Dimension Reduction

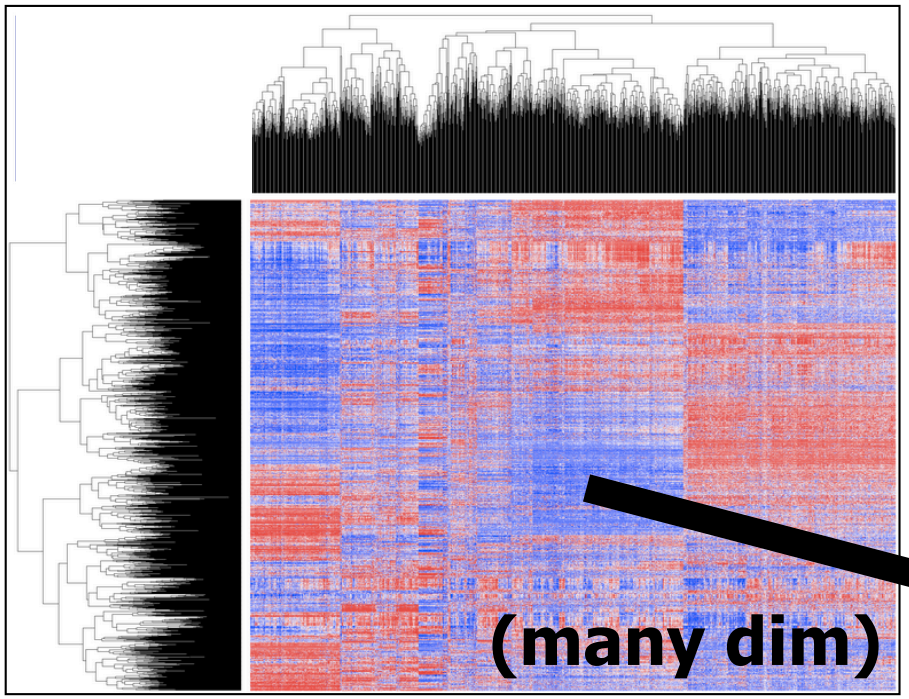
Dimensionality reduction (1)

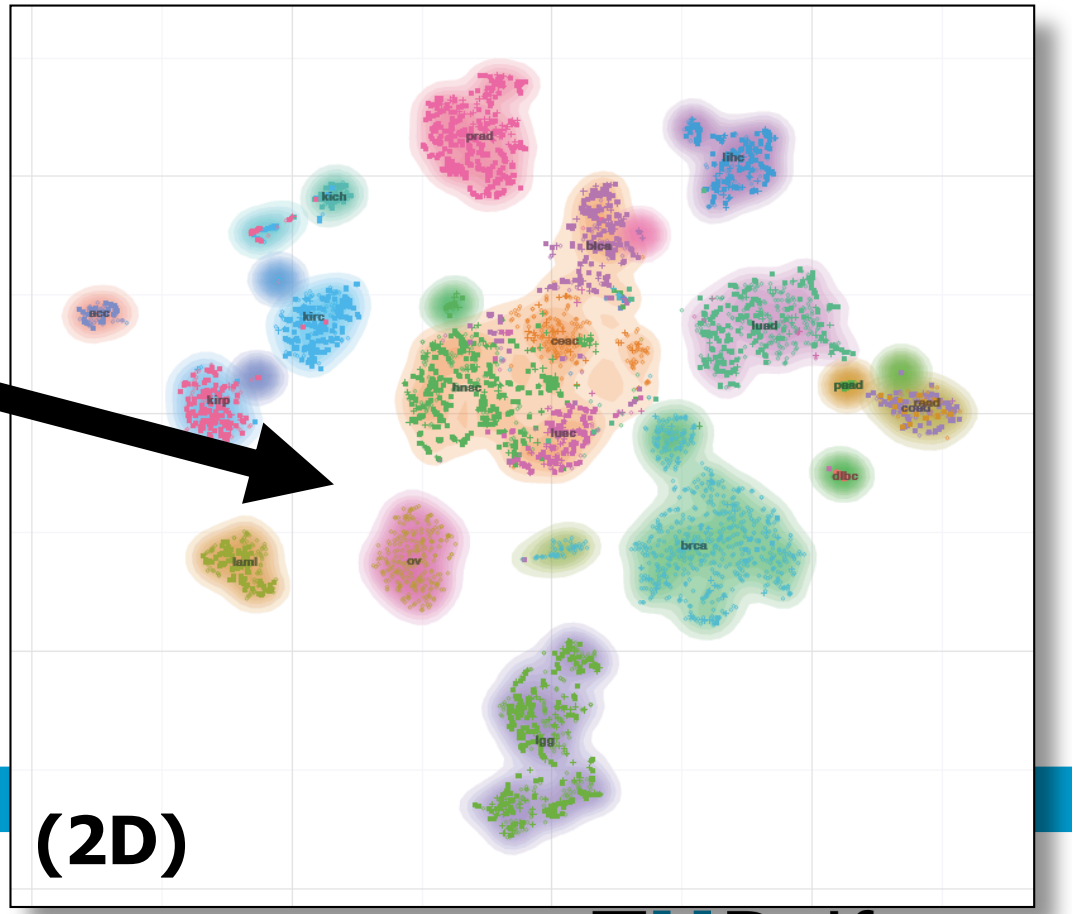
- **Many data sets are *high-dimensional*: each instance contains many features**

- **Why do we want to reduce data dimensionality?**
  - Make storage or processing of data easier
  - (Visual) discovery of hidden structure in the data
  - Intrinsic dimensionality might be smaller
  - Remove redundant and noisy features

Dimensionality reduction (2)

Visual discovery of data structure



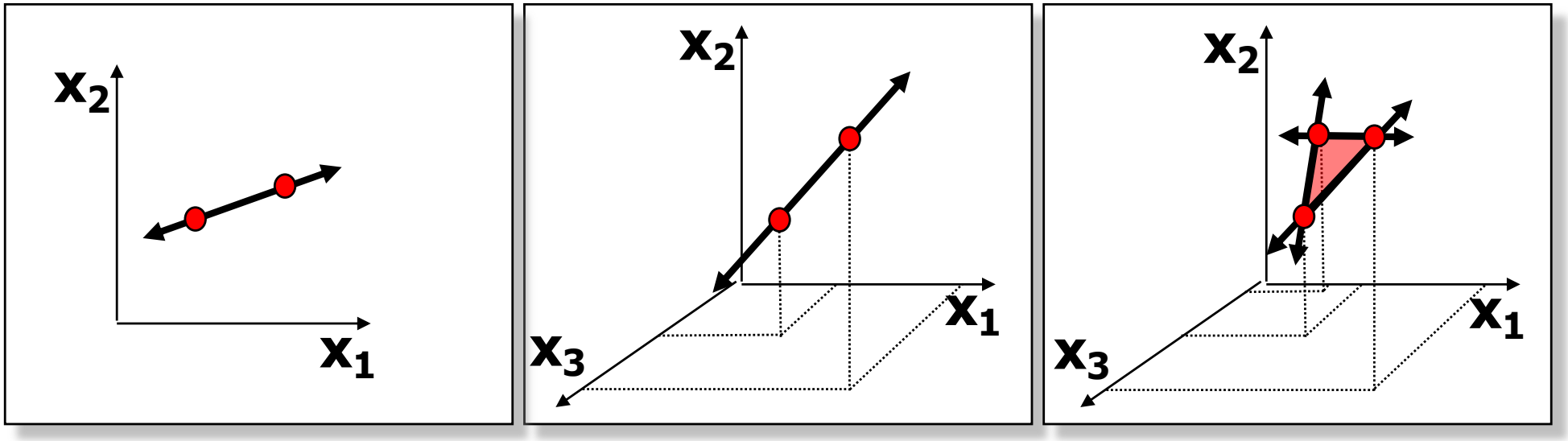


The Delft Bioinformatics Lab

TU Delft

Dimensionality reduction (3)

Intrinsic dimensionality



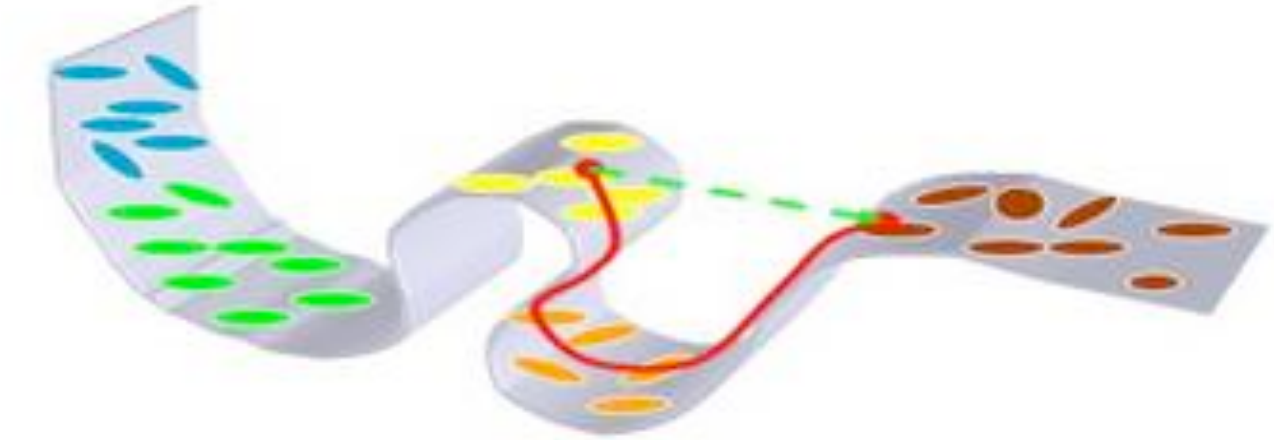
2 objects, 2 dimensions ➔ 1 dimension	2 objects, 3 dimensions ➔ 1 dimension	3 objects, 3 dimensions ➔ 2 dimension

Maximum number of dimensions: #objects -1

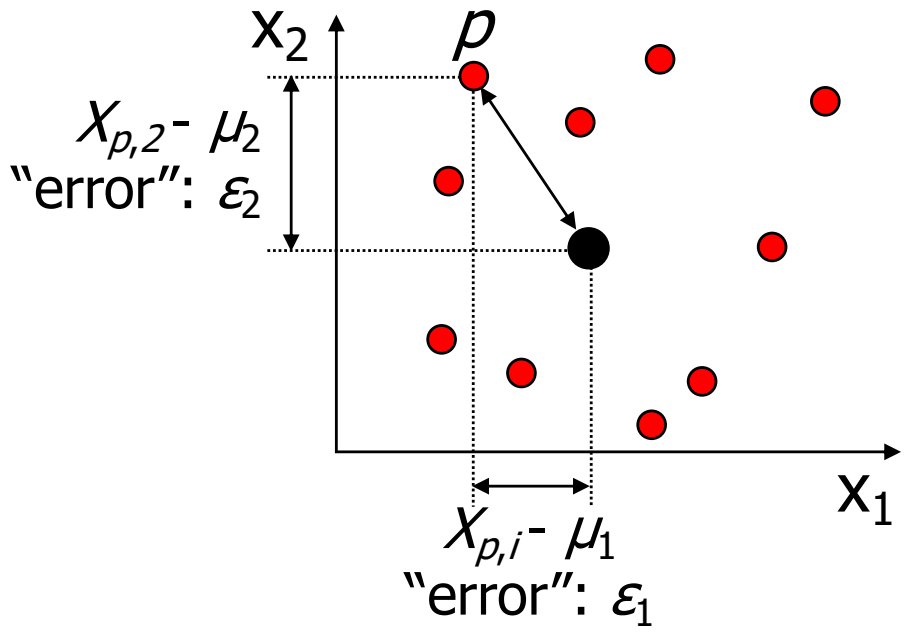
Dimensionality reduction (4)

Intrinsic dimensionality

Data may lie also be spread across a lower dimensional space (manifold)



Dimensionality reduction (5)
Accumulation of noise



In *D* dimensions distance becomes:

$$\sum_{i=1}^{D}(x_{p,i} - \mu_p)^2$$

If a feature *i* is not relevant than can be considered as error

$$\epsilon_i = (x_{p,i} - \mu_p)$$

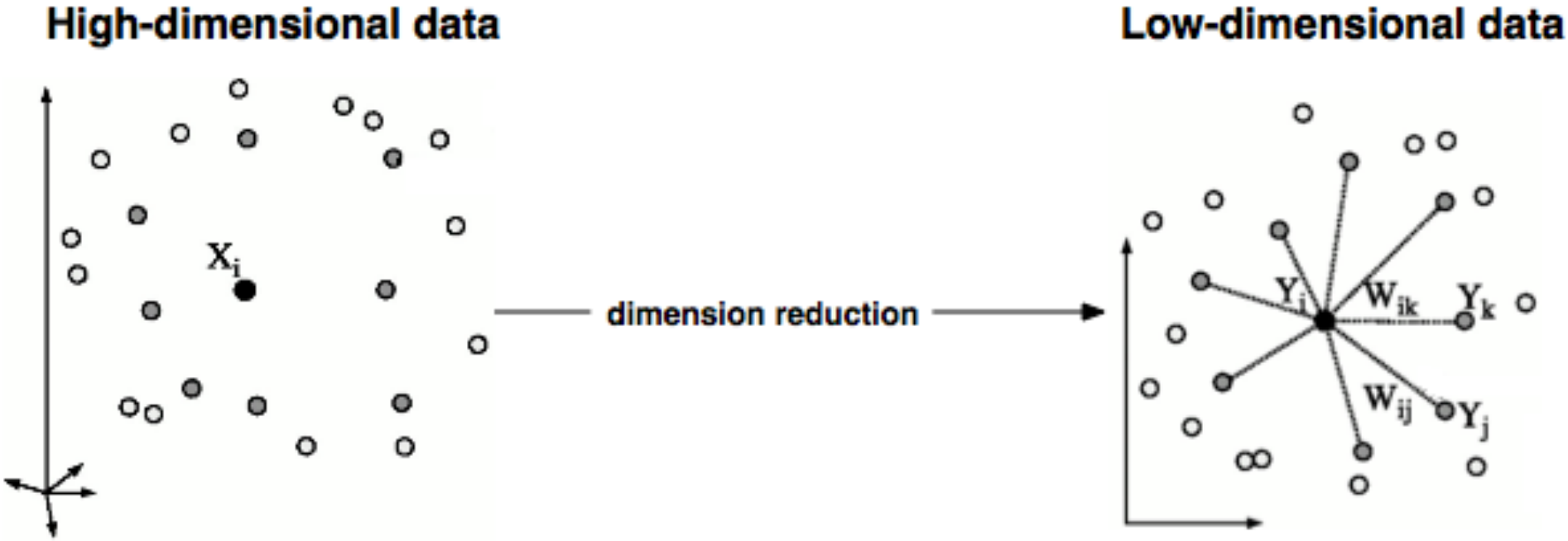
When there are *K* non-relevant features: accumulation of errors:

$$\sum_{j=1}^{K}(\epsilon_j)^2$$

If *K* >> *D-K* than error becomes dominating

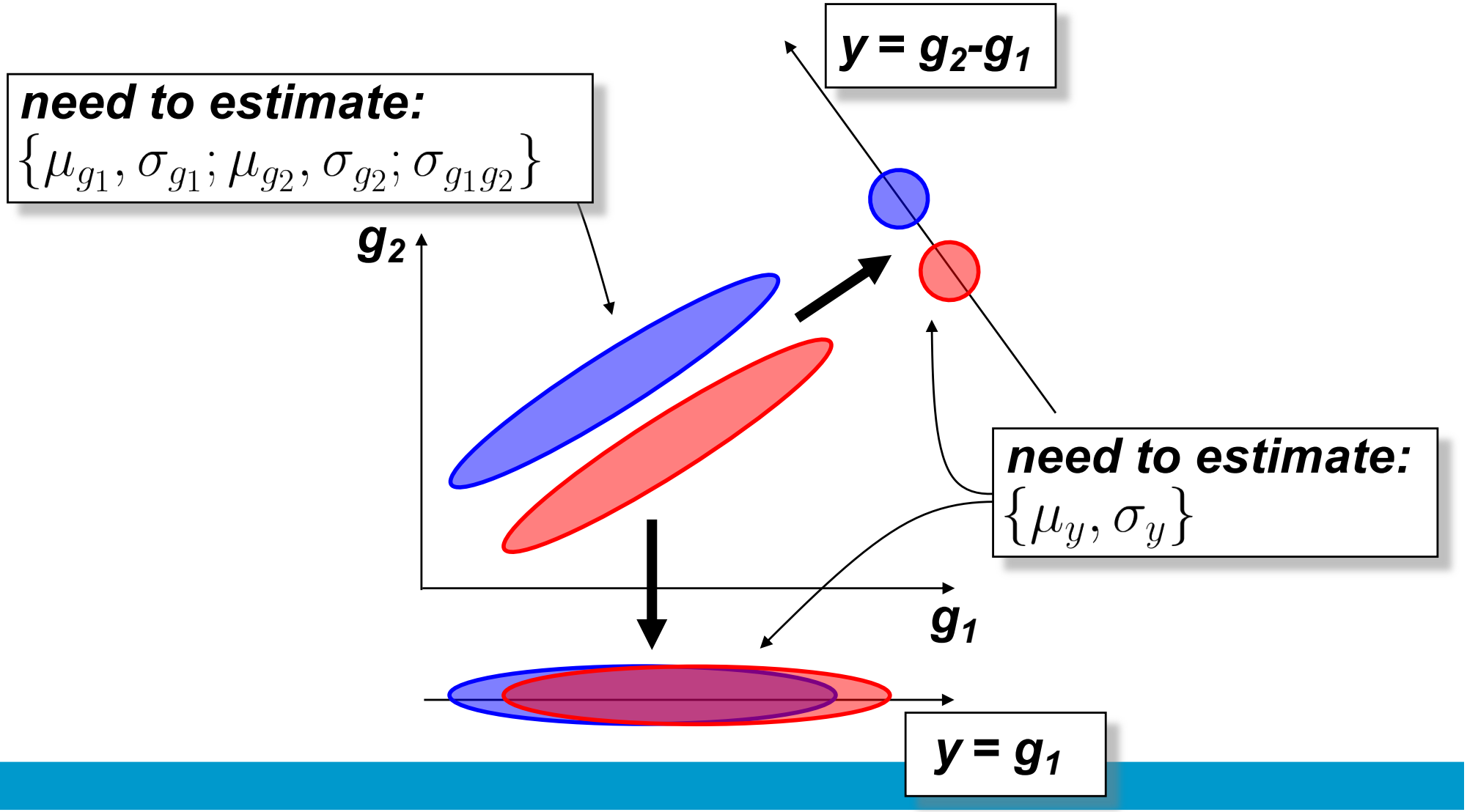
Dimensionality reduction (6)

Transform high-dimensional data to data of lower dimensionality, whilst *preserving the structure* in the original data as good as possible:

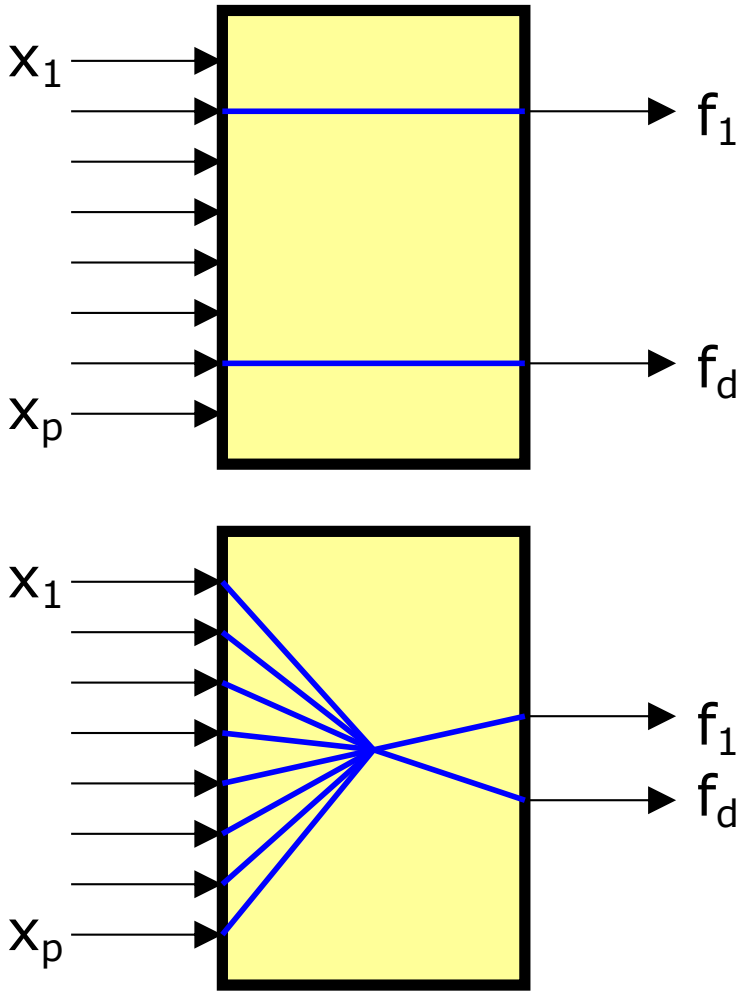


$$y_i = f(X)$$

Projecting to lower dimensions

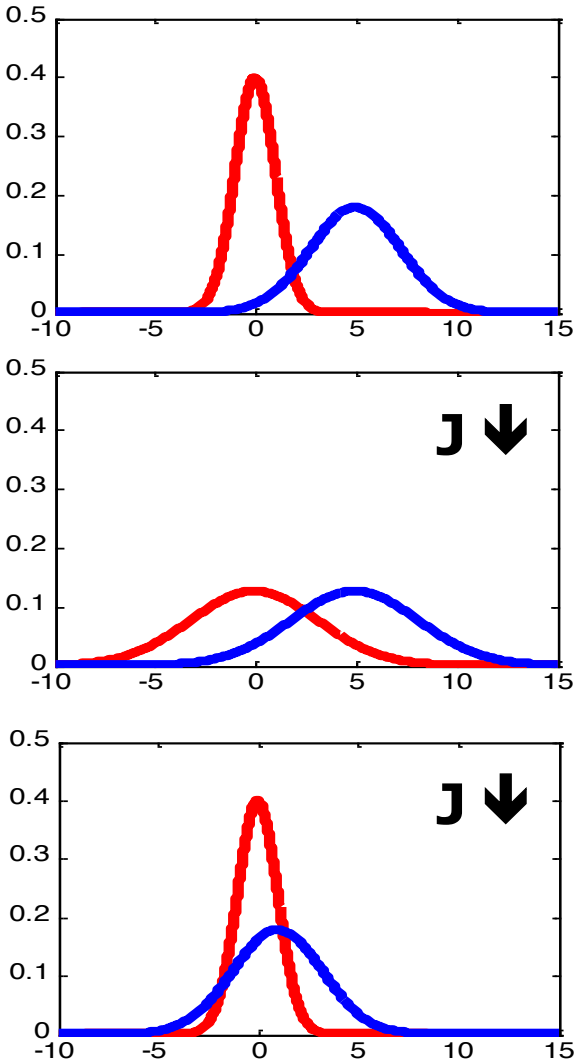


Feature reduction



- **Find most discriminating features**
- **Feature filtering (selection)**
  - Select *d* features out of *p* features
  - Interpretable, but, expensive and approximate
- **Feature mapping (extraction)**
  - Map *p* features to *d* features
  - Fast and optimal, but, still needs all original features
- **More advanced techniques**
  - Multi-variate selection criteria
  - Search algorithms
  - Supervised mappings
  - Non-linear mapping

Feature filtering

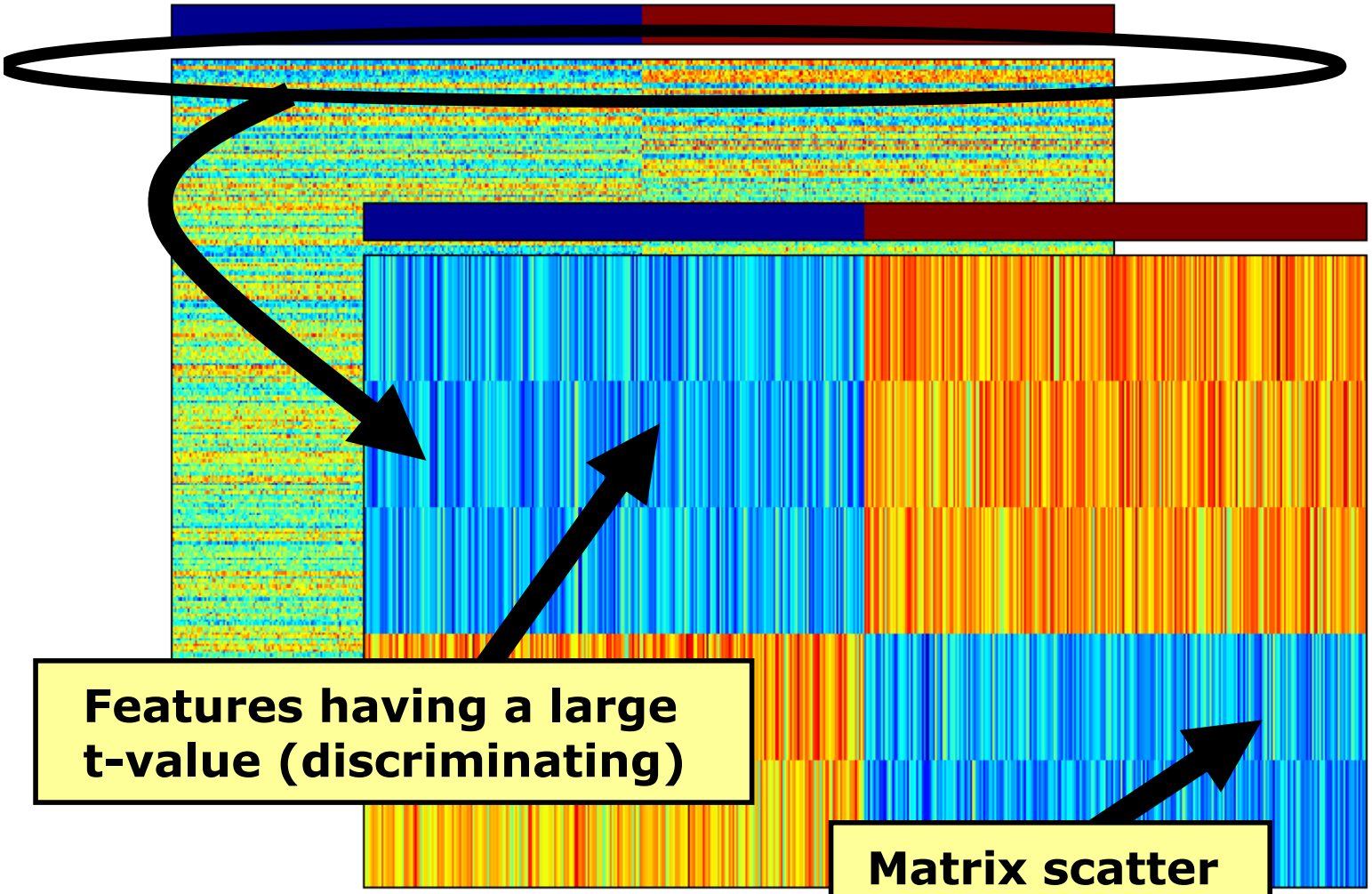


- **Define criterion *J* that expresses *separability* of classes**
- **T-test**
  - Test on difference in mean assuming Gaussian distribution

$$t = \frac{\mu_A - \mu_B}{\sqrt{\frac{\sigma_A^2}{n_A} + \frac{\sigma_B^2}{n_B}}}$$

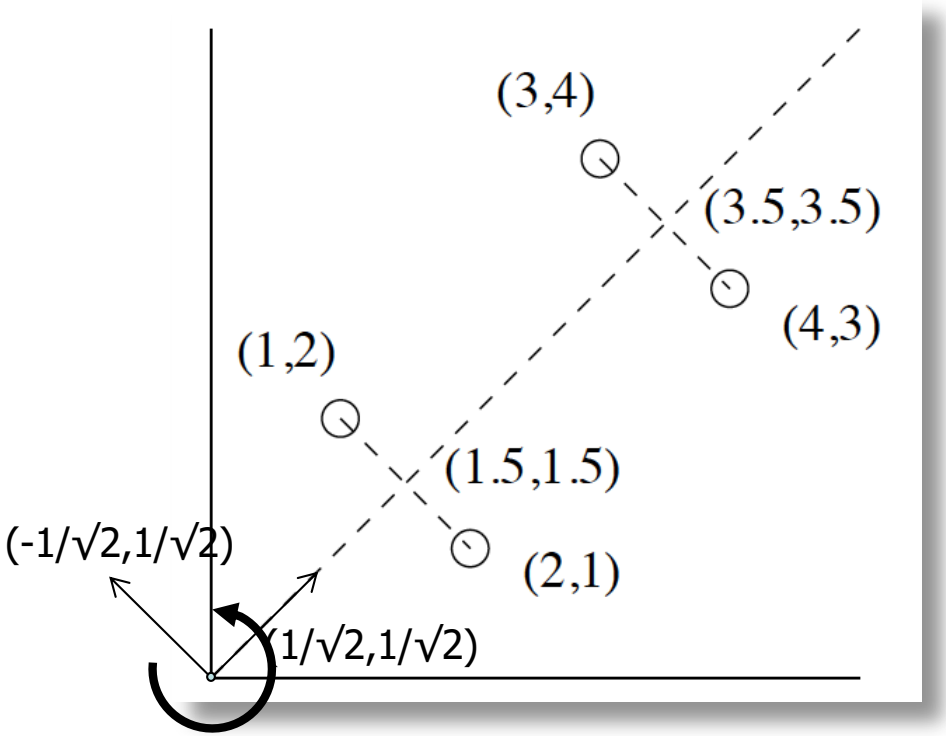
- **Feature filtering**
  - Rank individual features (on *J*)
  - Select the top-N features (scoring best on the criterion *J*)

Example: Feature filtering

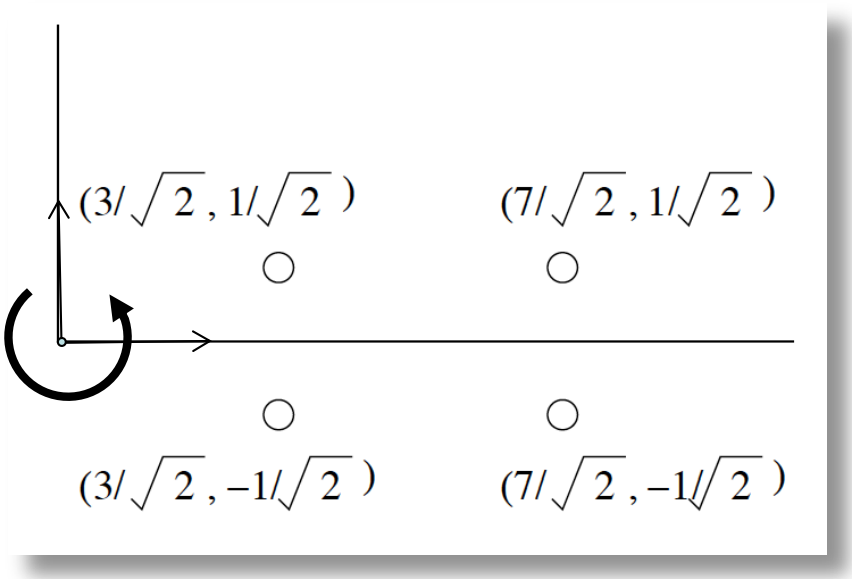


Principal component analysis

PCA is a *linear* techniques to reduce dimensions

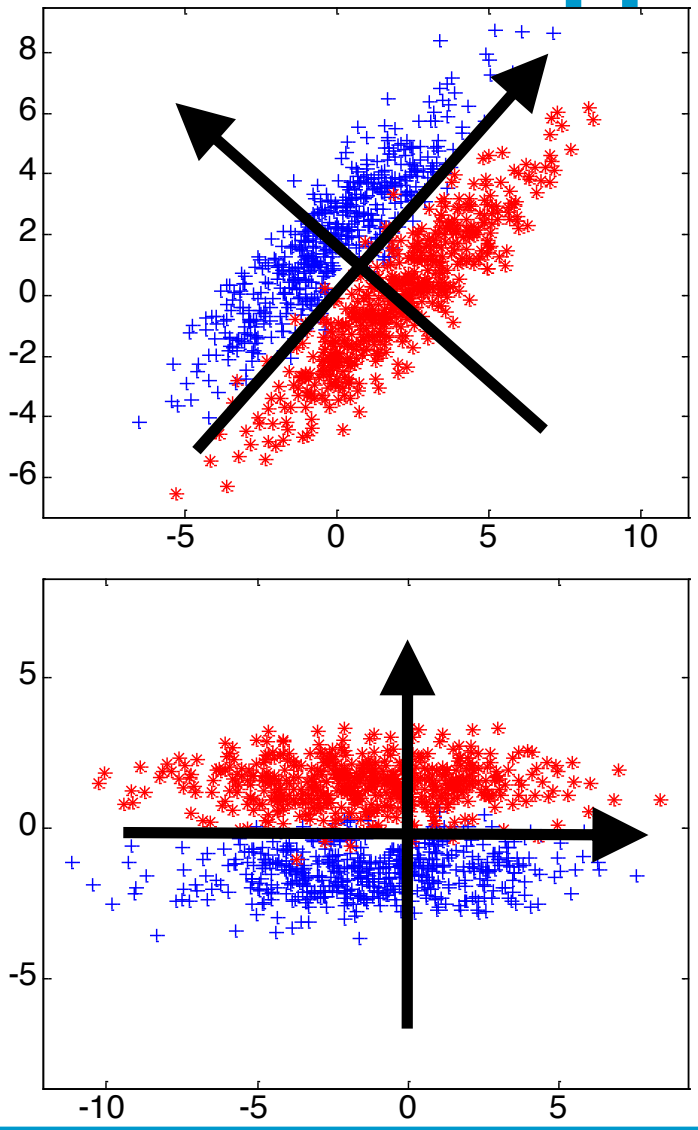


Rotate coordinate system such that variation in data is captures best



Project data on new coordinate system

Feature Mapping: PCA



- **Decorrelate data**
  Find rotation (& translation) of the space such that the data does not show correlations (linear relations)
- **Principal Component Analysis**
  Eigenvalue decomposition of the covariance matrix

$$Y = Xw \quad (\textit{rotation of the data})$$

$$Cov(Y) = I \quad (\textit{mapped data no correlation})$$

$$Cov(Y) = \left((Xw)^T(Xw)\right) = \left(w^T X^T\right)(Xw) = I$$

$$X^T X = \left(w^T\right)^{-1}(w)^{-1} = \left(ww^T\right)^{-1}$$

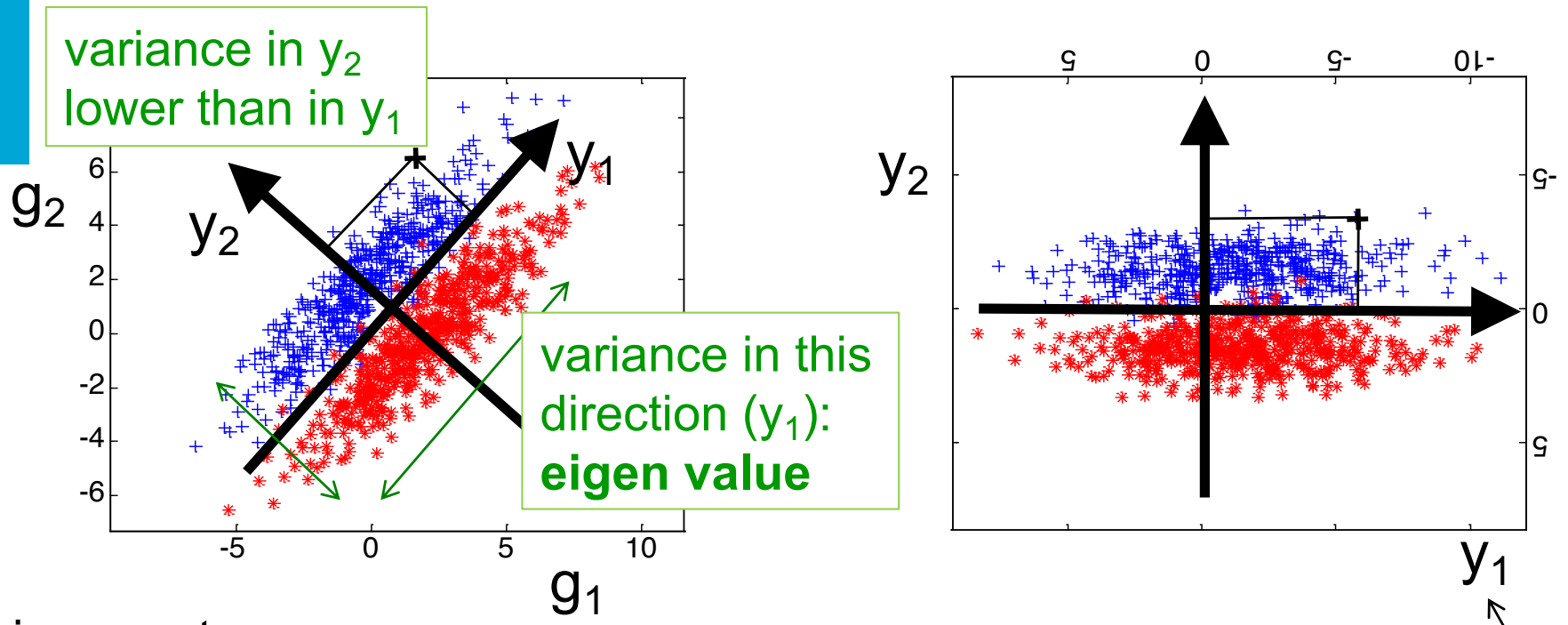
$$\left(X^T X\right)^{-1} = \left(\left(ww^T\right)^{-1}\right)^{-1} = ww^T$$

$$ww^T = \left(Cov(X)\right)^{-1} = \left(\Sigma_X\right)^{-1} = \left(E \Lambda E^T\right)^{-1} = E\Lambda^{-1}E^T$$

$$ww^T = \left(E\Lambda^{-\frac{1}{2}}\right)\left(\Lambda^{-\frac{1}{2}}E^T\right) = \left(E\Lambda^{-\frac{1}{2}}\right)\left(E\Lambda^{-\frac{1}{2}}\right)^T$$

$$Y = XE\Lambda^{-\frac{1}{2}}$$

PCA transforms the space



- eigen vector
- principal component
- eigen gene

$$y_1 = w_{11}g_1 + w_{12}g_2$$
$$y_2 = w_{21}g_1 + w_{22}g_2$$

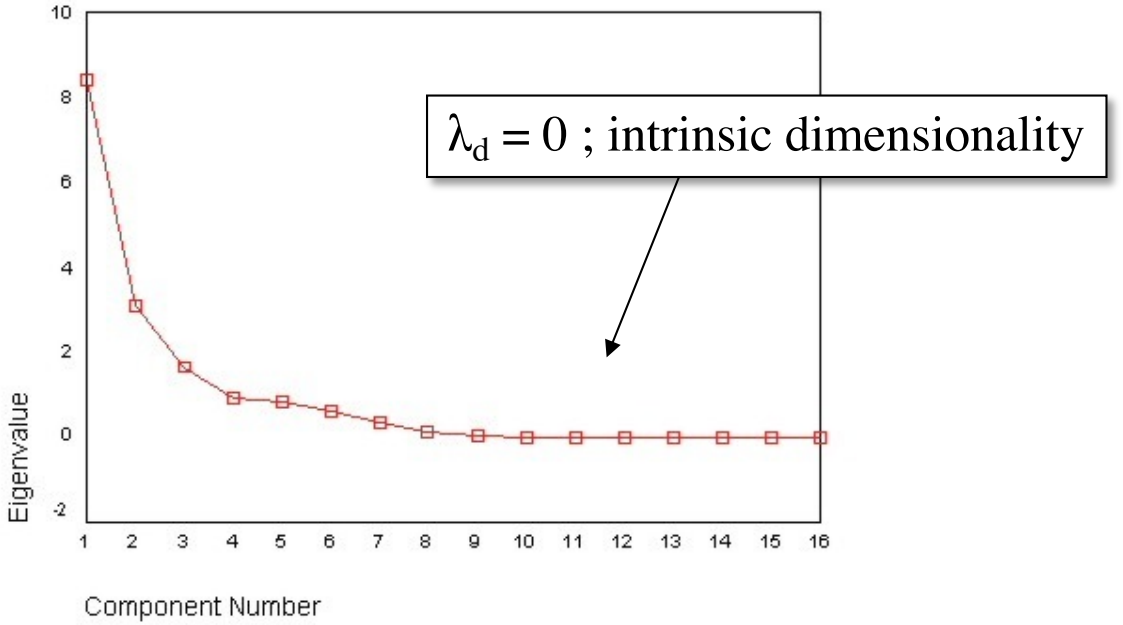
loading factor

(importance of gene in contributing to variance)

component showing the most variation

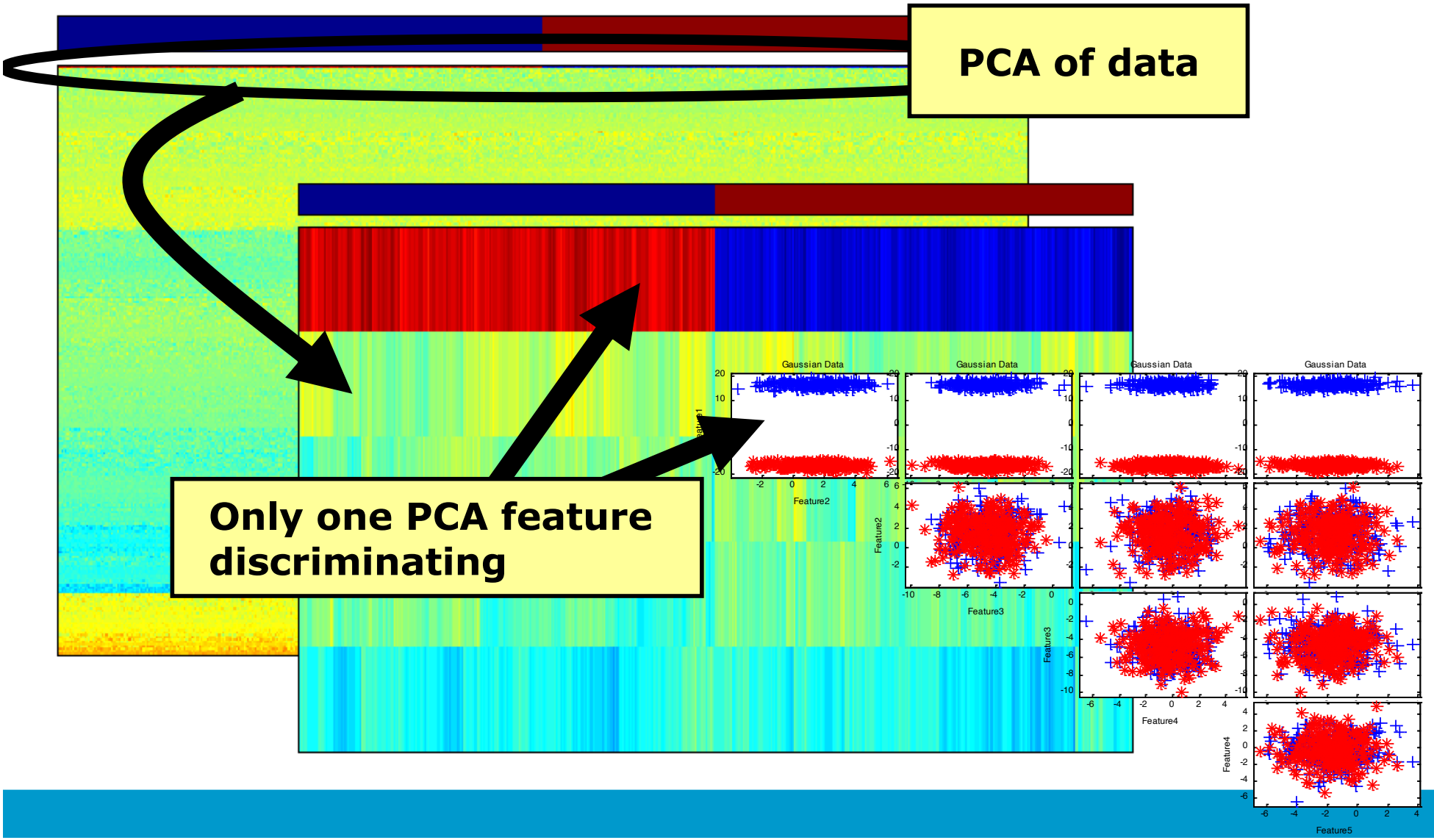
PCA scree plot

- *Scree plot* of eigenvalues shows amount of variance retained by the eigenvectors (*principal components, PCs*):



- First *K* PCs explain $\frac{\sum_{d=1}^{K} \lambda_d}{\sum_{d'=1}^{D} \lambda_{d'}} \times 100\%$ of variance

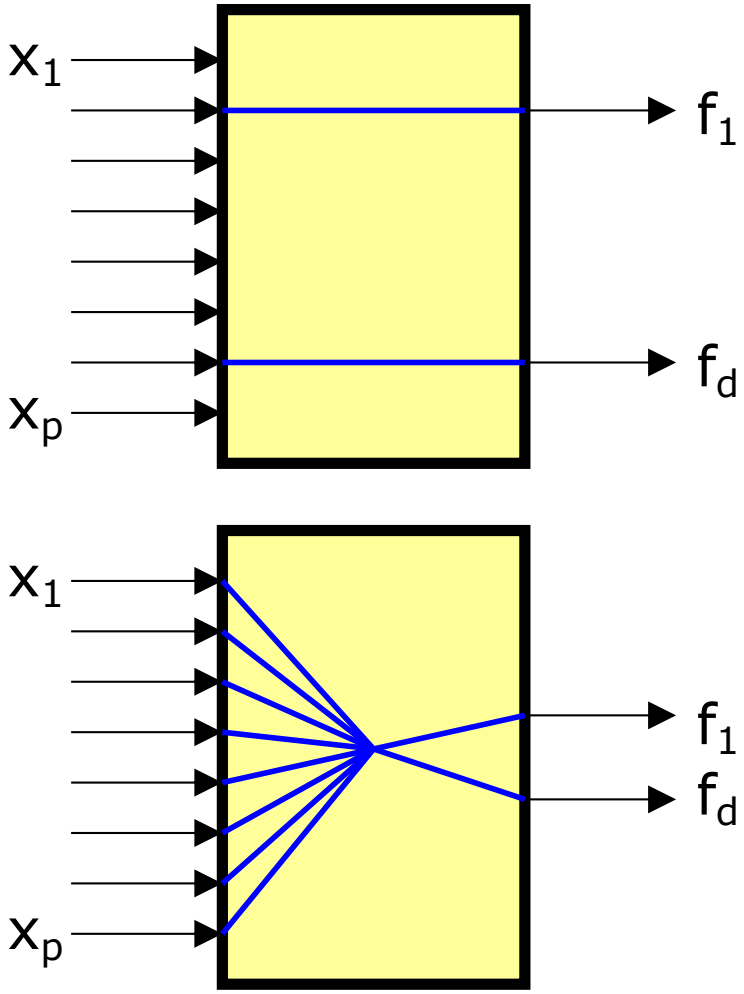
Example: PCA



The Delft Bioinformatics Lab

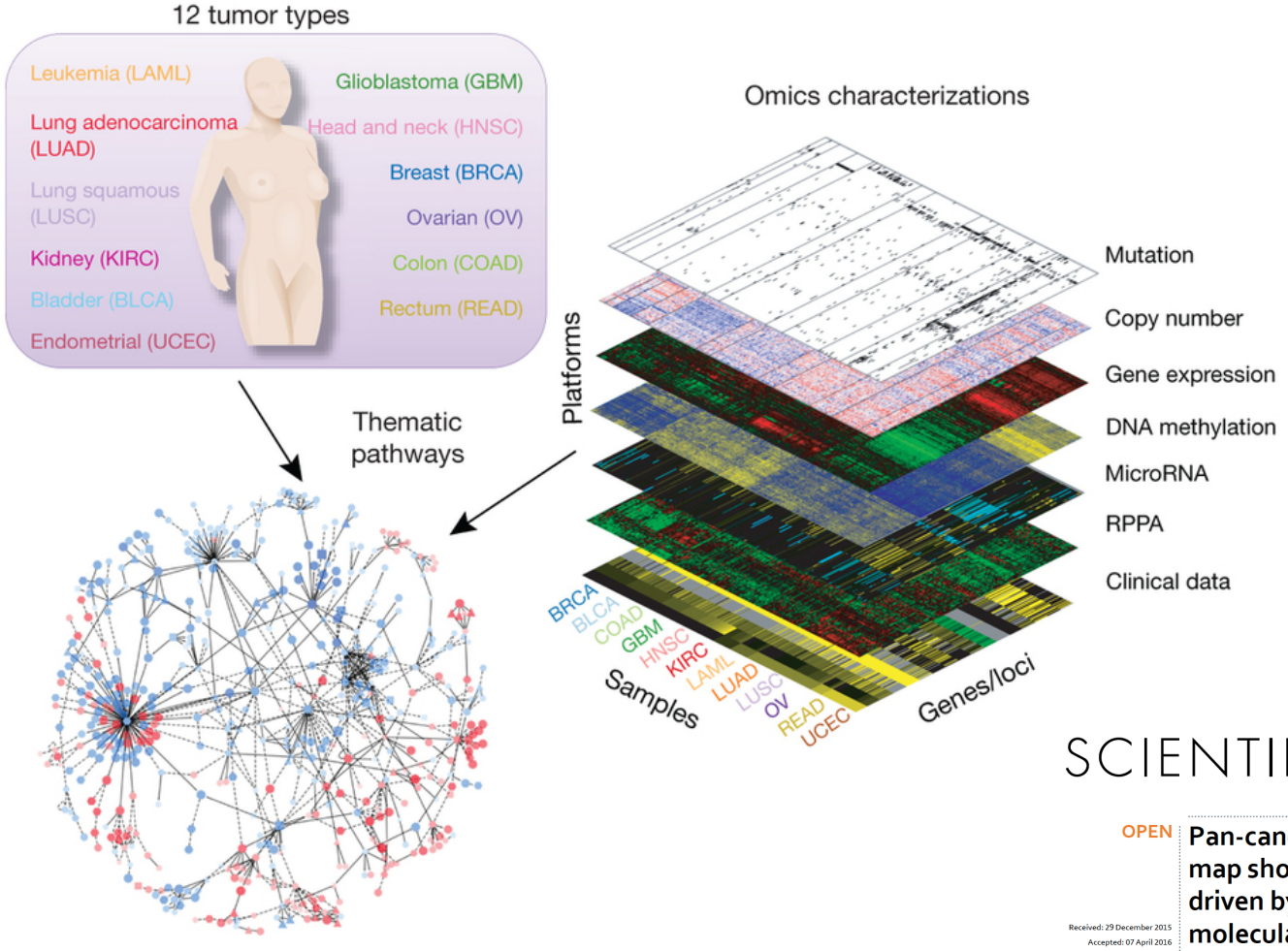
TU Delft

Feature reduction (2)



- **But: How many features to use ?**
- **Naïve approach**
  - Feature filtering: threshold the t-values (retained separability)
  - Feature mapping: threshold the eigenvalues of the covariance matrix (variance retained)
- **Or, more advanced**
  - based on some performance that can be reached on the basis of the final *d* features

Example: TCGA data (1)



SCIENTIFIC REPORTS

OPEN

Pan-cancer subtyping in a 2D-map shows substructures that are driven by specific combinations of molecular characteristics

Received: 29 December 2015
Accepted: 07 April 2016
Published: 25 April 2016

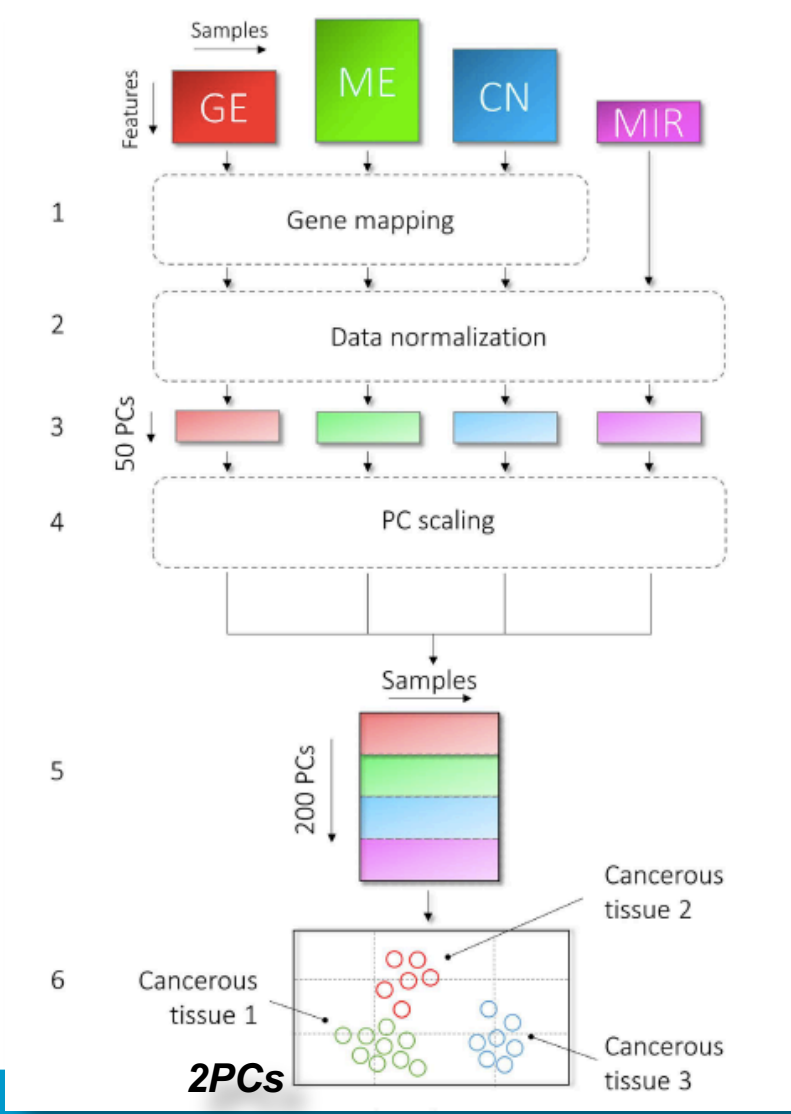
Erdogan Taskesen[1], Sjoerd M. H. Huisman[1,2], Ahmed Mahfouz[1,2], Jesse H. Krijthe[1], Jeroen de Ridder[1], Anja van de Stolpe[1], Erik van den Akker[1], Wim Verheagh[1] & Marcel J. T. Reinders[1]

Taskesen, Reinders et al. Scientific Reports. 2016.

The Delft Bioinformatics Lab

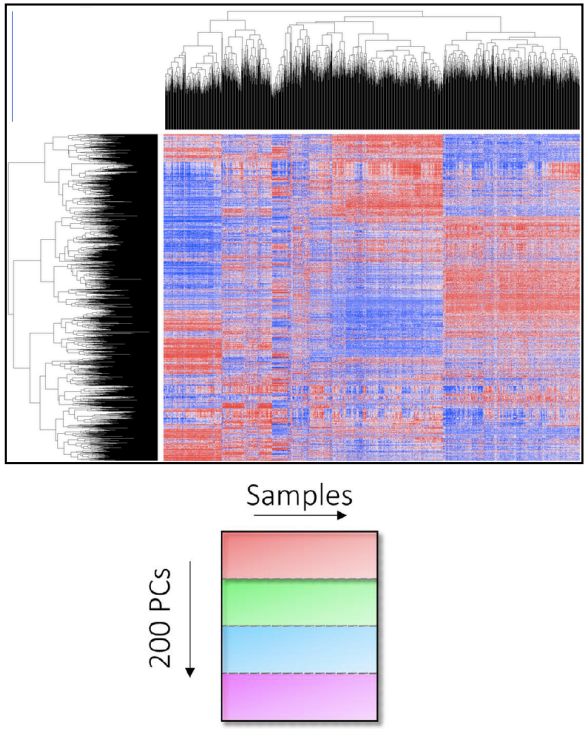
TU Delft

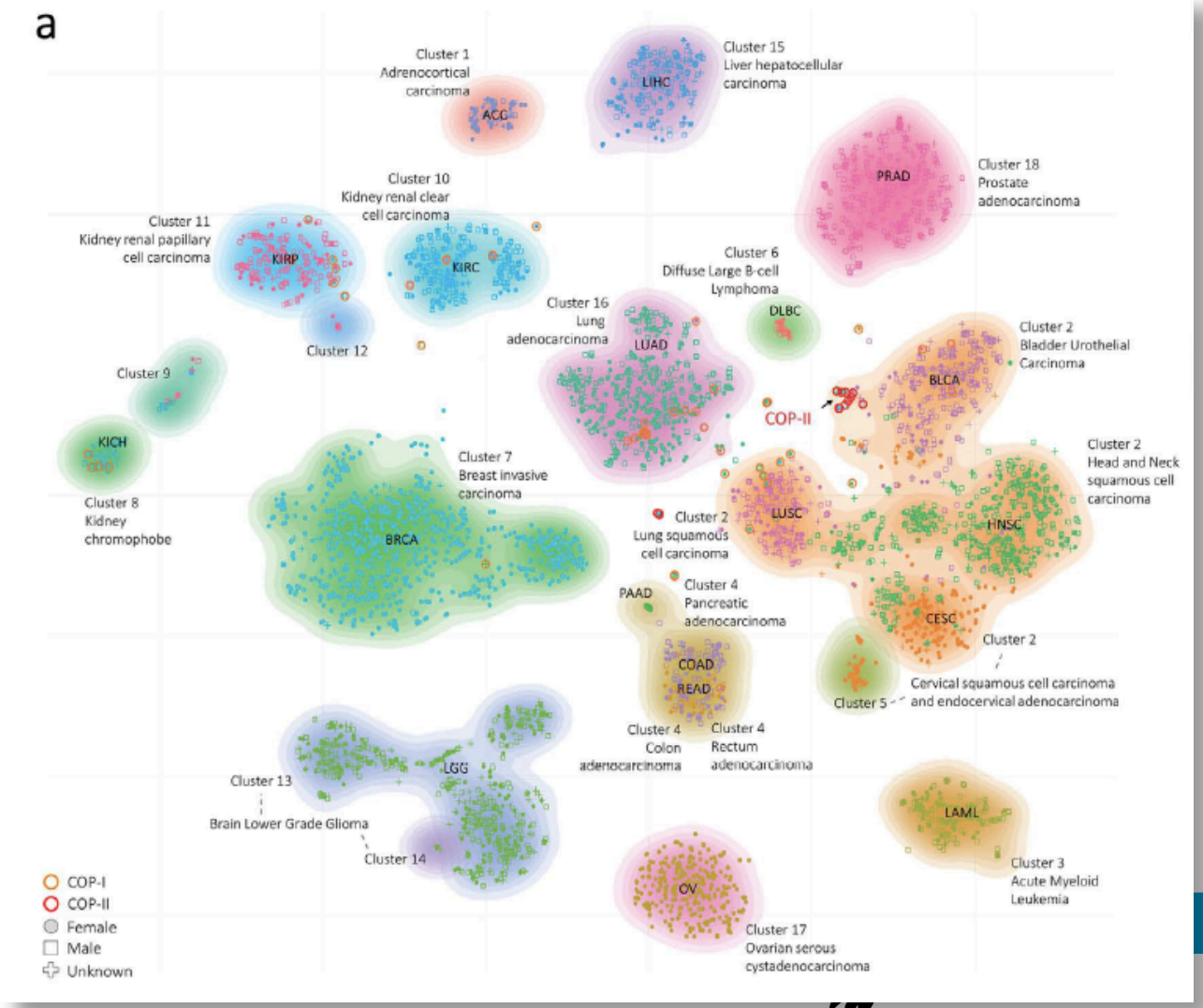
Example: TCGA data (2)



Dimension reduction (4)

Visualize in scatter plot





The Delft Bioinformatics Lab

TU Delft

Dimension reduction: Summary

- Dimensionality reduction can be helpful to visually inspect data and to remove uninformative information
- There are two ways: selection and extraction
- Principal Component Analysis is a linear approach for feature extraction which removes correlations between the features
- First eigenvectors are the most important, and loading factors indicate which (original) features contribute to these PCs