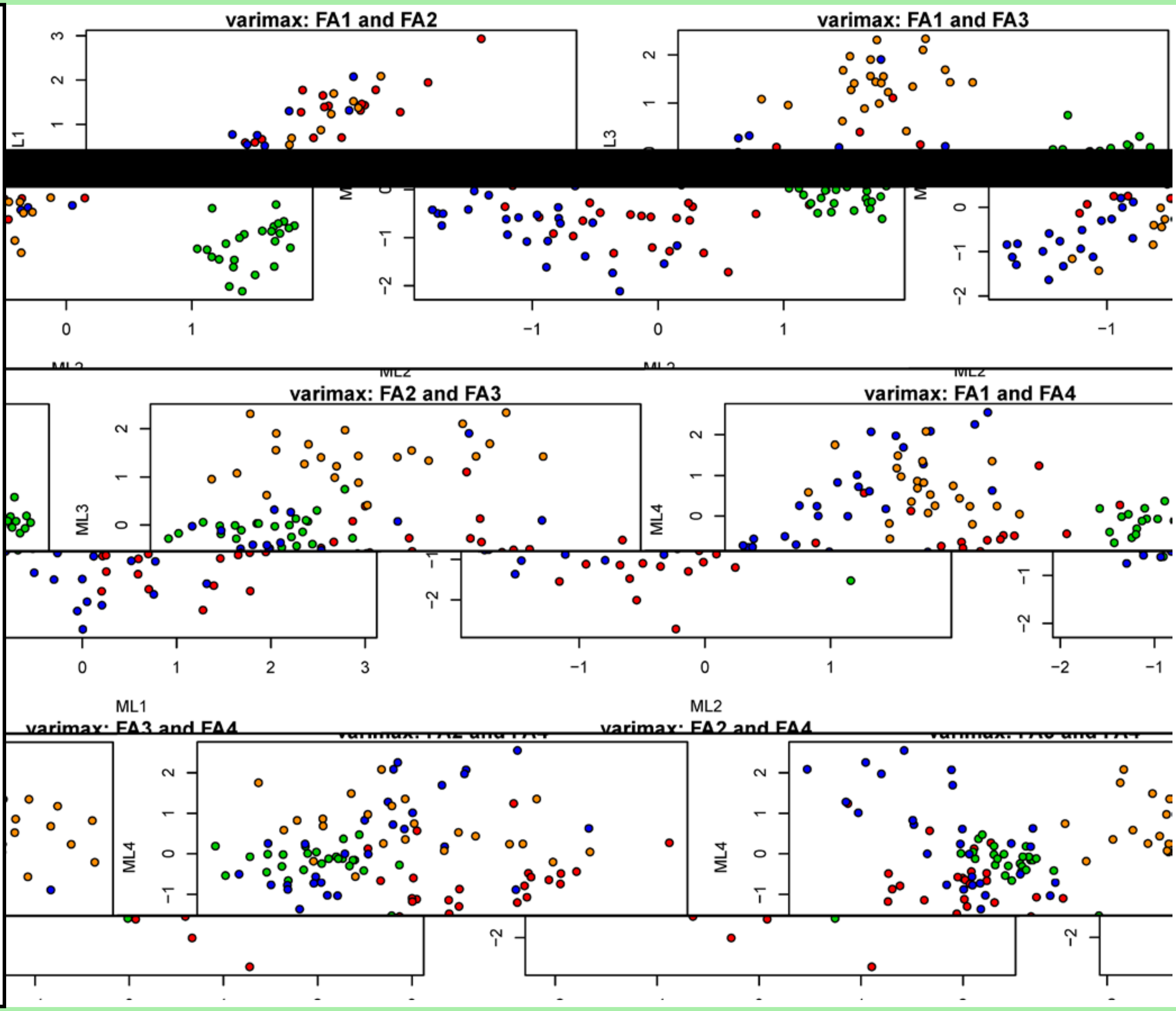
FA3 separates colon (orange)

FA4 separates part of lung (blue)

Factor Analysis

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

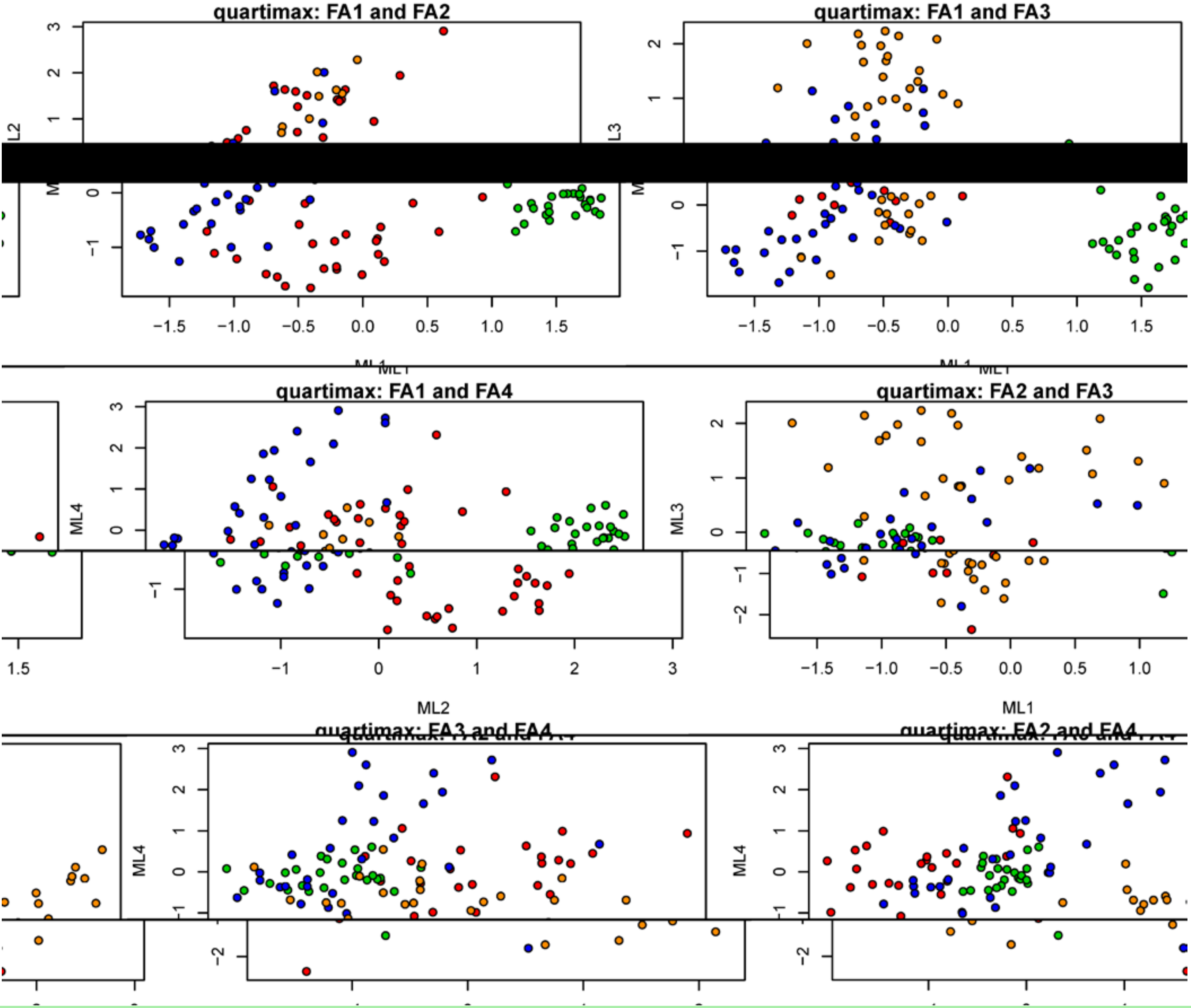


[redacted]

FA with varimax rotation

separation is slightly worse that without rotations

Factor Analysis



FA with quartimax rotation

result is similar to varimax rotation

separation is slightly worse that without rotations

Scaling and Projection Methods

projection of the data to a low-dimensional space (“scaling”)

- visualize the data

- represent the data in a low-dimensional space for further processing:
    - model selection using low-complex model classes
    - low-dimensional representation can capture only the main structures
    - noise and outliers are not represented

Projection Pursuit

Projection pursuit:
least Gaussian ("interesting") projections of the data

how to define non-Gaussianity?

covariance and mean given: Gaussian distribution maximizes the entropy

Objective: minimize $H(t)$ for $t = \boldsymbol{w}^T \boldsymbol{x}$
t is normalized to zero mean and unit variance

This is difficult to optimize
→ finding unimodal super-Gaussians
→ finding multimodal distributions

Other criteria are given for ICA: kurtosis and different contrast functions which measure non-Gaussianity

Multidimensional Scaling

Multidimensional Scaling (MDS):

projection to a low-dimensional space while keeping the distances between data points

$$\begin{aligned} \boldsymbol{y}_i &= f(\boldsymbol{x}_i; \boldsymbol{w}) \\ \delta_{ij} &= \| \boldsymbol{x}_i - \boldsymbol{x}_j \| \\ d_{ij} &= \| \boldsymbol{y}_i - \boldsymbol{y}_j \| \end{aligned}$$

goal: $d = \delta$

$$R_1(d, \delta) = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2} \propto \sum_{i<j} (d_{ij} - \delta_{ij})^2$$

“Kruskal's measure” penalizes large errors

$$R_2(d, \delta) = \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$$

fractional (relative) errors

$$R_3(d, \delta) = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}} \propto \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}$$

“Sammon mapping” compromise

$\propto$ means factors constant in the parameters $\boldsymbol{w}$

Multidimensional Scaling

derivatives used in gradient based methods:

$$\frac{\partial}{\partial \boldsymbol{y}_k} R_1(d, \delta) \;=\; \frac{2}{\sum_{i<j} \delta_{ij}^2} \sum_{j \neq k} \left(d_{kj} \;-\; \delta_{kj}\right) \frac{\boldsymbol{y}_k \;-\; \boldsymbol{y}_j}{d_{kj}}$$

$$\frac{\partial}{\partial \boldsymbol{y}_k} R_2(d, \delta) \;=\; 2 \sum_{j \neq k} \frac{d_{kj} \;-\; \delta_{kj}}{\delta_{kj}^2} \frac{\boldsymbol{y}_k \;-\; \boldsymbol{y}_j}{d_{kj}}$$

$$\frac{\partial}{\partial \boldsymbol{y}_k} R_3(d, \delta) \;=\; \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} \;-\; \delta_{kj}}{\delta_{kj}} \frac{\boldsymbol{y}_k \;-\; \boldsymbol{y}_j}{d_{kj}}$$

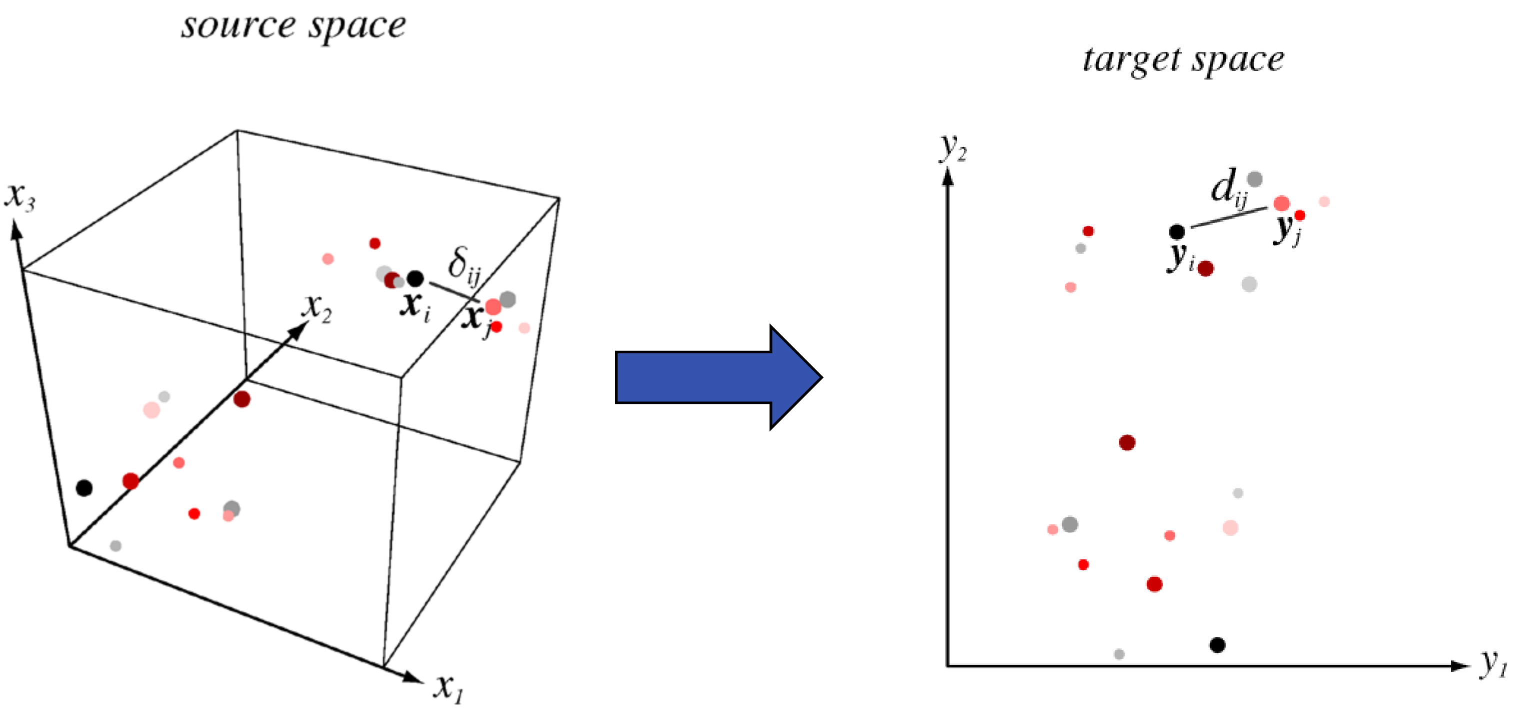
R viewed as potential function → derivatives are forces

Multidimensional Scaling

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

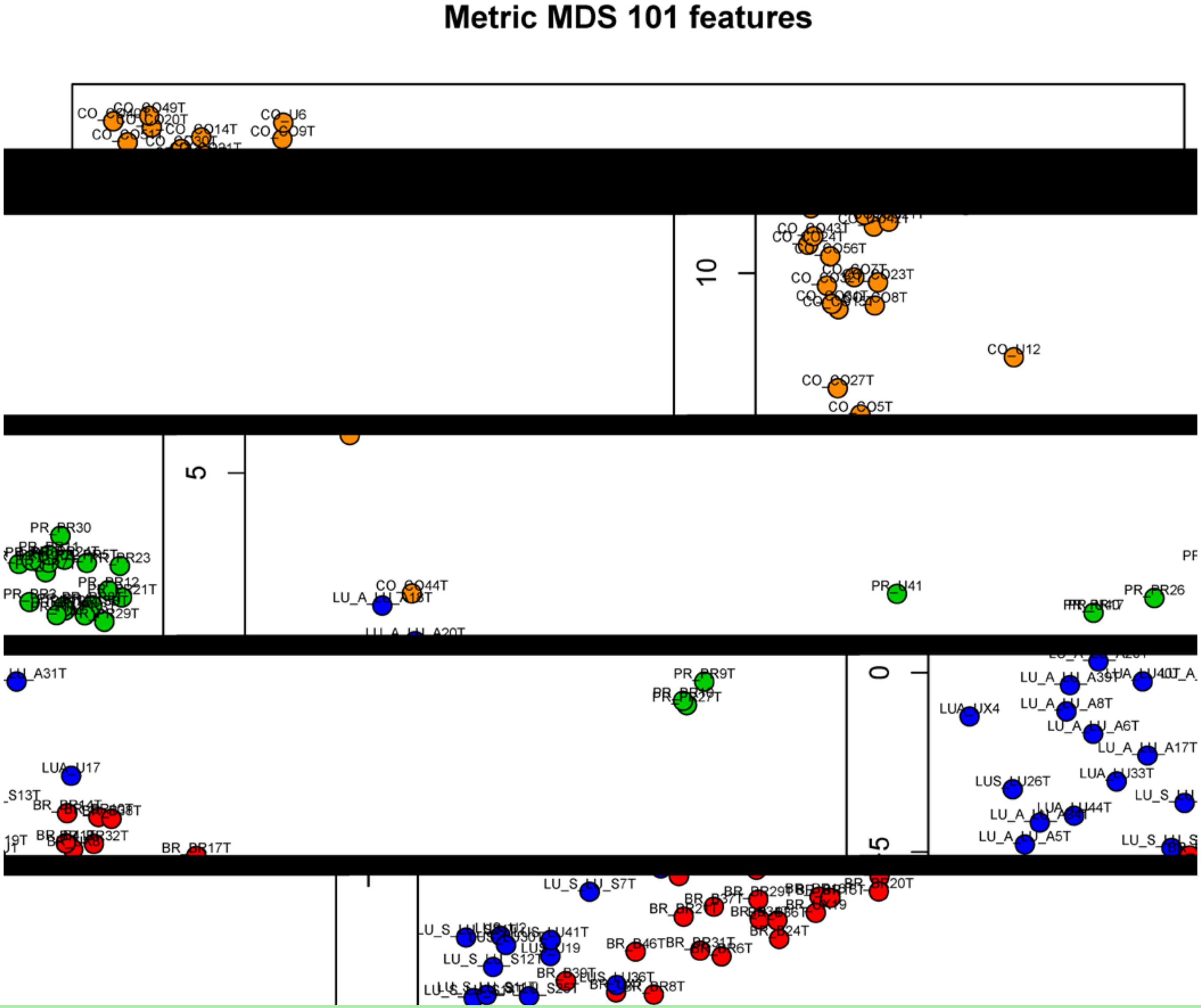
example from Duda, 2001, multidimensional scaling from a 3-dimensional space to a 2-dimensional space (right).



Multidimensional Scaling

metric multidimensional scaling or principal coordinates analysis is applied to the multiple tissue data set.

Multidimensional Scaling



[redacted]

[redacted]

[redacted]

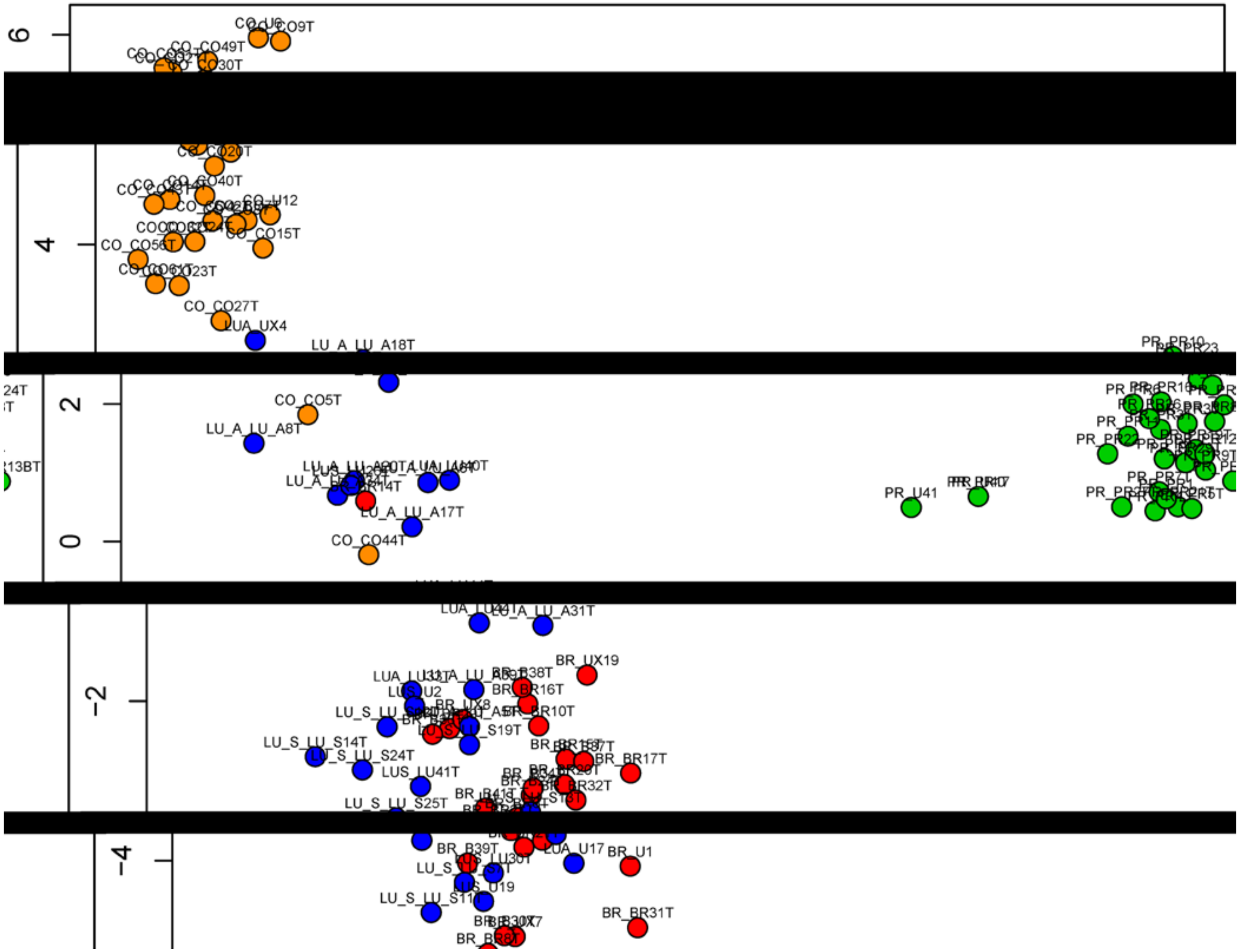
[redacted]

MDS for multiple tissues

the 101 features with largest variance are selected

Multidimensional Scaling

Metric MDS 13 features



[redacted]

[redacted]

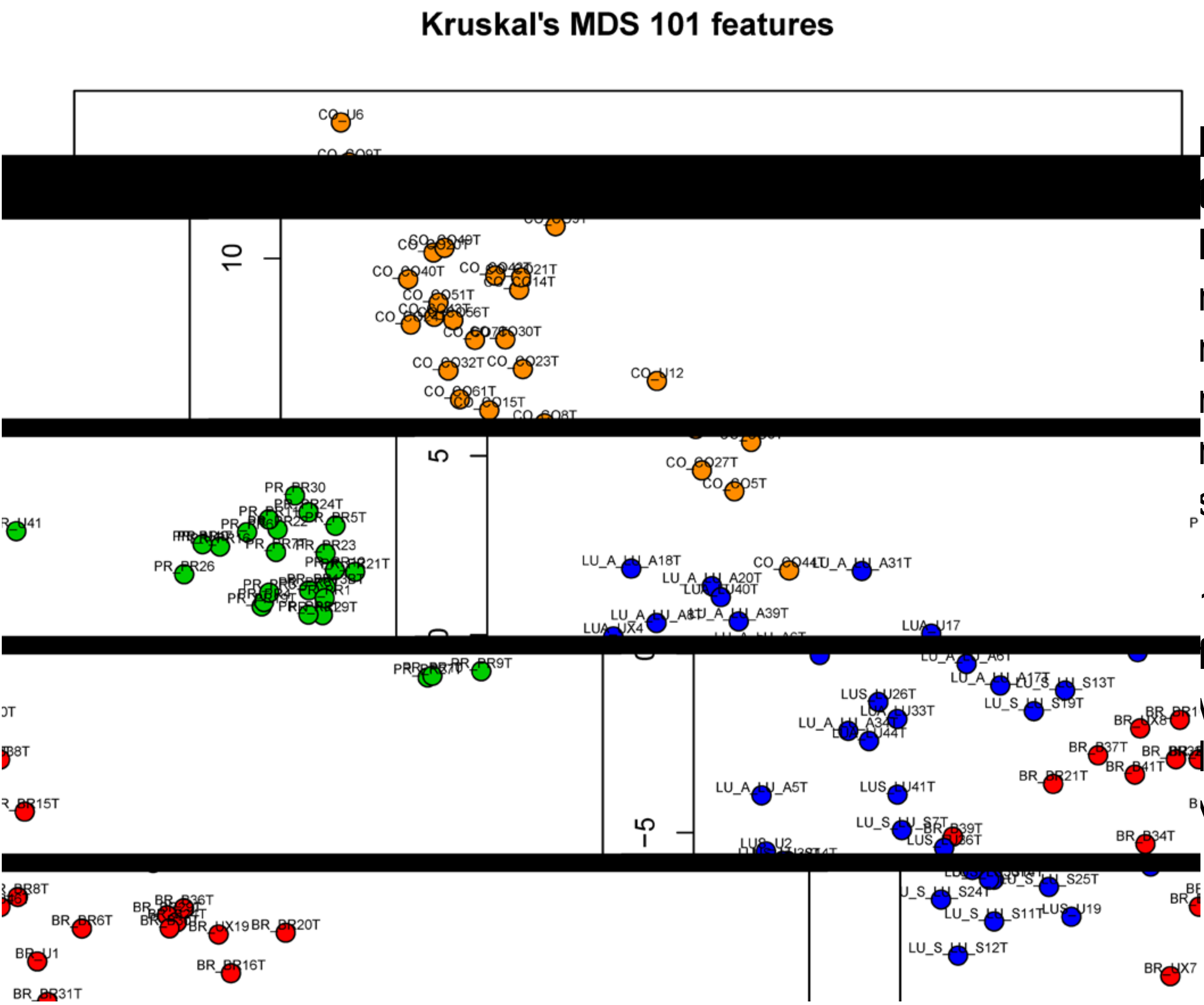
[redacted]

[redacted]

MDS for multiple tissues

the 13 features with largest variance are selected

Multidimensional Scaling



Kruskal's MDS 101 features

[redacted]

[redacted]

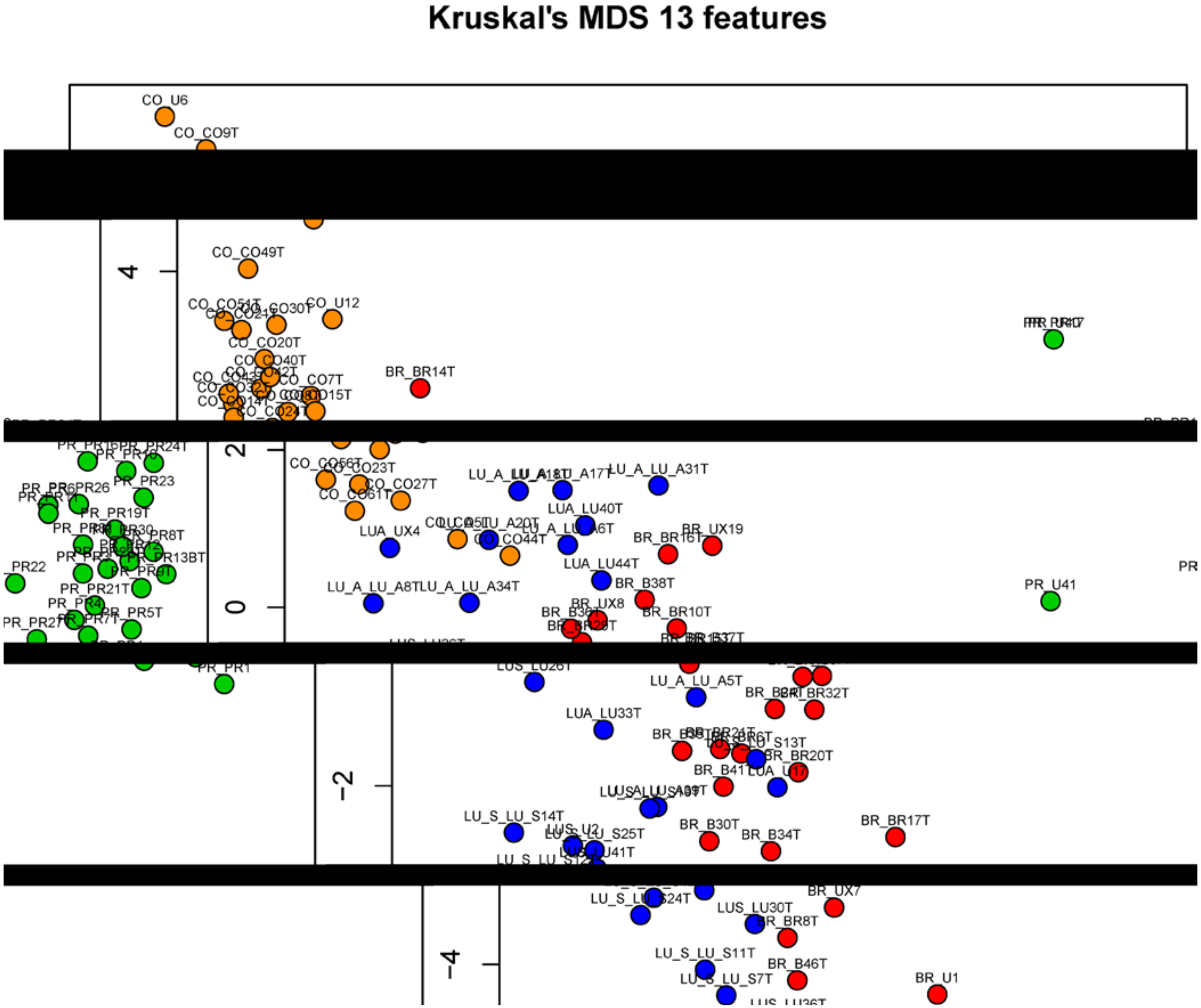
[redacted]

[redacted]

Multiple tissues: Kruskal's non-metric multidimensional scaling

101 features with the largest variance

Multidimensional Scaling

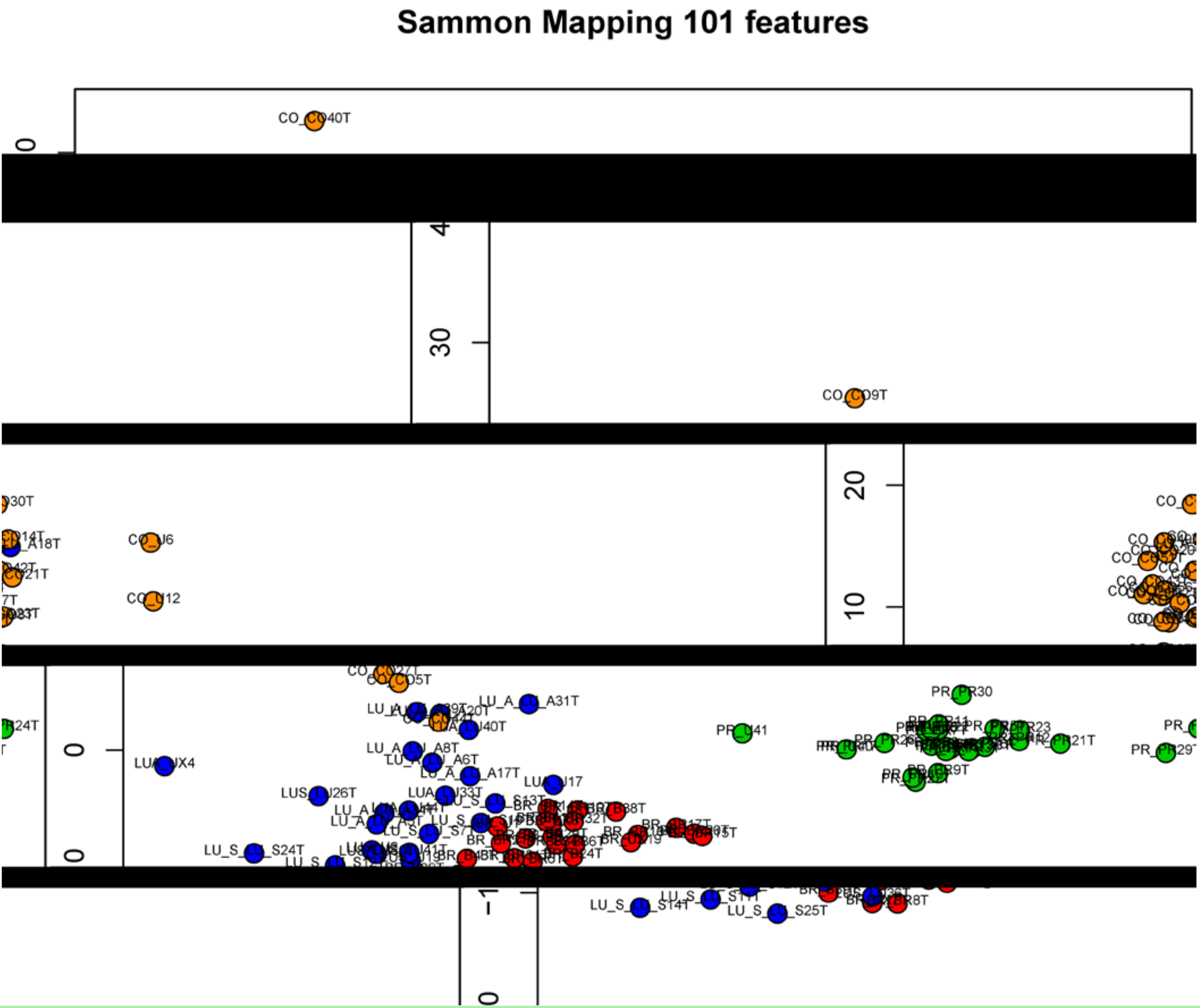


[redacted]
[redacted]
[redacted]
[redacted]

Multiple tissues: Kruskal's non-metric multidimensional scaling

13 features with the largest variance

Multidimensional Scaling


[redacted]

Multiple tissues: Sammon's non-linear mapping

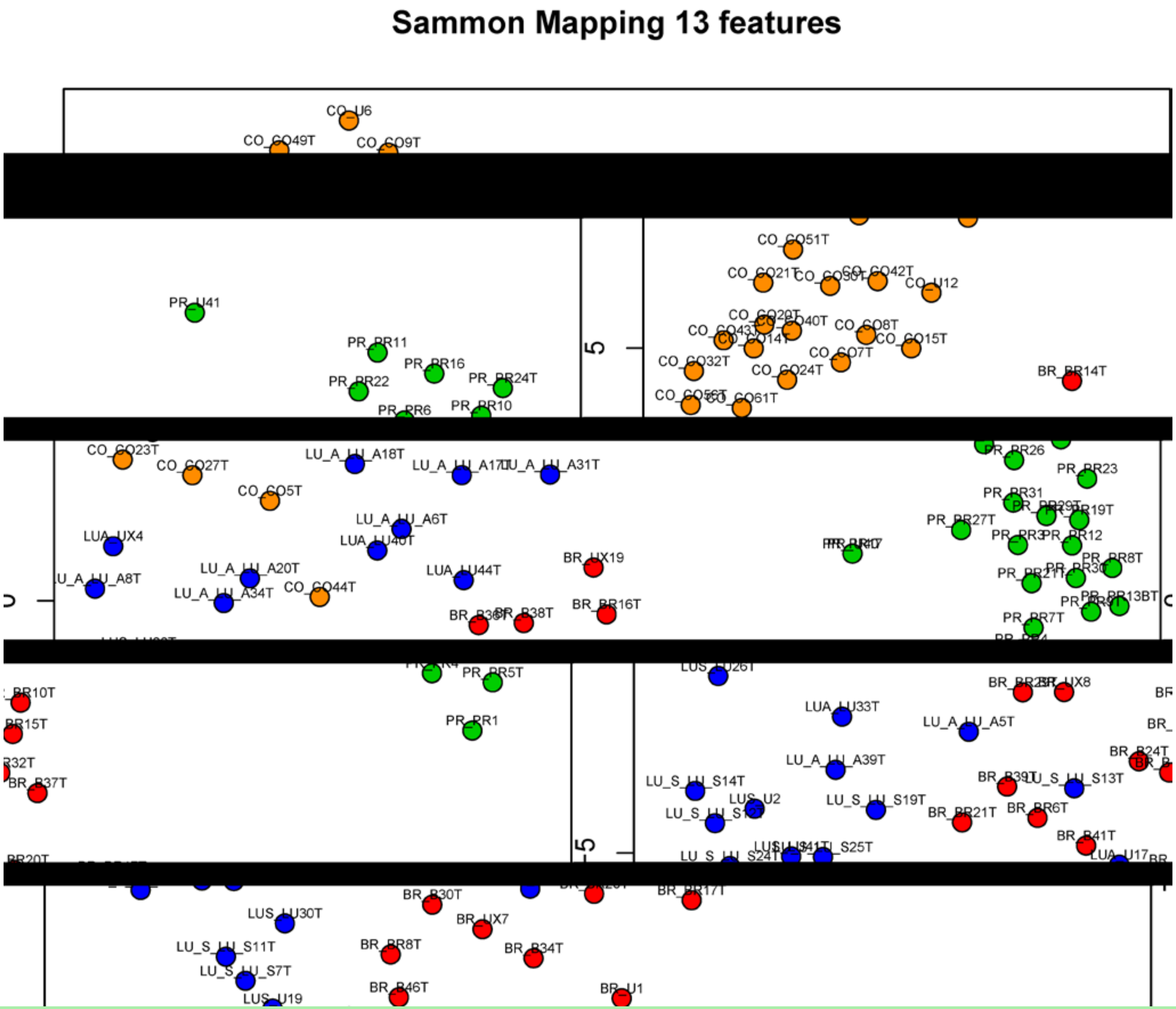
101 features with the largest variance

worse than metric or Kruskal's measure

Multidimensional Scaling

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples



[redacted]

[redacted]

[redacted]

[redacted]

Multiple tissues: Sammon's non-linear mapping

13 features with the largest variance

clusters are more spread out

Non-negative Matrix Factorization

Non-negative matrix factorization (NFM)
is a matrix factorization method where all matrix entries are assumed to be positive

the non-negativity constraints make the representation of the observations purely additive: a parts-based representation, where parts are added to the observation but not subtracted (e.g. images)

$$\boldsymbol{X} \in \mathbb{R}^{n \times m} \qquad \boldsymbol{Y} \in \mathbb{R}^{n \times l} \qquad \boldsymbol{U} \in \mathbb{R}^{m \times l}$$

$$0 \leq X_{ij} \qquad 0 \leq Y_{ik} = [\boldsymbol{y}_k]_i \qquad 0 \leq U_{jk} = [\boldsymbol{u}_k]_j$$

$$\boldsymbol{X} \ = \ \boldsymbol{Y} \ \boldsymbol{U}^T \ = \ \sum_{k=1}^{l} \boldsymbol{y}_k \ \boldsymbol{u}_k^T$$

Non-negative Matrix Factorization

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

Objective 1: Kullback-Leibler divergence (positive matrices)

$$\mathrm{D}(\boldsymbol{A} \parallel \boldsymbol{B}) \ = \ \sum_{ij} \left(A_{ij} \ \log \frac{A_{ij}}{B_{ij}} \ + \ A_{ij} \ - \ B_{ij} \right) \quad \text{for} \quad \sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$$

minimize the Kullback-Leibler divergence $\mathrm{D}(\boldsymbol{X} \parallel \boldsymbol{Y} \ \boldsymbol{U}^T)$ by gradient descent gives:

$$Y_{ik} \ = \ Y_{ik} \, \frac{\sum_{j=1}^{m} U_{jk} \ X_{ij} \ / \ \left(\boldsymbol{Y} \ \boldsymbol{U}^T\right)_{ij}}{\sum_{j=1}^{m} U_{jk}}$$

$$U_{jk} \ = \ U_{jk} \, \frac{\sum_{i=1}^{n} Y_{ik} \ X_{ij} \ / \ \left(\boldsymbol{Y} \ \boldsymbol{U}^T\right)_{ij}}{\sum_{i=1}^{n} Y_{ik}}$$

Non-negative Matrix Factorization

Objective 2: Euclidean distance (Frobenius norm):

$$\|\boldsymbol{X} - \boldsymbol{Y}\,\boldsymbol{U}^T\|_F^2 \qquad \|\boldsymbol{A} - \boldsymbol{B}\|_F^2 = \sum_{ij} (A_{ij} - B_{ij})^2$$

$$Y_{ik} = Y_{ik} \frac{(\boldsymbol{X}\,\boldsymbol{U})_{ik}}{(\boldsymbol{Y}\,\boldsymbol{U}^T\,\boldsymbol{U})_{ik}}$$

multiply $\boldsymbol{X} = \boldsymbol{Y}\,\boldsymbol{U}^T$ from right by $\boldsymbol{U}$

$$U_{jk} = U_{jk} \frac{(\boldsymbol{Y}^T\,\boldsymbol{X})_{kj}}{(\boldsymbol{Y}^T\,\boldsymbol{Y}\,\boldsymbol{U}^T)_{kj}}$$

multiply $\boldsymbol{X} = \boldsymbol{Y}\,\boldsymbol{U}^T$ from left by $\boldsymbol{Y}^T$

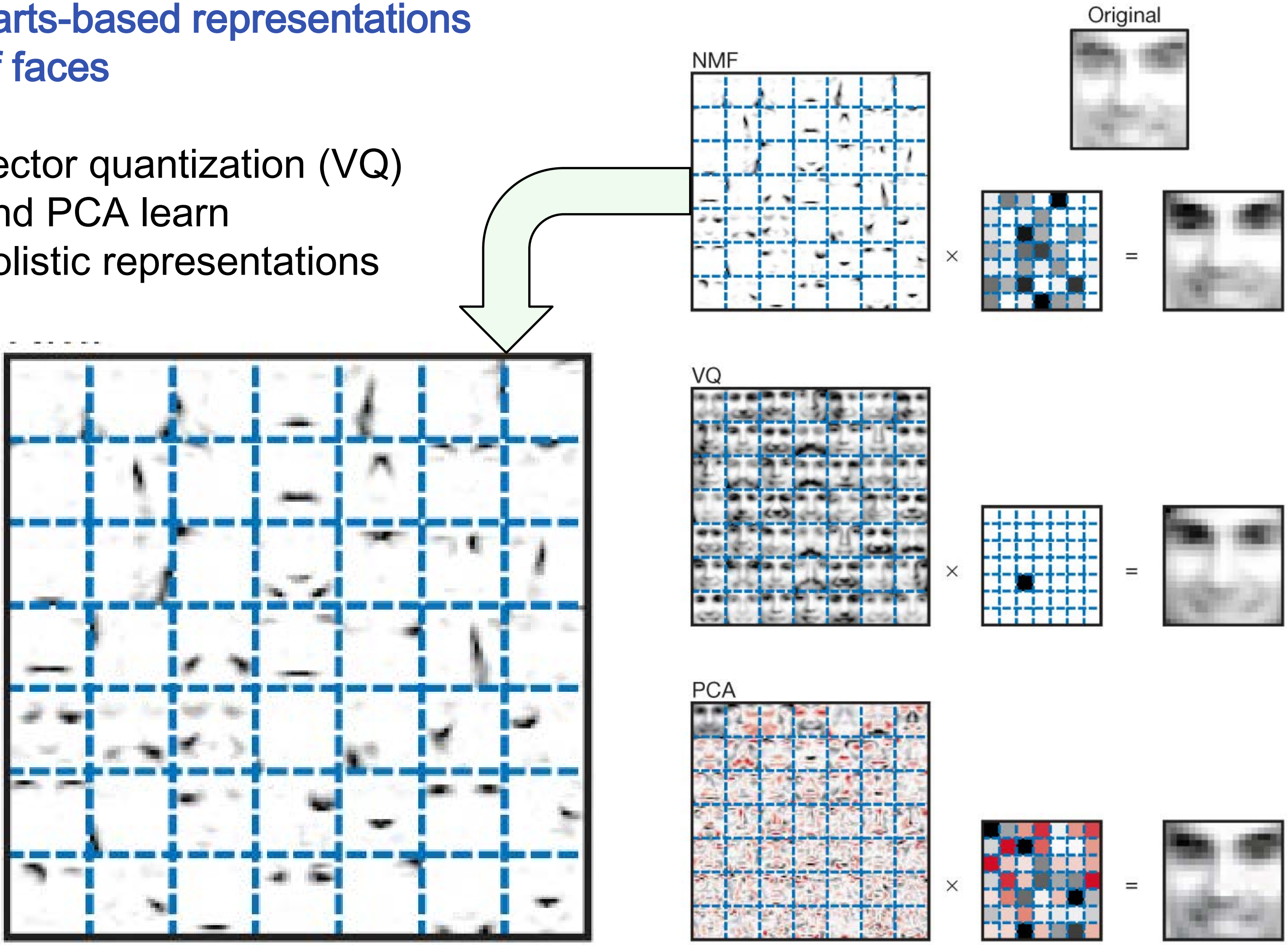
For a fixed point, the left and the right hand side have to be equal

NFM has been extended to sparse NFM (both decomposition matrices); sparse $\boldsymbol{Y}$ → few parts are present
spares $\boldsymbol{U}$ → few measurements indicate part
For example gene expression: part = pathway, few genes in pathway and few pathways are active in a sample

Non-negative Matrix Factorization

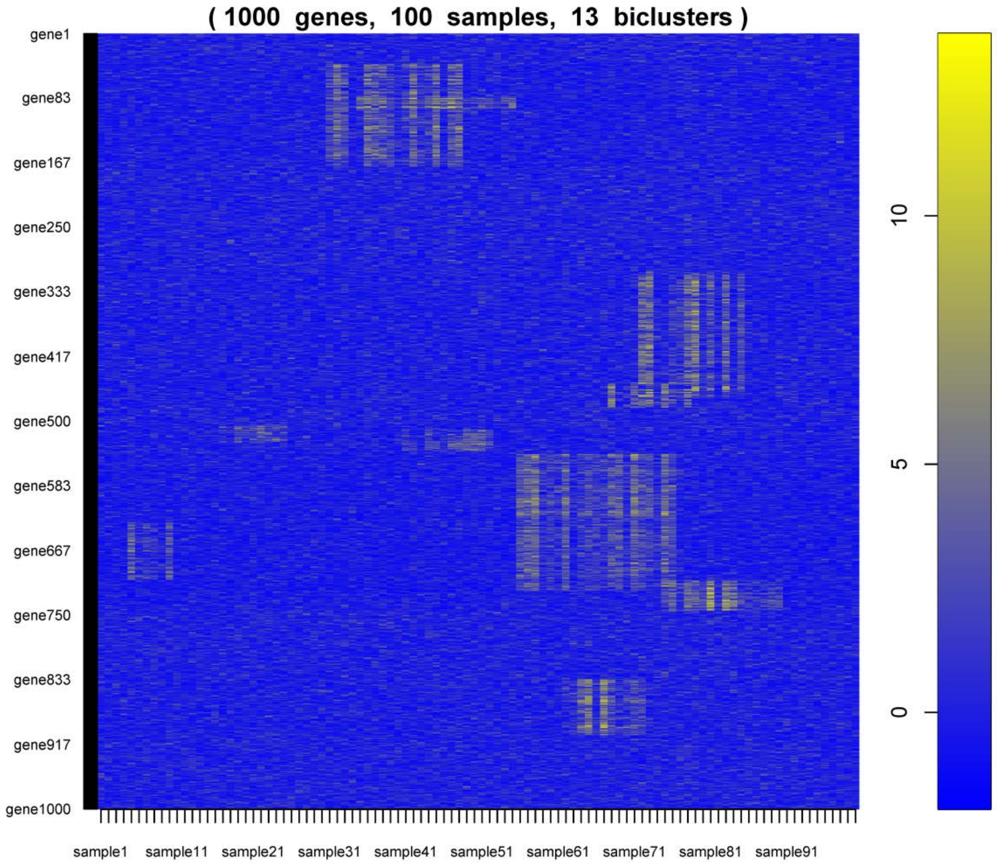
parts-based representations of faces

vector quantization (VQ) and PCA learn holistic representations



Non-negative Matrix Factorization

positive toy data generated by `FABIA` biclustering package



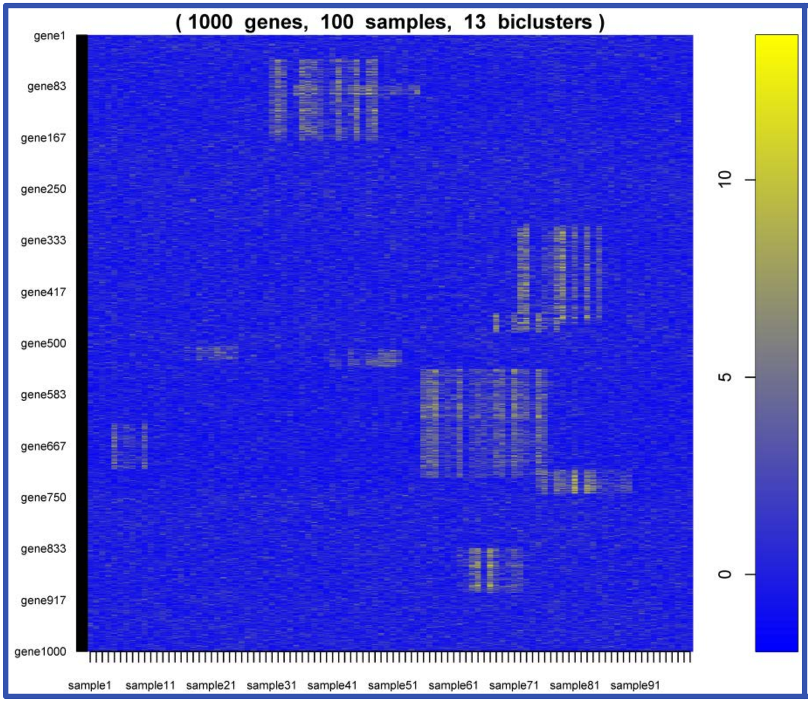
Non-negative Matrix Factorization

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

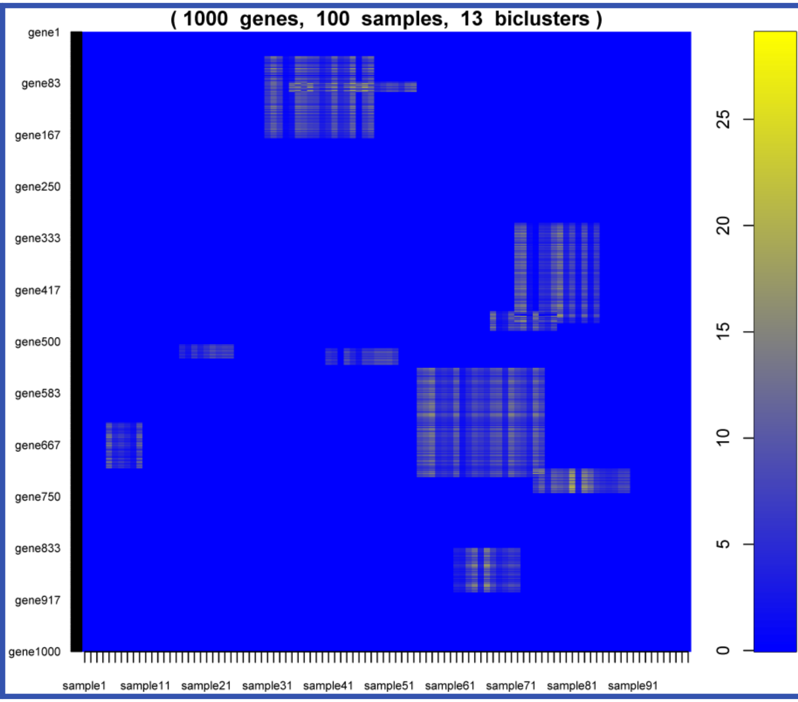
6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

The data contains blocks of patterns. For visualization purposes only, the blocks are constructed by adjacent row or column elements (blocks are the parts).

noisy data

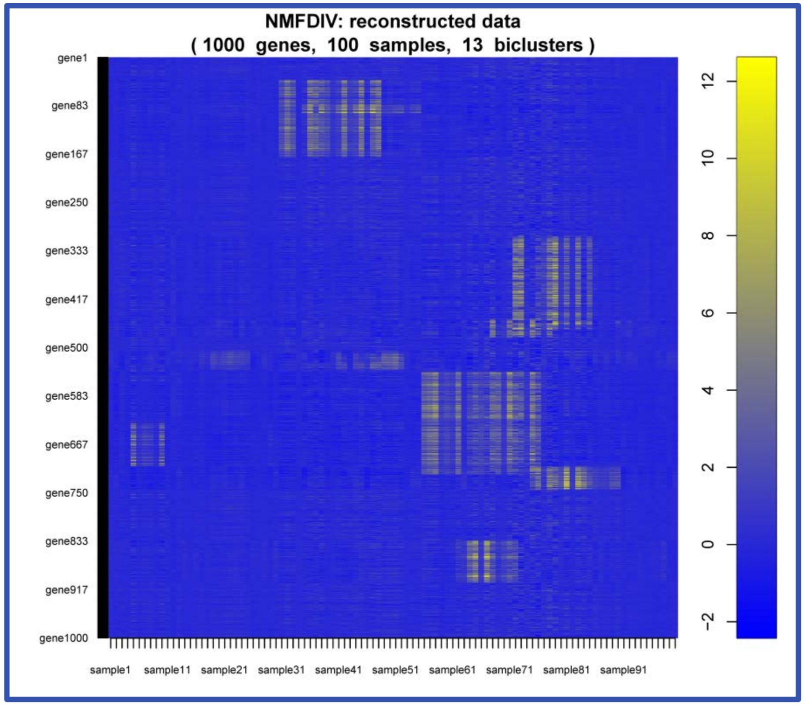


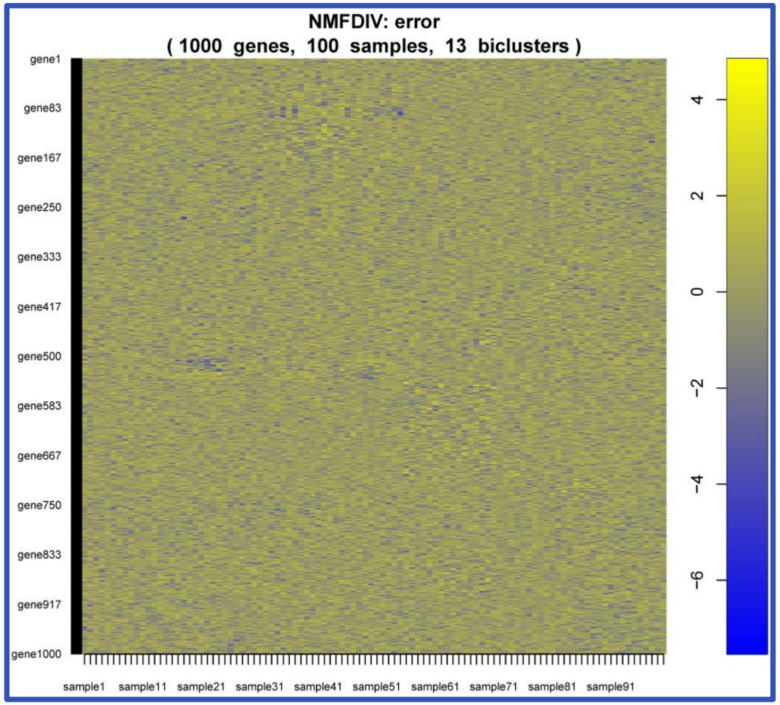
noise-free data



Non-negative Matrix Factorization

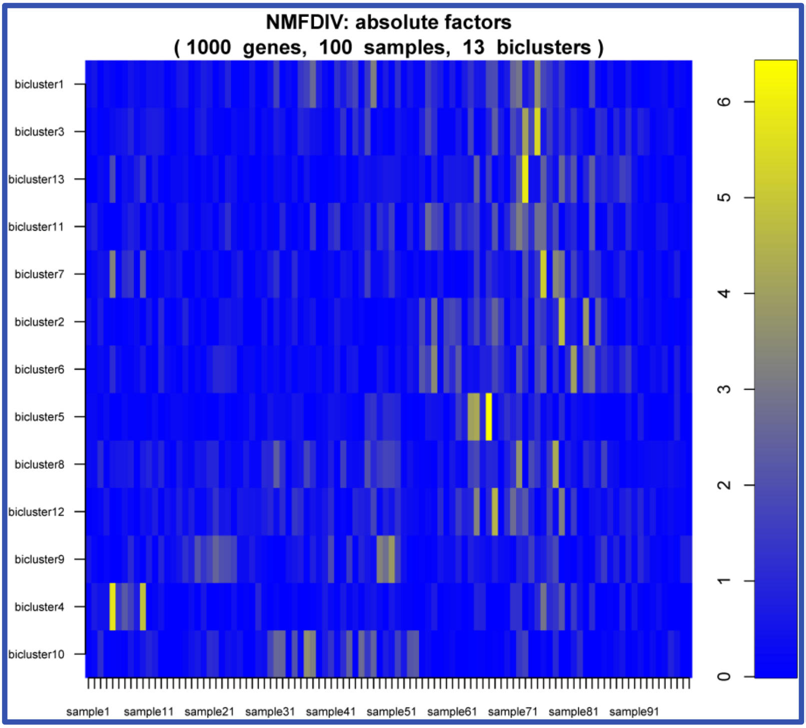
NMF Kullback-Leibler divergence using the `fabia` package



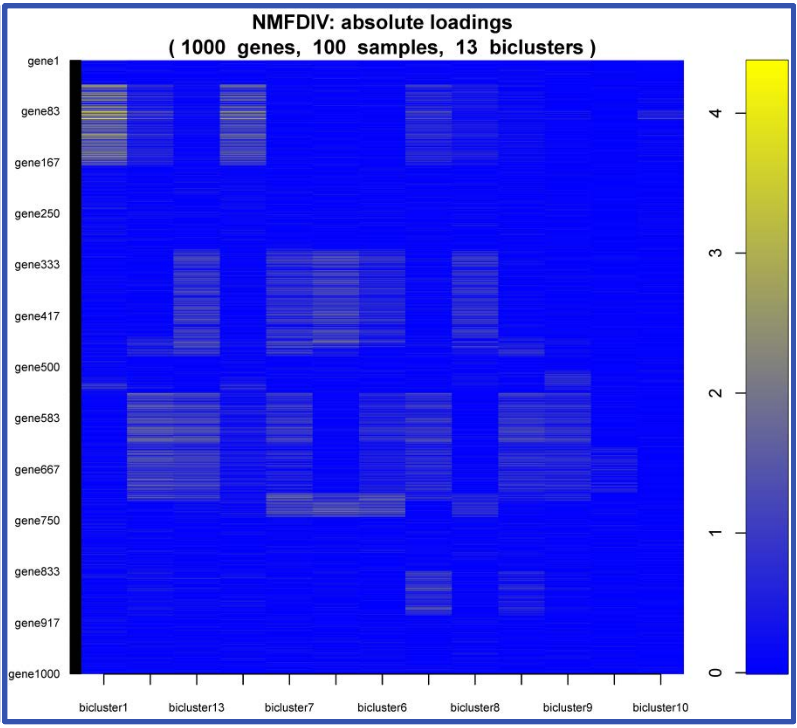


Non-negative Matrix Factorization

NMF Kullback-Leibler divergence



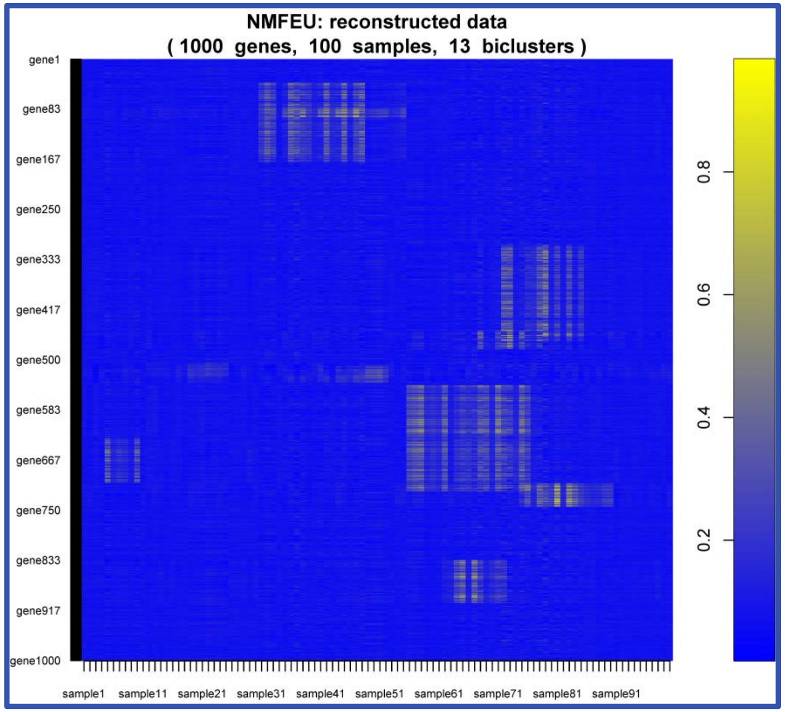
NMFDIV: absolute factors
(1000 genes, 100 samples, 13 biclusters)

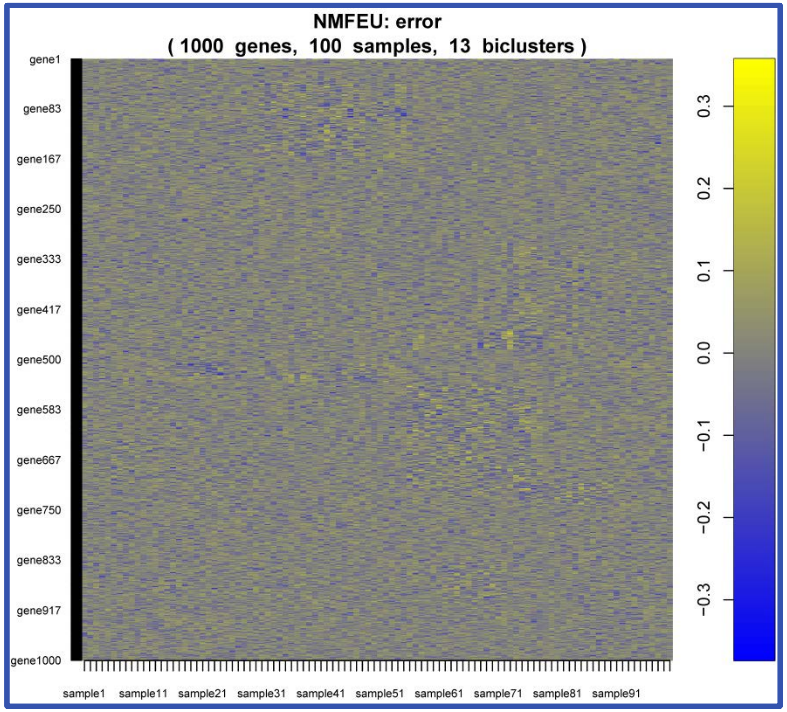


NMFDIV: absolute loadings
(1000 genes, 100 samples, 13 biclusters)

Non-negative Matrix Factorization

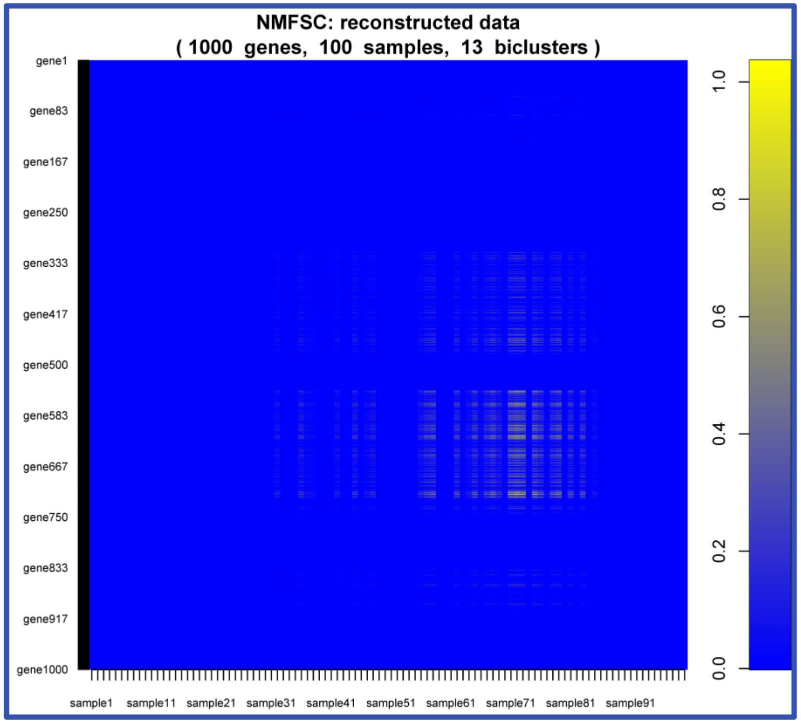
NMF Euclidean distance: `fabia` package

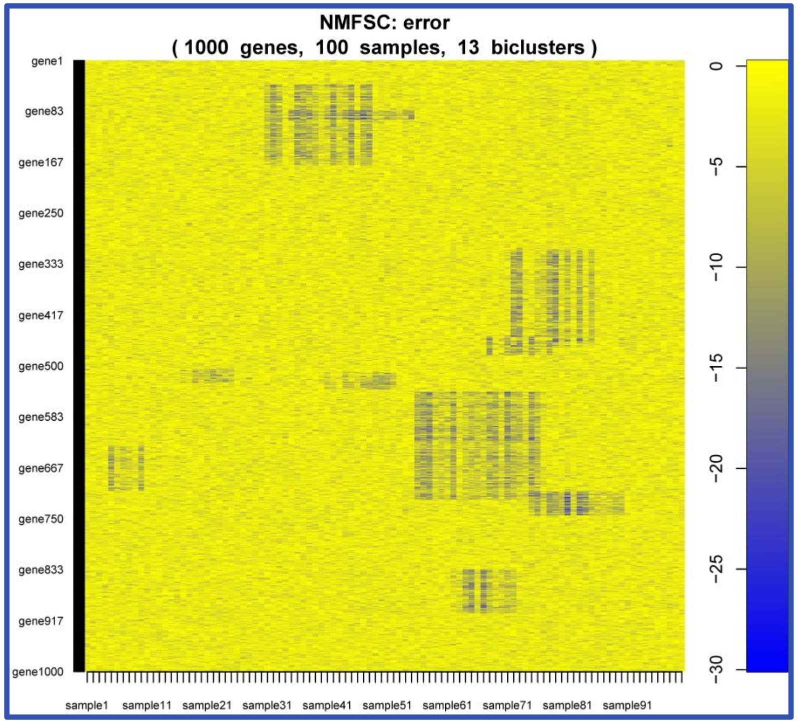




Non-negative Matrix Factorization

NMF with sparseness constraints using the `fabia` package



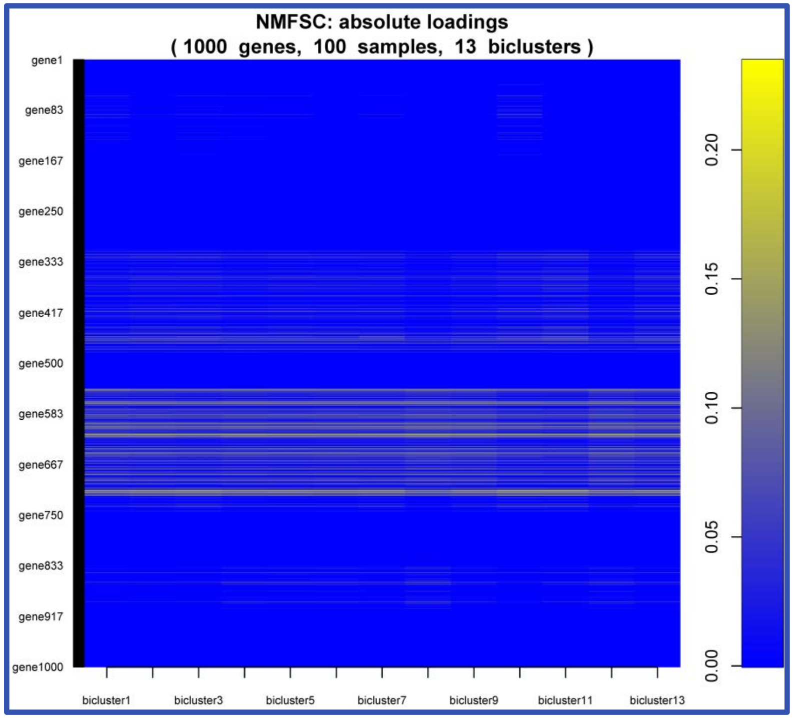


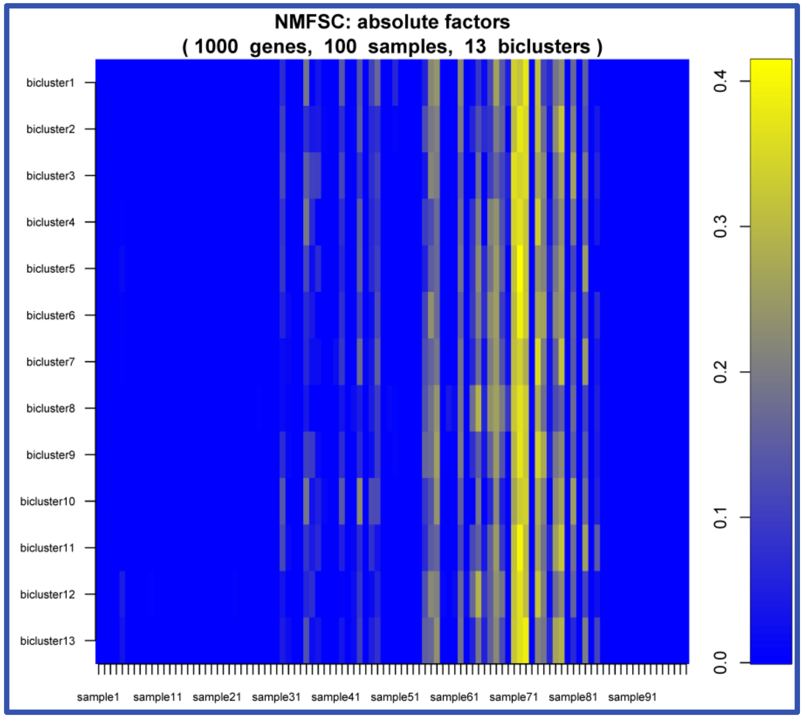
Non-negative Matrix Factorization

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

NMF sparseness constraint.
Not all blocks are detected: too much sparseness enforced
→ difficult to properly adjust the sparseness parameter



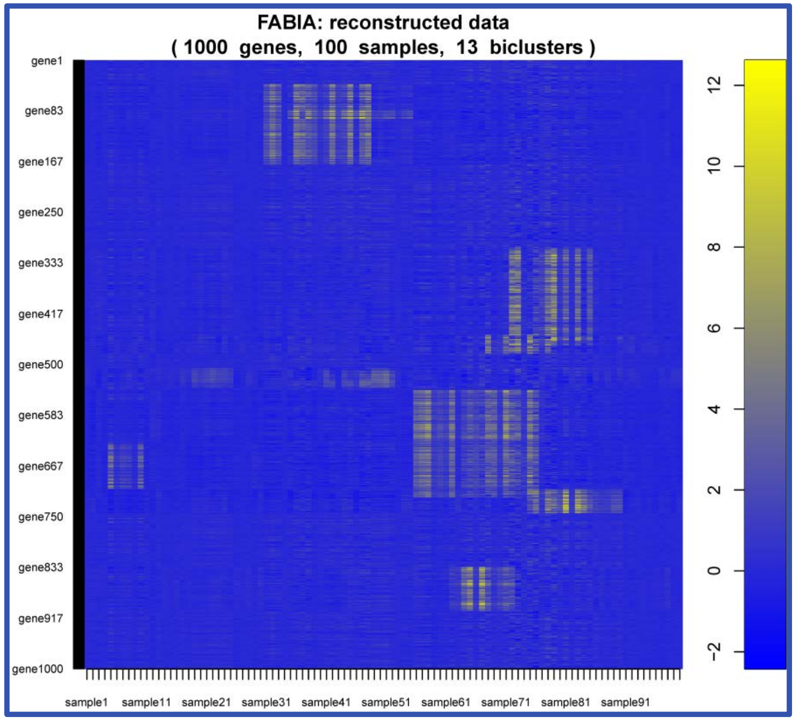


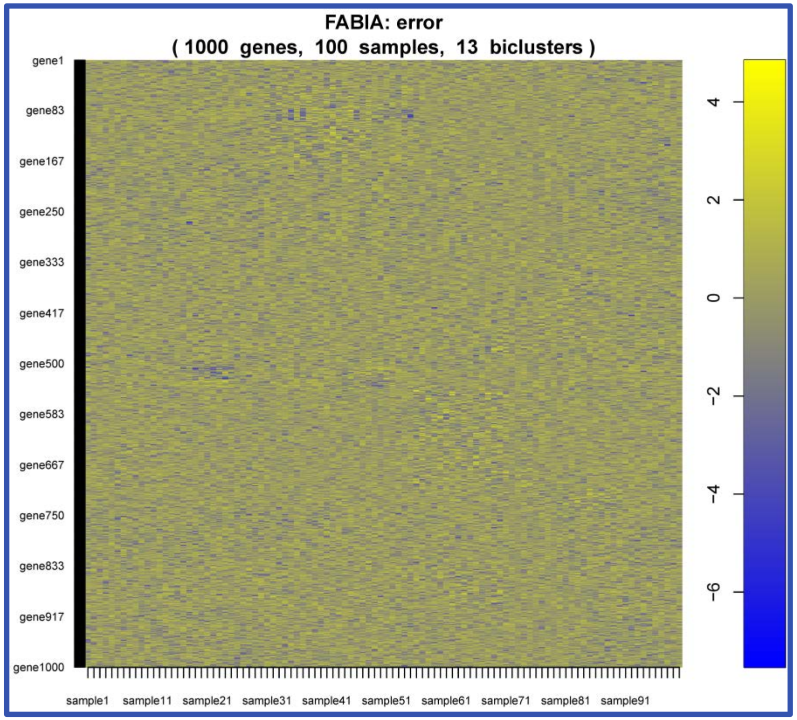
Non-negative Matrix Factorization

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

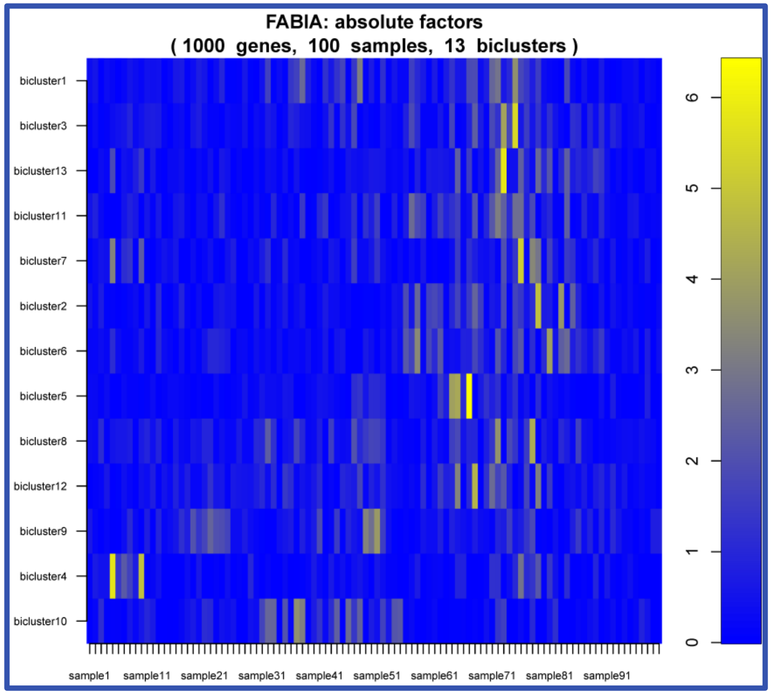
biclustering with FABIA

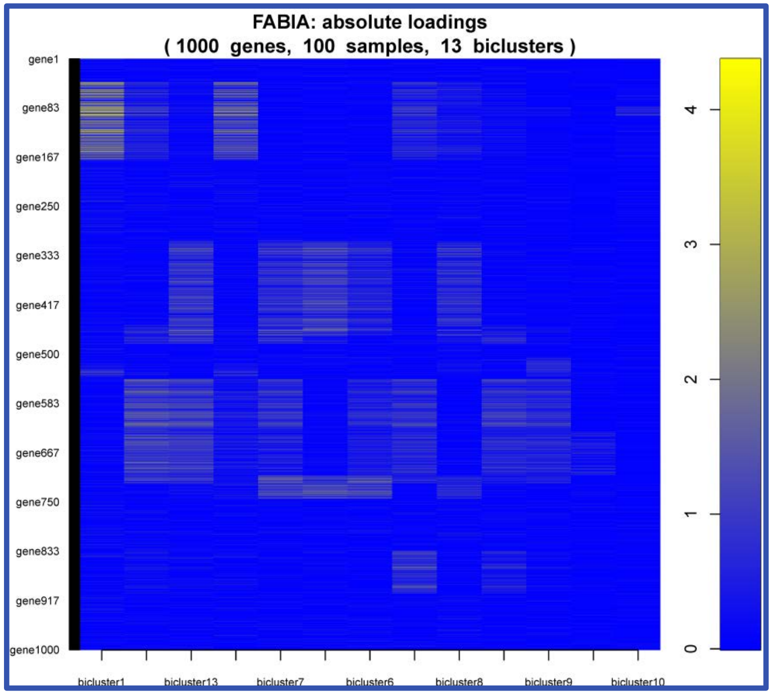

FABIA: reconstructed data
(1000 genes, 100 samples, 13 biclusters)


FABIA: error
(1000 genes, 100 samples, 13 biclusters)

Non-negative Matrix Factorization

biclustering with FABIA





Locally Linear Embedding

Locally linear embedding (LLE) computes low-dimensional, neighborhood-preserving embeddings / representations.
LLE performs nonlinear mappings. The objective is

$$\varepsilon(\boldsymbol{W}) \;=\; \sum_i \left\| \boldsymbol{x}_i \;-\; \sum_{j=1}^{k} W_{ij} \boldsymbol{x}_j \right\|^2 \qquad \sum_{j=1}^{k} W_{ij} \;=\; 1$$

Optimized by constrained least squares using neighbors $\boldsymbol{x}_j$ of $\boldsymbol{x}_i$
The solutions of this problem are invariant to rotations, rescalings, and translations of $\boldsymbol{x}_i$

Down-projection optimizes where the W_{ij} are fixed

$$\Phi(\boldsymbol{Y}) \;=\; \sum_i \left\| \boldsymbol{y}_i \;-\; \sum_{j=1}^{k} W_{ij} \boldsymbol{y}_j \right\|^2$$

The representation of $\boldsymbol{x}_i$ by its neighbors is transferred to $\boldsymbol{y}_i$

$$\Phi(\boldsymbol{Y}) \;=\; \sum_{ij} M_{ij} \; \boldsymbol{y}_i^T \boldsymbol{y}_j$$

δ_{ij} :1 for $\boldsymbol{i}$=$\boldsymbol{j}$, 0 otherwise

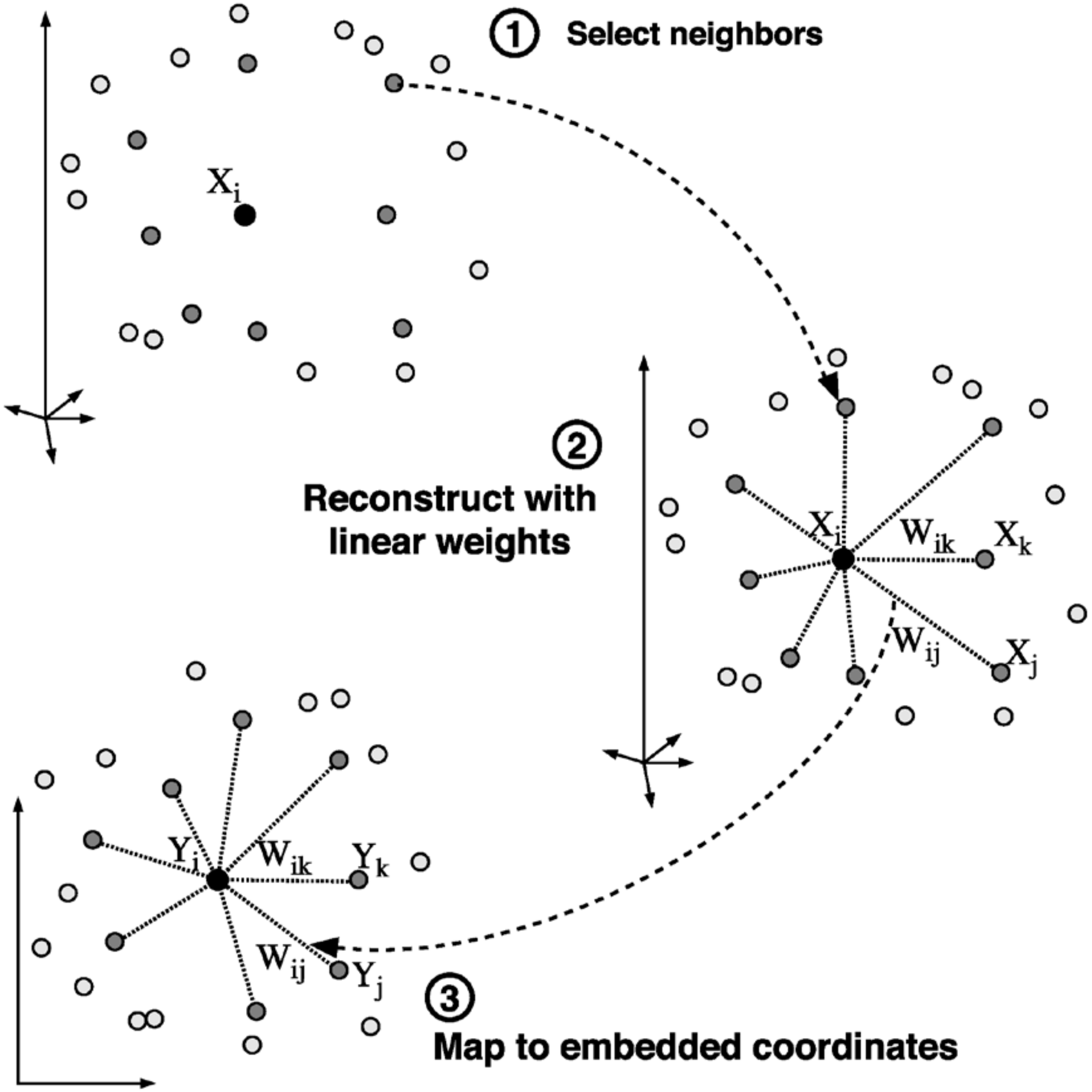
$$M_{ij} \;=\; \delta_{ij} - W_{ij} \;-\; W_{ji} \;+\; \sum_k W_{ki} \, W_{kj}$$

$$\boldsymbol{M} \;=\; (\boldsymbol{I} \;-\; \boldsymbol{W})^T \; (\boldsymbol{I} \;-\; \boldsymbol{W})$$

optimal embedding: bottom $\boldsymbol{d}$ eigenvectors of $\boldsymbol{M}$, except the last one

Locally Linear Embedding

steps of the LLE method



Locally Linear Embedding

```
  Given: $\boldsymbol{X}$: $n$ by $m$ matrix consisting of $n$ data items in $m$ dimensions, dimension of embedding space $l$, $k$ number of neighbors, distance measure
Find neighbors in $\boldsymbol{X}$ space
  for $(i = 1\ ;\ i\ \geq\ n\ ;\ i++)$ do
    compute the distance from $\boldsymbol{x}_i$ to every other point $\boldsymbol{x}_j$
    find the $k$ smallest distances
    assign the corresponding points to be neighbors of $\boldsymbol{x}_i$
  end for
Solve for reconstruction weights $\boldsymbol{W}$
  for $(i = 1\ ;\ i\ \geq\ n\ ;\ i++)$ do
    create matrix $\boldsymbol{Z}$ consisting of all neighbors of $\boldsymbol{x}_i$ [d]
    subtract $\boldsymbol{x}_i$ from every row of $\boldsymbol{Z}$
    compute the local covariance $\boldsymbol{C} = \boldsymbol{Z}^T \boldsymbol{Z}$ [e]
    solve linear system $\boldsymbol{C}\boldsymbol{w} = \mathbf{1}$ for $\boldsymbol{w}$ [f]
    set $W_{ij} = 0$ if $j$ is not a neighbor of $i$
    set the remaining elements in the $i$-th row of $\boldsymbol{W}$ equal to $\boldsymbol{w}/\sum_j (w_j)$;
  end for
Compute embedding coordinates $\boldsymbol{Y}$ using weights $\boldsymbol{W}$
  create sparse matrix $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^T (\boldsymbol{I} - \boldsymbol{W})$
  find bottom $l + 1$ eigenvectors of $\boldsymbol{M}$ (corresponding to the $d + 1$ smallest eigenvalues)
  set the $q$-th column of $\boldsymbol{Y}$ to be the $q + 1$ smallest eigenvector (discard the bottom eigenvector $\mathbf{1} = (1, 1, 1, 1 \ldots)$ with eigenvalue zero)
Result $\boldsymbol{Y}$: $n$ by $l$ matrix consisting of $l < m$ dimensional embedding coordinates.
```

Locally Linear Embedding

[a] Notation $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ denote the i-th row of $\boldsymbol{X}$ and $\boldsymbol{Y}$ (in other words the data and embedding coordinates of the i-th point),
$\boldsymbol{M}^T$ denotes the transpose of matrix $\boldsymbol{M}$,
$\boldsymbol{I}$ is the identity matrix,
$\mathbf{1}$ is a column vector of all ones

[b] This can be done in a variety of ways, for example above we compute the k nearest neighbors using Euclidean distance. Other methods such as epsilon-ball include all points within a certain radius or more sophisticated domain specific and/or adaptive local distance metrics.

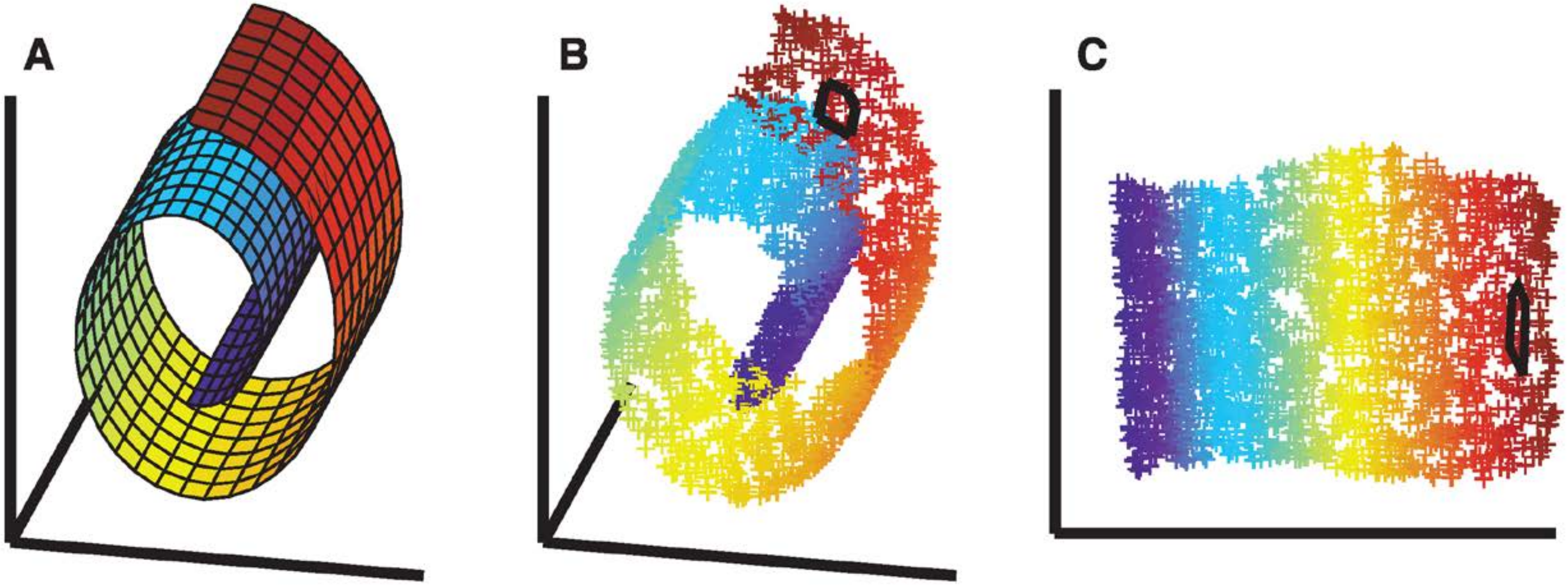
[c] Even for simple neighborhood rules like KNN or epsilon-ball using Euclidean distance, there are highly efficient techniques for computing the neighbors of every point, such as KD trees.

[d] $\boldsymbol{Z}$ consists of all rows of $\boldsymbol{X}$ corresponding to the neighbors of $\boldsymbol{x}_i$ but not $\boldsymbol{x}_i$ itself

[e] If $k > m$, the local covariance will not have full rank, and it should be regularized by setting $\boldsymbol{C} = \boldsymbol{C} + \epsilon \boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix and ϵ is a small constant of order 1e-3 trace($\boldsymbol{C}$). This ensures that the system to be solved in step 2 has a unique solution.

Locally Linear Embedding

LLE for Swiss Roll data

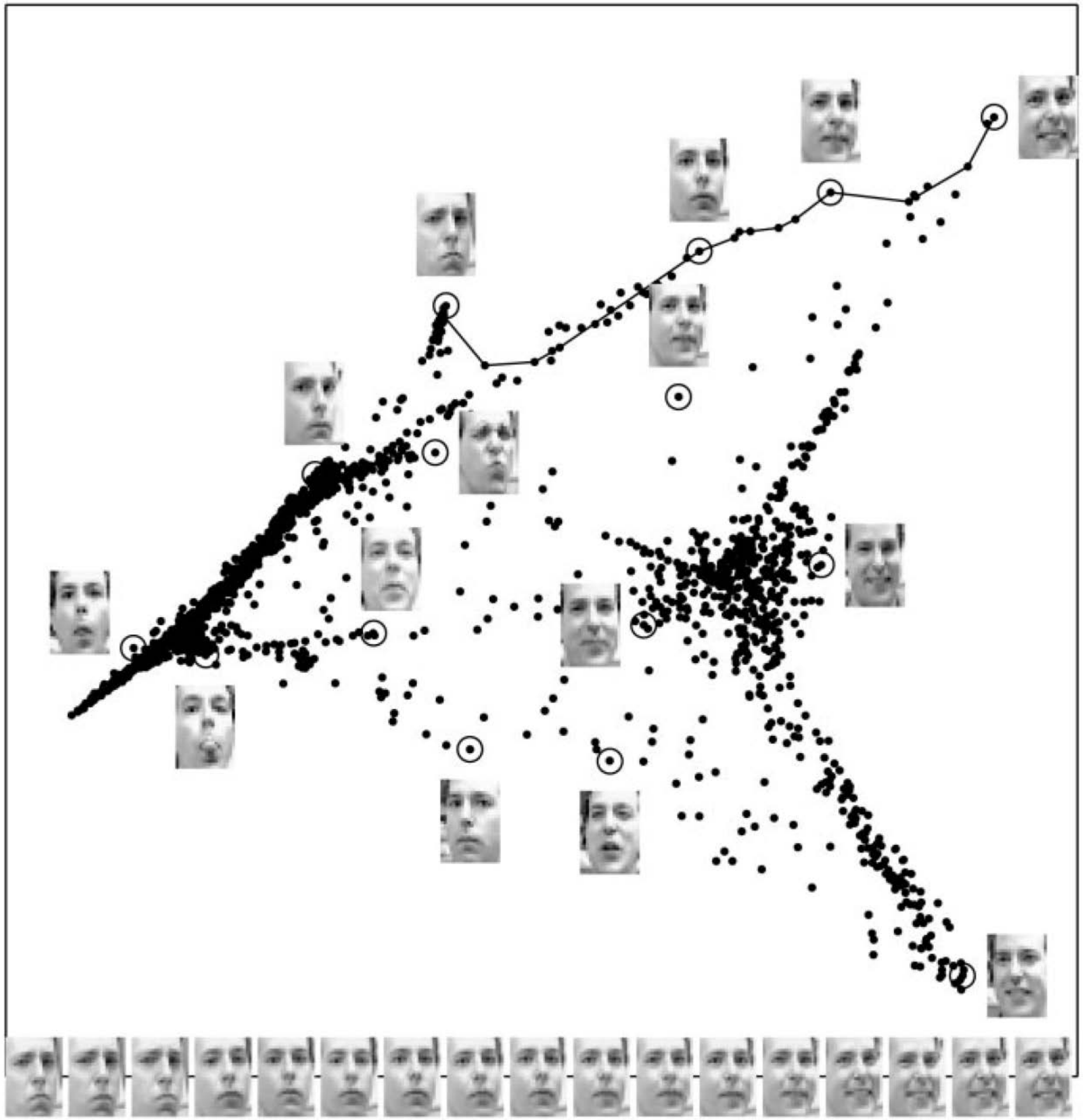


Locally Linear Embedding

5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

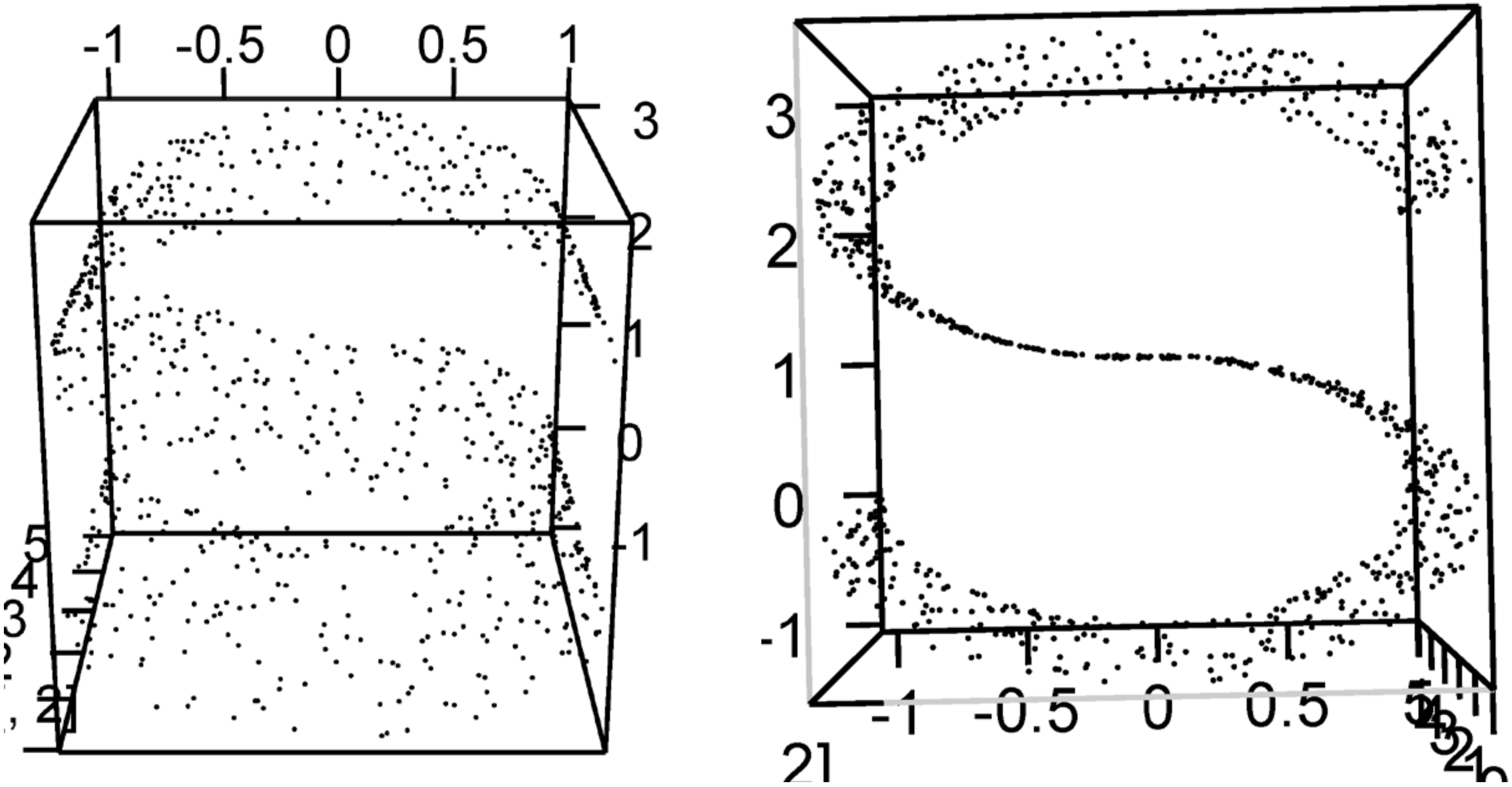
6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

LLE for face images



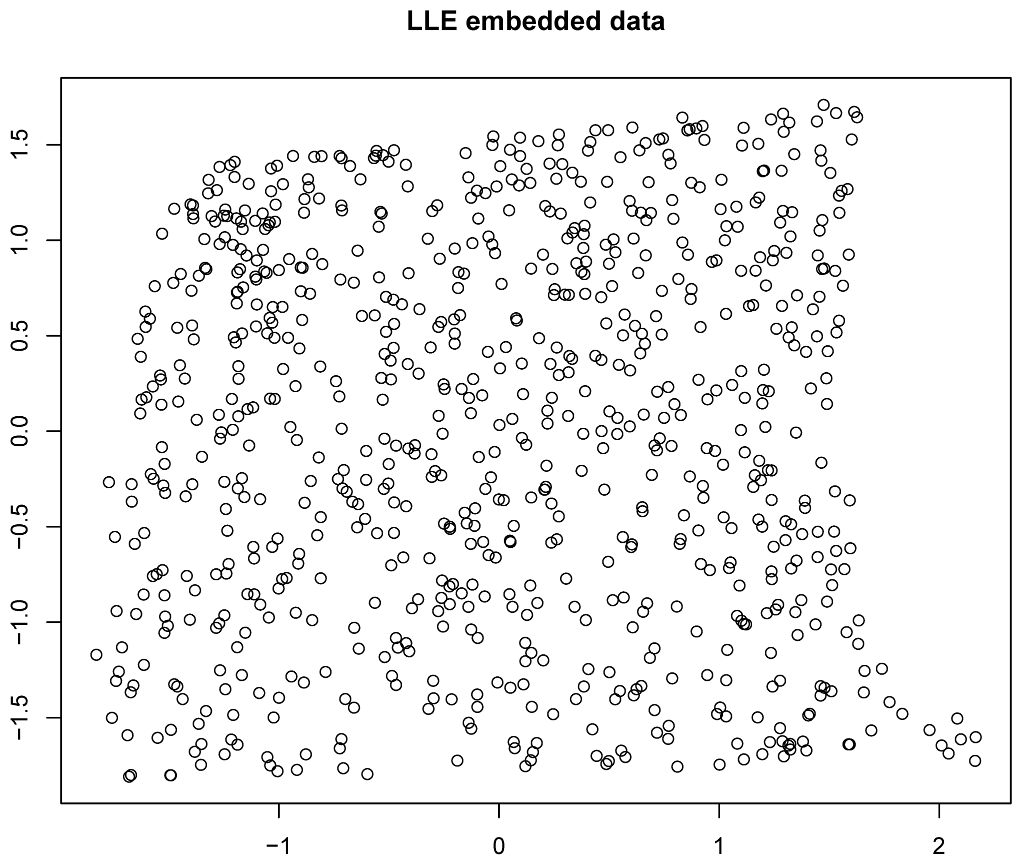
Locally Linear Embedding

LLE on the ”S” curve data



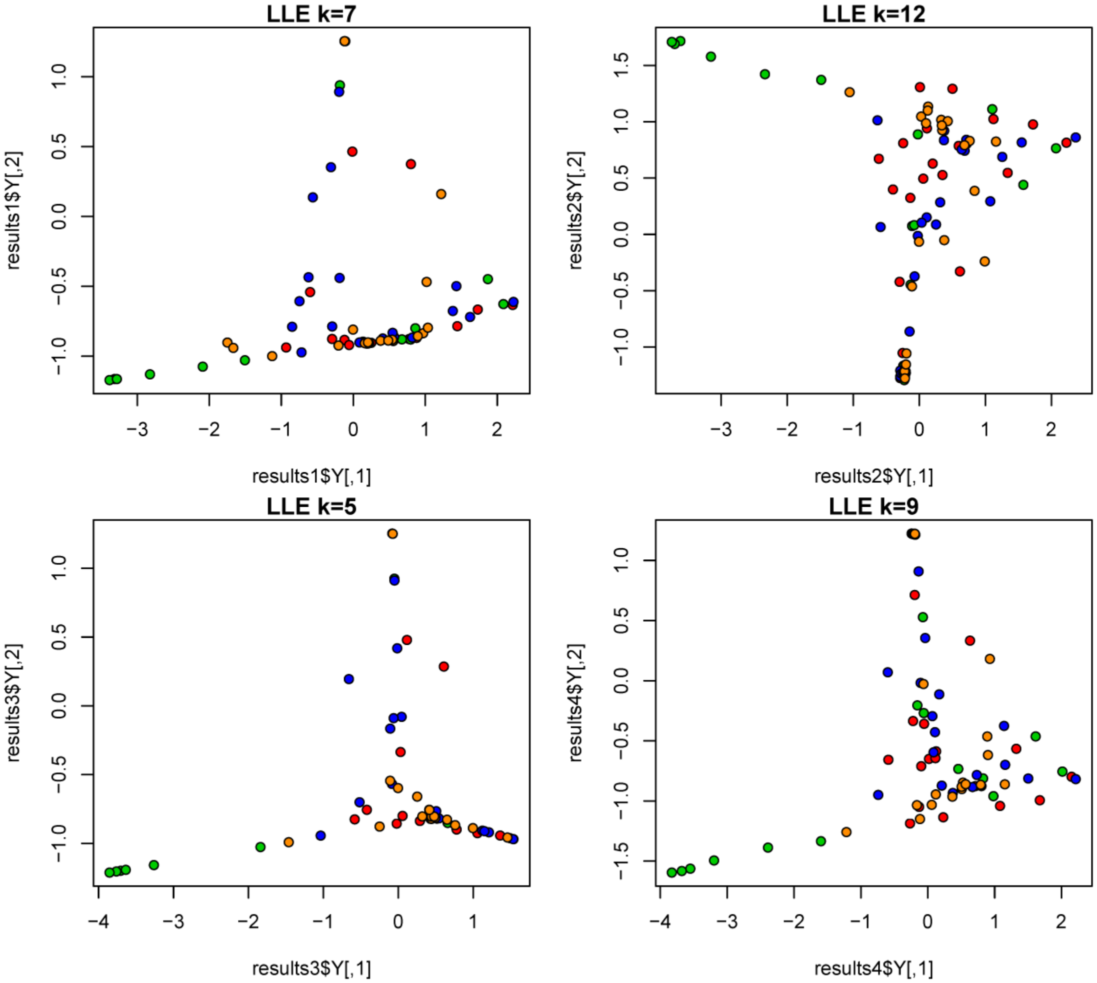
Locally Linear Embedding

LLE on the ”S” curve data



Locally Linear Embedding

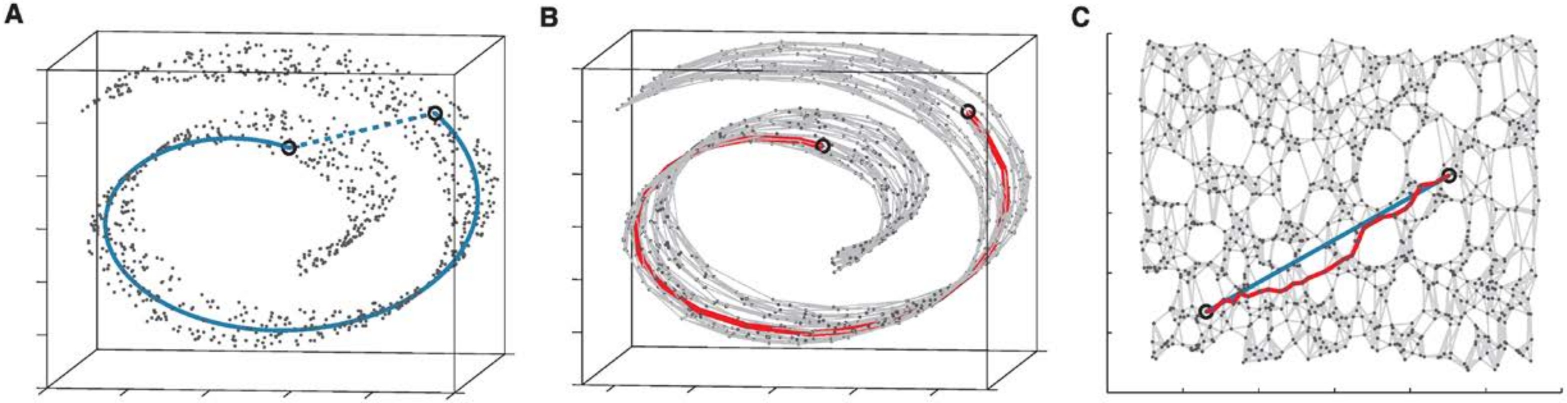
LLE applied to multiple tissues: 101 features with largest variance.
results are worse than with other methods: observations not on manifold



Isomap

Isomap is a low-dimensional embedding method
which computes a quasi-isometric, low-dimensional embedding.
Isomap is similar to LLE and a non-linear projection

- geodesic distance induced by a neighborhood
- geodesic distances:
  - shortest distances on a manifold
  - shortes path by Dijkstra's algorithm
  - sum of edge weights
- largest l eigenvectors of geodesic distance matrix are coordinates in projected space



Isomap

doubly centered geodesic distance matrix $\tau(\boldsymbol{D})$

$$\tau(\boldsymbol{D}) \;=\; -\,\frac{1}{2}\;\boldsymbol{H}\;\boldsymbol{D}^2\;\boldsymbol{H}$$

where $\boldsymbol{D}^2 = D^2_{ij} = D^2_{ji}$ is the element-wise square of the geodesic distance matrix

$\boldsymbol{H}$ is the centering matrix $\boldsymbol{H} \;=\; \boldsymbol{I}_n \;-\; \frac{1}{n}\;\mathbf{1}\;\mathbf{1}^T$ $\qquad \mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$

objective of Isomap: $\mathrm{E} \;=\; \left\| \tau(\boldsymbol{D}_X) \;-\; \tau(\boldsymbol{D}_Y) \right\|_{L^2}$

$\boldsymbol{D}_Y$ matrix of Euclidean distances in the projected space
$\boldsymbol{D}_X$ matrix of geodesic distances
τ converts distances to inner products

The objective can be minimized by setting the coordinates $\boldsymbol{y}_i$ to the top $\boldsymbol{l}$ eigenvectors of the matrix $\tau(\boldsymbol{D}_X)$

Isomap

Given: distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between pairs from n data points in an m-dimensional space X, parameter k or parameter e

Construct neighborhood graph

Define the graph G over all data points by connecting points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ if they are closer than e (e-Isomap), or if i is one of the k nearest neighbors of j (k-Isomap). Closeness and neighborhood is measured by $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Set edge lengths equal to $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Compute shortest paths by Floyd's algorithm

```
for $(i = 1 \; ; \; i \; \geq \; n \; ; \; i++)$ do
  for $(j = 1 \; ; \; j \; \geq \; n \; ; \; j++)$ do
    Initialize $d_G(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ if $i, j$ are linked by an edge; $d_G(\boldsymbol{x}_i, \boldsymbol{x}_j) = \infty$, otherwise.
  end for
end for
for $(k = 1 \; ; \; k \; \geq \; n \; ; \; k++)$ do
  for $(i = 1 \; ; \; i \; \geq \; n \; ; \; i++)$ do
    for $(j = 1 \; ; \; j \; \geq \; n \; ; \; j++)$ do
      $d_G(\boldsymbol{x}_i, \boldsymbol{x}_j) = \min\{d_G(\boldsymbol{x}_i, \boldsymbol{x}_j), d_G(\boldsymbol{x}_i, \boldsymbol{x}_k) + d_G(\boldsymbol{x}_k, \boldsymbol{x}_j)\}$.
    end for
  end for
end for
define shortest path matrix $\boldsymbol{D}_X$ by $[\boldsymbol{D}_X]_{ij} = d_G(\boldsymbol{x}_i, \boldsymbol{x}_j)$
```

Construct l-dimensional embedding

Compute λ_p as the p-th eigenvalue (in decreasing order) of the matrix $\tau(\boldsymbol{D}_X)$, and v_{pi} as the i-th component of the p-th eigenvector.

set $y_{ij} = \sqrt{\lambda_i} v_{ji}$.

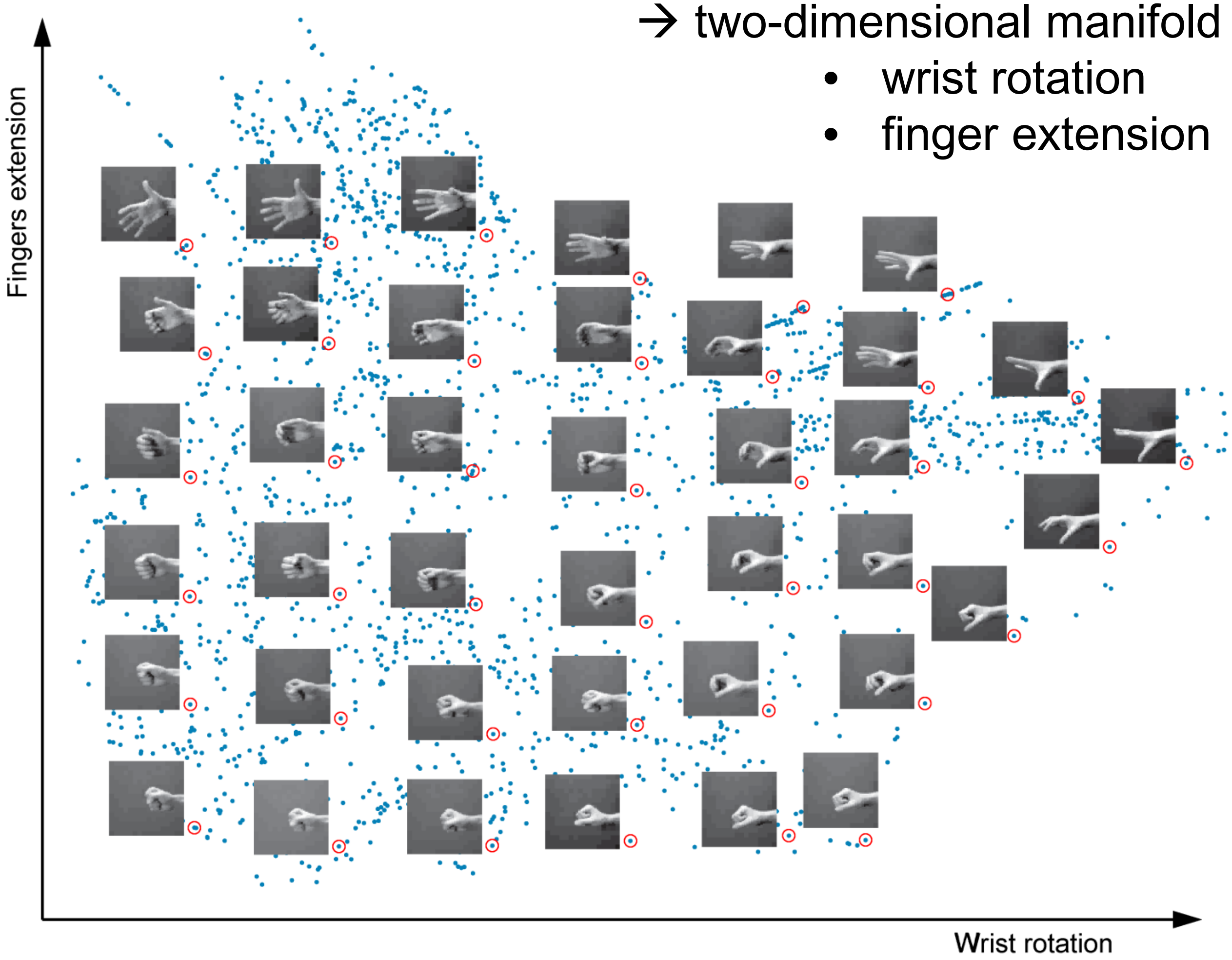
Result $\boldsymbol{Y}$: coordinate vectors $\boldsymbol{y}_i$ in a l-dimensional ($l < m$) Euclidean space Y

Isomap

Isomap (k=6), n=2000 images of a hand opening and closing movements at different wrist orientations

→ two-dimensional manifold

- wrist rotation
- finger extension

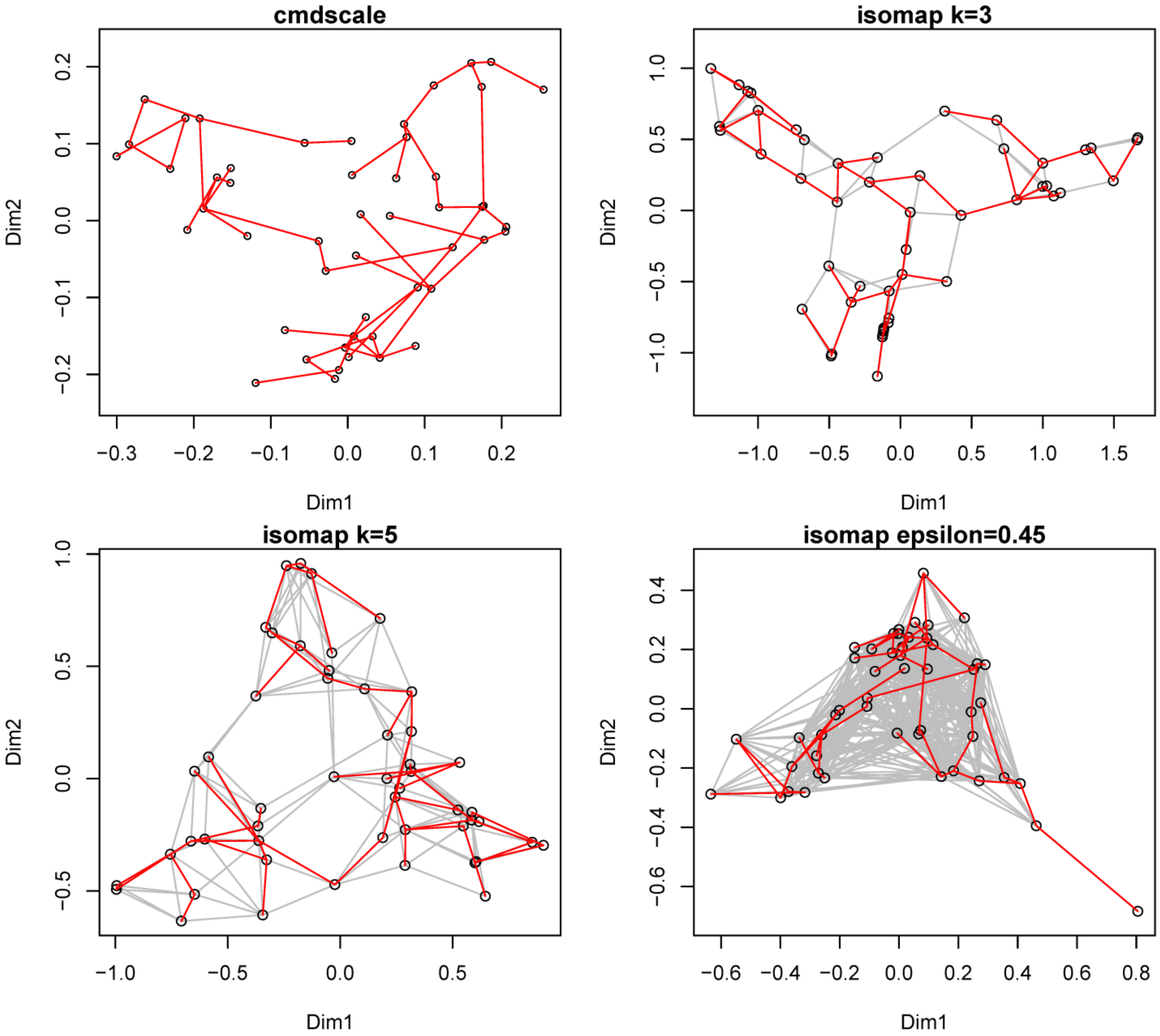


Isomap

tree counts in 1-hectare plots in the Barro Colorado Island

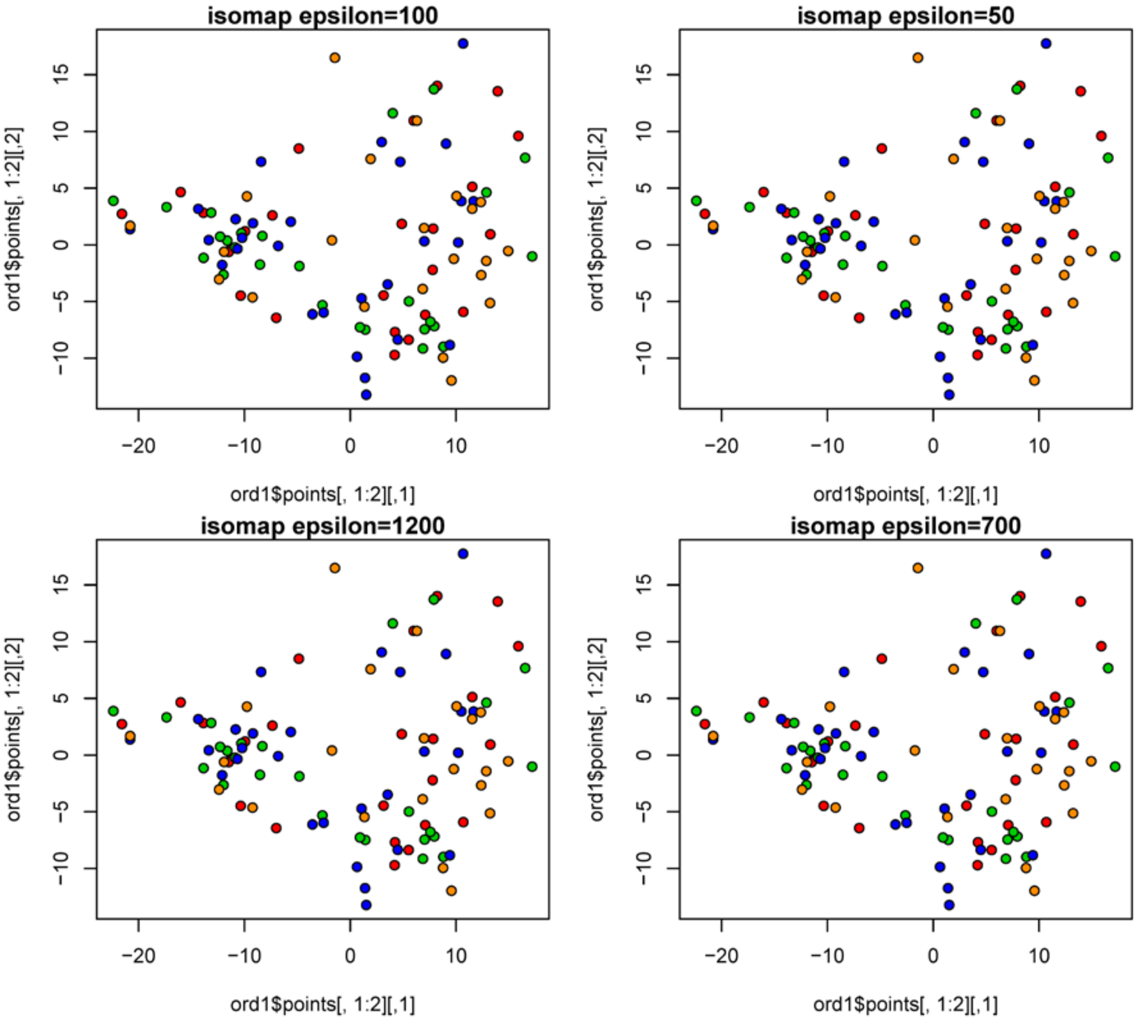
- 50 plots of 1 hectare with counts of trees on each plot
- quadrants are located in a regular grid
- 225 tree species (at least 10 cm in diameter at breast height)
- counts in each one hectare square of forest

Isomap


cmdscale
isomap k=3
isomap k=5
isomap epsilon=0.45
Dim1
Dim2
−0.3
−0.2
−0.1
0.0
0.1
0.2
−1.0
−0.5
0.5
1.0
1.5
−0.6
−0.4
0.4
0.6
0.8

Isomap

Isomap for the multiple tissues data: results not as good as with other methods: observations not on a manifold

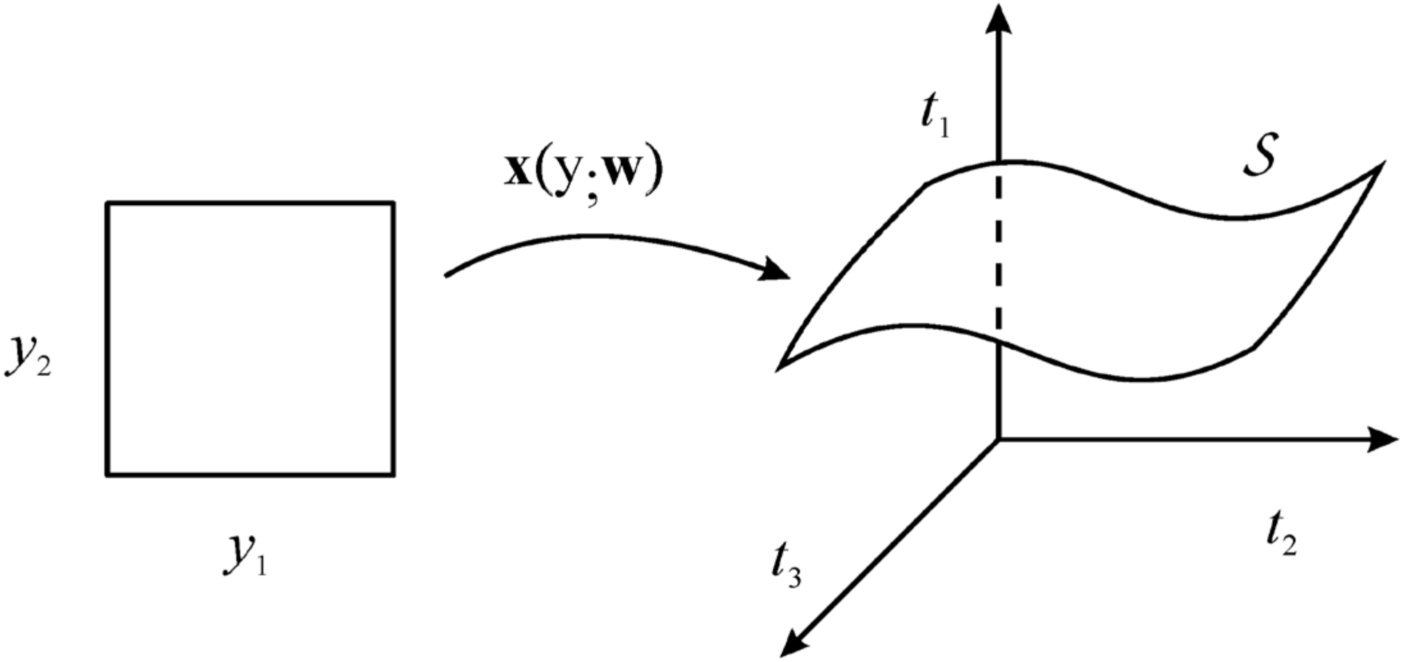


Generative Topographic Mapping

generative topographic mapping (GTM) is a non-linear latent variable model as an alternative to SOMs to overcome their disadvantages.

GTM is similar to factor analysis as is also maps from the latent space to the observations space.

Latent variables $\boldsymbol{y} \in \mathbb{R}^l$ are mapped to observations $\boldsymbol{x} \in \mathbb{R}^m$, $m > l$

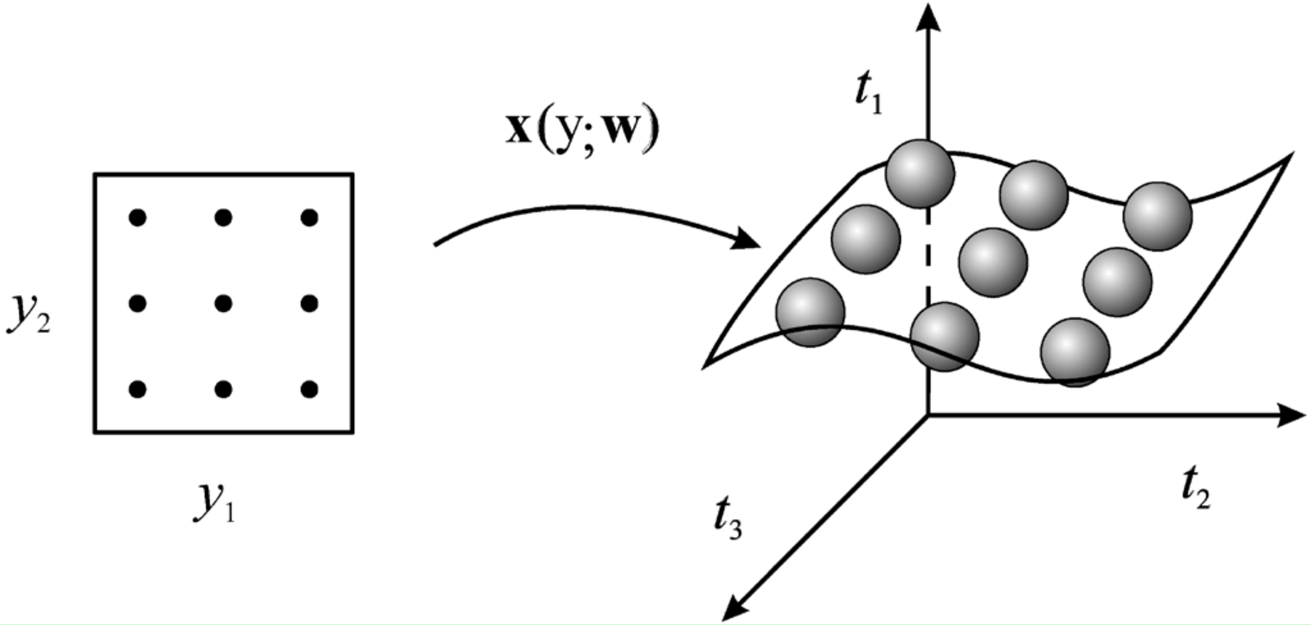


Generative Topographic Mapping

latent-variable space: distribution $p(\boldsymbol{y})$
observation space: distribution $p(\boldsymbol{x} \mid \boldsymbol{w})$

If points are mapped from a l-dimensional to a m-dimensional space: probability masses would vanish → Gaussian ball in m-dim. space:

$$p(\boldsymbol{t} \mid \boldsymbol{y}, \boldsymbol{w}, \beta) \; = \; \left(\frac{\beta}{2\pi} \right)^{m/2} \exp - \frac{\beta}{2} \, \| \boldsymbol{x}(\boldsymbol{y}, \boldsymbol{w}) \; - \; \boldsymbol{t} \|^2$$



Generative Topographic Mapping

distribution in the m-dimensional space is obtained by integrating over all $\boldsymbol{x}$ that contribute to a density at $\boldsymbol{t}$:

$$p(\boldsymbol{t} \mid \boldsymbol{w}, \beta) \ = \int p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{w}, \beta) \ p(\boldsymbol{x}) \ d\boldsymbol{x}$$

For data points $\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$, the log likelihood is

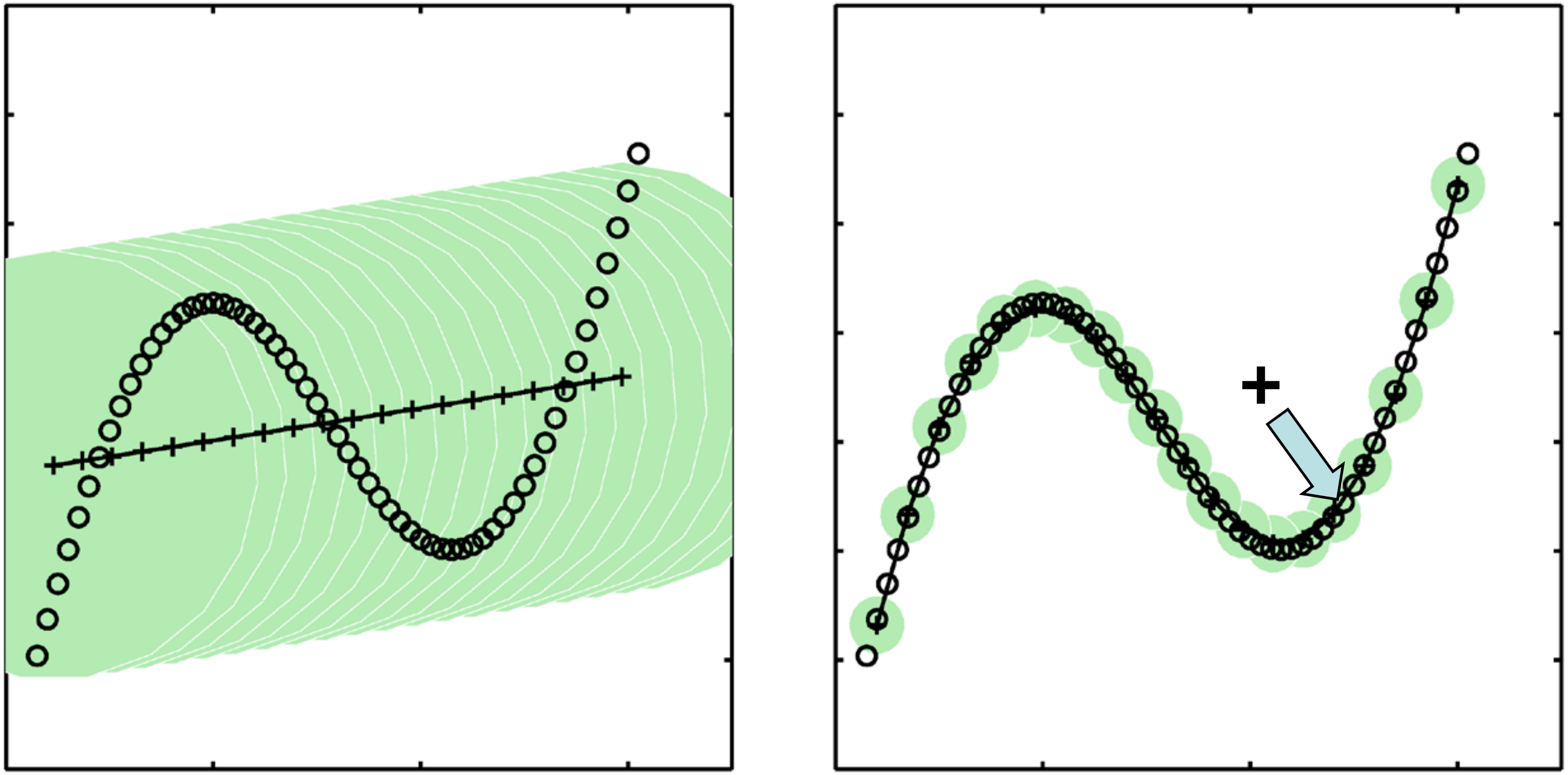
$$\log \mathcal{L} \ = \ \sum_{i=1}^{n} \ln p(\boldsymbol{t}_i \mid \boldsymbol{w}, \beta)$$

$$p(\boldsymbol{y}) \ = \ \frac{1}{L} \sum_{j=1}^{L} \delta(\boldsymbol{y} \ - \ \boldsymbol{y}_j)$$

$$p(\boldsymbol{t} \mid \boldsymbol{w}, \beta) \ = \ \frac{1}{L} \sum_{j=1}^{L} p(\boldsymbol{t} \mid \boldsymbol{y}_j, \boldsymbol{w}, \beta)$$

kernel density estimate or constraint Gaussian mixture model in the m-dim. space with centers mapped from an l-dim. space

Generative Topographic Mapping



Toy problem involving data (o) generated from a 1-dimensional curve embedded in 2 dimensions, together with the projected latent points (+) and their Gaussian noise distributions (filled circles). The initial configuration is shown on the left, and the result on the right.

t-Distributed Stochastic Neighbor Embedding

t-distributed stochastic neighbor embedding (t-SNE) models each high-dimensional observations by a two- or three-dimensional representation: similar observations are represented by nearby projections and dissimilar observations distant representations.

stochastic neighbor embedding (SNE) the similarity is the conditional probability $p_{j|i}$ that $\boldsymbol{x}_i$ would pick $\boldsymbol{x}_j$ as its neighbor

For $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ we obtain $p_{j|i} \ = \ \dfrac{\exp(-\|\boldsymbol{x}_i \ - \ \boldsymbol{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\boldsymbol{x}_i \ - \ \boldsymbol{x}_k\|^2 / 2\sigma_i^2)}$

For low-dimensional projections a conditional probability is computed

$$q_{j|i} \ = \ \frac{\exp(-\|\boldsymbol{y}_i \ - \ \boldsymbol{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\boldsymbol{y}_i \ - \ \boldsymbol{y}_k\|^2)}$$

objective is the Kullback-Leibler divergence between P and Q:

$$KL(P||Q) \ = \ \sum_{i \neq j} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

minimized by gradient descent

t-Distributed Stochastic Neighbor Embedding

objective for the SNE:

- difficult to optimize
- crowding problem

  For example in ten dimensions, it is possible to have 11 data points that are mutually equidistant but there is no way to model this faithfully in a two-dimensional map

t-distributed stochastic neighbor embedding, solves these SNE problems by

- objective of the SNE is symmetrized → simpler gradients
- objective uses Student's t-distribution → heavy-tailed which reduces the crowding problem and simplifies the optimization

t-Distributed Stochastic Neighbor Embedding

Symmetry:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

Heavy-tails using the Student's t-distribution:

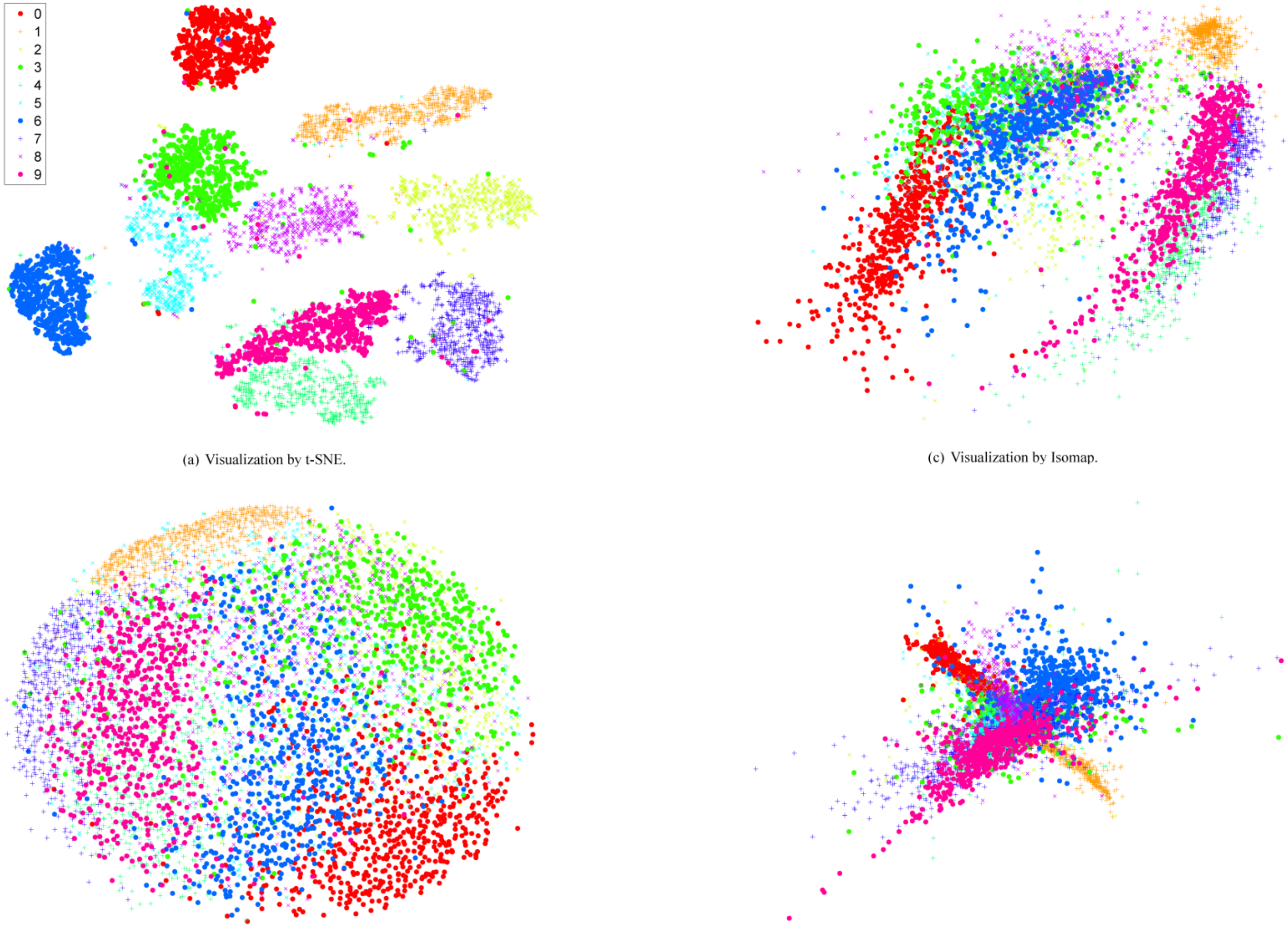
$$q_{ij} = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|\boldsymbol{y}_k - \boldsymbol{y}_l))^{-1}}$$

Optimization via gradient descent

t-Distributed Stochastic Neighbor Embedding

6,000 handwritten digits from the MNIST data set:

t-SNE is compared to Sammon's mapping, Isomap, and LLE.



(a) Visualization by t-SNE.

(c) Visualization by Isomap.

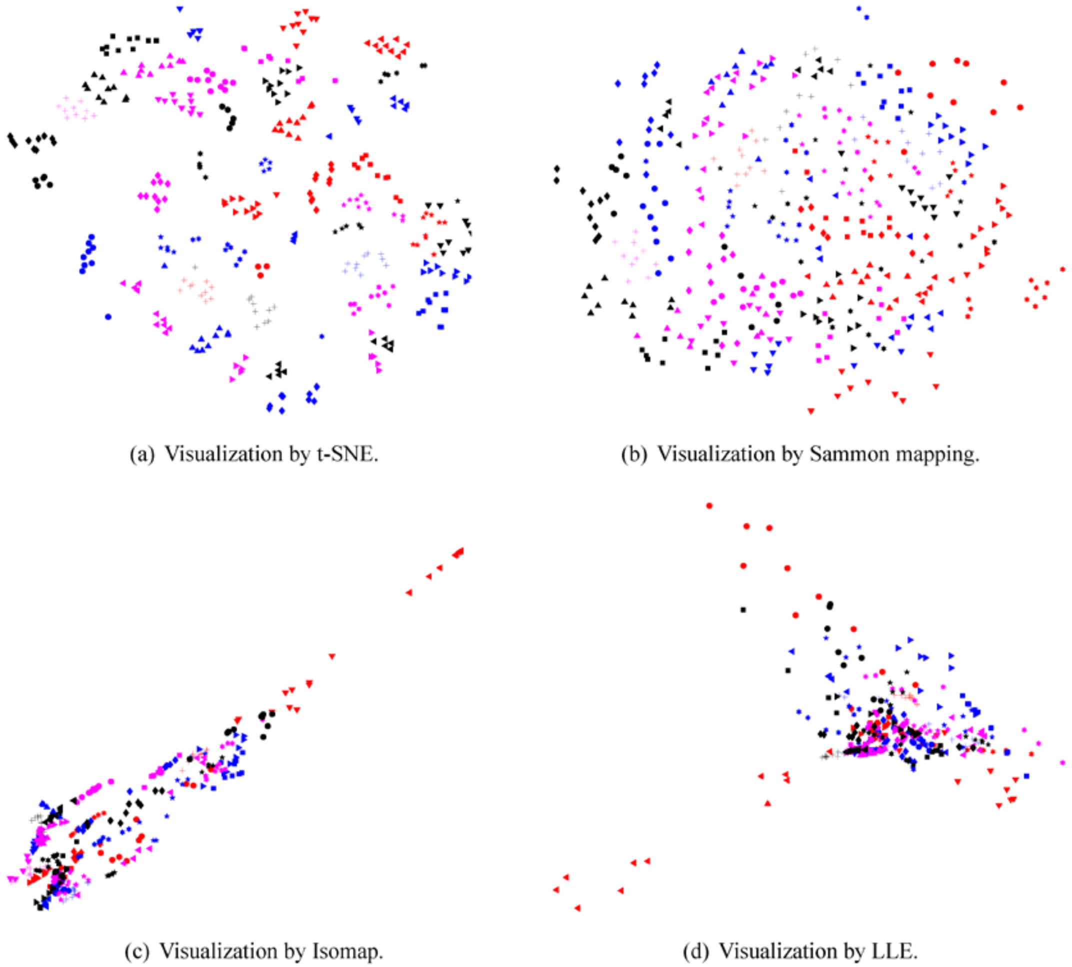
(b) Visualization by Sammon mapping.

(d) Visualization by LLE.

t-Distributed Stochastic Neighbor Embedding

faces from the Olivetti data base:

t-SNE is compared to Sammon's mapping, Isomap, and LLE.


(a) Visualization by t-SNE.

(b) Visualization by Sammon mapping.

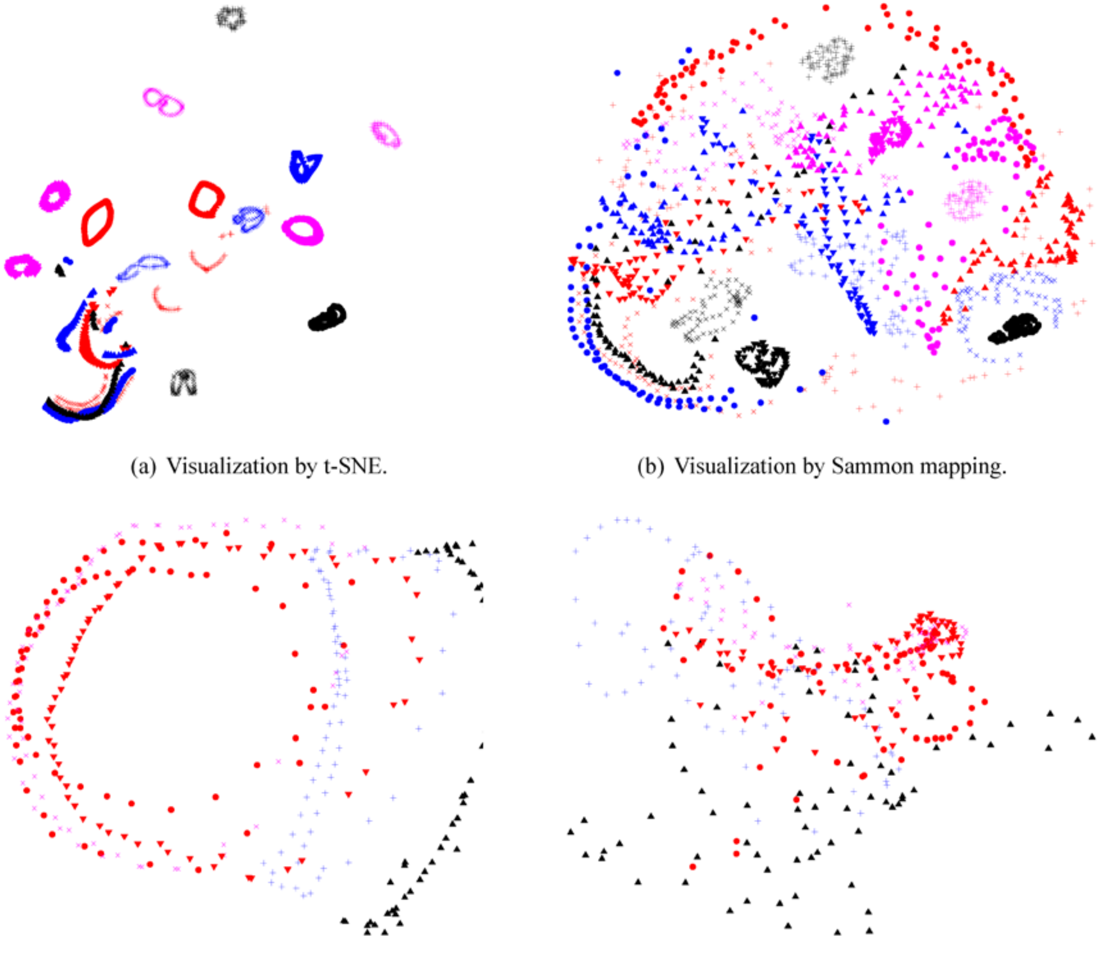
(c) Visualization by Isomap.

(d) Visualization by LLE.

t-Distributed Stochastic Neighbor Embedding

COIL-20 data set:

t-SNE is compared to Sammon's mapping, Isomap, and LLE.



(a) Visualization by t-SNE.

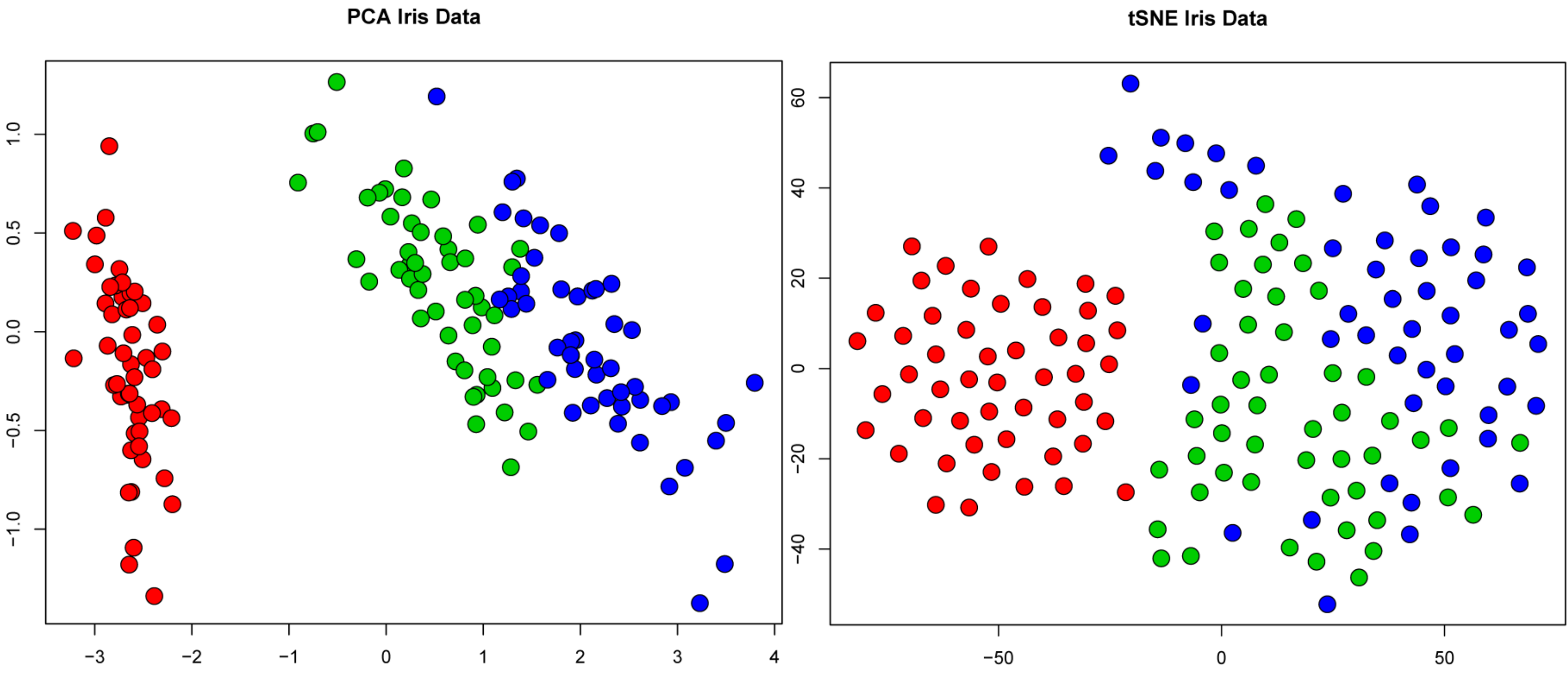
(b) Visualization by Sammon mapping.

(c) Visualization by Isomap.

(d) Visualization by LLE.

t-Distributed Stochastic Neighbor Embedding

iris data set



t-Distributed Stochastic Neighbor Embedding

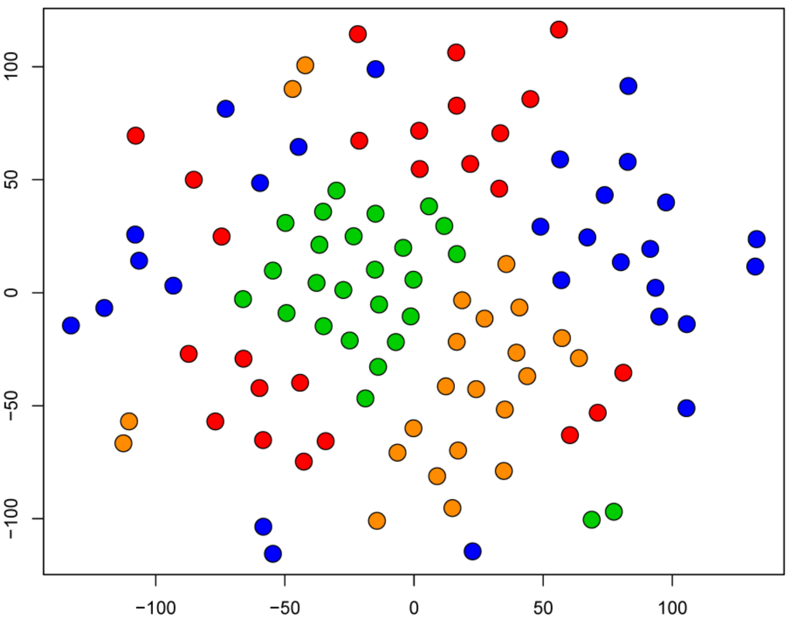
5 Factor Analysis
5.1 The Model
5.2 MLE FA
5.3 FA vs. PCA & ICA
5.4 Artificial Examples
5.5 Real Examples
5.5.1 Iris
5.5.2 Multiple Tissues

6 Scaling & Projection
6.1 Projection Pursuit
6.2 Multidim. Scaling
6.2.1 The Method
6.2.2 Examples
6.3 Non-negative Matrix Factorization
6.3.1 The Method
6.3.2 Examples
6.4 Locally Linear Embedding
6.4.1 The Method
6.4.2 Examples
6.5 Isomap
6.5.1 The Method
6.5.2 Examples
6.6 Topographic Maps
6.6.1 The Method
6.6.2 Examples
6.7 t-D. Stochastic Neighbor Embedding
6.7.1 The Method
6.7.2 Examples
6.8 SOM
6.8.1 The Method
6.8.2 Examples

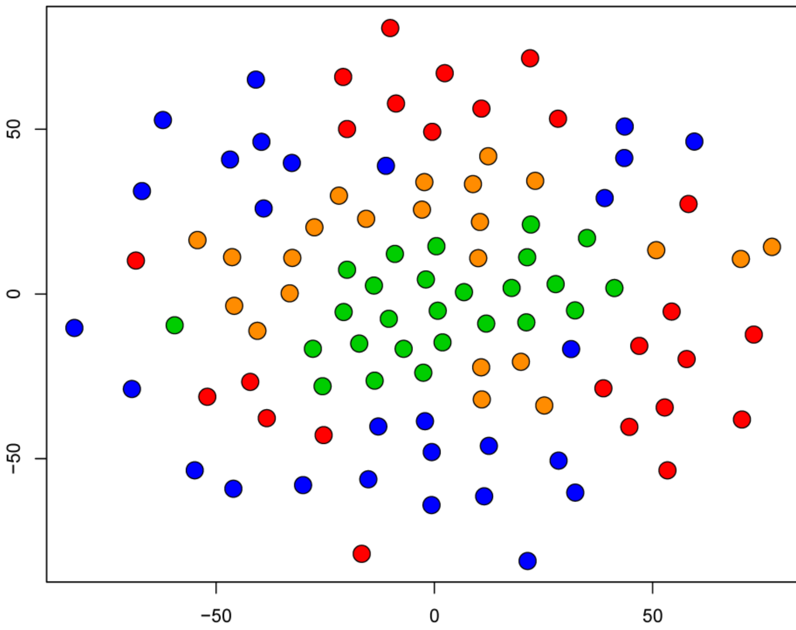
multiple tissues data: features with largest variance

The results are not as good as with other methods because the observations are not located on a 2-dimensional manifold.

tSNE Multiple Tissues Data: perplexity=30



tSNE Multiple Tissues Data: perplexity=50



Self-Organizing Map

Self-Organizing Map (SOM) or Kohonen map:
SOMs comprise two objectives:

- clustering (see next subsection)
- down-projecting.

For SOMs the objective function cannot be expressed as a single scalar function like an energy or an error function.

lack of a scalar objective / cost function:

- no theoretical basis for choosing learning parameters
- no ensurance to achieve topographic ordering
- no proofs of convergence
- models cannot be compared
- overfitting not detected
- stopping of training is difficult to determine
- quality of the solution is hard to assess
- no probability density for further processing by other methods

Self-Organizing Map

$\boldsymbol{y}_k \in \mathbb{R}^l$ equidistantly fill a hypercube associated with $\boldsymbol{w}_k \in \mathbb{R}^m$, which are the parameters of the SOM

Data points that are neighbors should be neighbors in the projection preserve neighborhood relation: topologically ordered maps (TOMs)

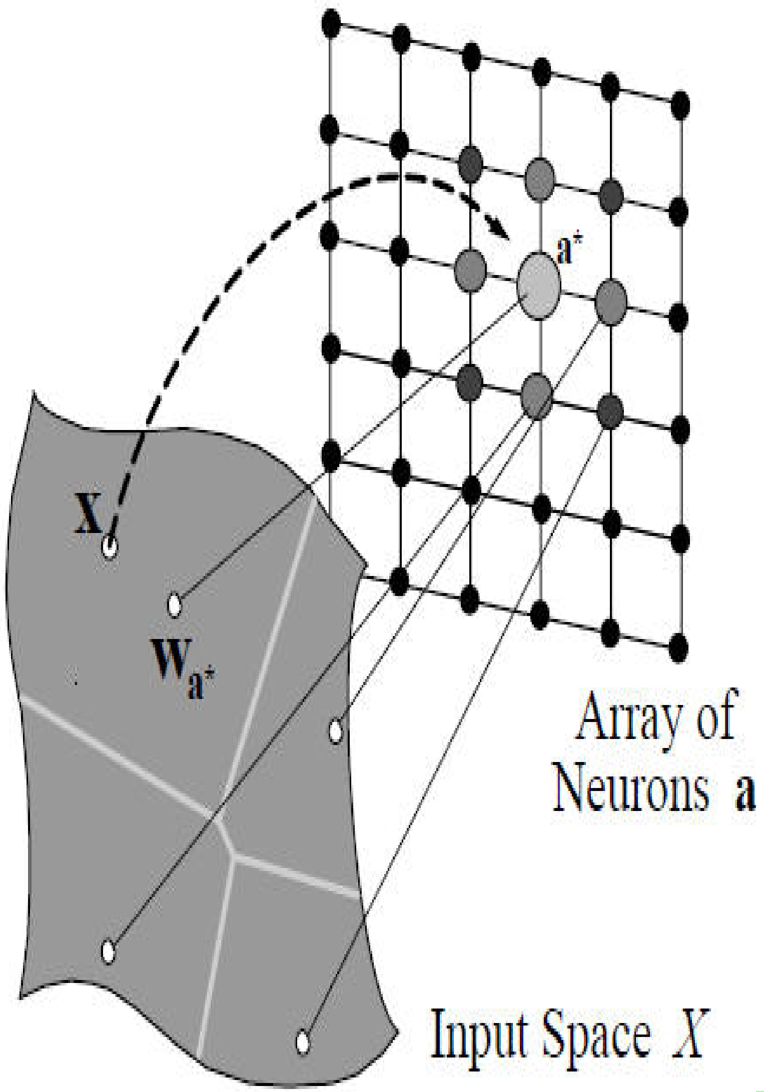
on-line update rule:

$$k \;=\; \arg\max_{s} \; \boldsymbol{x}^T \boldsymbol{w}_s$$

$$(\boldsymbol{w}_t)^{\text{new}} \;=\; \boldsymbol{w}_t \;+\; \eta \, \delta \left(\| \boldsymbol{y}_t \;-\; \boldsymbol{y}_k | \| \right) \; (\boldsymbol{x} \;-\; \boldsymbol{w}_t)$$

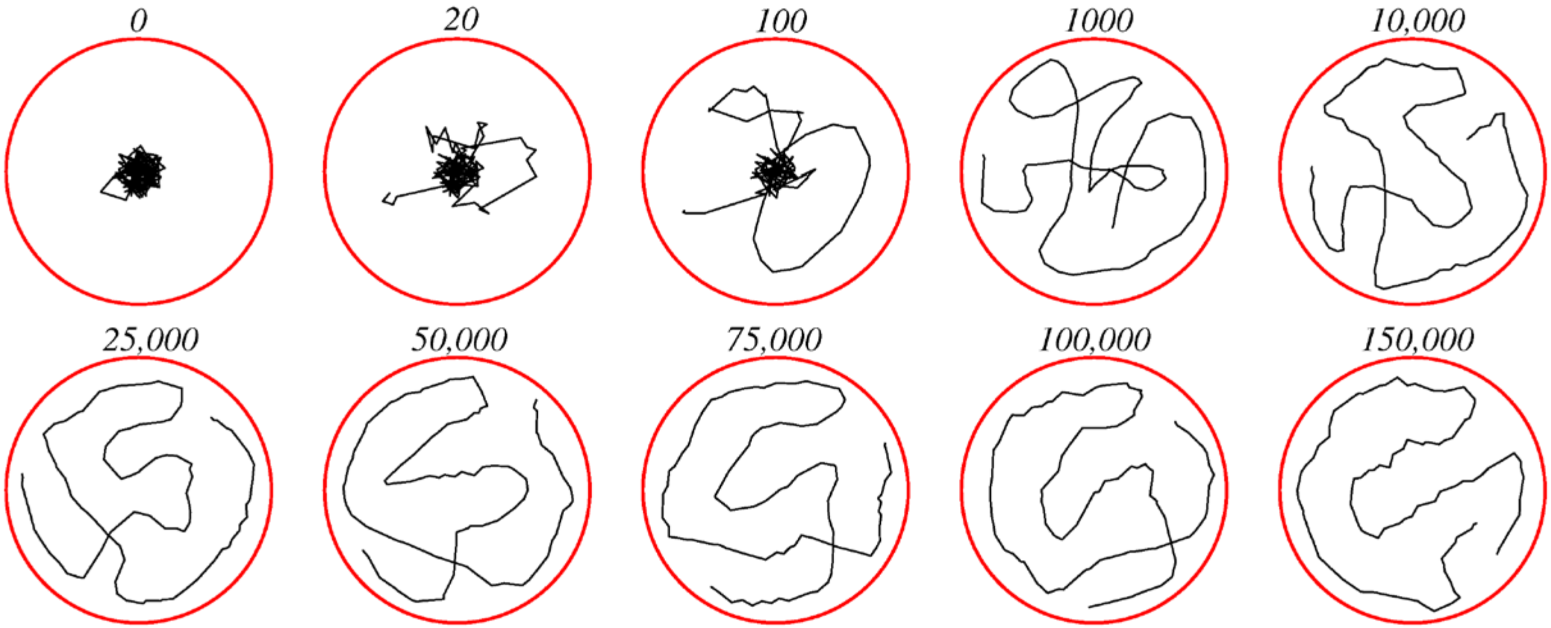
where η is the learning rate, δ is the window function which is largest for $\boldsymbol{y}_t \;=\; \boldsymbol{y}_k$ and is decreasing with the distance to $\boldsymbol{y}_k$.

Self-Organizing Map



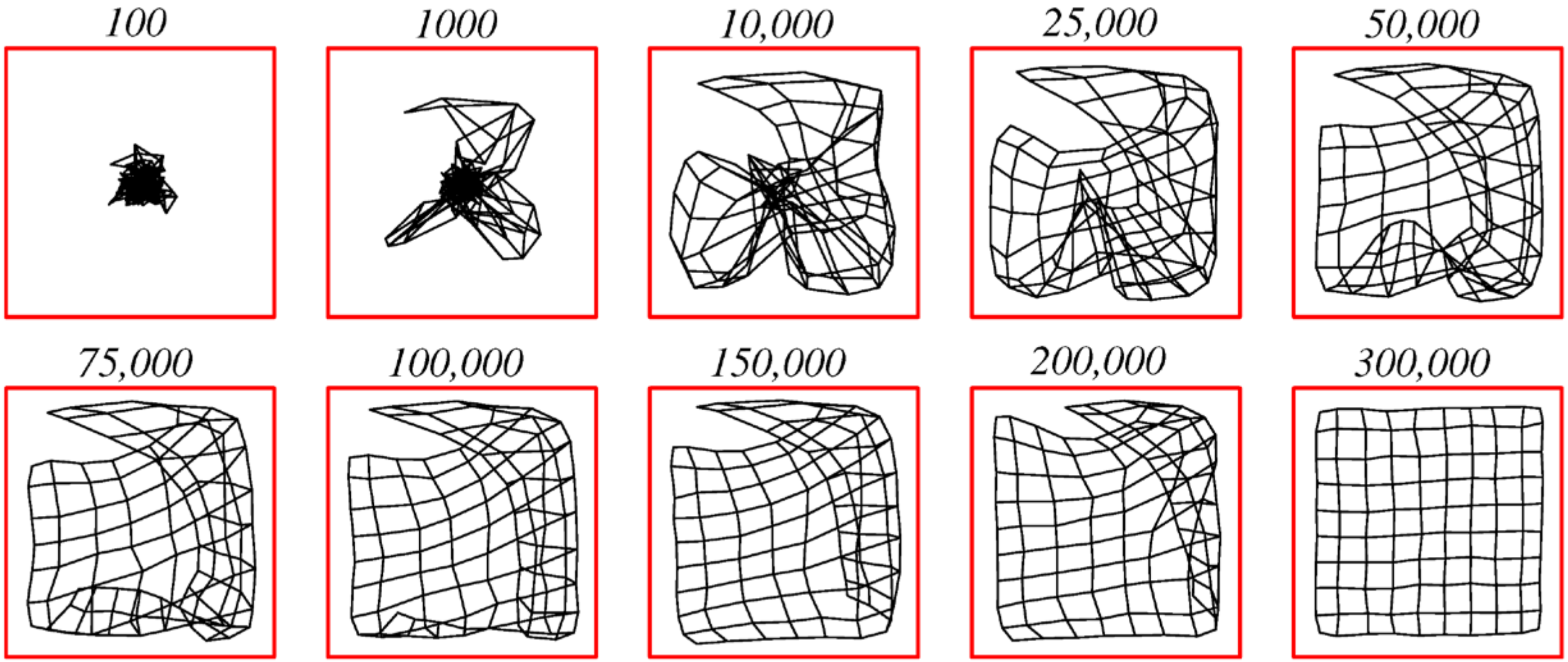
Self-Organizing Map

Example 1: one-dimensional representation of a two-dimensional space



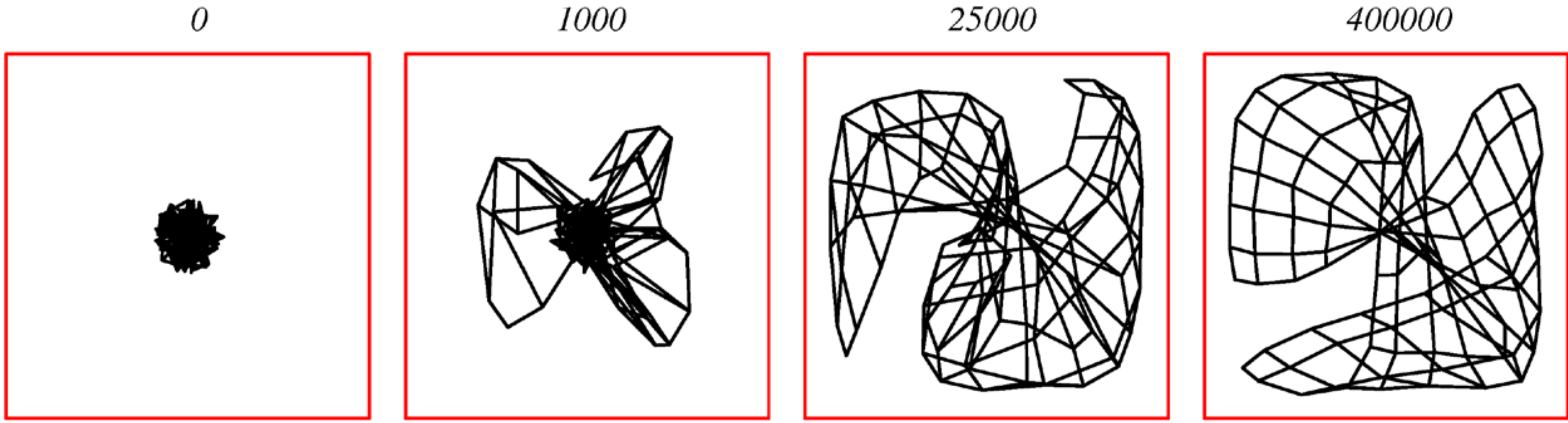
Self-Organizing Map

Example 2: square data space to a square (grid) representation space



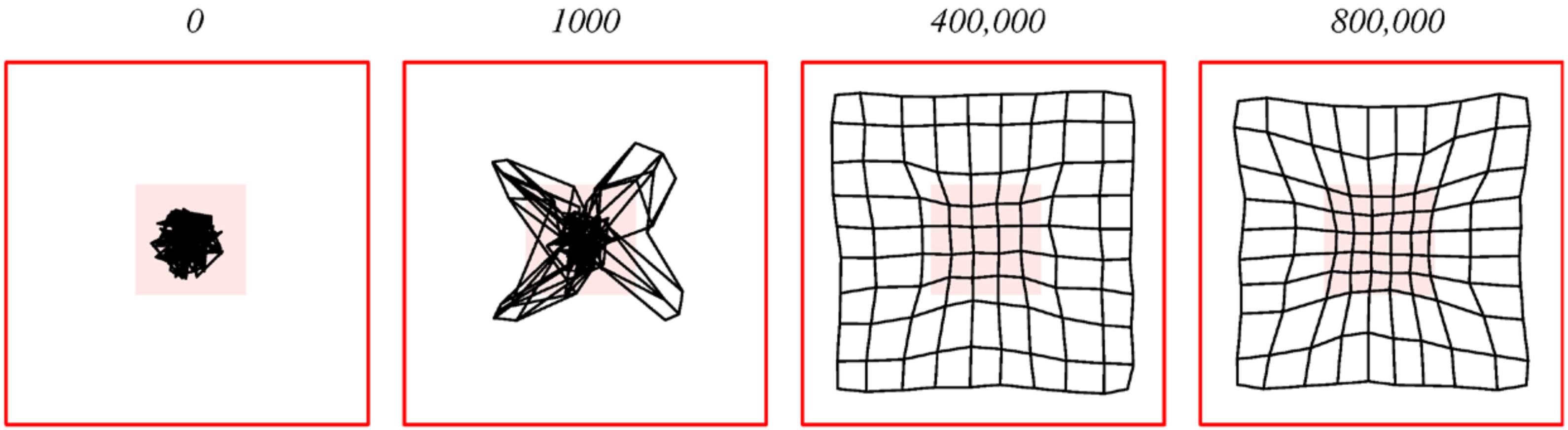
Self-Organizing Map

Example 2 with different initialization. Kinks in the map do not vanish even if more patterns are presented – that is a local minimum



Self-Organizing Map

Example 2 with a non-uniformly sampling: the density at the center was higher than at the border


0
1000
400,000
800,000