Machine Learning for Bioinformatics & Systems Biology

0. On-line background

Perry Moerland *Amsterdam UMC, University of Amsterdam*

Marcel Reinders *Delft University of Technology*

Lodewyk Wessels *Netherlands Cancer Institute*

Some material courtesy of Robert Duin, David Tax & Dick de Ridder

Modelling …. Learning from examples

Machine learning

- Wikipedia:
  - "the scientific study of **algorithms** and **statistical models** that computer systems use to perform a specific task without using explicit instructions, relying on **patterns** and **inference** instead … Machine learning algorithms build a **mathematical model** based on **sample data**, known as "**training data**", in order to make **predictions** or **decisions** without being explicitly programmed to perform the task.“
- Christopher M. Bishop:
  - "**Pattern recognition** has its origins in **engineering**, whereas **machine learning** grew out of **computer science**. However, these … can be viewed a two facets of the same **same field**"

Machine learning (2)

- The construction of **approximate, generalizing (predictive) models** by **learning from examples**, for problems for which *no full physical model is known* (yet)
- Focus in this course will be on **classification** and **statistical machine learning**, not (so much) on *regression, structural/syntactic* pattern recognition and *reinforcement learning*.

- Related areas
  - Applied statistics
  - Pattern recognition
  - Artificial intelligence
  - Computer vision
  - Data mining

Face recognition, no physical model

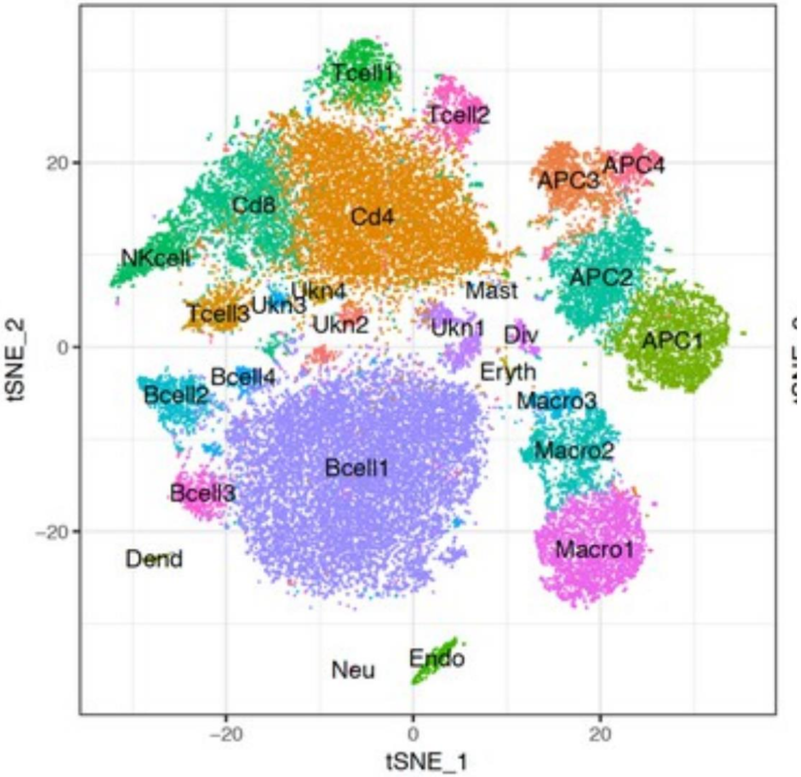
Machine learning (3)

- Examples:
  - Computer vision: license plate reading, people counting, face detection, smart cameras, ...
  - Signal processing: thermostat, speech/speaker recognition, ...
  - Information retrieval: Google, Amazon, automated translation, ...
  - Biometrics: fingerprint recognition, iris scan, signature verification...
  - Defensive: friend-or-foe recognition, target tracking, ...
  - Medicine: interpreting scans, diagnostic systems, ...

BioSB

Machine learning (4)

- Bioinformatics:
  - Gene (function) prediction, SNP prioritization, …
  - Diagnosis/prognosis, biomarker discovery, …
  - Network inference: PPI, metabolic networks, …
  - Cell-type identification, …
  - Etc.



Goal

- After having followed this course, the student has a good understanding of a wide range of machine learning techniques and is able to recognize what method is most applicable to data analysis problems (s)he encounters in bioinformatics and systems biology applications.
- Many problems are in fact machine learning problems!

BioSB

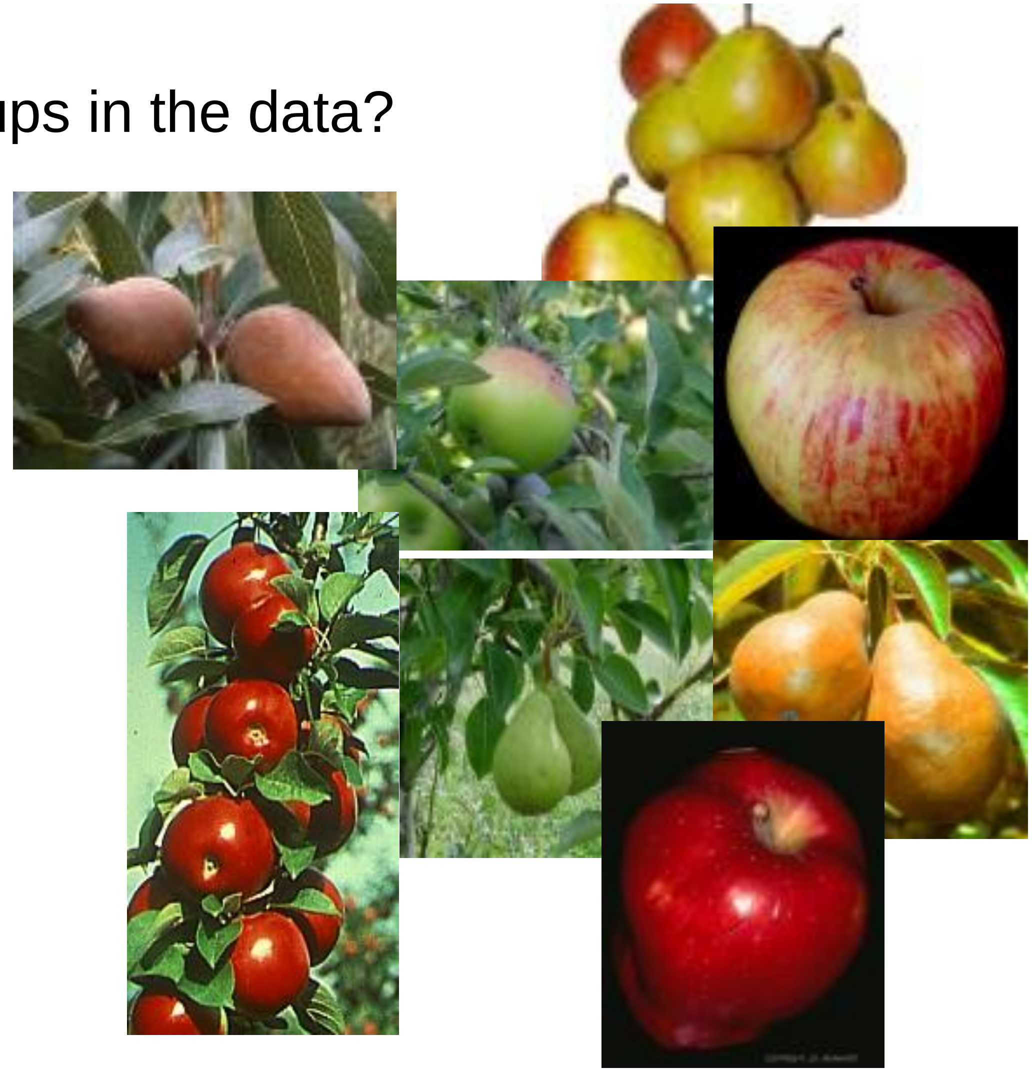
Machine learning (5)

- Finding structure in data
  - Outlier/anomaly detection
  - Clustering
  - Dimensionality reduction,
    selecting useful (combinations of) features
  - Regression
  - Classification
  - ...

- All aimed at *generalisation*:
  making a prediction for data you have not yet seen

Clustering

- Can we find natural groups in the data?
- E.g. red vs green fruit

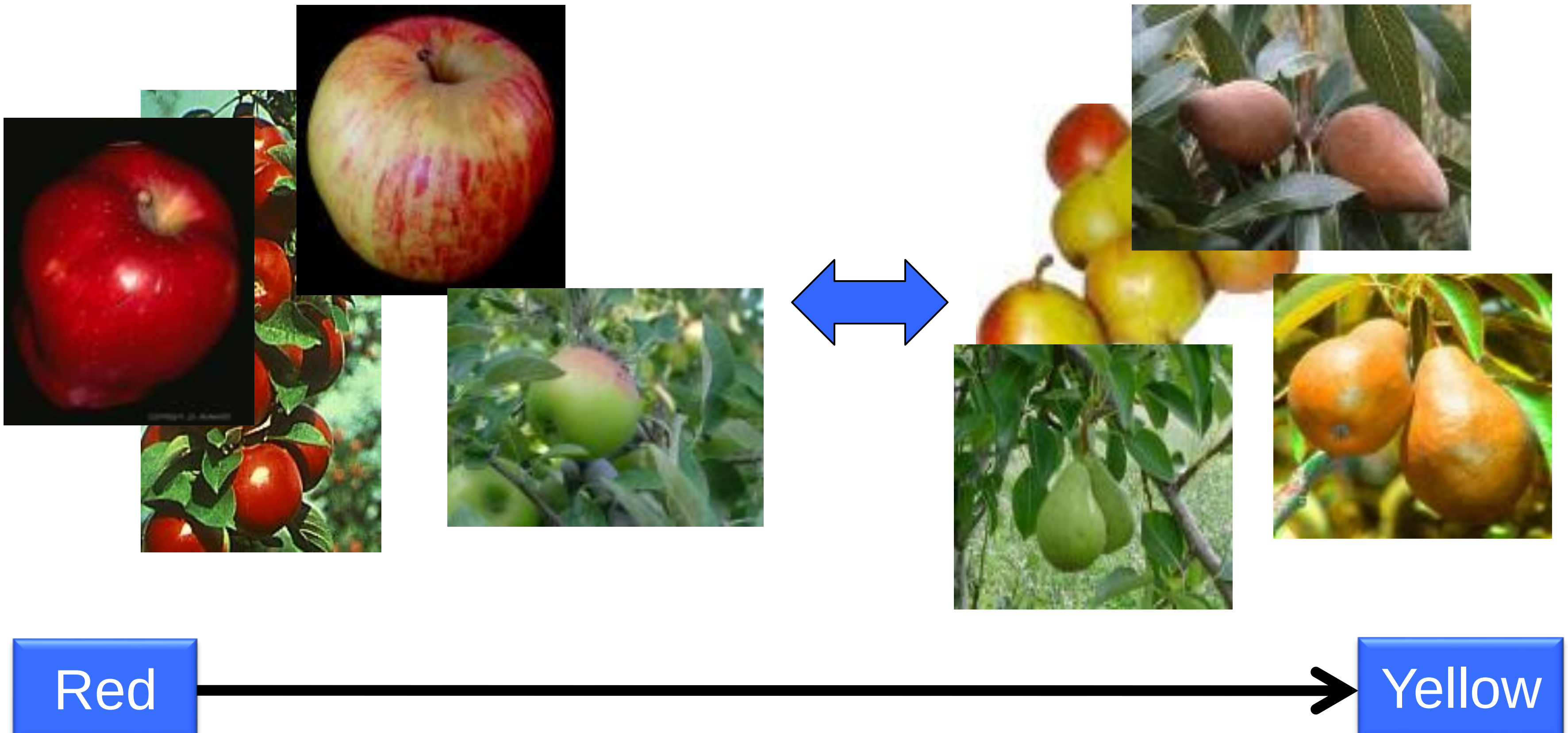


Outlier detection

- Can we find strange objects?

Dimensionality reduction

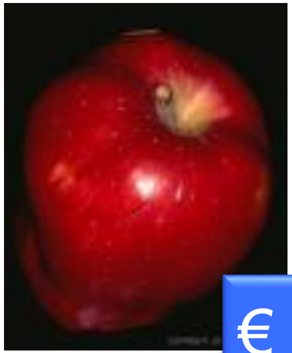
- Can we find predictive measurements?



Regression

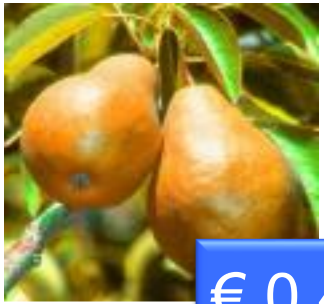
- Can we predict real-valued outputs?

€ 0.25



€ 0.30

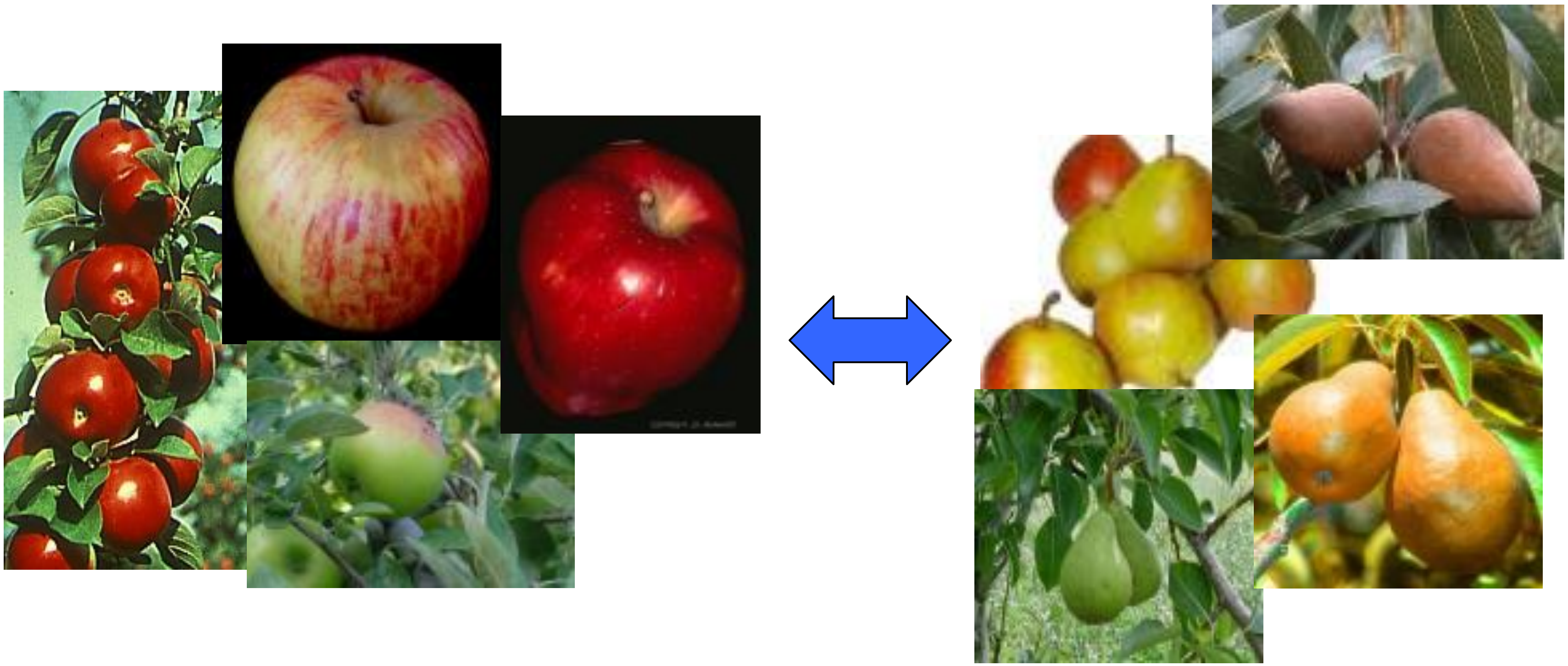
€ 0.40



€ 0.45

Classification

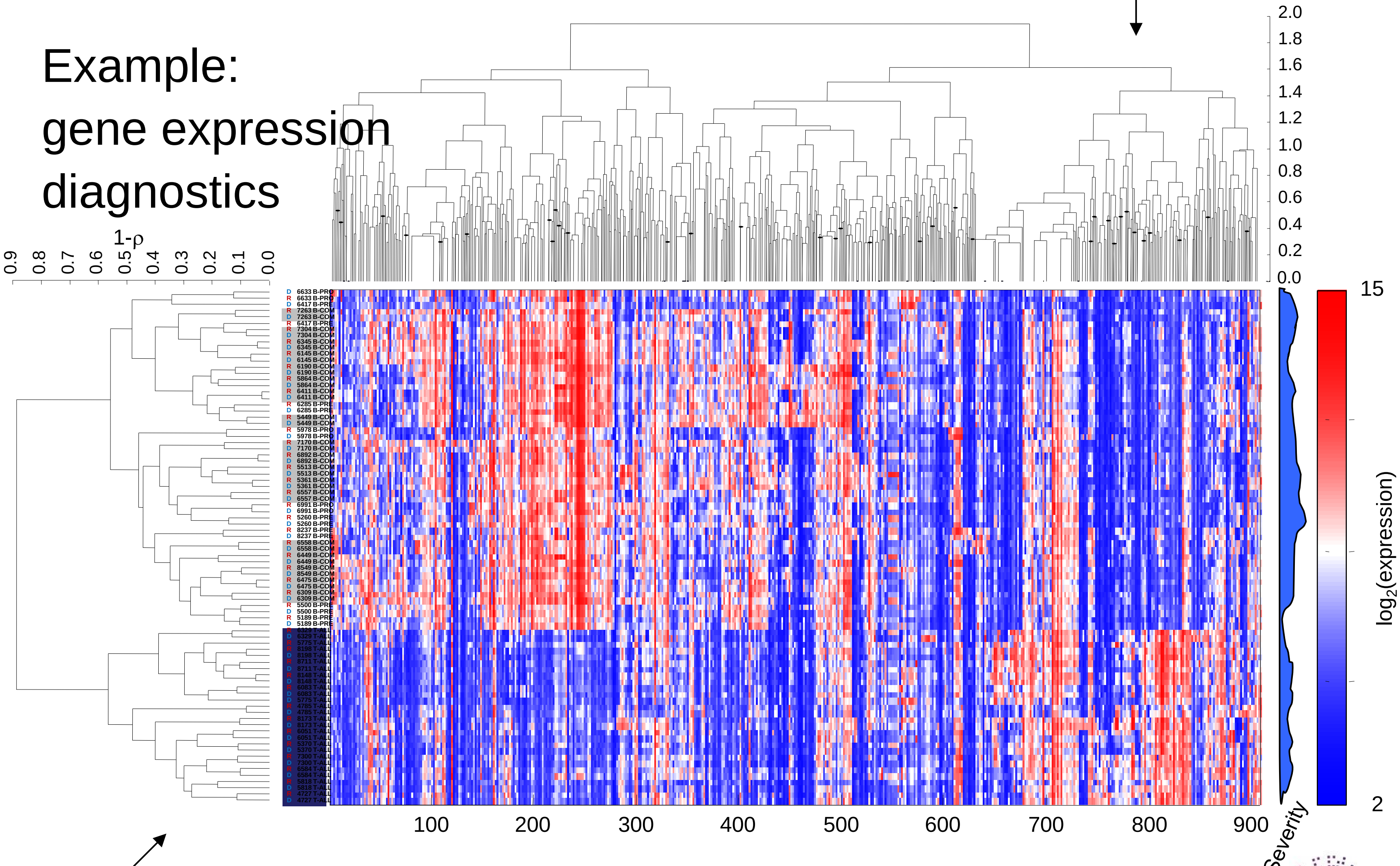
- Can we distinguish apples from pears?



Machine learning in bioinformatics

- Example:
  gene expression
  diagnostics

genes



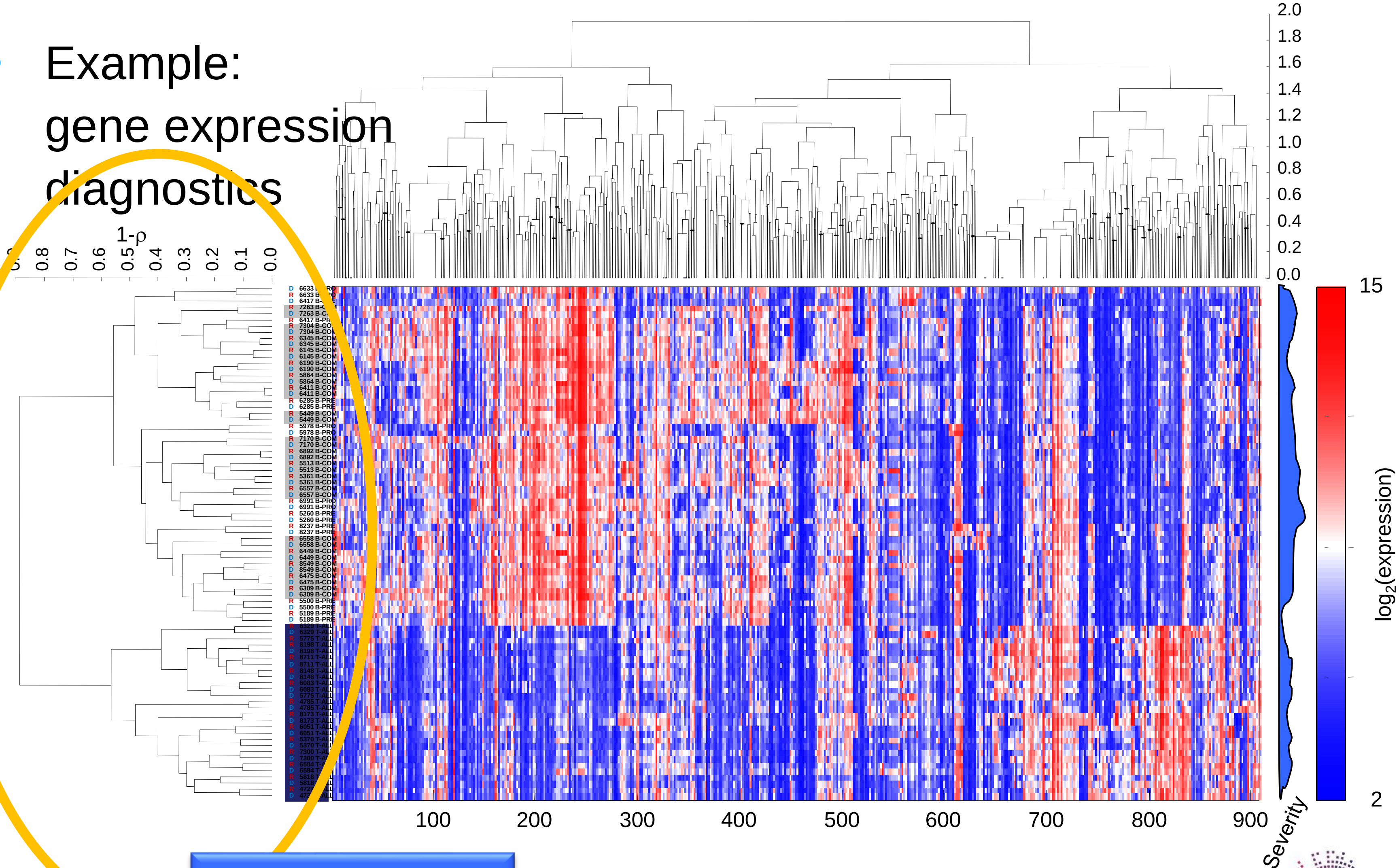
samples

diagnosis/relapse in childhood leukemia

BioSB

Machine learning in bioinformatics

- Example: gene expression diagnostics

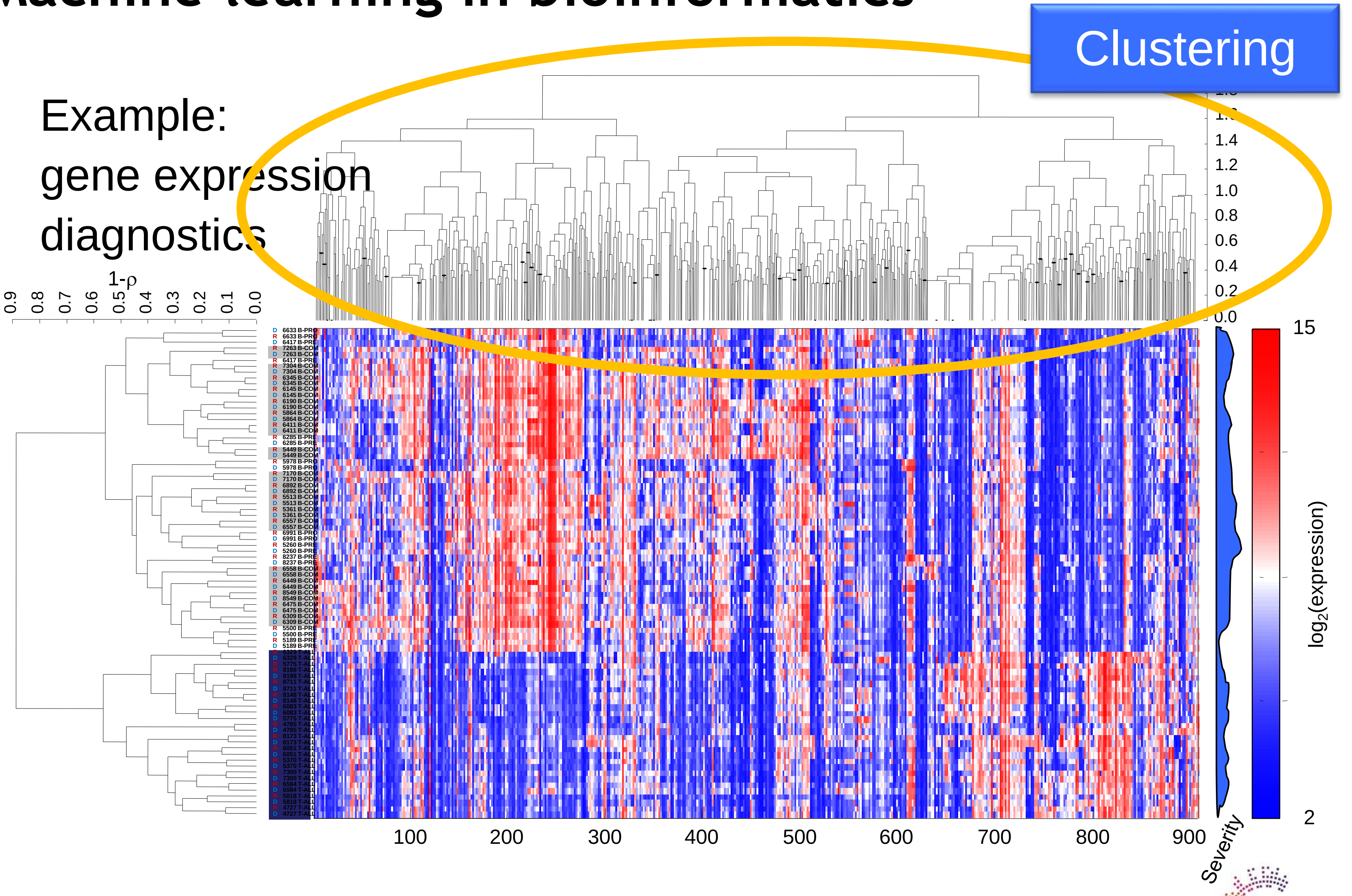


Clustering

Clustering of patients: similar subtypes of disease

Machine learning in bioinformatics

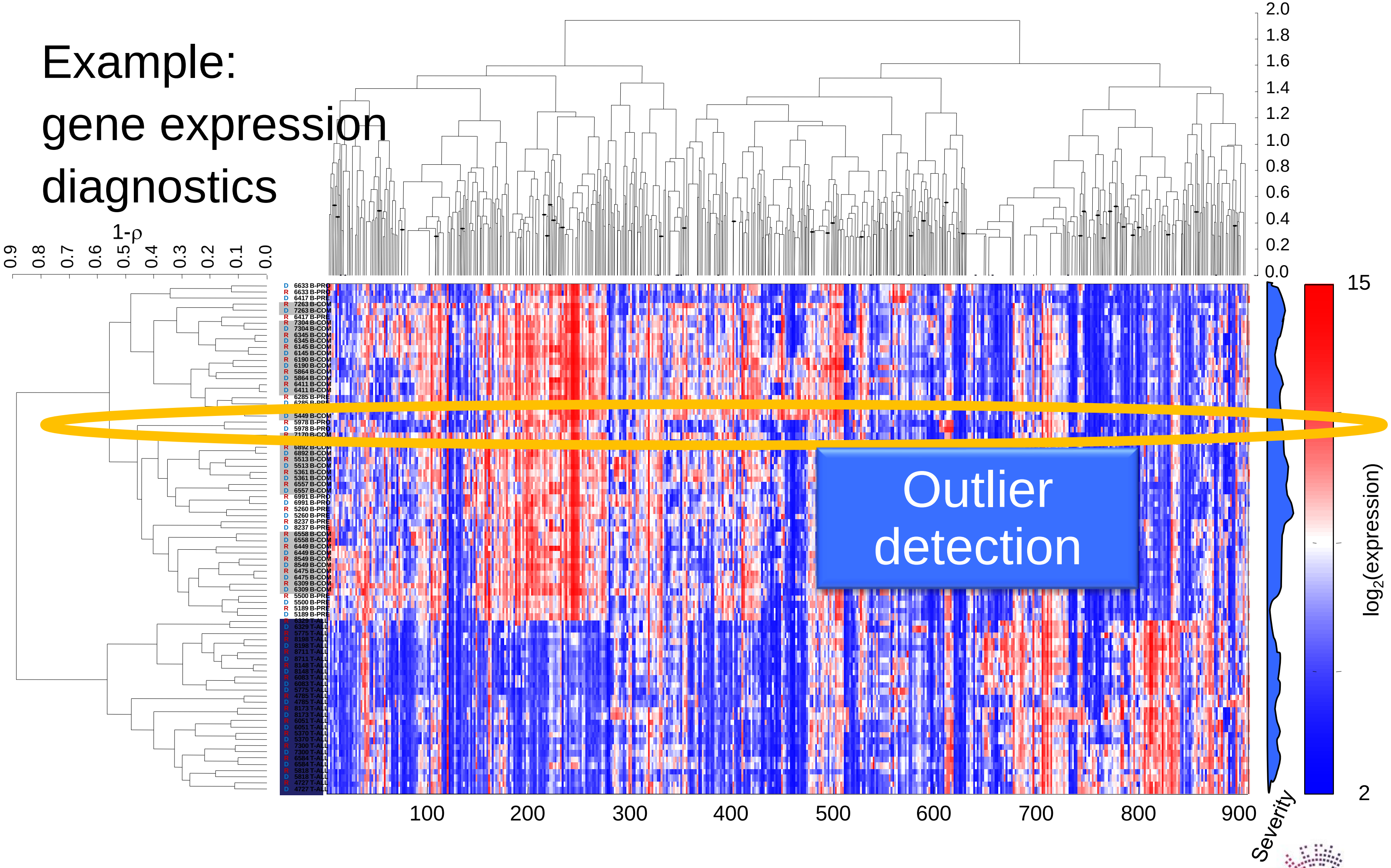
- Example:
  gene expression
  diagnostics



Clustering of genes: similar 'disruptive' processes

Machine learning in bioinformatics

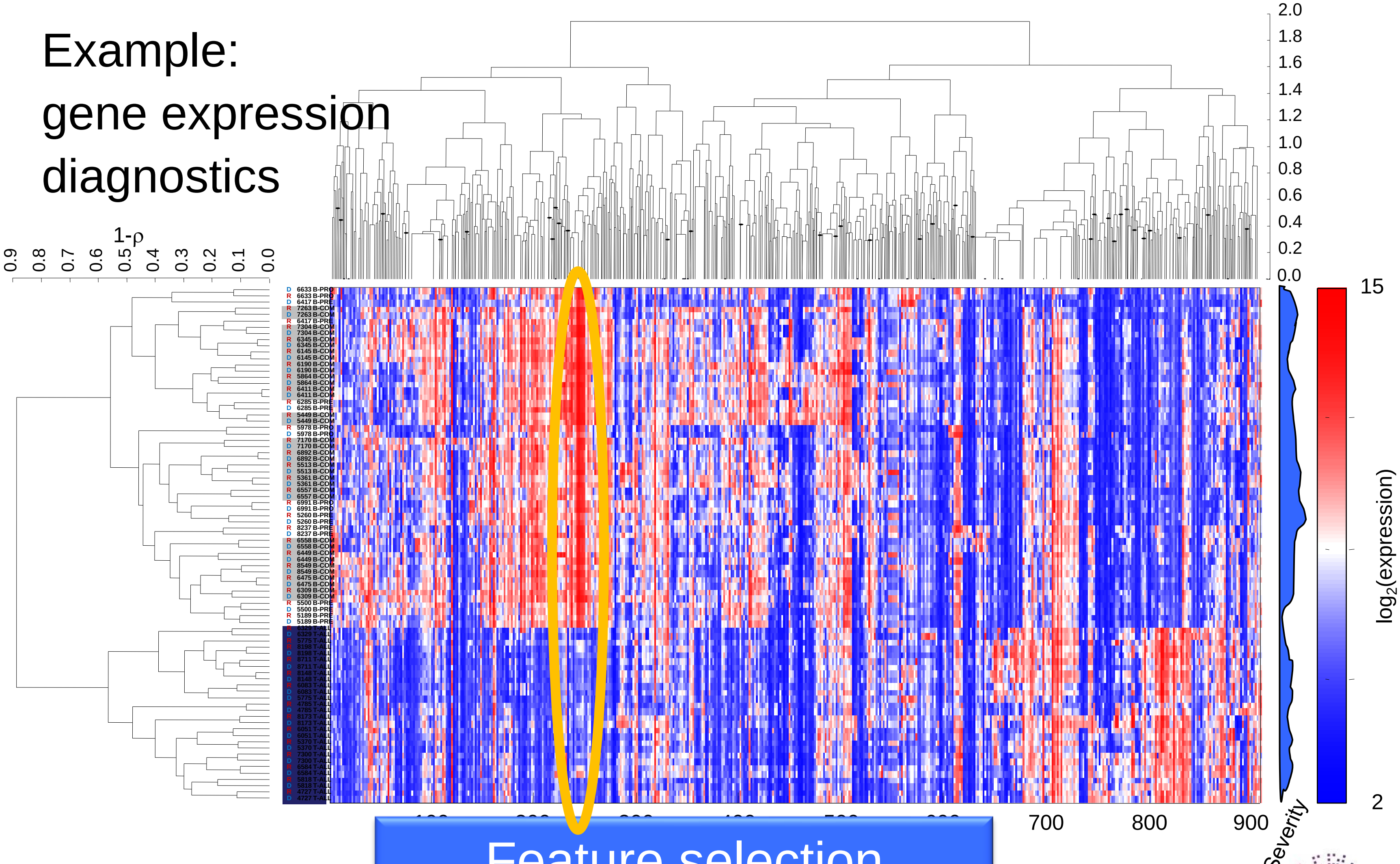
- Example:
  gene expression
  diagnostics



Technical error / rare patient-rare genetic background

Machine learning in bioinformatics

- Example:
  gene expression
  diagnostics

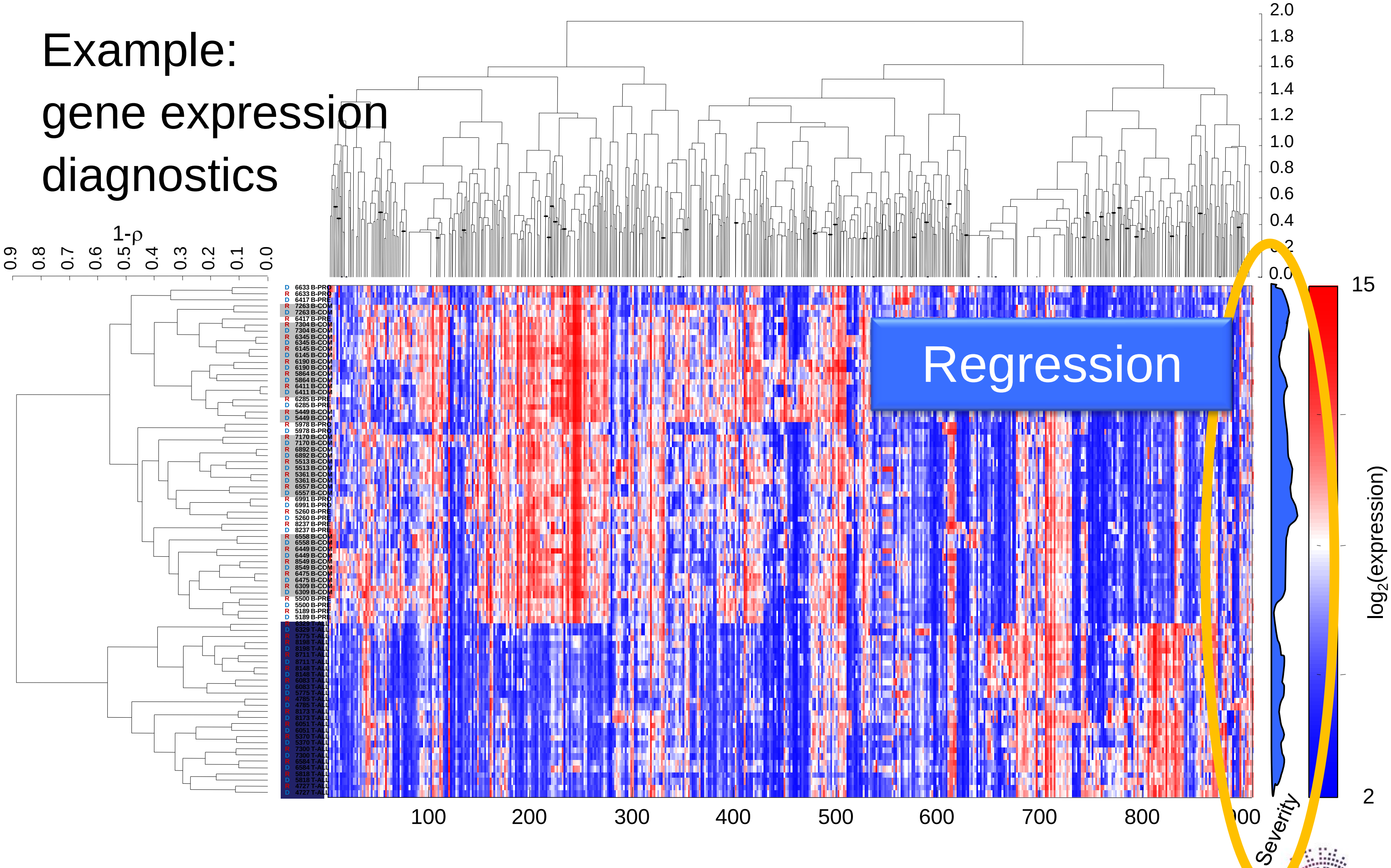


Feature selection
(dimensionality reduction)

Potential biomarkers

Machine learning in bioinformatics

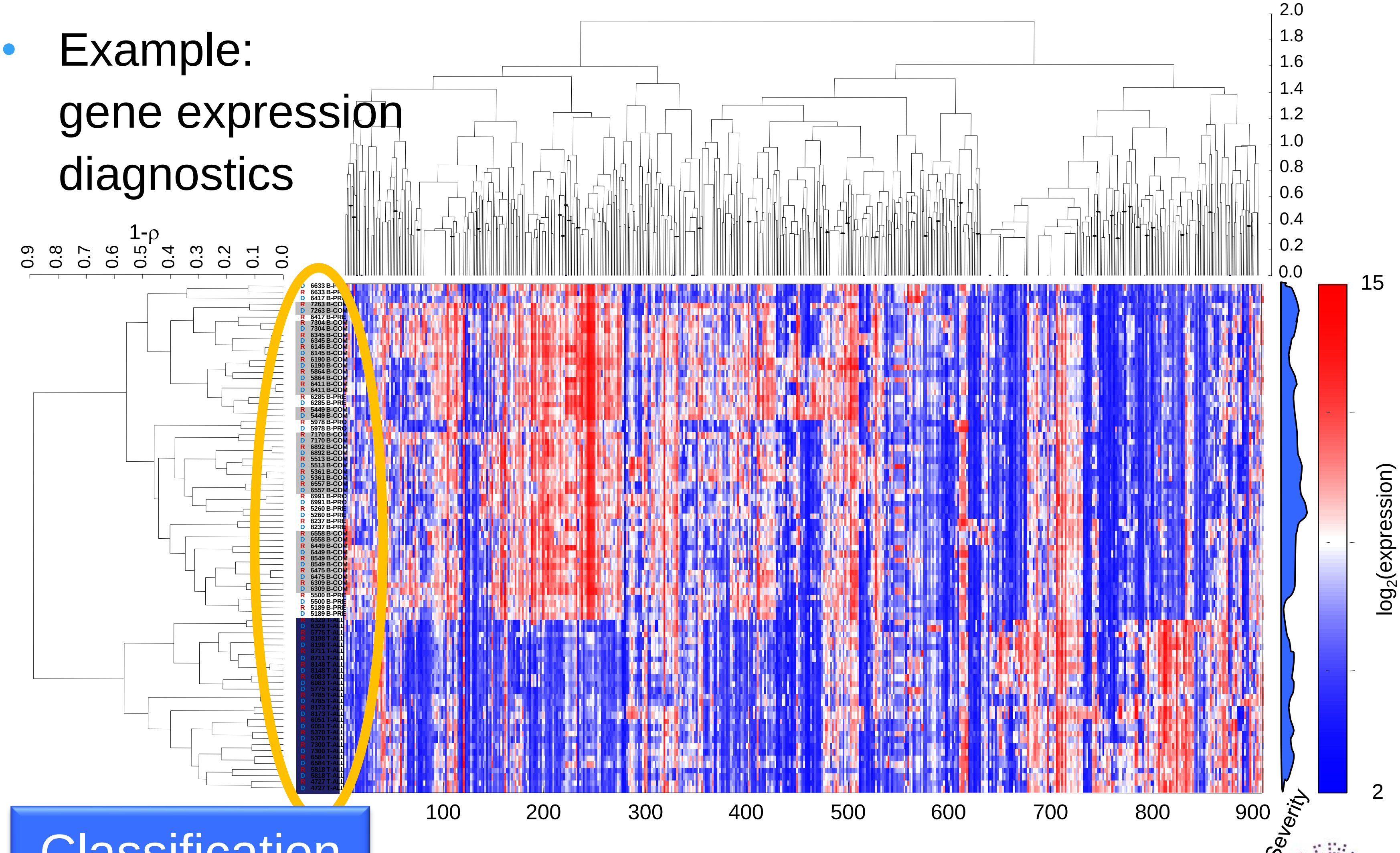
- Example:
  gene expression
  diagnostics



E.g. Predicting survival time

Machine learning in bioinformatics

- Example:
  gene expression
  diagnostics



Classification

E.g. Predicting metastasis

Machine learning in bioinformatics (2)

- Tools applicable to any type of biological data

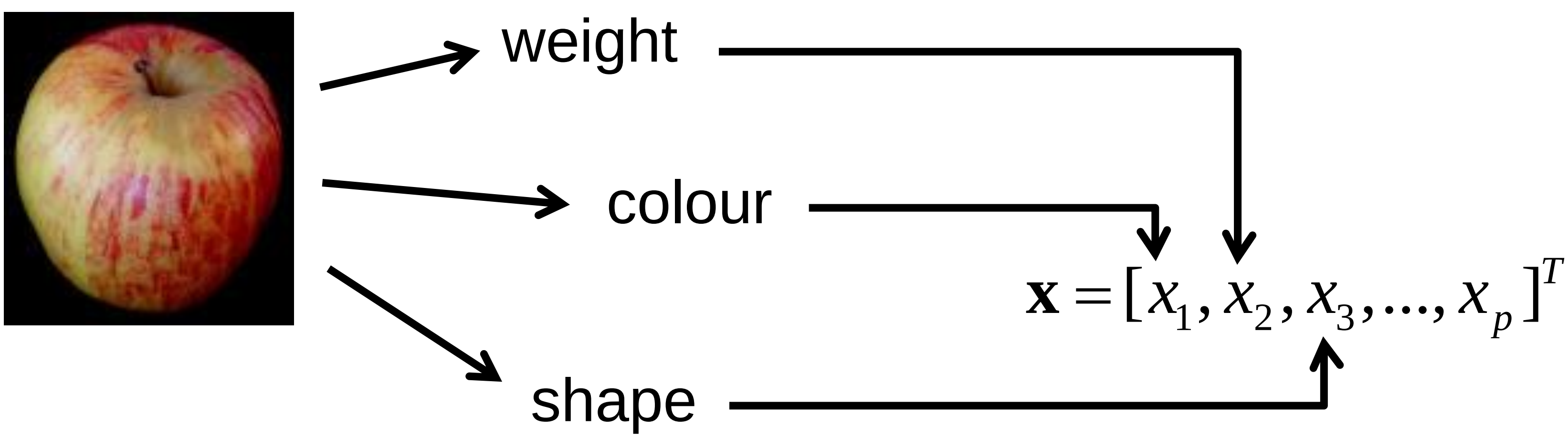
- Examples:
  - Protein sequence data:
    - Clustering: finding orthologous groups
    - Classification: prediction of EC number, subcellular localization, …
    - Regression: predicting secondary structure
  - TF binding data (ChIP):
    - Clustering: finding functional gene groups
    - Classification: predicting gene annotation
    - Regression: finding cis-regulatory modules
  - …

BioSB

Terminology

Measurements and features

- To automate these tasks, we have to find a mathematical *representation* of objects
- Objects are usually represented by *features*, i.e. sets of useful *measurements* obtained from some *sensors*



Measurements and features (2)

- This course assumes measurements as given, i.e. sensor accuracy etc. are not *explicitly* modeled

- However,
  - in general measurements will never be perfect
  - objects within a class will vary intrinsically

- Hence, we need statistics to model all variation

This is important!
If we know everything and there is no noise, you'll need different algorithms/models

Datasets

- A *dataset* is a set of measurements on many objects

- For clustering:

Object	Weight	Colour
Apple #1	25	36
Apple #2	20	34
Apple #3	35	40
Pear #1	35	55
Pear #2	37	55
Pear #3	40	57
Pear #4	36	41

Datasets

- A *dataset* is a set of measurements on many objects

- For regression:

Object	Weight	Colour	Price
Apple #1	25	36	0.21
Apple #2	20	34	0.17
Apple #3	35	40	0.33
Pear #1	35	55	0.41
Pear #2	37	55	0.26
Pear #3	40	57	0.35
Pear #4	36	41	0.29

BioSB

Datasets

- A *dataset* is a set of measurements on many objects

- For classification:

Object	Weight	Colour	Label
Apple #1	25	36	A
Apple #2	20	34	A
Apple #3	35	40	A
Pear #1	35	55	P
Pear #2	37	55	P
Pear #3	40	57	P
Pear #4	36	41	P

Datasets

- A *dataset* is a set of measurements on many objects

- For classification:

Object	Weight	Colour	Label
Apple #1	25	36	A
Apple #2	20	34	A
Apple #3	35	40	A
Pear #1	35	55	P
Pear #2	37	55	P
Pear #3	40	57	P
Pear #4	36	41	P

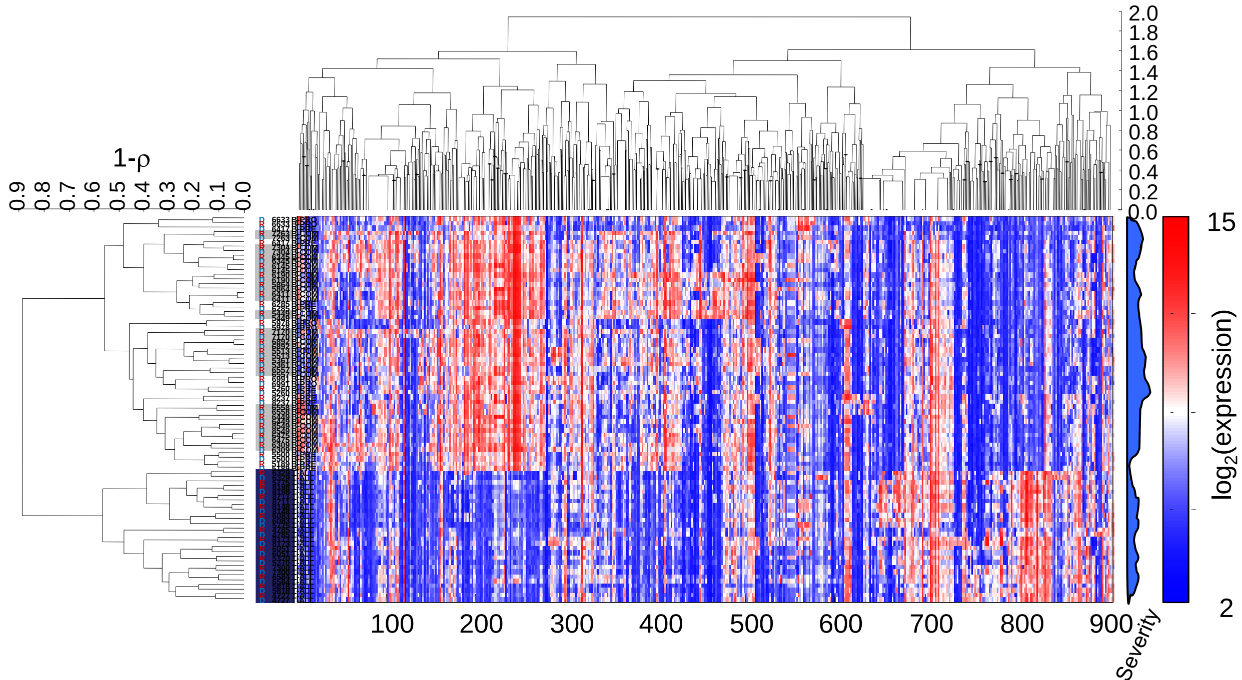
object

dataset

measurement **feature** **labels**

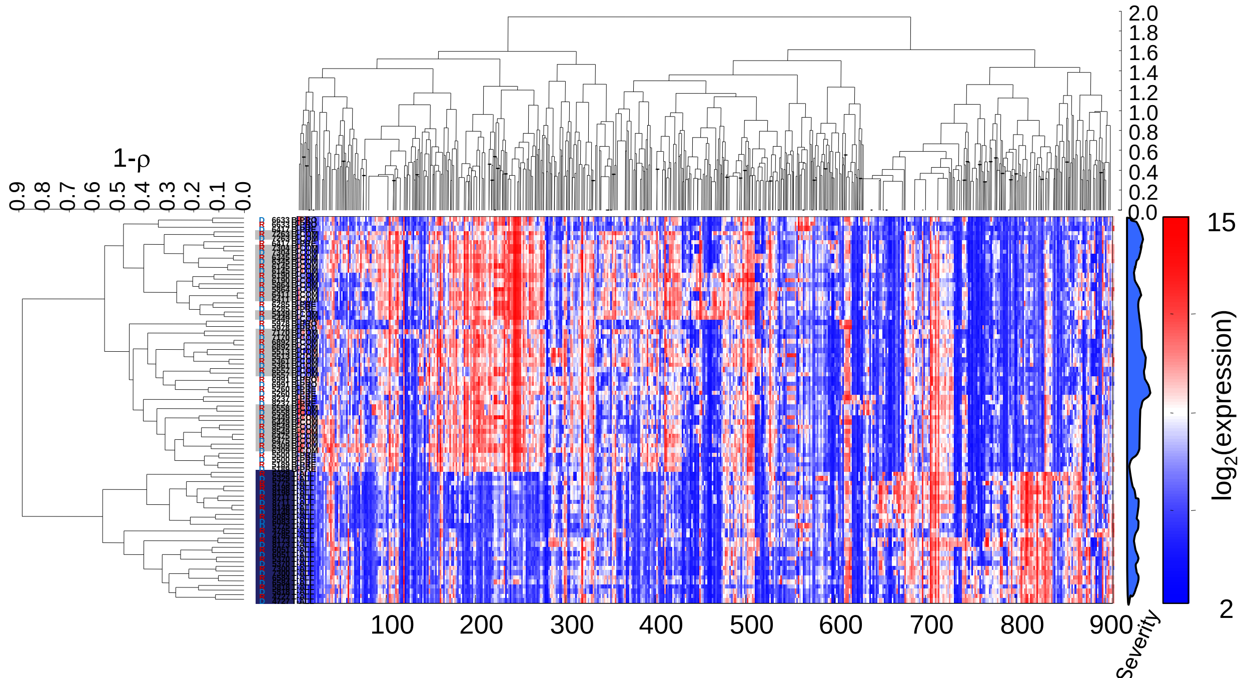
Datasets (2)

- What objects, labels/targets and features are depends on the problem...
- Gene expression-based diagnostics:
  - object: patient
  - feature: gene expression, copy number, mutational pattern, ….
  - label: relapse; regressor/dependent variable: survival time



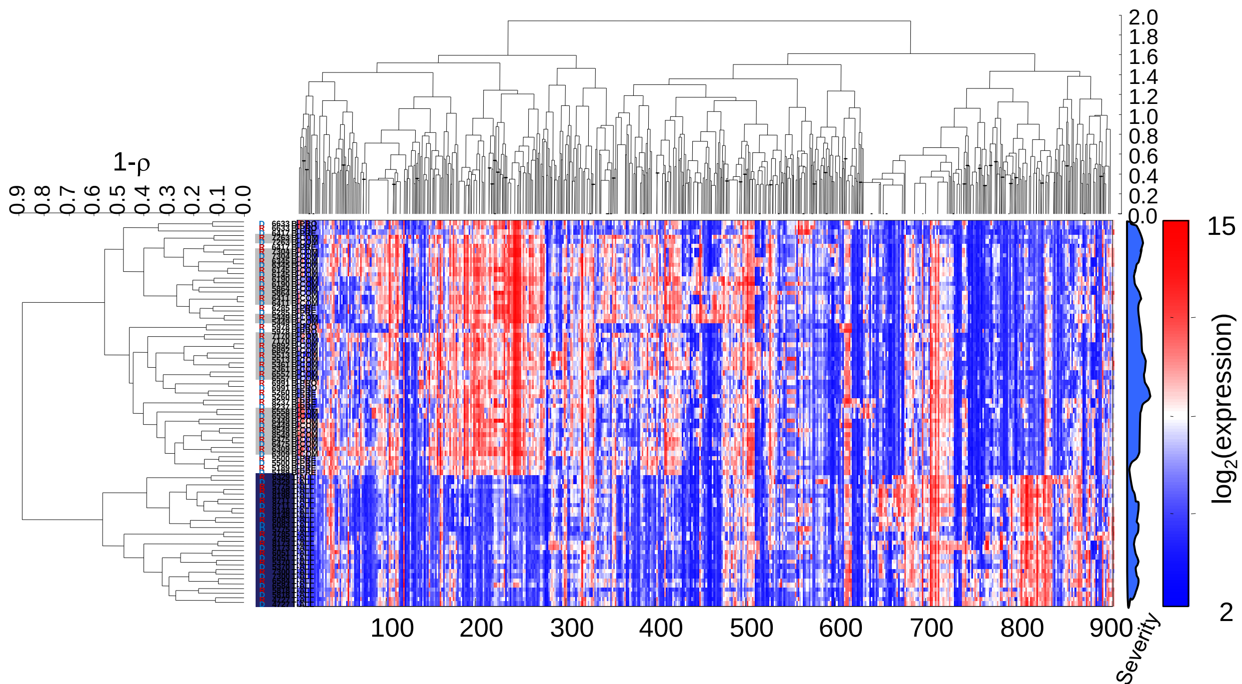
Datasets (2)

- What objects, labels/targets and features are depends on the problem...
- Protein-protein interactions:
  - object: protein PAIR
  - feature: gene expression correlation, difference in annotation, …
  - label: complex or not; regressor/dependent variable: binding strength



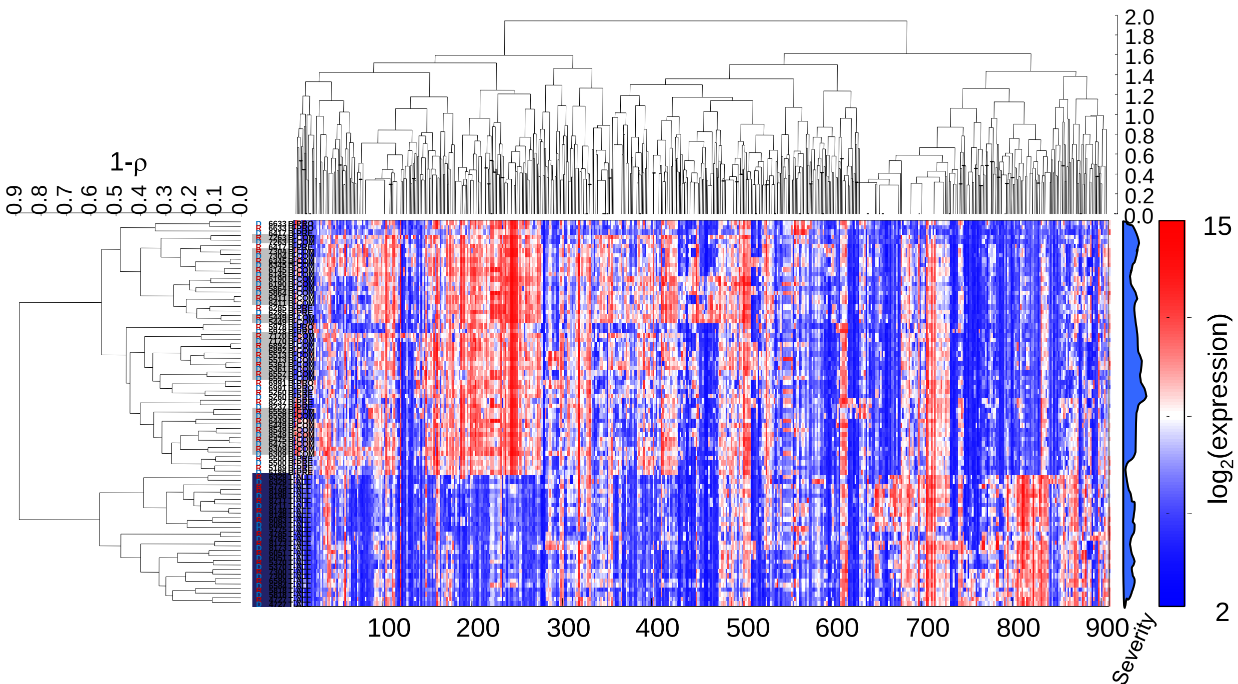
Datasets (2)

- What objects, labels/targets and features are depends on the problem...
- Gene prediction:
  - object: gene
  - feature: sequence (representation), conservation of sequence, …
  - label: gene or not; regressor/dependent variable: conservation



Datasets (2)

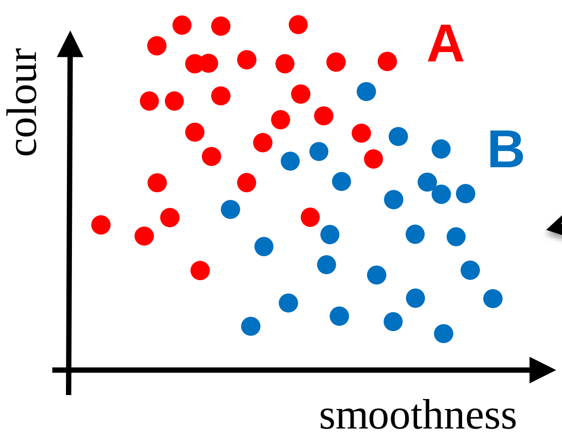
- What objects, labels/targets and features are depends on the problem…
- TFBS detection:
  - object: location on genome
  - feature: ChIP-seq, sequence features, distance to TSS …
  - label: TFBS or not; regressor/dependent variable: specificity

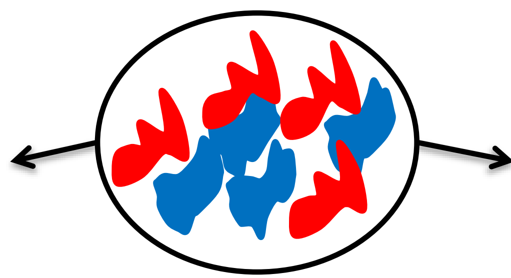


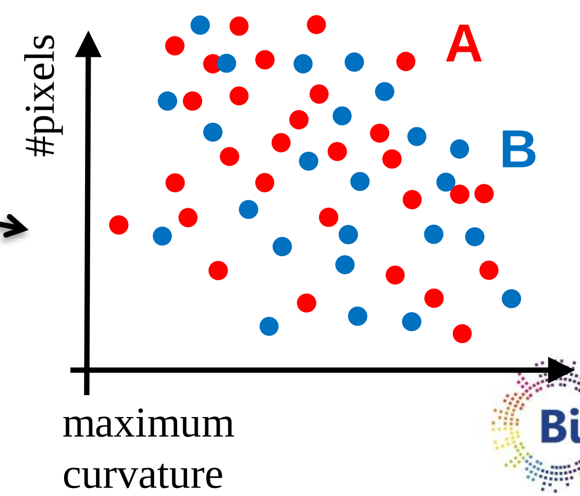
Measurements and features (3)

- Problems
  - simple
  - knowledge present
  - a few good features
  - almost separable classes (classification) or a linear relation (regression)

  - complex
  - lack of knowledge
  - many poor features
  - overlapping classes (classification) or highly non-linear relation (regression)







BioSB

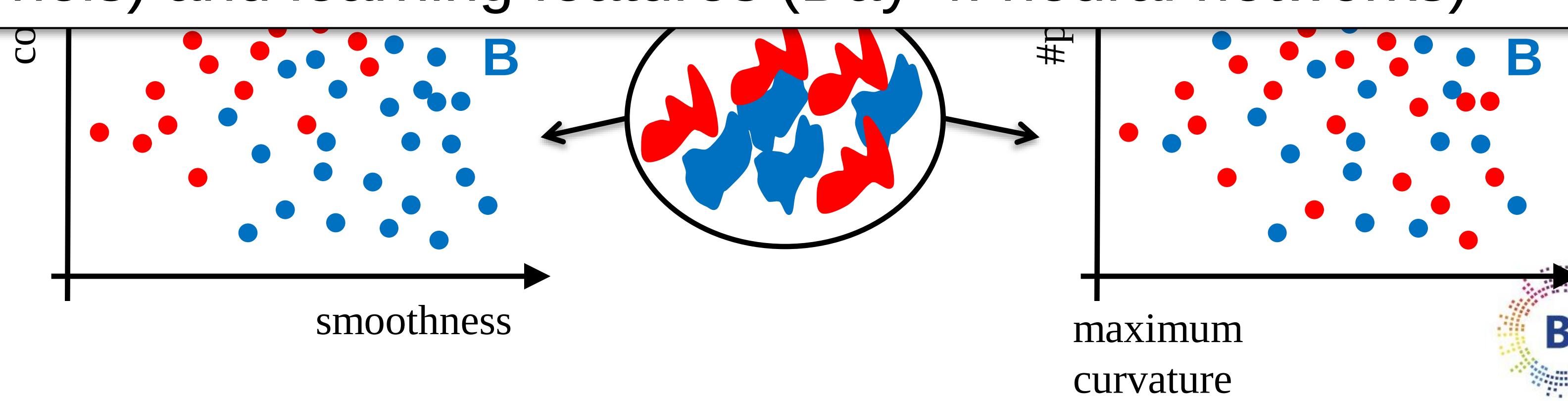
Measurements and features (3)

- Problems
  - simple
  - knowledge present
  - a few good features

- complex
- lack of knowledge
- many poor features

Features (object representations) are important!

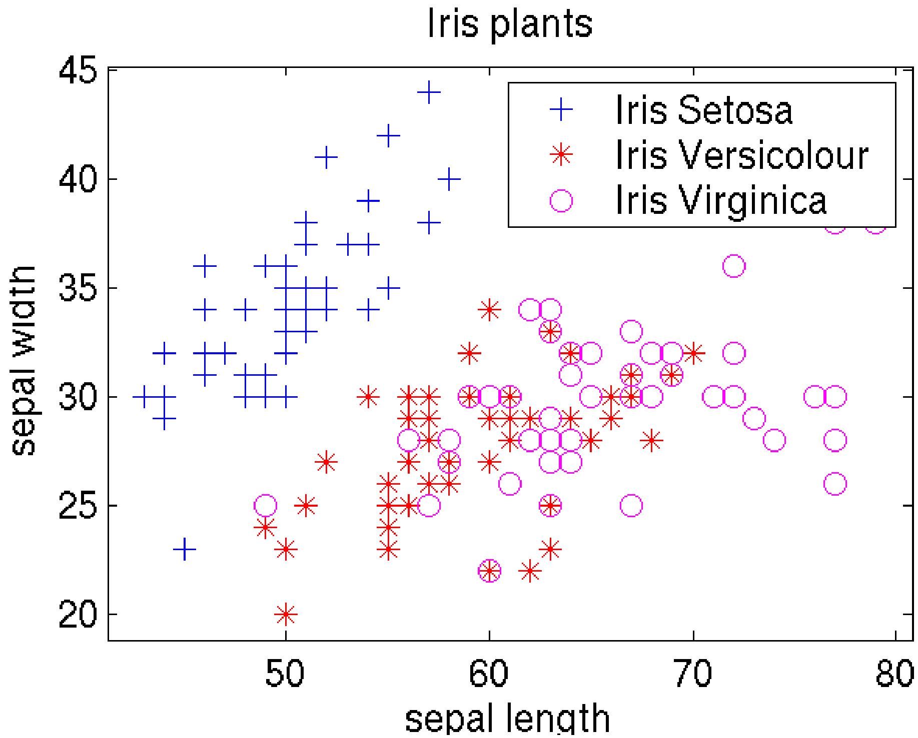
We don’t deal too much with which features are measured, although we will touch upon derived features (Day 4: kernels) and learning features (Day 4: neural networks)



Feature space

- We can interpret objects as vectors in a vector space

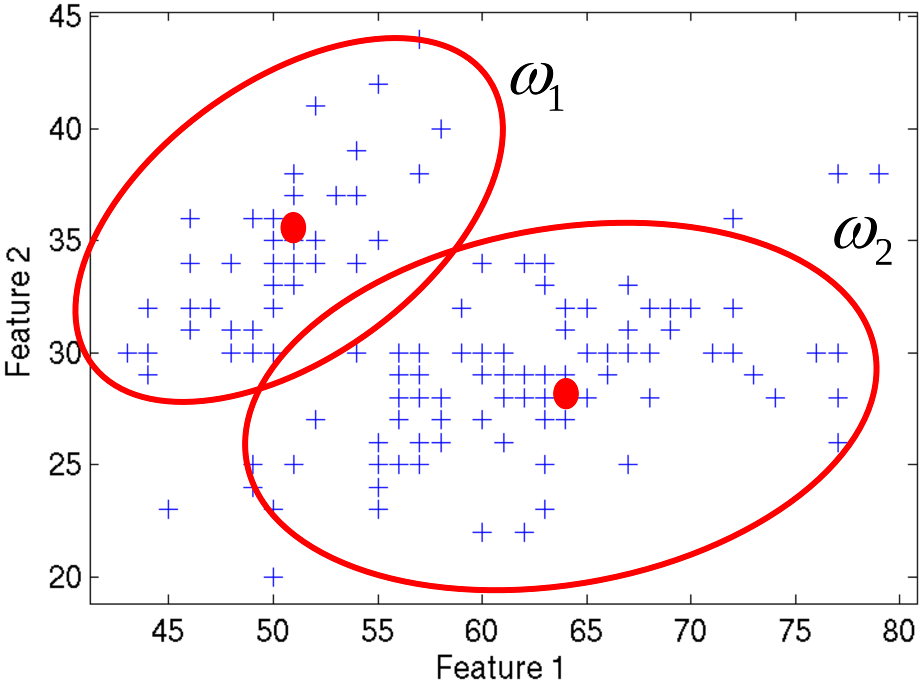
$$\mathbf{x} = [x_1, x_2, x_3, \ldots, x_p]^T$$



Iris flower dataset, introduced by **Ronald Fisher (famous statistician)** in 1936 as an example of discriminant analysis

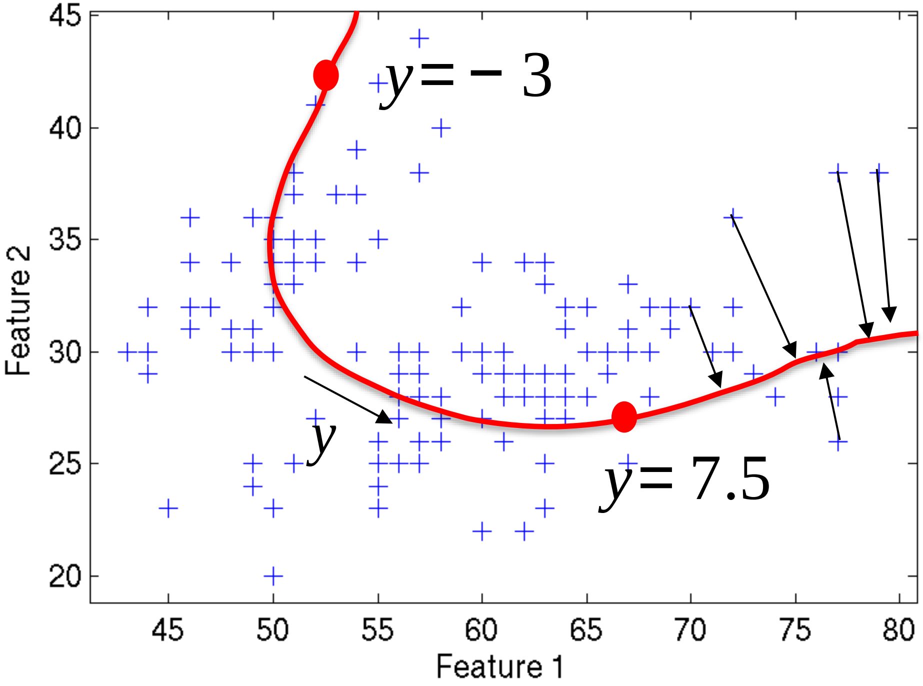
Clustering

- Given unlabeled data $\boldsymbol{x}$,
  find labels ω for natural groups in the data



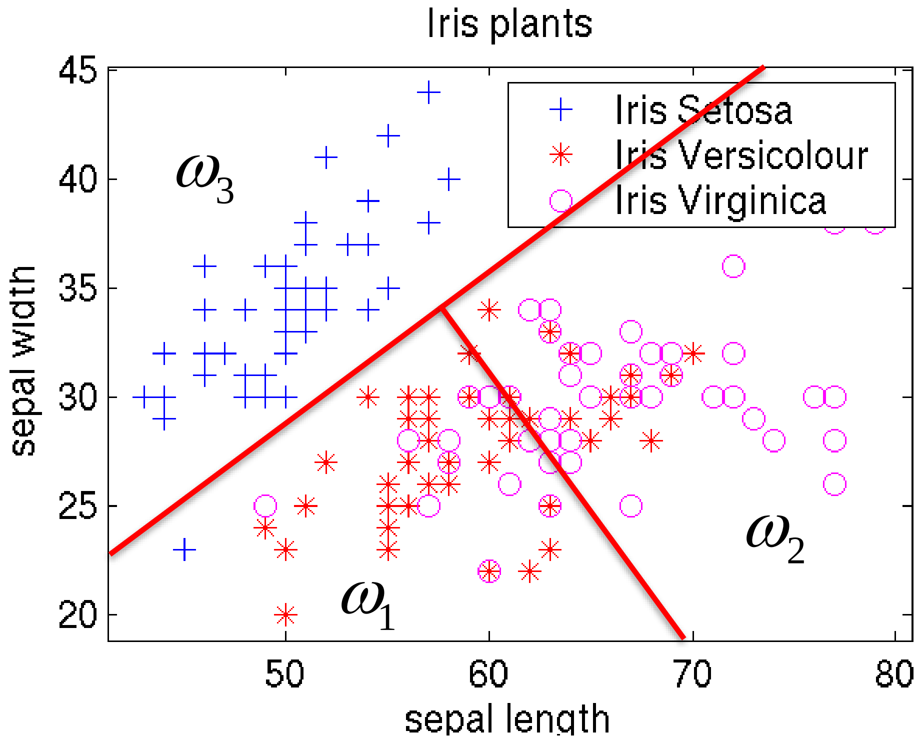
Dimensionality reduction

- Given unlabeled data $\boldsymbol{x}$,
  map it to a lower dimensional feature vector y



Classification

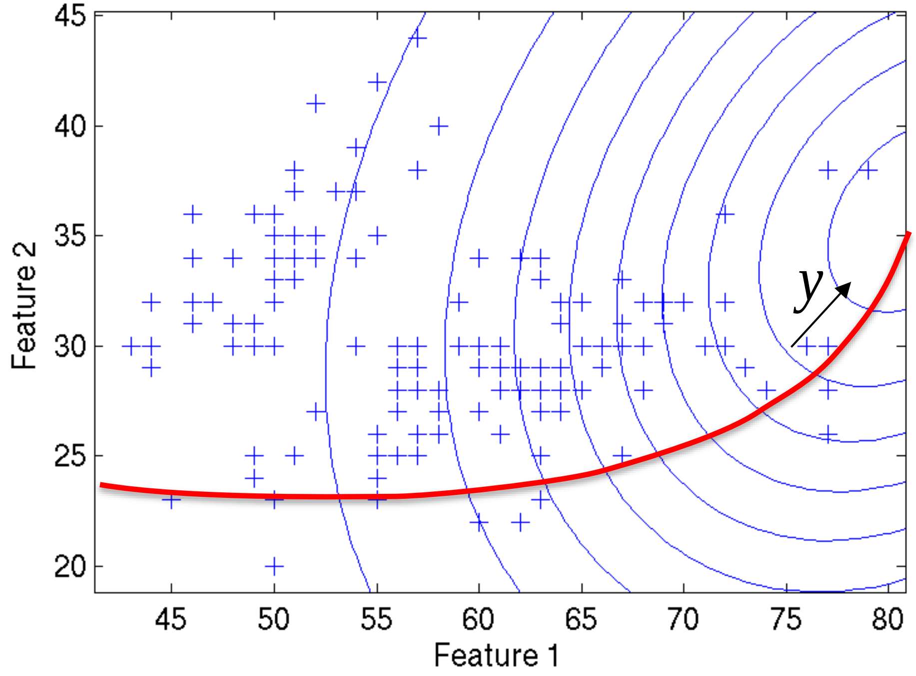
- Given labeled data $\boldsymbol{x}$,
  assign each point in feature space to a class ω_i
  (in effect partitioning the feature space)



BioSB

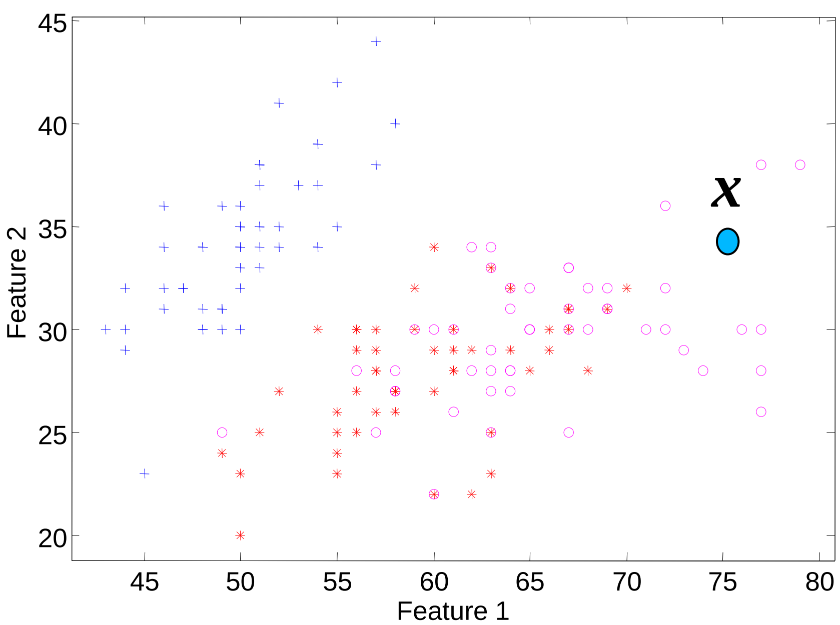
Regression

- Given labeled data $\boldsymbol{x}$,
  assign each point in feature space a real-valued output y



General model

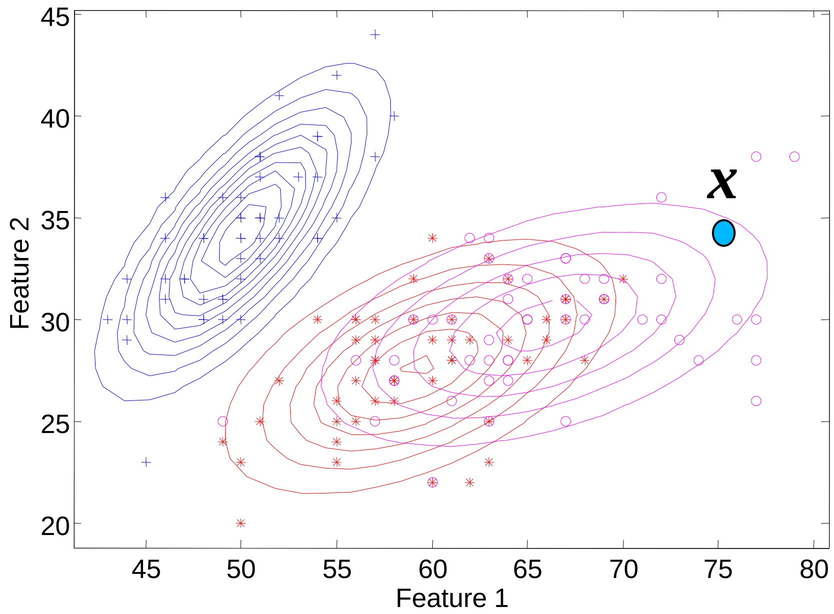
- Construct a model $f(\mathbf{x})$ that outputs ω or y
- This model should be fit to the data



$$f(\mathbf{x}) = \omega \text{ or } f(\mathbf{x}) = y$$

General model (2)

- Construct a model $f(\boldsymbol{x})$ that outputs ω or y
- This model should be fit to the data
- Ideally, we know $p(y \mid \boldsymbol{x})$ or $p(\omega \mid \boldsymbol{x})$ over the entire feature space



$$p(y \mid \boldsymbol{x})$$

or

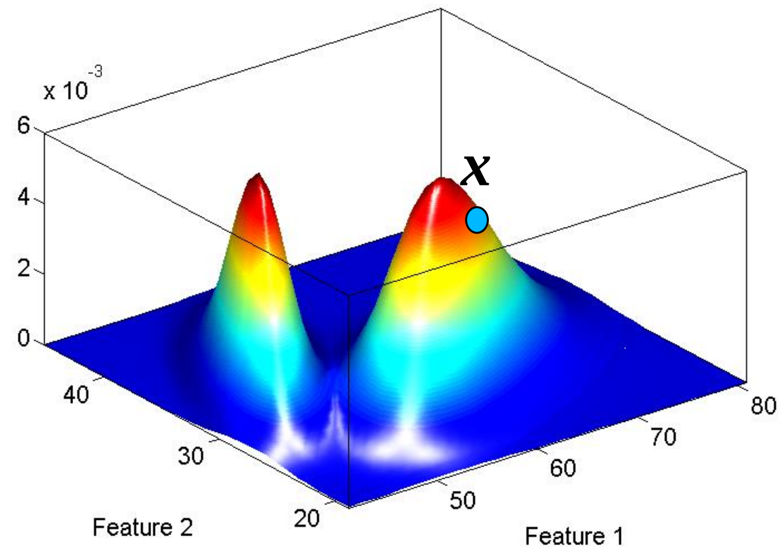
$$p(\omega \mid \boldsymbol{x})$$

$\Rightarrow$ $f(\boldsymbol{x}) = \omega$ or $f(\boldsymbol{x}) = y$

if we know the probability distributions, we can make the most informed decision

General model (3)

- Construct a model $f(\boldsymbol{x})$ that outputs ω or y
- This model should be fit to the data
- Ideally, we know $p(y \mid \boldsymbol{x})$ or $p(\omega \mid \boldsymbol{x})$ over the entire feature space



$p(y \mid \boldsymbol{x})$
or
$p(\omega \mid \boldsymbol{x})$

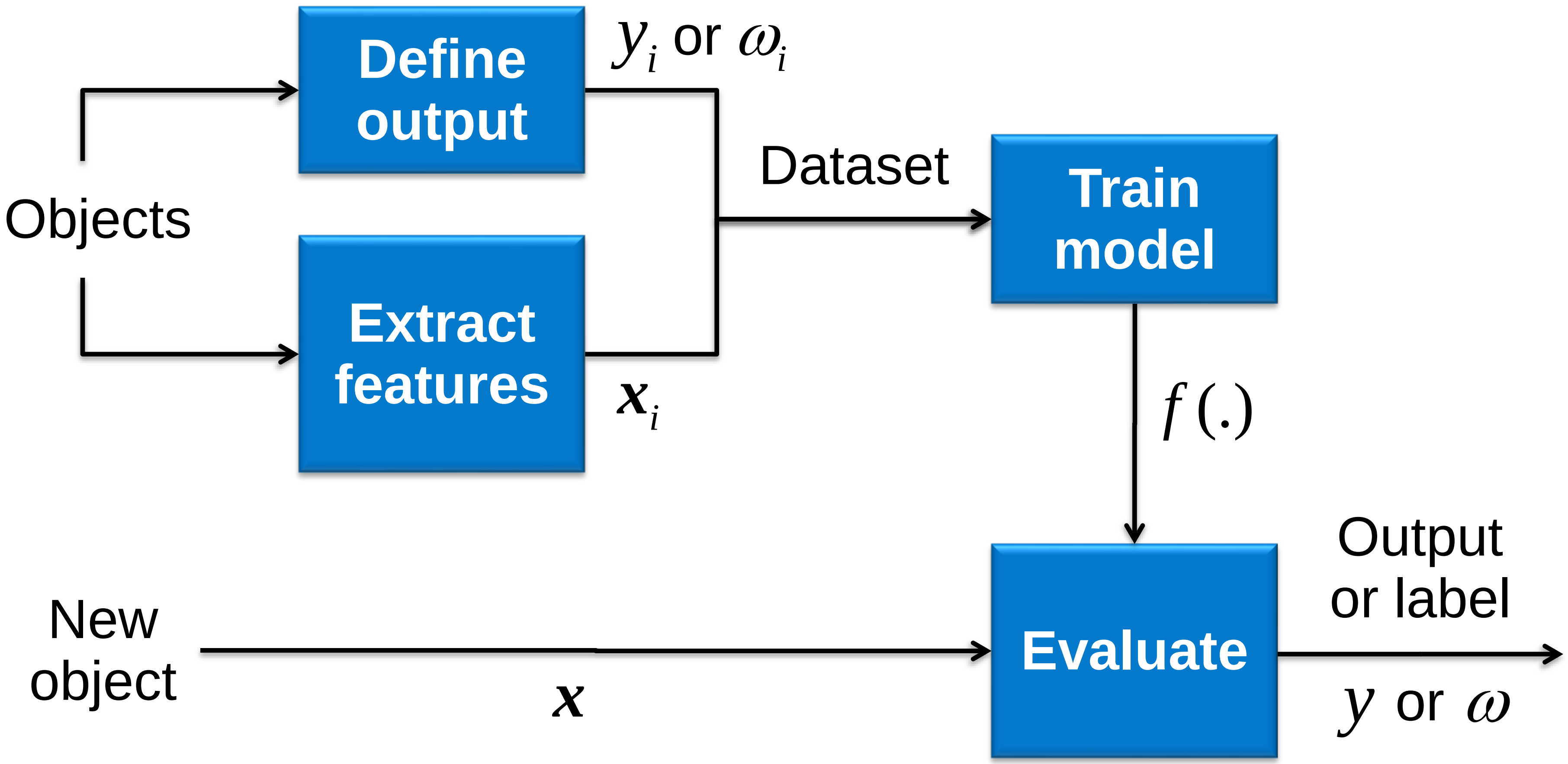
$f(\boldsymbol{x}) = \omega$ or $f(\boldsymbol{x}) = y$

if we know the probability distributions, we can make the most informed decision

General model (4)

- Clustering: find cluster labels ω given object $\mathbf{x}$
  fit model using dataset $\{\mathbf{x}_i\}$ $\quad p(\omega\,|\,\mathbf{x})$
- Dimensionality reduction: find mapping y given object $\mathbf{x}$
  fit model using dataset $\{\mathbf{x}_i\}$ $\quad p(y\,|\,\mathbf{x})$
- Classification: find class labels ω given object $\mathbf{x}$
  fit model using dataset $\{\mathbf{x}_i, \omega_i\}$ $\quad p(\omega\,|\,\mathbf{x})$
- Regression: find target y given object $\mathbf{x}$
  fit model using dataset $\{\mathbf{x}_i, y_i\}$ $\quad p(y\,|\,\mathbf{x})$

Machine learning pipeline



BioSB

Statistics

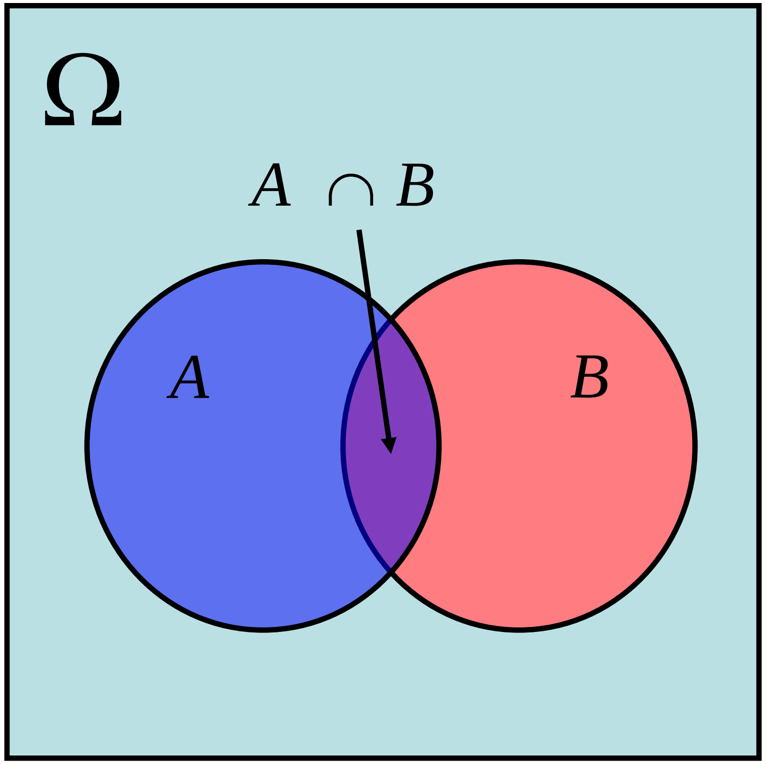
Required background

- The course is aimed at PhD students with a background in bioinformatics, systems biology, computer science or a related field, and life sciences. A working knowledge of basic statistics and linear algebra is assumed.

- Self-assessment; if you have problems, read the primers
- Now, a brief recap

Recall: probability

- Ω : all possible outcomes (sample space)
  e.g. the number of eyes on a dice: 1, 2, 3, 4, 5, 6
- $A \in \Omega$: event
  e.g. "throwing a 3"
- P : probability measure
  - $0 \leq P(A) \leq 1$
  - $P(\Omega) = 1$
  - $P(A \cup B) =$
    $P(A) + P(B) - P(A \cap B)$
  - E.g. $P(A) = 1/6$



Recall: probability (3)

- Subjective approach:
  "the probability of A is a number between 0 and 1
  indicating how likely people believe A to be true"

- Frequentist approach:
  "the probability of A is a number between 0 and 1
  indicating the average ratio of A being true in
  a large number of repeated experiments"

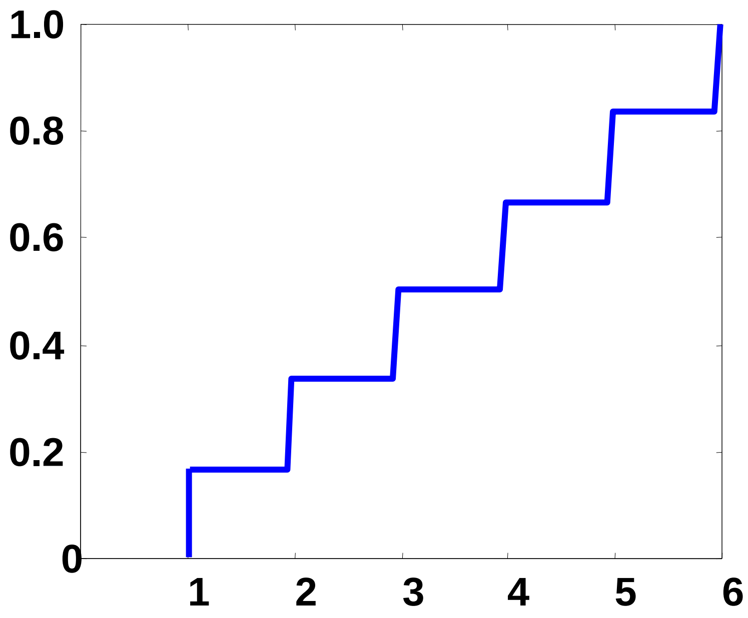
- Is really a philosophical debate...
  the "right" approach depends on the problem
  and the data available

problems (can) arise in interpretation: what does it mean?

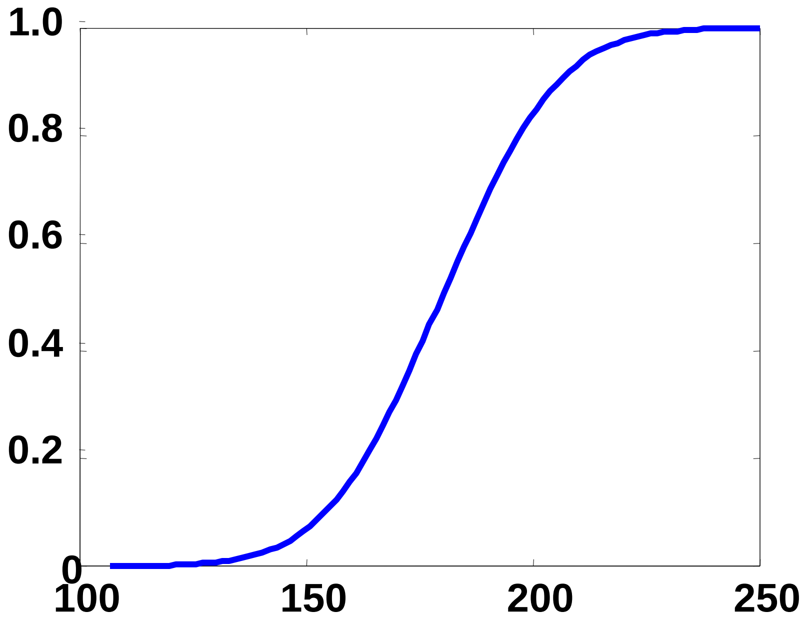
BioSB

Recall: CDFs

- Cumulative distribution function
- $P_X(x) = F(x)$: probability that $X \leq x$, $\Re \rightarrow [0,1]$



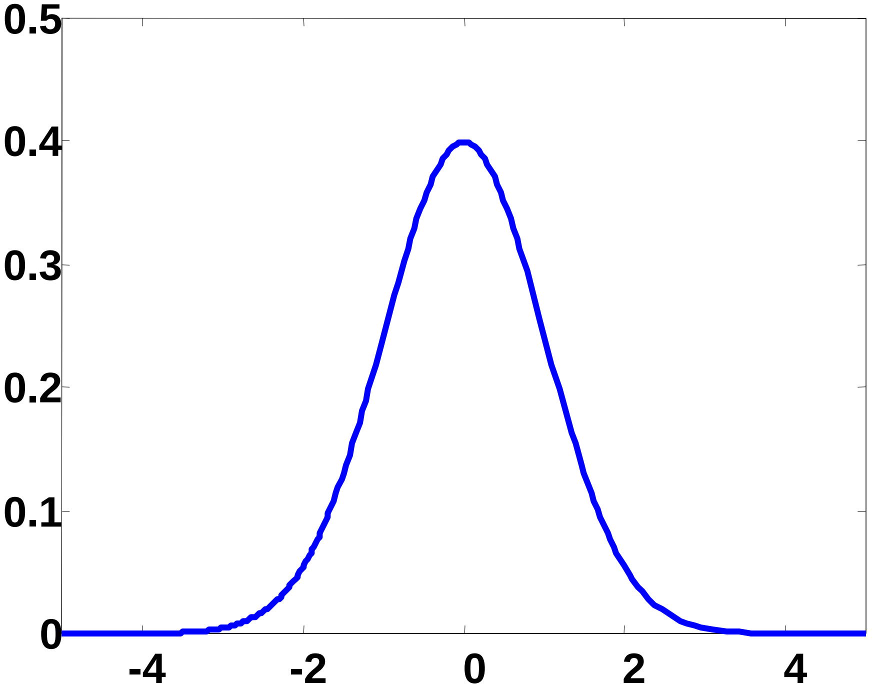
e.g. 10,000 dice throws



10,000 body lengths

Recall: PDFs

- $p(x) = \dfrac{dP(x)}{dx}$: probability density function
  - $p(x) \geq 0$
  - $\int_{-\infty}^{\infty} p(x)dx = 1$
  - $\int_a^b p(x)dx = P(a \leq x \leq b)$



- **$p(x)$ is not the probability of X being x !**

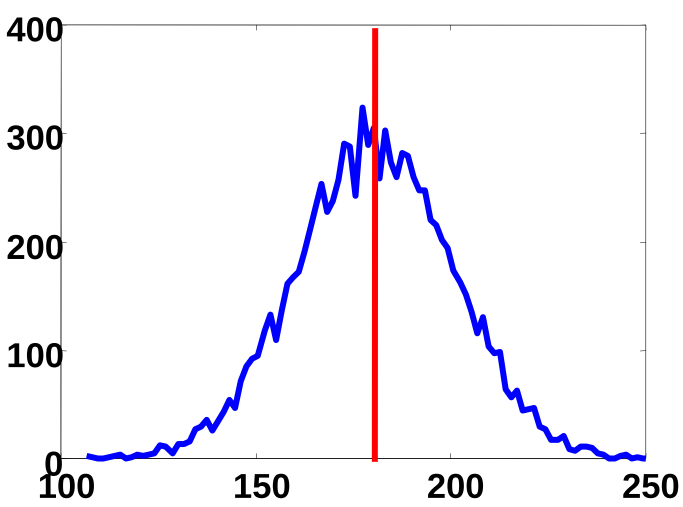
Recall: expectation

- Expectation: mean of distribution,

$$\mu = \mathrm{E}[X] = \int_{-\infty}^{\infty} x \; p(x) \; dx$$

- Note: expectations are over entire distributions; on data sets $\{x\}$ we can only estimate the mean,

$$m = \hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i$$



- $\mathrm{E}[c] = c$
- $\mathrm{E}\,[aX + bY] = a\,\mathrm{E}[X] + b\,\mathrm{E}[Y]$

Important to realize that estimates are always based on a finite dataset!
m is an estimate(!) of μ; that is why there is a hat!

BioSB

Recall: variance

- Variance: average deviation from expected value,

$$\sigma^2 = \mathrm{var}(X) = \int_{-\infty}^{\infty} (x-\mu)^2 \, p(x) \; dx$$

or

$$\sigma^2 = \mathrm{E}[(X - \mathrm{E}(X))^2] = \mathrm{E}[X^2] - (\mathrm{E}[X])^2$$

- σ is called the standard deviation

- $\mathrm{var}(X) \geq 0$
- $\mathrm{var}(c) = 0$
- $\mathrm{var}(aX) = a^2 \, \mathrm{var}(X)$

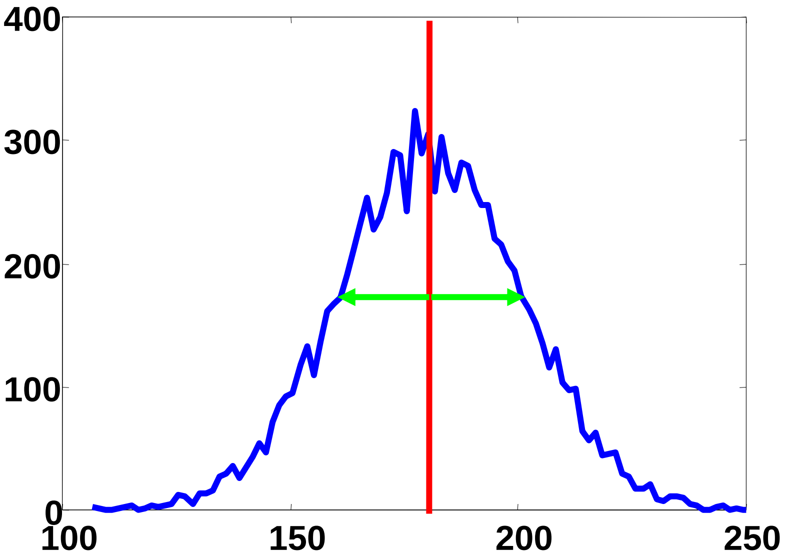
Recall: variance (2)

- Again, on data sets $\{x\}$ we can only estimate the variance:

$$s^2 = \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2$$

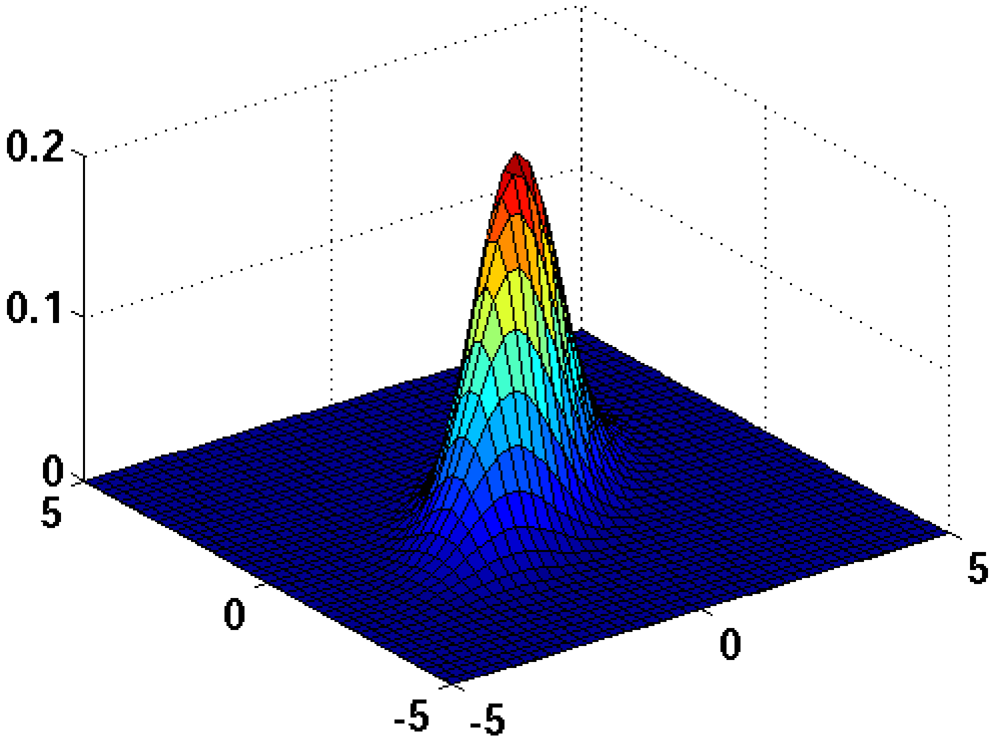
- Usually, this unbiased estimator is used:

$$s^2 = \hat{\sigma}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu})^2$$



Recall: joint distributions

- For $p > 1$ measurements $x = (x_1 , ..., x_p)$, joint distributions & densities:



Recall: covariance

- Covariance: measure of how two random variables vary together,

$$\mathrm{cov}(X,Y) = \mathrm{E}[(X-\mathrm{E}(X))(Y-E(Y))]$$
$$= \mathrm{E}[XY]-\mathrm{E}[X]\mathrm{E}[Y]$$

- Correlation: normalised covariance,

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}} \in [-1,1]$$

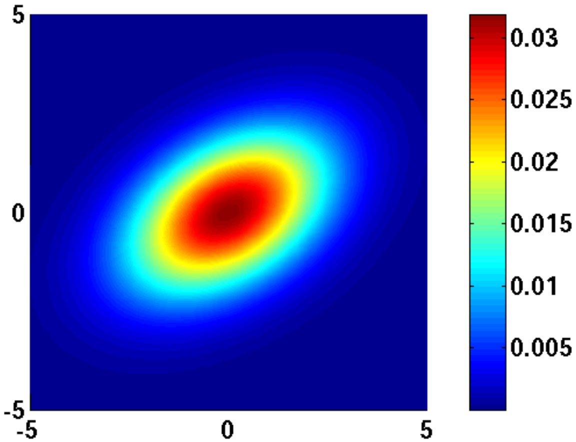
- $\mathrm{cov}(X,Y) = 0$: X and Y are uncorrelated

Recall: covariance (2)

- For a set of random variables $X_1 \dots X_p$,
  we can calculate a covariance matrix,

$$\Sigma = \begin{bmatrix} \text{cov}(X_1, X_1) & \text{cov}(X_1, X_2) & \dots & \text{cov}(X_1, X_p) \\ \text{cov}(X_2, X_1) & \dots & \dots & \text{cov}(X_2, X_p) \\ \dots & \dots & \dots & \dots \\ \text{cov}(X_p, X_1) & \text{cov}(X_p, X_2) & \dots & \text{cov}(X_p, X_p) \end{bmatrix}$$

e.g.

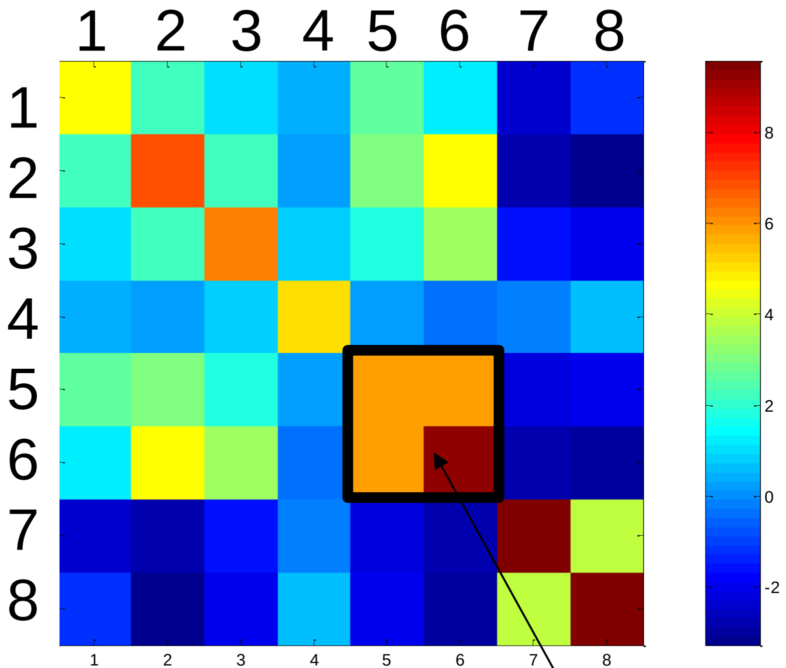


$$\Sigma = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

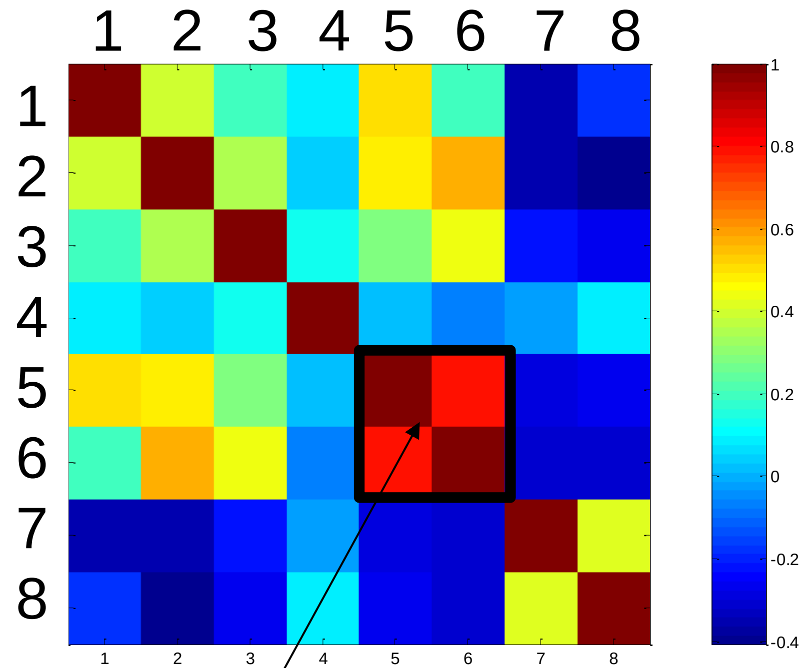
Pairwise covariance of all features!

Recall: covariance (3)

- Example: IMOX data (images of handwritten digits 1:8)



`imagesc(cov(+a))`

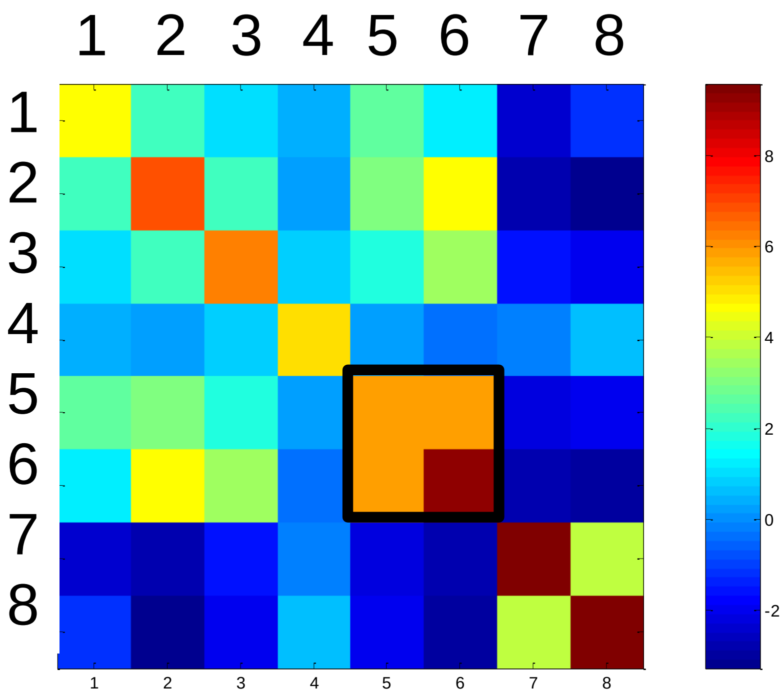


`imagesc(corrcoef(+a))`

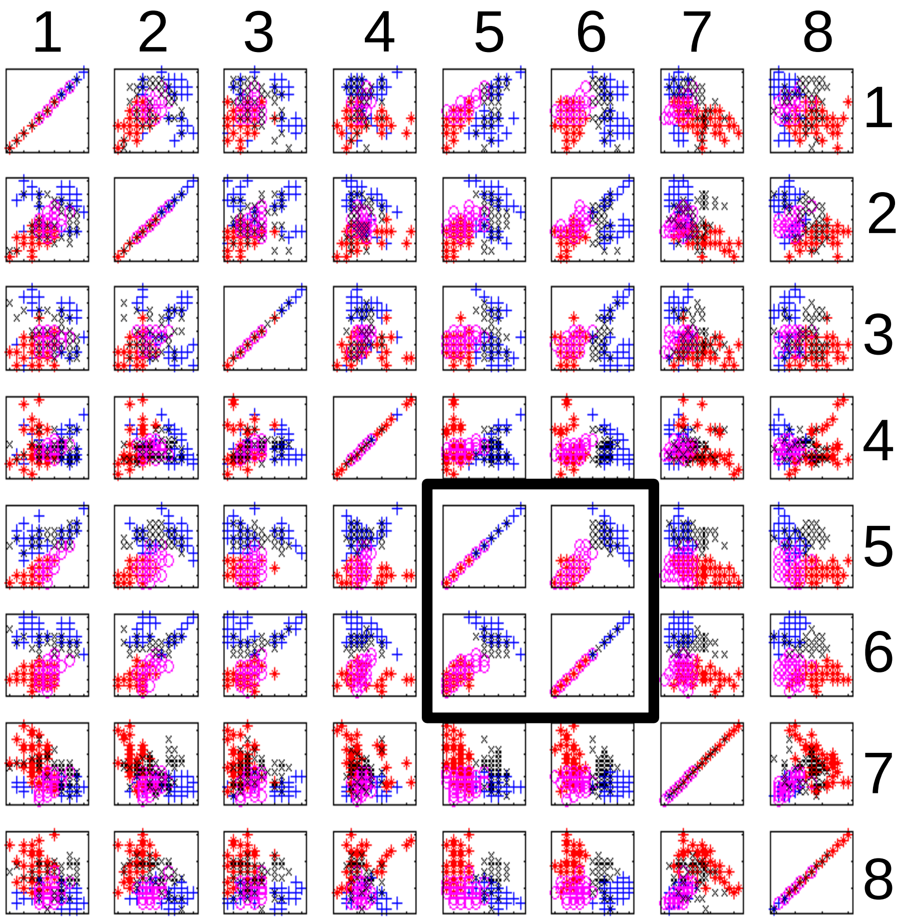
characters 5/6 are alike

Recall: covariance (4)

- Example: IMOX data



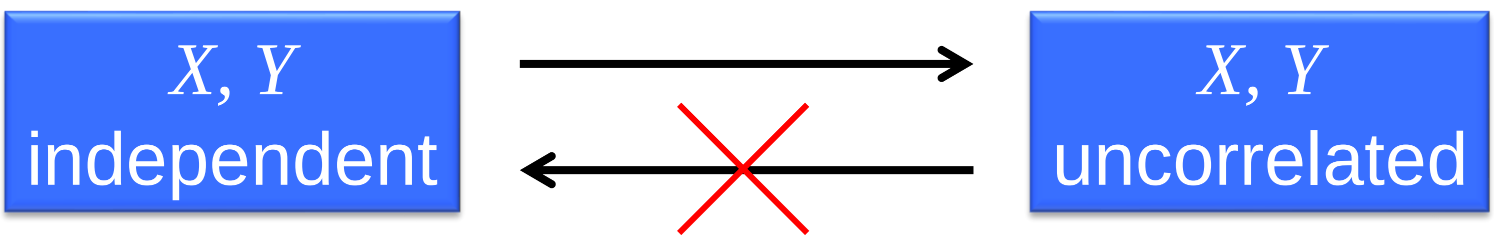
`imagesc(cov(+a))`



`scatterd(a,'gridded')`

Recall: independence

- Important concept: often needed as assumption!
- Two events A and B are independent iff
  $P(A \cap B) = P(A)\,P(B)$
- Two random variables X and Y are independent iff
  $p(x,y) = p(x)\,p(y)$



- Uncorrelated: "there's no *linear* dependence"
  Independent: "there's no dependence at all"

Recall: Bayes' theorem

- Conditional probability of A given B,

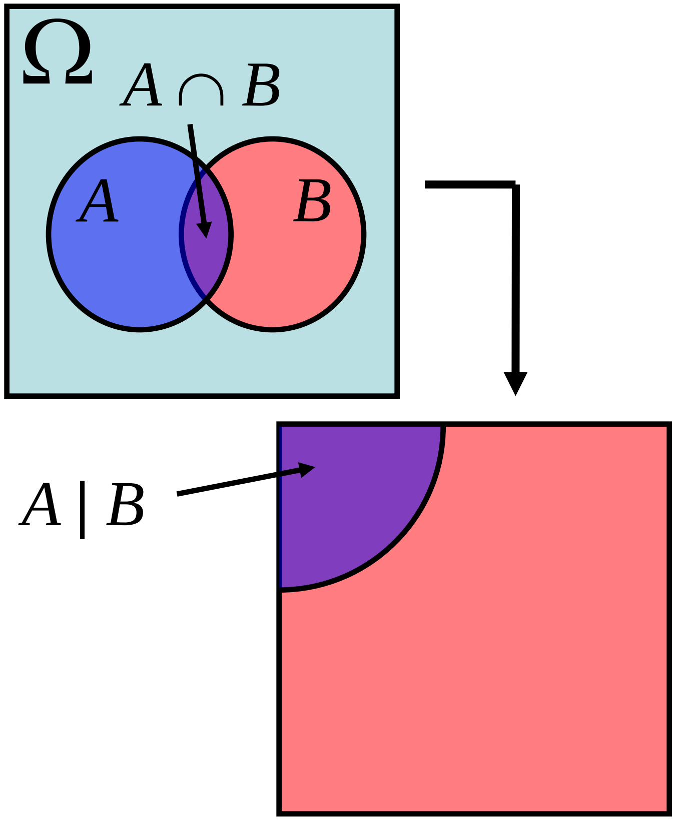
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- As a consequence,

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

- Bayes' theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



Bayes' theorem (2)

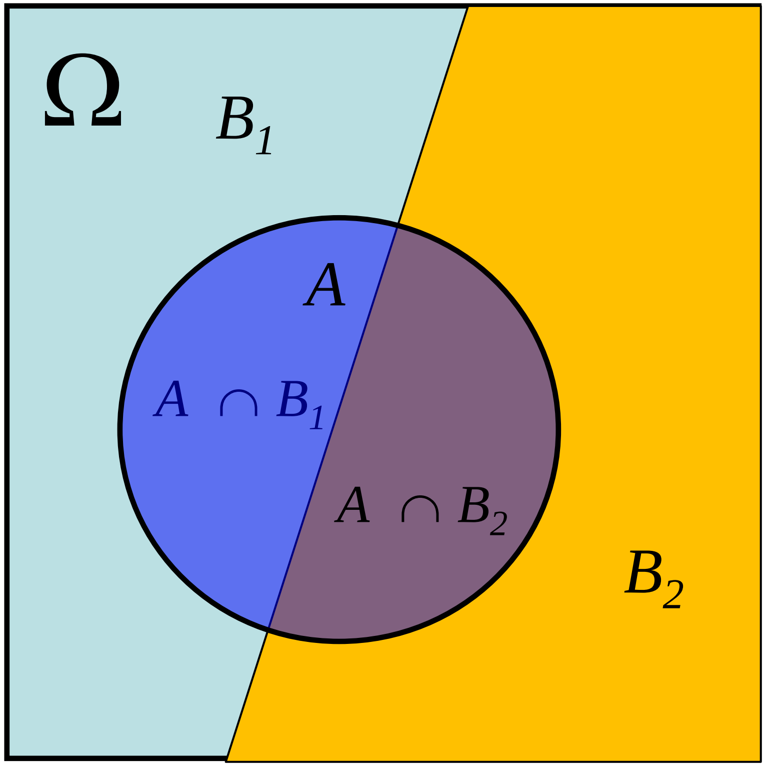
- Bayes' theorem is very useful, but controversial:
  - reverses causality
  - introduces subjective (prior) probabilities

$$P(cause \mid effect) = \frac{P(effect \mid cause)P(cause)}{P(effect)}$$

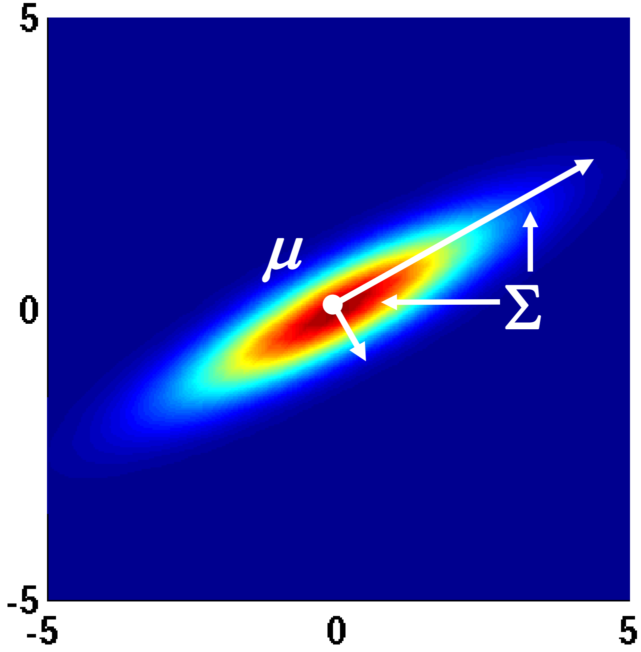
- **… but the cornerstone of pattern recognition and machine learning**
  - $P(disease|temperature) = \frac{P(temperature|disease)P(disease)}{P(temperature)}$
  - What is P (disease)? How to measure / know?

Recall: total probability

- Total probability:
- $P(A) = \sum_{\forall B_i} P(A \cap B_i)$
- $P(A) = \sum_{\forall B_i} P(A|B_i)P(B_i)$



Multivariate Gaussian distribution



$$\Sigma = \begin{bmatrix} 3 & 1\tfrac{1}{2} \\ 1\tfrac{1}{2} & 2 \end{bmatrix}$$

- p - dimensional density:

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi^{p}\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\mu}$: mean

$\boldsymbol{\Sigma}$: covariance matrix

Special properties

- The Gaussian distribution is a special case:

X, Y independent $\rightleftarrows$ X, Y uncorrelated

- Proof: if uncorrelated, **Σ** is diagonal (σ_1 ... σ_p)

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi^p \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) \right)$$

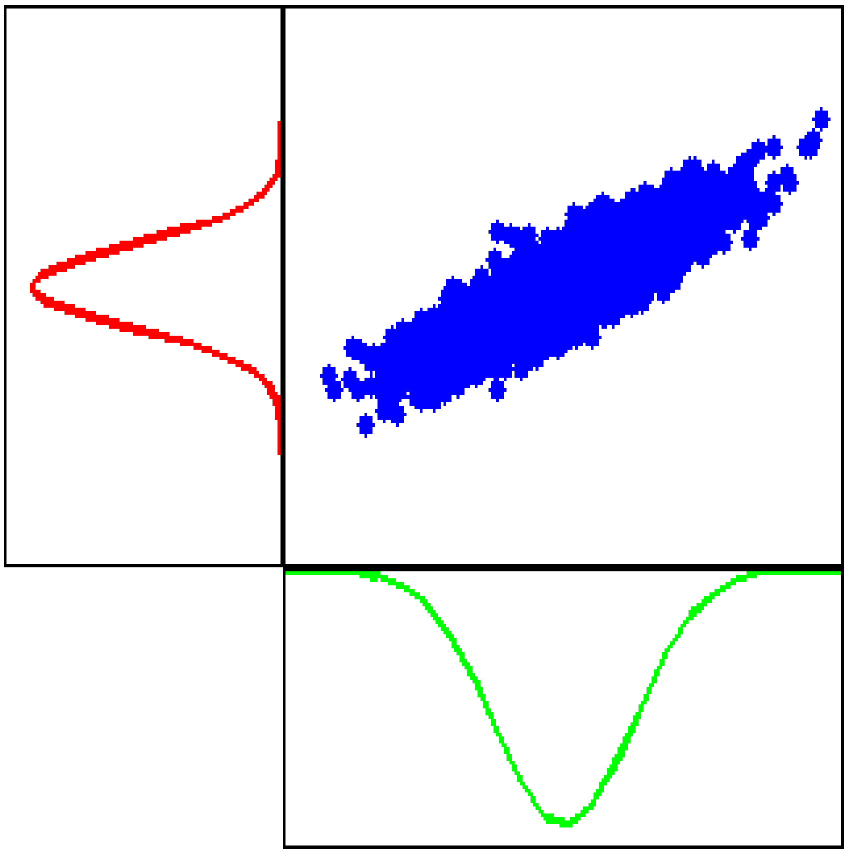
$$= \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2}(x_1-\mu_1)^{\mathrm{T}} \sigma_1^{-2} (x_1-\mu_1) \right) \times \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2}(x_2-\mu_2)^{\mathrm{T}} \sigma_2^{-2} (x_2-\mu_2) \right)$$

$$\times \ldots \times \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{1}{2}(x_p-\mu_p)^{\mathrm{T}} \sigma_p^{-2} (x_p-\mu_p) \right) = p(x_1)p(x_2)\ldots p(x_p)$$

BioSB

Special properties (2)

- Any projection of a high-dimensional Gaussian is itself again Gaussian

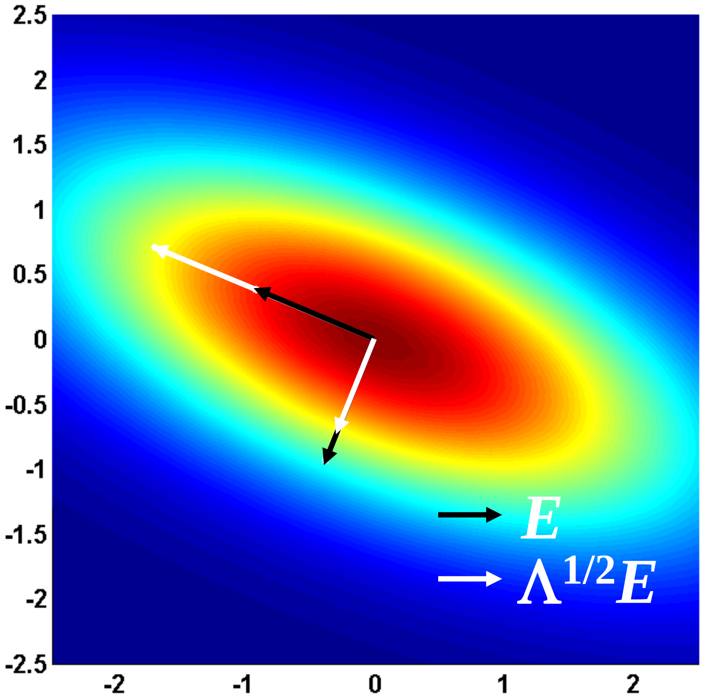


Sphering

- Eigenanalysis on a $p \times p$ covariance matrix $\mathbf{\Sigma}$:
  solve for $i = 1, ..., p$
  1. $\det(\mathbf{\Sigma} - \lambda_i \mathbf{I}) = 0$
  2. $(\mathbf{\Sigma} - \lambda_i \mathbf{I})\, \boldsymbol{e}_i = 0$
- $\mathbf{\Sigma} = \mathbf{E}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{E}$
- The $\boldsymbol{e}_i$ are the eigenvectors,
  stored as the columns of matrix $\mathbf{E}$;
  they correspond to the main axes of the Gaussian
- The λ_i are the eigenvalues,
  stored on the diagonal of matrix Λ;
  they correspond to the lengths of the main axes

Sphering (2)

- Covariance matrix determines shape of density

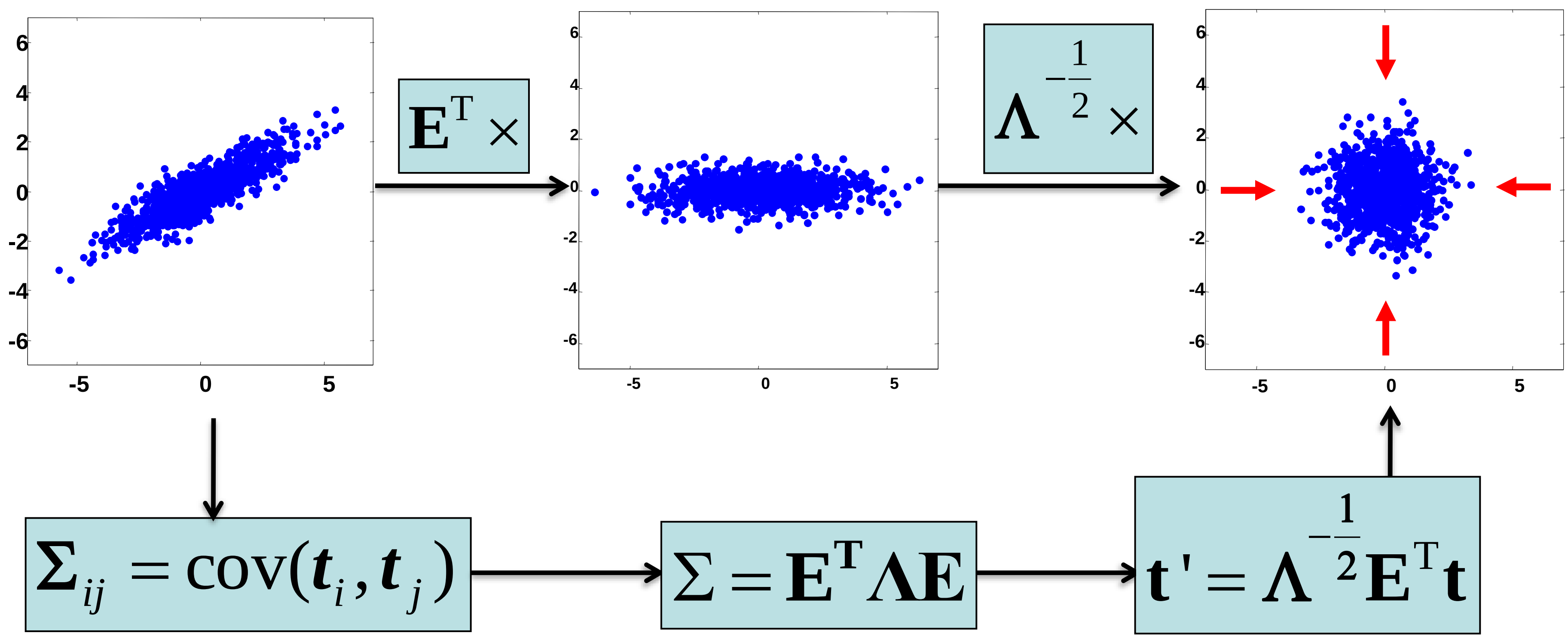


- Eigenvectors correspond to main axes of Gaussian, e.g.

$$\boldsymbol{\Sigma} = \begin{bmatrix} 3 & -1 \\ -1 & 1 \end{bmatrix} \longrightarrow \mathbf{E} = \begin{bmatrix} -0.92 & -0.38 \\ 0.38 & -0.92 \end{bmatrix} \quad \boldsymbol{\Lambda} = \begin{bmatrix} 3.41 & 0 \\ 0 & 0.59 \end{bmatrix}$$

Sphering (3)

- Eigenanalysis of covariance matrix can be used to “sphere” or “whiten” data:



- After sphering, $\mathbf{\Sigma}_{ij} = \mathbf{I}$

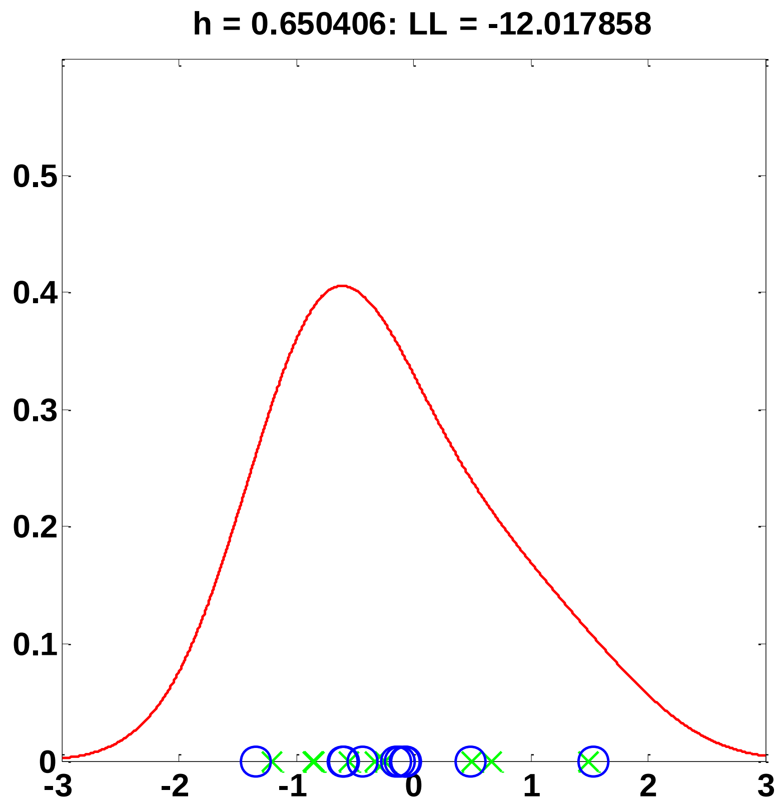
BioSB

Cross-validation

Cross-validation

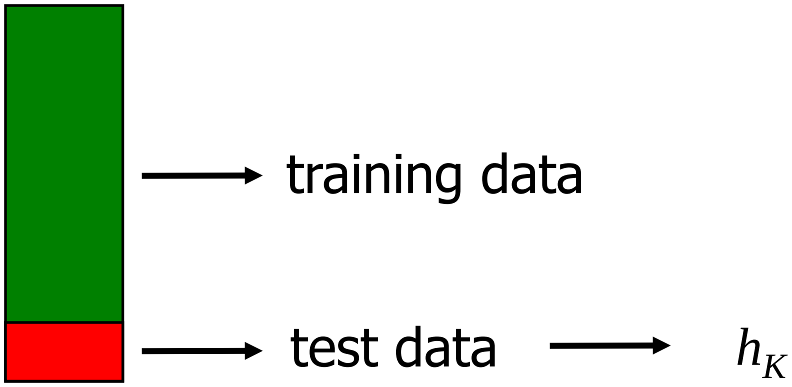
- Solution:
  - Split data into *training set* and *validation set*
  - Optimise h w.r.t. likelihood of validation set, given Parzen model trained on training set

- Problems:
  - Uses a lot of valuable data
  - Sensitive to split of data



Cross-validation (2)

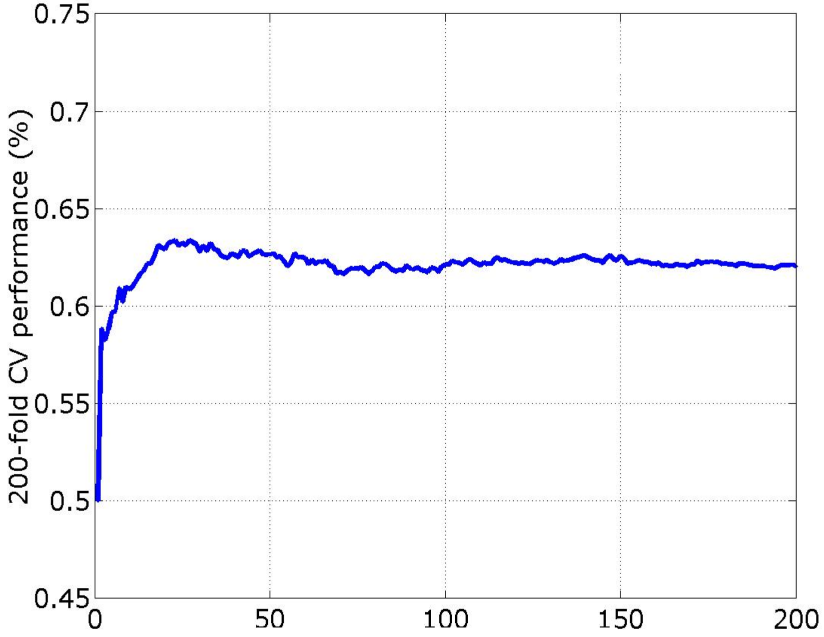
- Better solution: *K*-fold crossvalidation
  - Split data into K parts ($K = n$: leave-one-out)
  - Repeat K times:
    - Find h using $(K - 1)$ parts for training and 1 part for testing
  - Use average of h's as kernel width



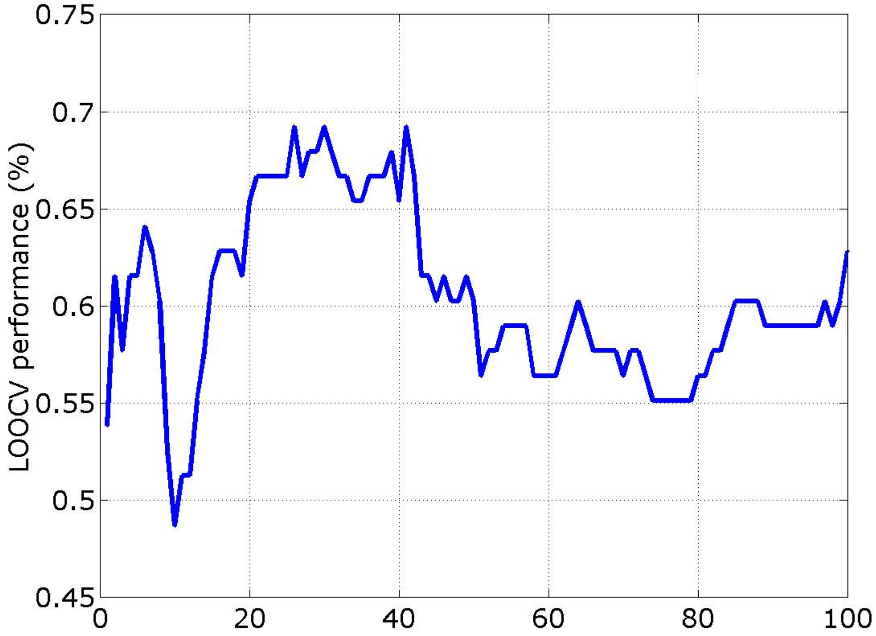
(will return)

Cross-validation (3)

- (Prefer) *K*-fold cross-validation over leave-one-out
  - Smoother (less variance) and more biased (conservative)



nr of features



nr of features

Bootstrap

- Alternative to cross-validation:
  - Repeat K times:
    - Draw n objects from the dataset, **with replacement** (some objects will be selected more than once)
    - Test using objects that were not selected

- Cross-validation and bootstrap estimates are *biased*
  - They are conservative (i.e. too pessimistic) because they do not use all data available

You want to get an estimate when you fit on complete/all data.
CV/Bootstrap are thus biased wrt fitting on complete data!

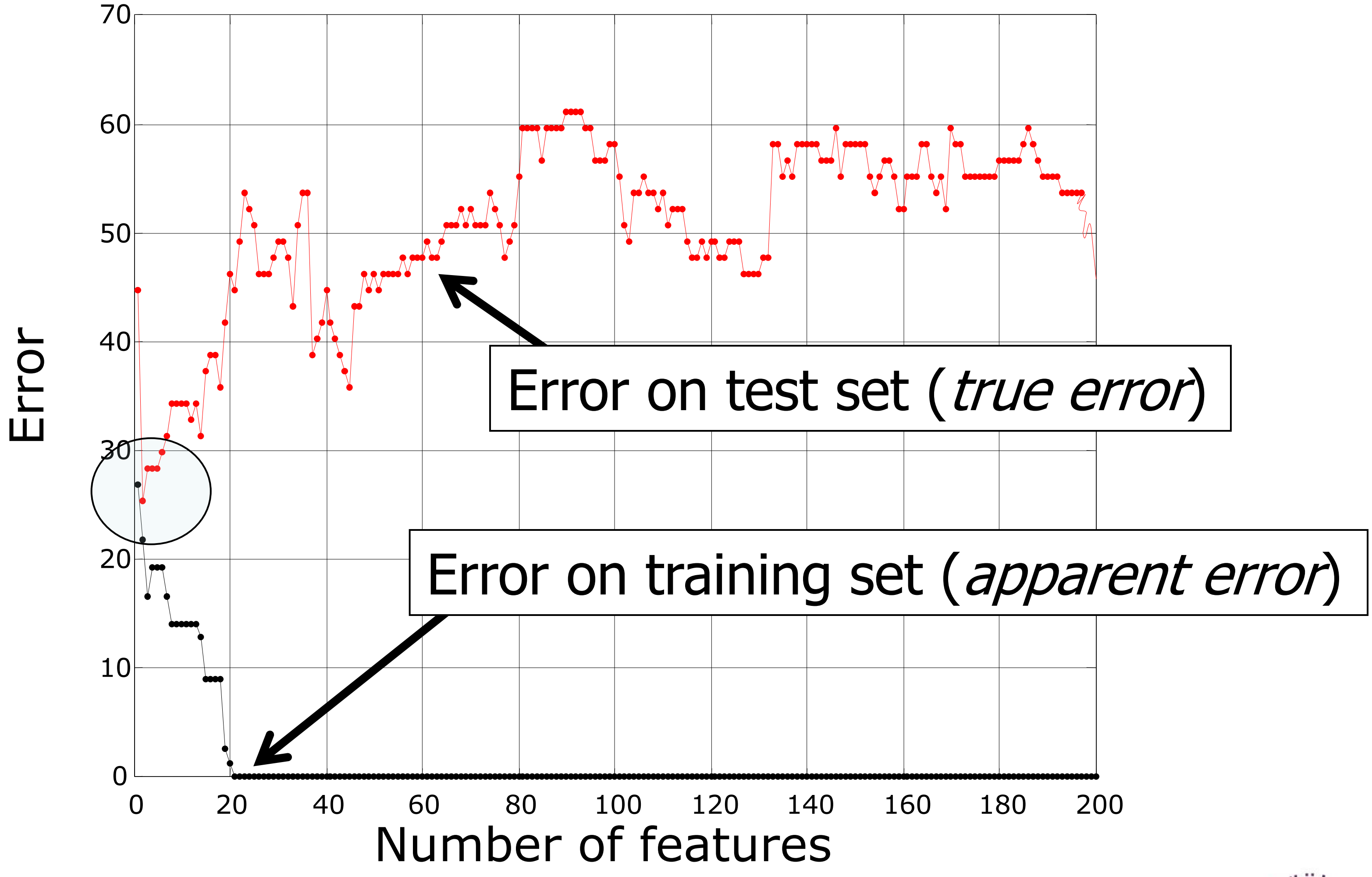
BioSB

Training, test and validation sets

- Terminology:
  - A *training set* is used to estimate parameters
  - An optional *validation set* is used to optimize parameter settings, e.g. by calculating classifier error on this set
  - **A *test set* is only used to judge performance of the entire classifier (only used once!)**

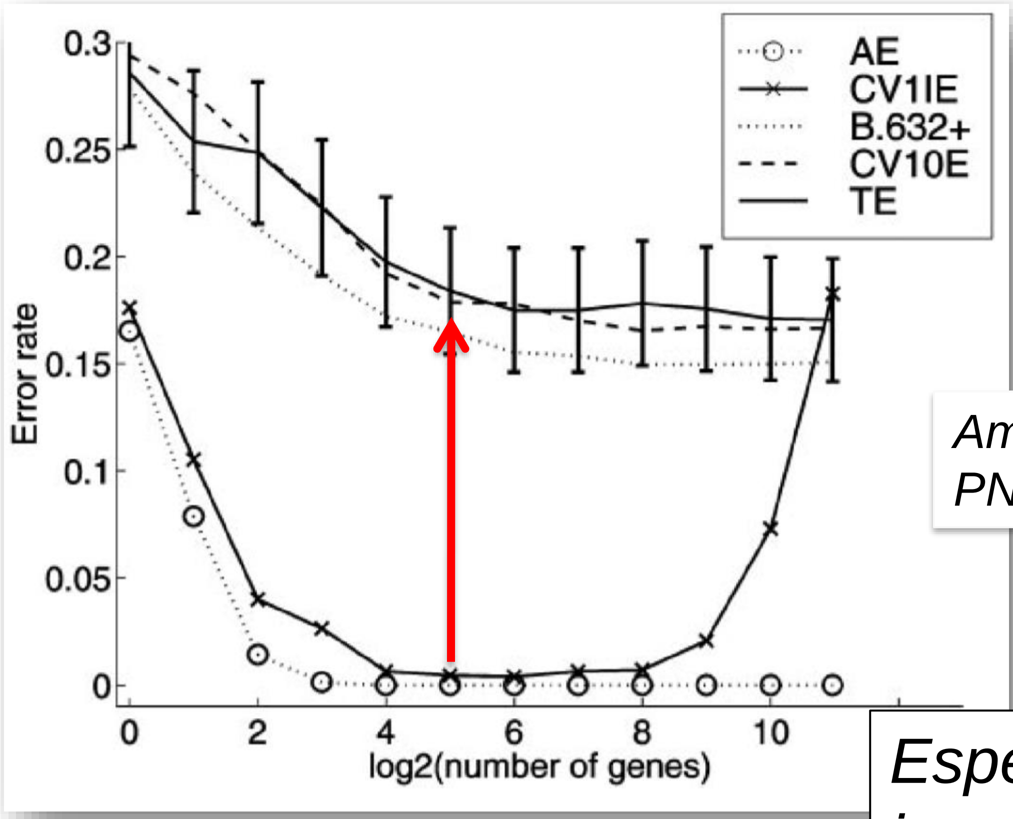
- Error estimates:
  - On training set: *apparent error*
  - On test set: *true error*

Training, test and validation sets (2)



Training, test and validation sets (3)

- The test set should *never* be used to set any parameters! This leads to biased estimates of performance -- in practice we may do much worse than we predict

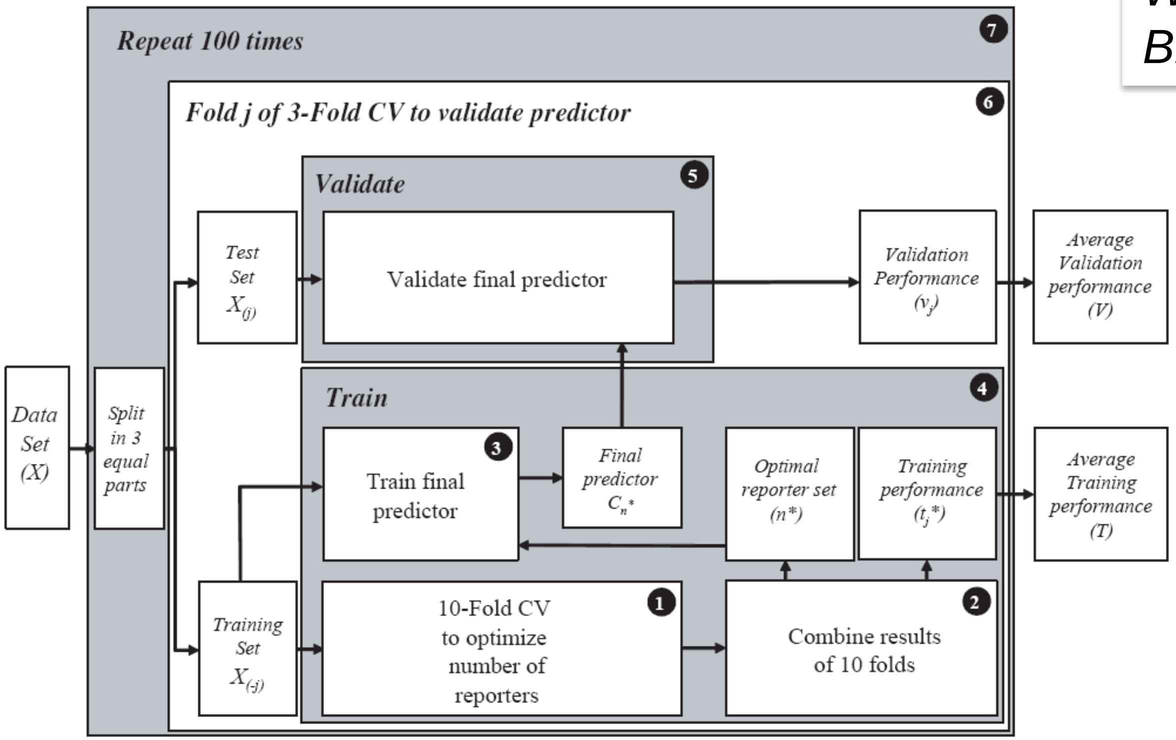


Ambroise et al., PNAS 2002

Especially in bioinformatics i.e. p>>n problems

Training, test and validation sets (4)

- Can lead to complicated schemes for estimating parameters, e.g. double/nested cross-validation loops



Wessels et al., Bioinformatics 2005

BioSB