# Classification with PAM and Random Forest

Markus Ruschhaupt

Practical Microarray Analysis 2008 - Heidelberg

**dkfz.** **GERMAN CANCER RESEARCH CENTER IN THE HELMHOLTZ ASSOCIATION**

# Two roads to classification

- Given: patient profiles already diagnosed by an expert.

- Task: infer a general rule to diagnose new patients.

- Basically, there are two ways to solve this task

  1. model class probabilities
  
     QDA, LDA, ...
  
  2. model class boundaries directly
  
     Optimal Separating Hyperplanes, Random Forest

## What's the problem?

In classification you have to trade off
- underfitting versus overfitting
- bias versus variance.

Curse of dimensionality! In 12'000 dimensions even linear methods are very complex! High variance!

# Simplify your models

# Discriminant analysis and gene selection

# Comparing Gaussian likelihoods

- Assumption: each group of patients is well described by a normal density.

- Training: estimate mean and covariance matrix for each group.

- Prediction: assign new patient to group with higher likelihood.

- Constraints on covariance structure lead to different forms of discriminant analysis.

# Discriminant analysis in a nutshell



QDA

LDA

DLDA
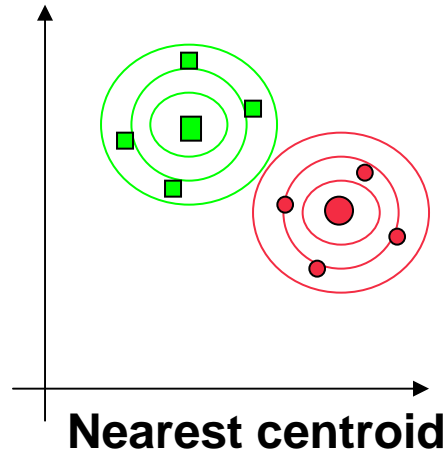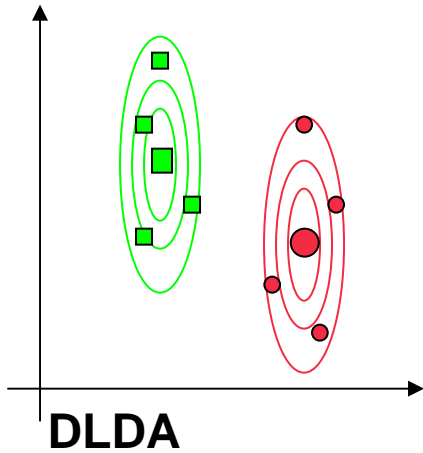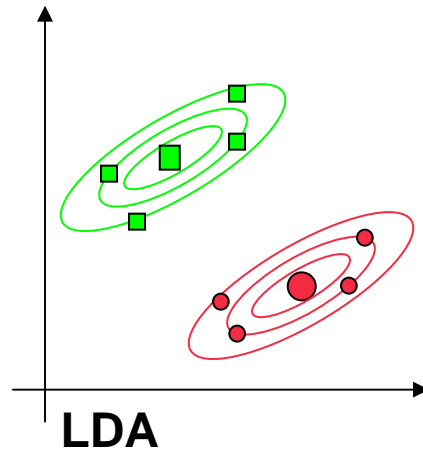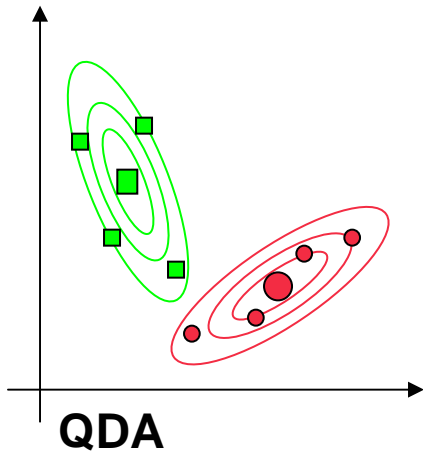
Nearest centroid

Characterize each class by mean and covariance structure.

1. Quadratic D.A. different COVs

2. Linear D.A. requires same COVs.

3. **Diagonal linear D.A.** same diagonal COVs.

4. Nearest centroids forces COVs to $\sigma^2\,I$.

# Discriminant analysis in a nutshell

Choose k that maximizes the linear discriminant function

$$\delta_k(x) = x^t \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^t \Sigma^{-1} \mu_k + \log \pi_k$$

The given data

Covariance matrix:
same for all groups

Group centroid:
one for each group

Prior class probability

For DLDA: Choose k that minimizes

$$\delta_k(x) = \sum_g \frac{(x_g - (\mu_k)_g)^2}{\sigma_g^2} - 2 \log \pi_k$$

variance of gene g

**Feature selection**

Next simplification:

Base the classification only on a small number of genes.
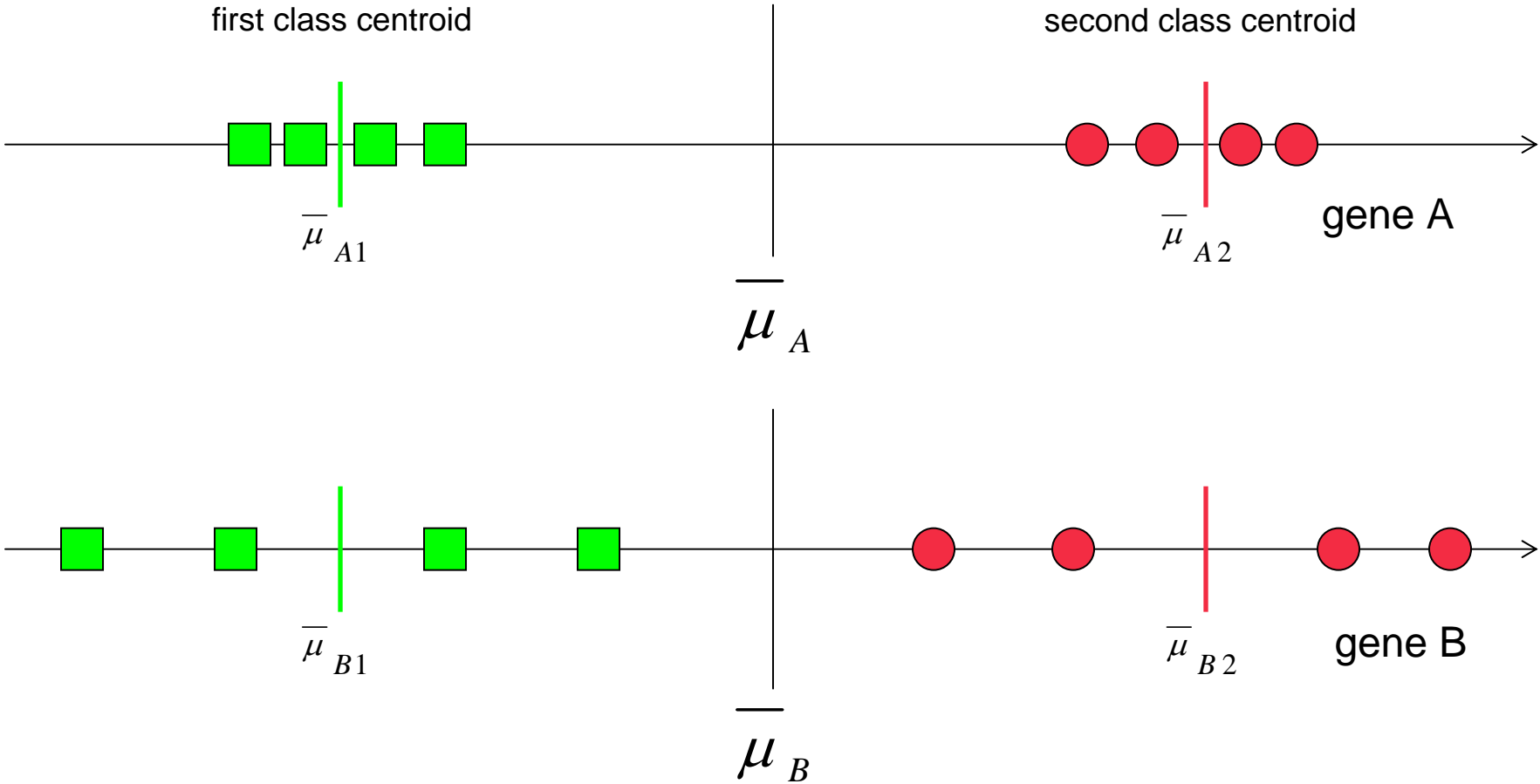
Feature selection: Find the most discriminative genes.

1. Filter: Rank genes according to discriminative power
   by t-statistic, Wilcoxon, ...
   Use only the first k genes for classification. Discrete, hard thresholding.

2. Shrinkage: Continously shrink genes until only a few have influence on classification.
   Example: Nearest Shrunken Centroids.

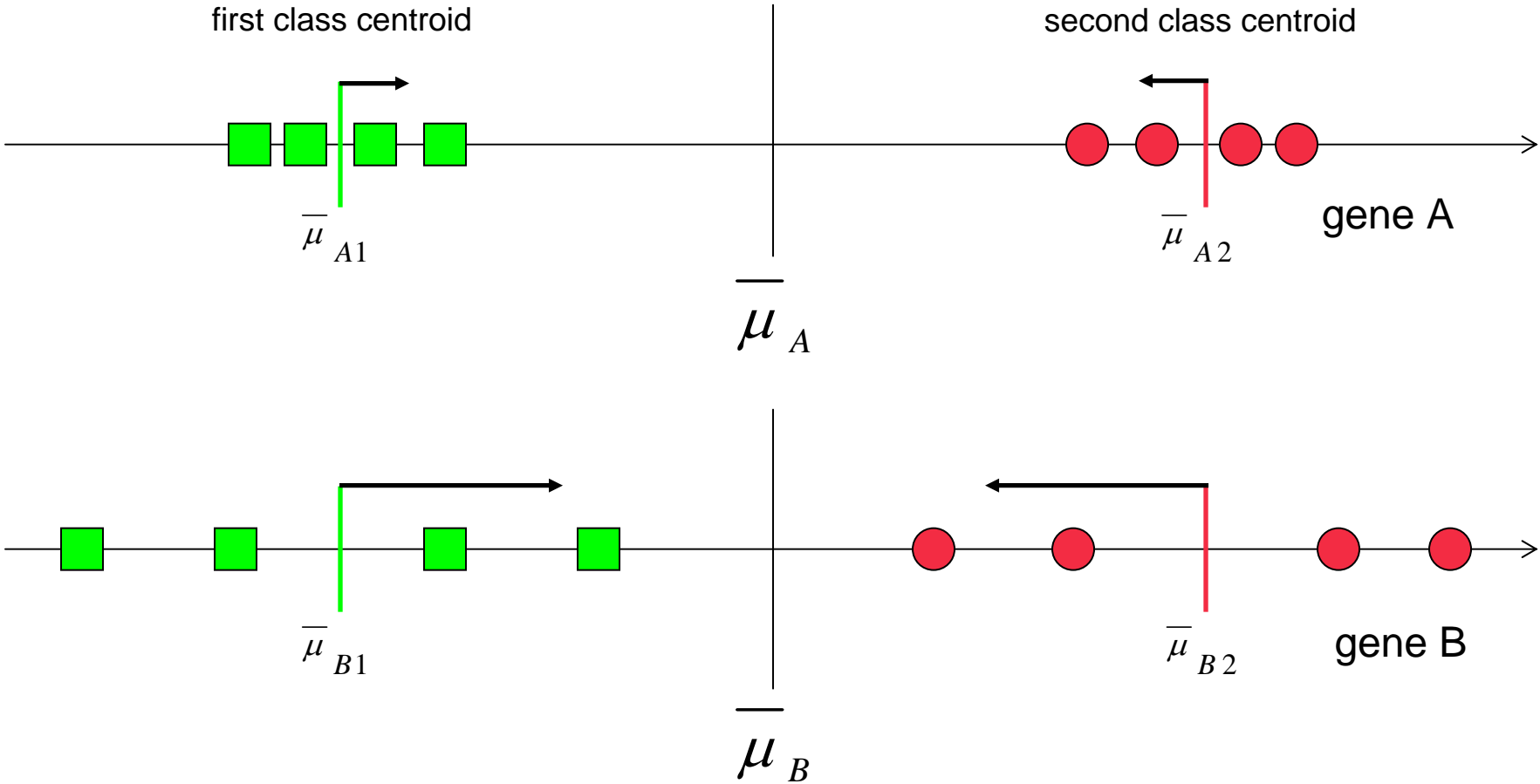# Shrunken Centroids

overall centroid

first class centroid

second class centroid

$$\overline{\mu}_{A1}$$

$$\overline{\mu}_{A2}$$

gene A

$$\overline{\mu}_{A}$$

$$\overline{\mu}_{B1}$$

$$\overline{\mu}_{B2}$$

gene B

$$\overline{\mu}_{B}$$

# Shrunken Centroids

overall centroid

first class centroid

second class centroid

$$\overline{\mu}_{A1}$$

$$\overline{\mu}_{A2}$$

gene A

$$\overline{\mu}_{A}$$

$$\overline{\mu}_{B1}$$

$$\overline{\mu}_{B2}$$

gene B

$$\overline{\mu}_{B}$$

# Nearest Shrunken Centroids

The difference between the group centroid for gene i and class k and the overall centroid can be written as follows:

$$\overline{\mu}_{gk} - \overline{\mu}_g = \sqrt{1/n_k + 1/n} \cdot (s_g + s_0) \cdot d_{gk}$$
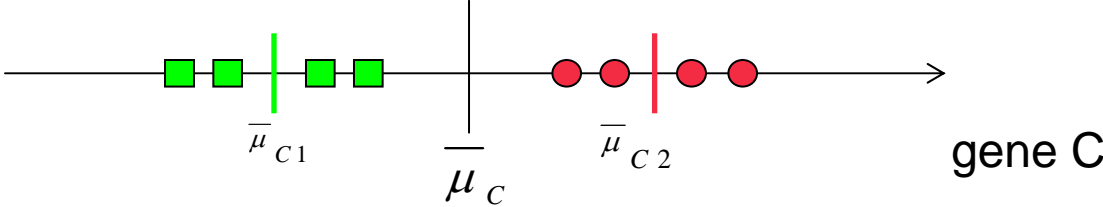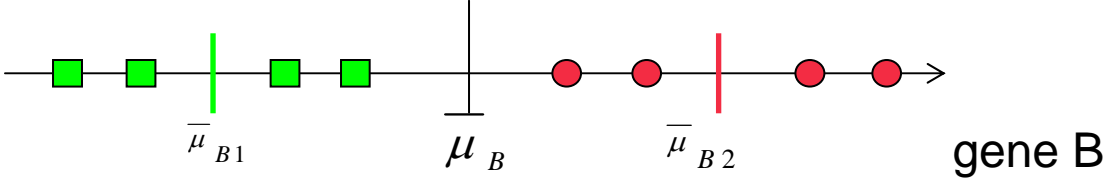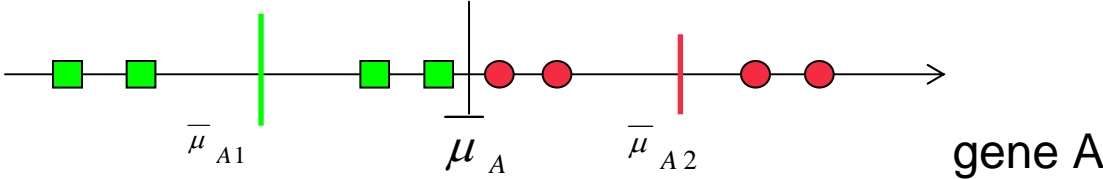
where $s_i$ is the pooled within-class standard deviation of gene i and $s_0$ is an offset to guard against genes with low expression levels.

Shrinkage: Each $d_{gk}$ is reduced by $\Delta$ in absolute value, until it reaches zero. Genes with $d_{gk}=0$ for all classes do not contribute to the classification.
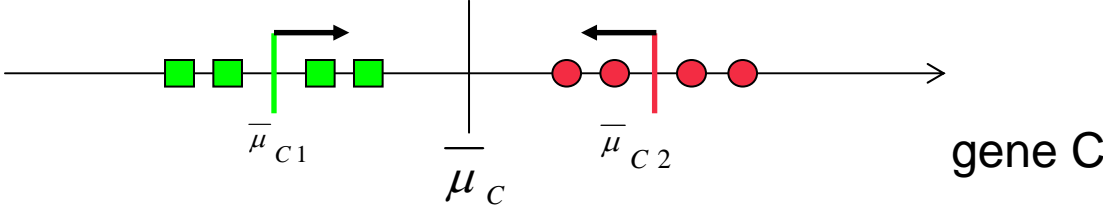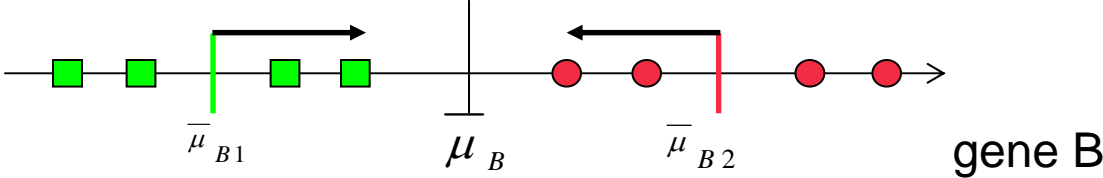
$$\overline{\mu}'_{gk} - \overline{\mu}_g = \left(\overline{\mu}_{gk} - \overline{\mu}_g\right) - \Delta \cdot \sqrt{1/n_k + 1/n} \cdot (s_g + s_0)$$

(Tibshirani et al., 2002)

dkfz.

# Shrunken Centroids



gene A

gene B

gene C

# Shrunken Centroids



gene A

gene B

gene C

$\overline{\mu}_{A1}$   $\mu_A$   $\overline{\mu}_{A2}$

$\overline{\mu}_{B1}$   $\mu_B$   $\overline{\mu}_{B2}$

$\overline{\mu}_{C1}$   $\overline{\mu}_C$   $\overline{\mu}_{C2}$

dkfz.

# Shrunken Centroids



gene A

gene B

gene C

$$\overline{\mu}_{A1} \qquad \overline{\mu}_{A} \qquad \overline{\mu}_{A2}$$

$$\overline{\mu}_{B1} \qquad \overline{\mu}_{B} \qquad \overline{\mu}_{B2}$$

$$\overline{\mu}_{C1} \qquad \overline{\mu}_{C2} \qquad \overline{\mu}_{C}$$

$$\overline{\mu}'_{B1} \qquad \overline{\mu}_{B} \qquad \overline{\mu}'_{B2}$$

$$\overline{\mu}'_{C1} \qquad \overline{\mu}_{C} \qquad \overline{\mu}'_{C2}$$

dkfz.

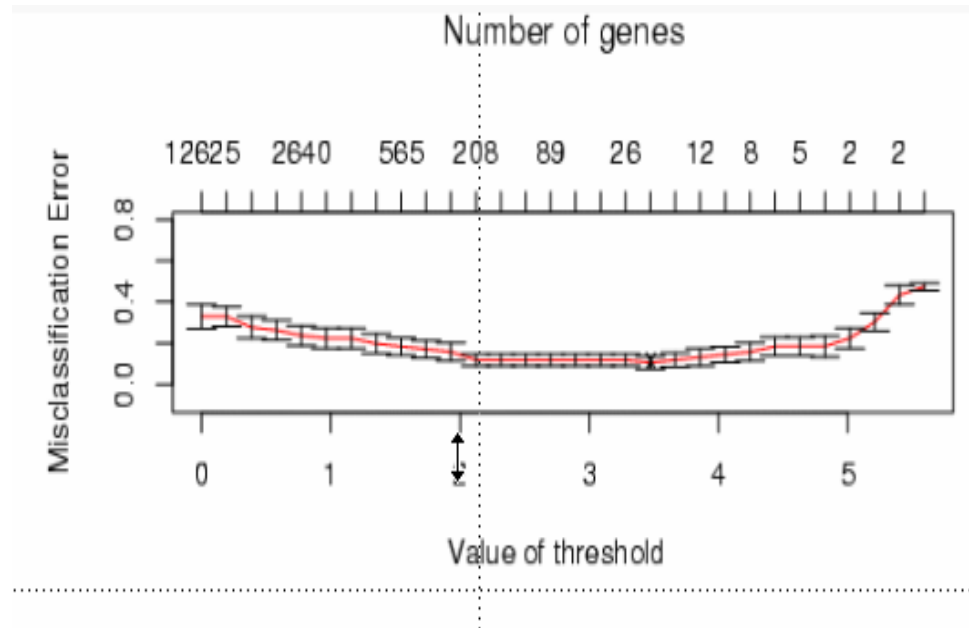# Shrunken Centroids



gene A

gene B

gene C

$$\delta_k \left( \begin{pmatrix} x_B^* \\ x_C^* \end{pmatrix} \right) = \frac{(x_B^* - \overline{\mu}'_{Bk})^2}{(s_B + s_0)^2} + \frac{(x_C^* - \overline{\mu}'_{Ck})^2}{(s_C + s_0)^2} - 2\log \pi_k$$
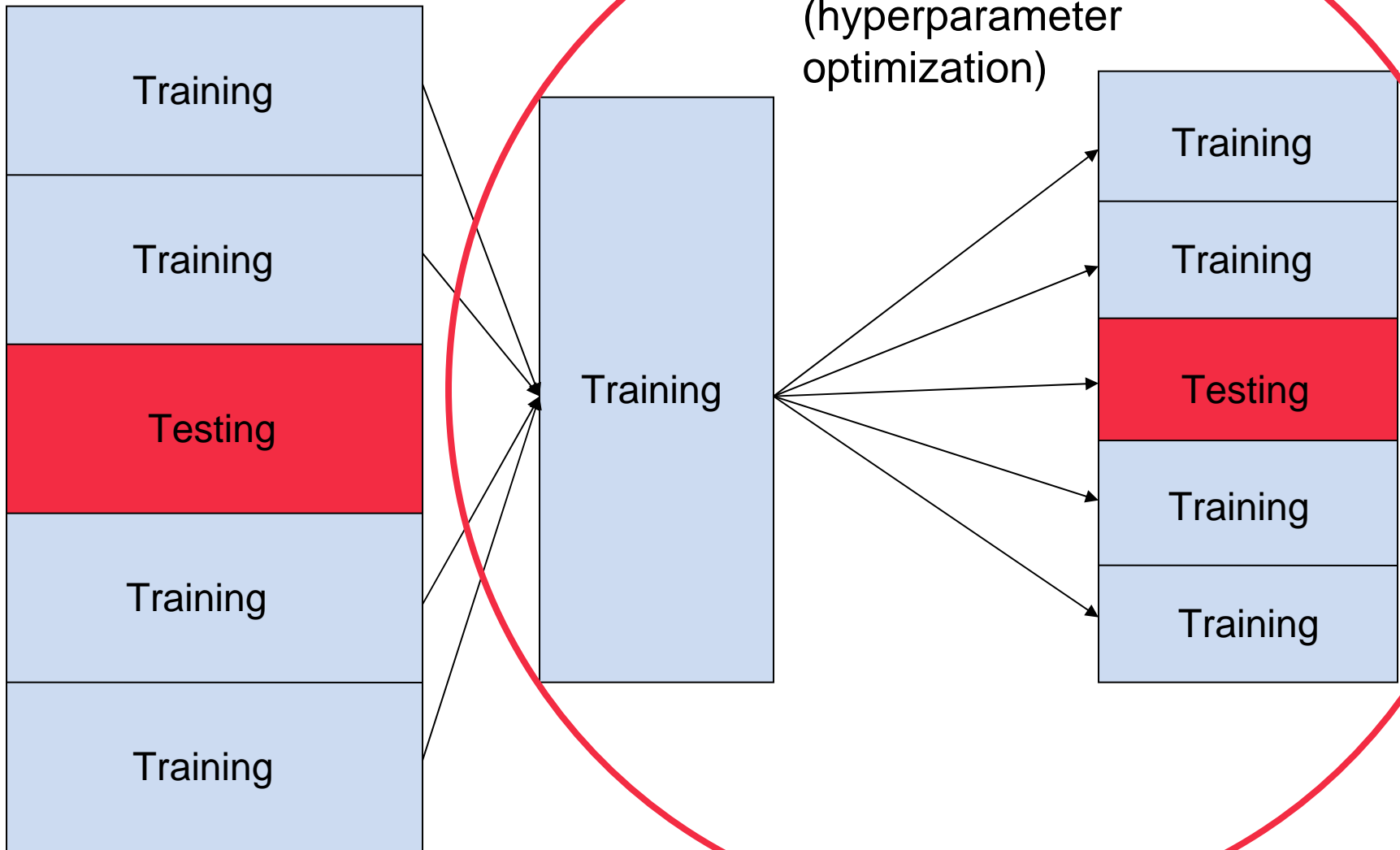
# Amount of shrinkage



**Small △, many genes, poor performance due to overfitting**

**High △, few genes, poor performance due to lack of information – *underfitting* -**

**The optimal △ is somewhere in the middle**

# Nested Cross-Validation



Model Selection (hyperparameter optimization)

Training

Training

Testing

Training

Training

Training

Training

Training

Training

Testing

Training

Training

dkfz.

# Choosing $\Delta$ with cross-validation

•Idea: given a set of possible $\Delta=\{\Delta_1,...,\Delta_n\}$ we want to estimate the misclassification rate for each $\Delta$ and choose the ‚best'. Use cross-validation to estimate the misclassification rate.
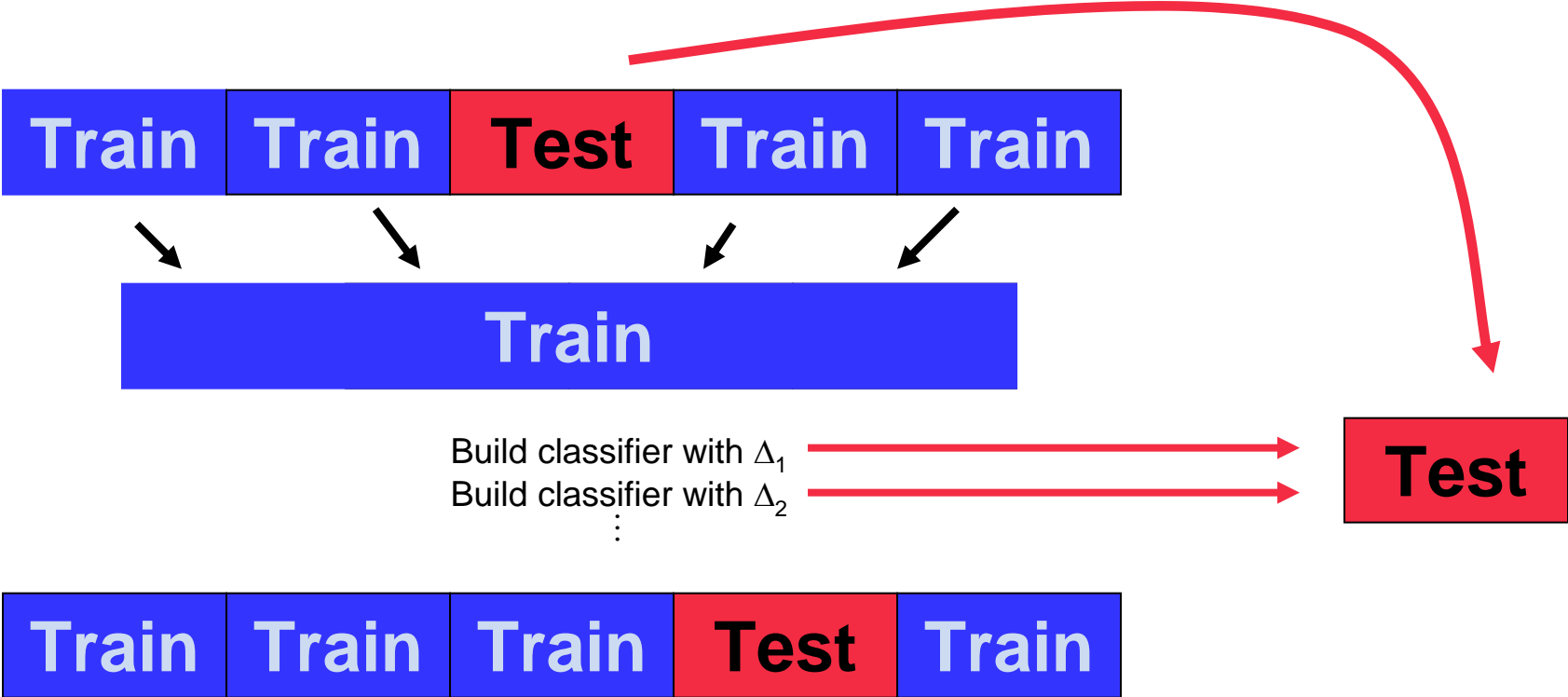
# Shortcomings of filter and shrinkage methods

- Filter and Shrinkage work only on single genes.
  They don't find interactions between groups of genes.

- Filter and Shrinkage methods are only heuristics.
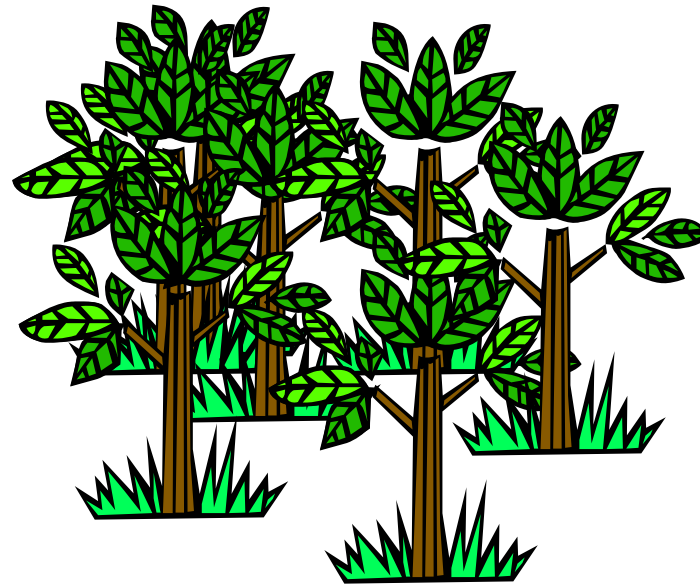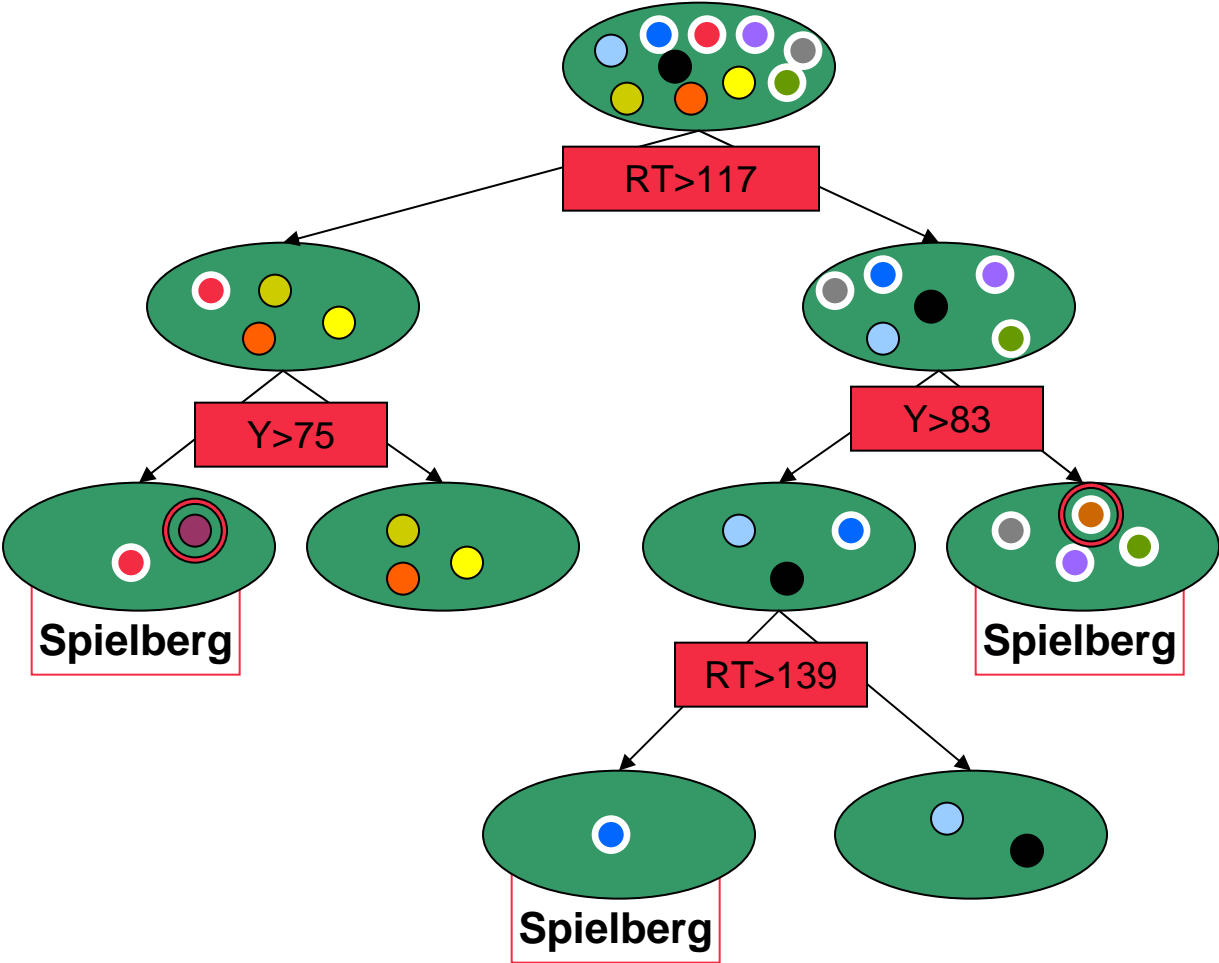  Search for best subset is infeasible for more than 30 genes.

dkfz.

# Random Forest

- Growing one tree

- Growing many trees (a forest)

# A binary tree – a classification example



| Title | | Year (Y) | Running time (RT) | Spielberg |
|---|---|---|---|---|
| Jaws | | 75 | 124 | X |
| Duel | | 71 | 90 | X |
| The Color Purple | | 85 | 154 | X |
| Schindler's List | | 93 | 195 | X |
| Jurassic Park | | 93 | 127 | X |
| Back to the future | | 85 | 111 | |
| The Godfather | | 72 | 175 | |
| Das Boot | | 81 | 150 | |
| Life of Brian | | 79 | 94 | |
| Stand by me | | 86 | 89 | |

| Hook | | 91 | 144 | X |
|---|---|---|---|---|
| Harold and Maude | | 71 | 91 | |

# A binary tree for classification
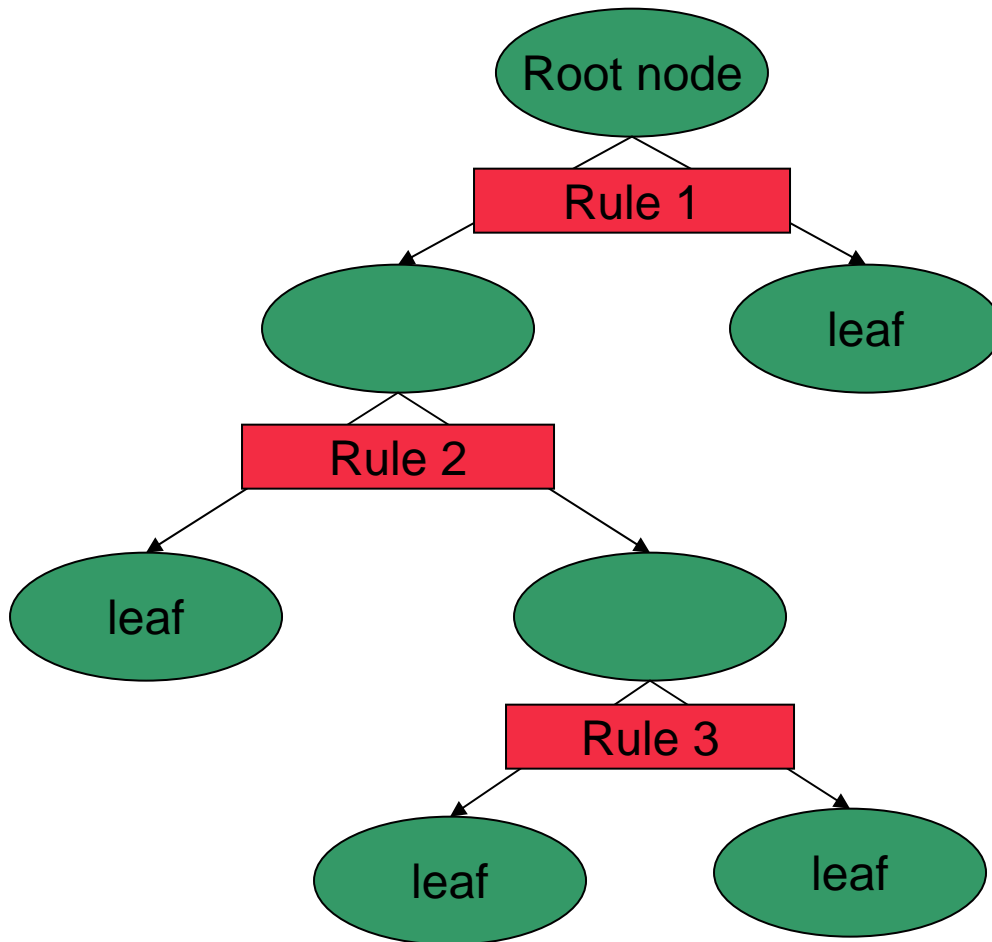


- The root node contains all samples.

- Each node contains a fraction of samples.

- Each rule splits up the samples into two groups.

- Every rule is of the form
  - $X > t$   for continuous X
  - $X \in A$  for categorial X

Only one variable per rule.

- Each leaf should be more or less pure (contains only samples of one type).

- A new sample is run through the tree and one looks for the leaf it ends up.

# A binary tree – a classification example
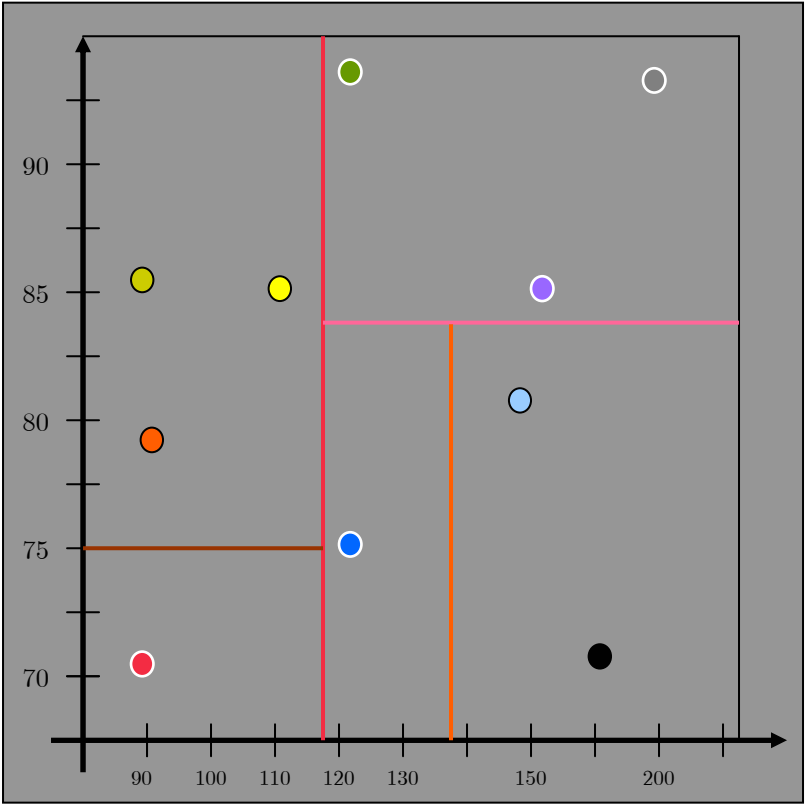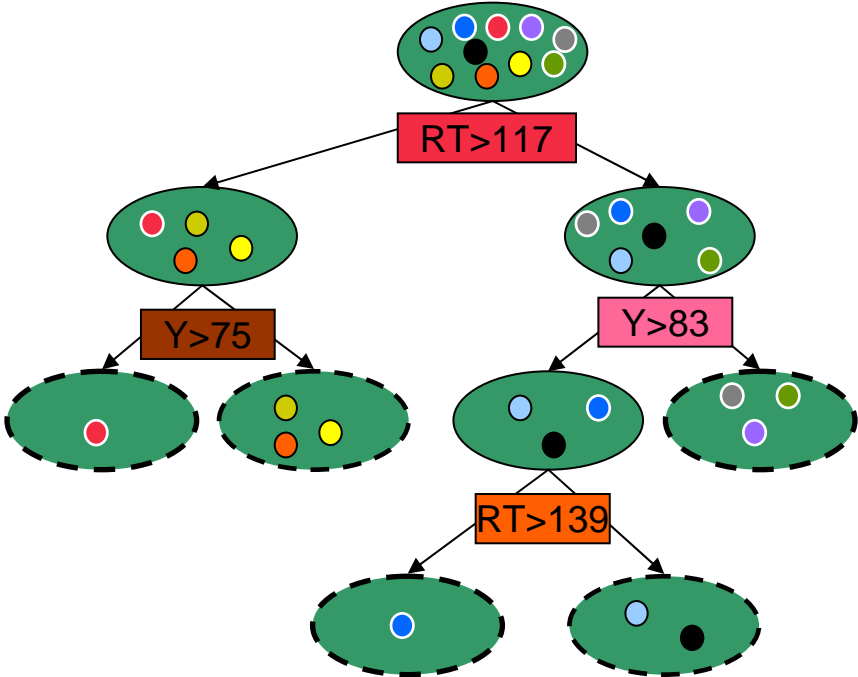


| Title | | Year(Y) | Running time (RT) | Spielberg |
|---|---|---|---|---|
| Back to the future | | 85 | 111 | |
| The Godfather | | 72 | 175 | |
| Das Boot | | 81 | 150 | |
| Life of Brian | | 79 | 94 | |
| Stand by me | | 86 | 89 | |

| Title | | Year(Y) | Running time (RT) | Spielberg |
|---|---|---|---|---|
| Jaws | | 75 | 124 | X |
| Duel | | 71 | 90 | X |
| The Color Purple | | 85 | 154 | X |
| Schindler's List | | 93 | 195 | X |
| Jurassic Park | | 93 | 127 | X |

dkfz.

# Not every partition can be achieved through a tree
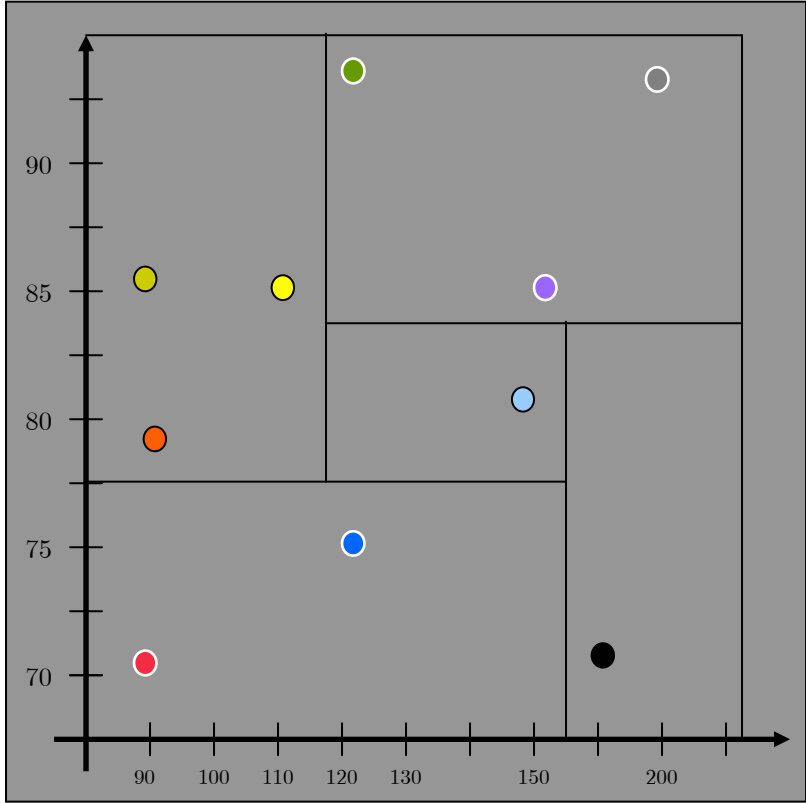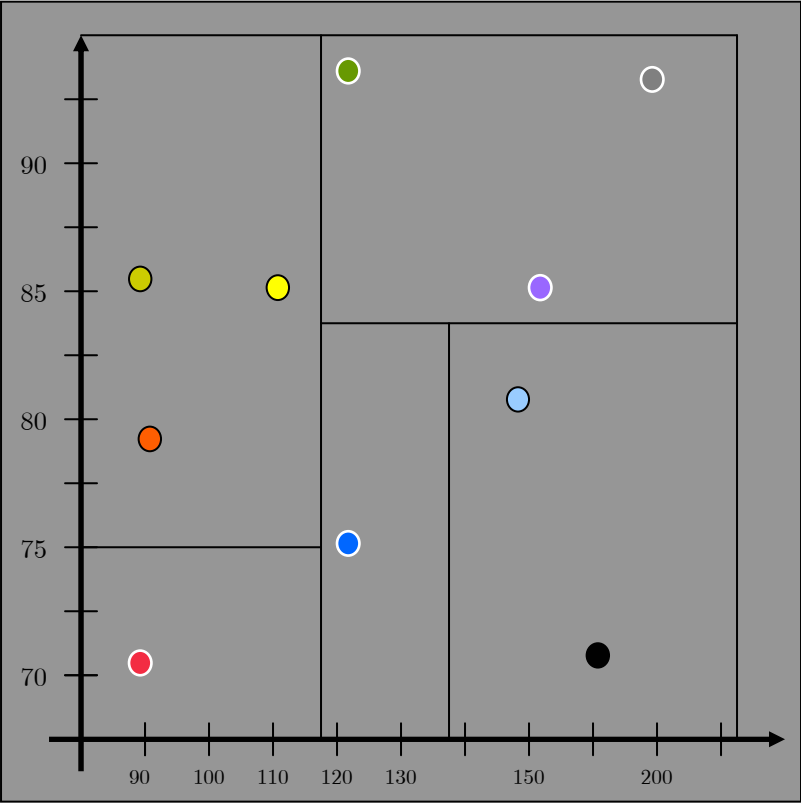
# Trees for microarray data

- Samples are microarray chips (containing many genes) divided into two or more classes, e.g. ER status +/- in breast cancer.

- Rules are based on the expression of one gene. Examples:
  - ERB2 is lower than 3.21
  - GAPDH is higher than 6.23

- You cannot base rules on more that one gene or two or more thresholds. Examples:
  - GAPDH is lower that ERB2
  - ERB2 is lower that 2.21 or higher than 5.32

# How to construct a tree?

- Choose a good node and a good gene/threshold pair ($x_i, t_i$) for each split („Splitting rule")
  - Which gene should we choose?
  - Which threshold should we use?


- Decide how many nodes your tree should have („Split-stopping rule","pruning").


- Assign a class to each leaf („Class assignment rule").

# Node splitting

fraction of samples in node u

fraction of samples in node w

impurity in node u

impurity in node w

impurity in node v

$$G=i(v) - (p_u\ i(u) + p_w\ i(w))$$

v

u    w
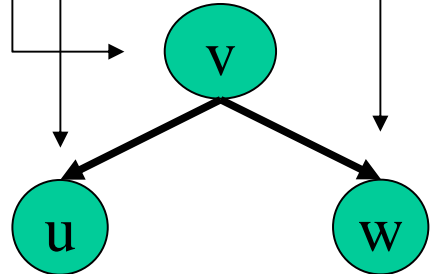
# Splitting rule

Impurity function $\Phi: [0,1]^k \rightarrow \Re$ ( k: number of different classes)

A low impurity function is achievable.

Attributes:

- minimal for (1,0,0,.....,0) and all permutations.

- maximal for (1/k,...,1/k)

- is symmetric function

  $\Phi(p_1,....p_n) = \Phi(\varphi(p_1),.. \varphi(p_n))$ for all permutations $\varphi$

There are various impurity functions: entropy, Gini index

# Gini index

**Gini index** $\Phi(p_1,...,p_k) = 1 - \sum p_i^2 = \sum p_i (1-p_i) = \sum \sum p_i p_j$
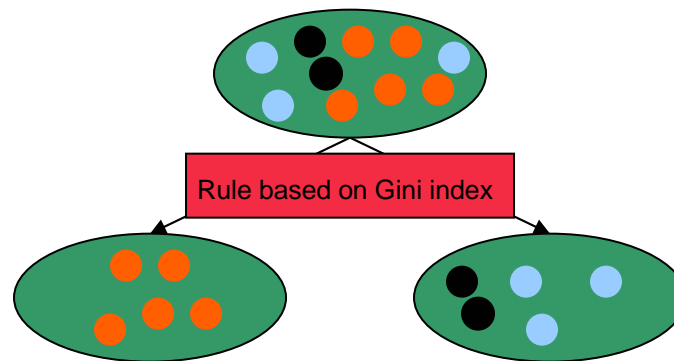
Minimum 0

Maximum 1- 1/k

Gini index tries to separate the largest class from the rest.



Rule based on Gini index

Gini index for two classes:
$$\Phi(p_1, p_2) = 2 p_1 p_2$$

dkfz.

## Using the impurity function

Define a function i(v) (v is a node of the tree)

$$i(v) = \Phi(P(1|v),....,P(k|v)).$$

P(j|v) := probability that you are member of class j if you are in node v

If the probabilities for all classes are equal one has

$$i(v) = \Phi(n_1(v)/n(v),....,n_k(v)/n(v)).$$

n (v) = all samples in node v

$n_j(v)$= all samples of class j in node v

Two classes, Gini index, equal probability:

$$i(v) = \frac{2\, n_1(v)\, n_2(v)}{n(v)^2}$$

# Goodness of fit – decrease of impurity

Determine gene x and threshold t that maximizes the following term:

$$G = i(v) - (p_u \, i(u) + p_w \, i(w))$$
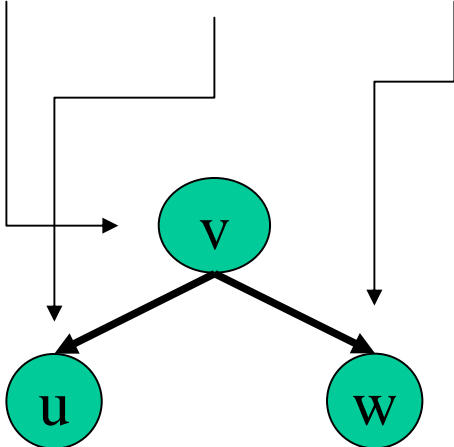
$p_u$ = fraction of samples in node u

$p_w$ = fraction of samples in node w

Because you have only a finite number of genes and a finite number of values, you have to test a finite number of pairs.

Two classes, Gini index, equal probability:

$$G = \frac{2}{n(v)} \cdot \left( \frac{n_1(v)\, n_2(v)}{n(v)} - \frac{n_1(u)\, n_2(u)}{n(u)} - \frac{n_1(w)\, n_2(w)}{n(w)} \right)$$

dkfz.

# Stop splitting rules – stop decision

- Decide at each node if the node should be split further or not. Is there a high decrease in the Gini index?
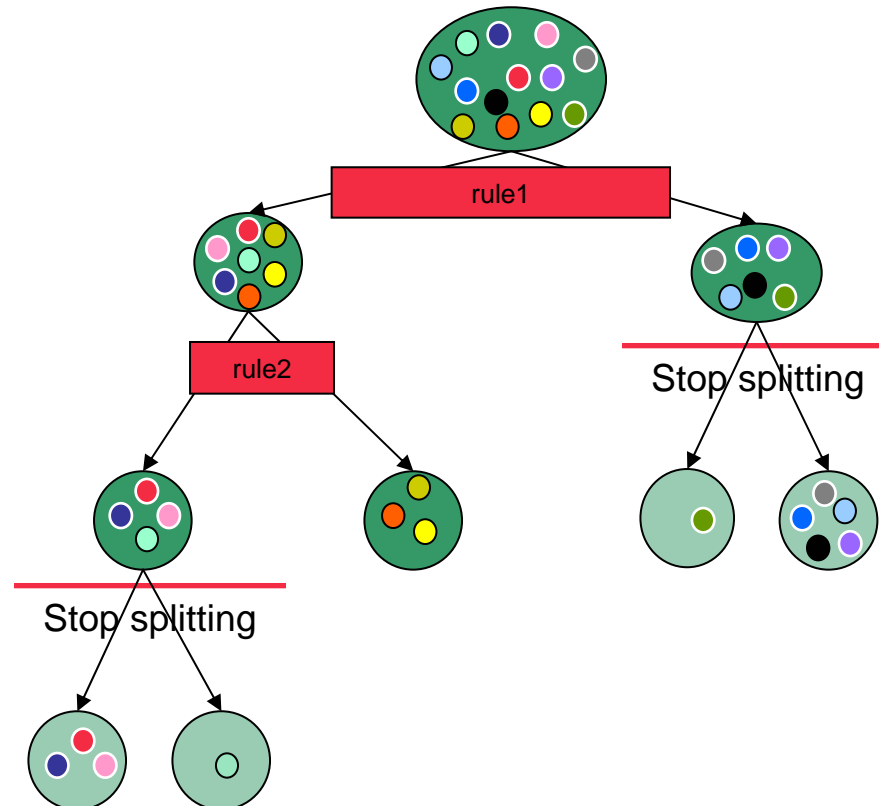
# Stop splitting rules - pruning

- First split until each node is pure or has a maximum number of elements, then prune the tree. Remove the splits with low decrease in the impurity function.

First step

Second step

Prune
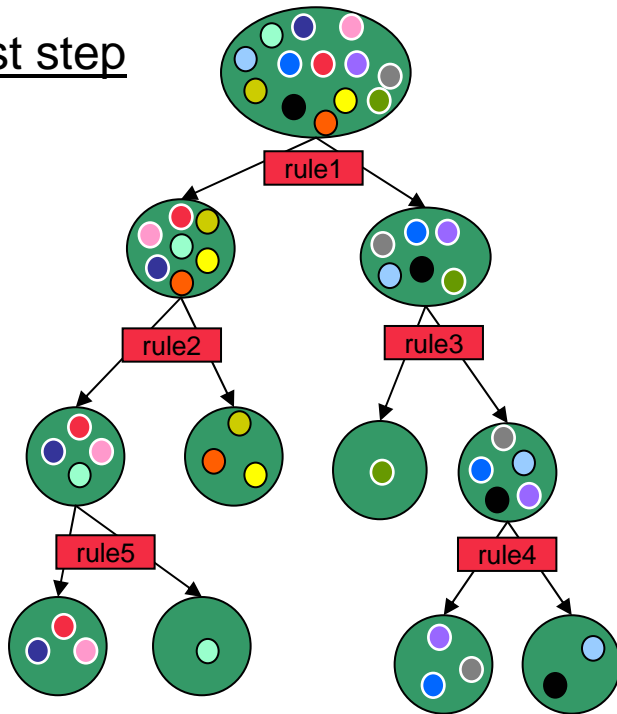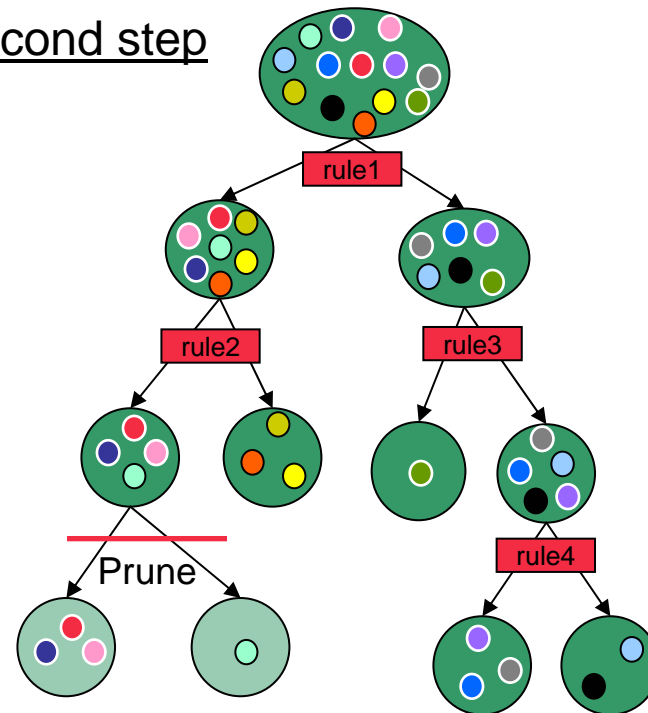
# Advantages and Disadvantages for trees

- Advantages

  - At first sight good interpretation

  - Can cope with any data structure or type

  - Uses conditional information effectively

  - Invariant under transformations of the gene expressions

- Disadvantages

  - not robust

  - classification performance is not that good

  - Genes can be hidden due to other genes

# From the tree to the forest

- General idea: combine collection of weak learners to construct a classification algorithm with better properties.

- Here: Construct a collection of trees and combine results of individual trees.

- Because the algorithm is deterministic, we have to introduce some kind of randomness. For each tree we will have two different sources of randomness:
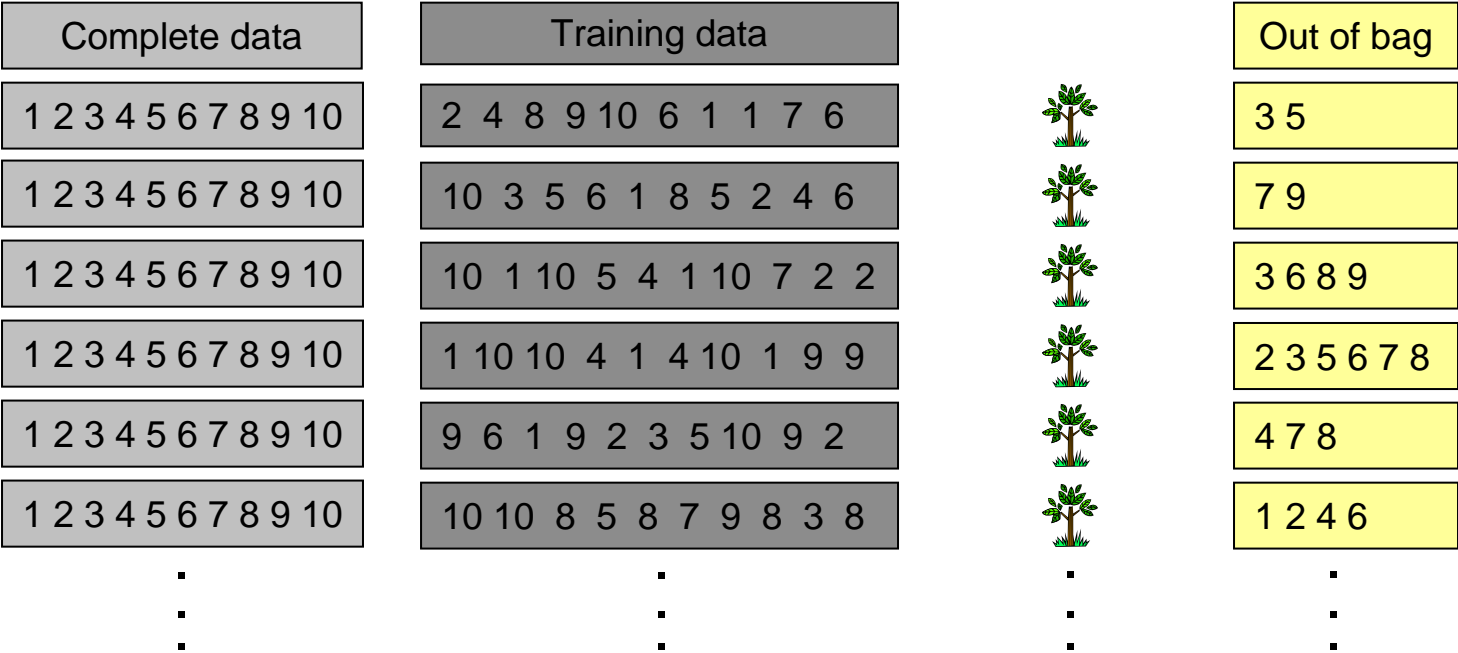  - random training set (bootstrap)
  - random gene selection

# Bootstrap

- Complete data :     K samples/patients
- Training data:     draw K times with replacement from the data set
- Out of bag:     sample that do not belong to the training set

Training data contains approximately 2/3 of the elements of the complete data set.

| Complete data | Training data | | Out of bag |
|---|---|---|---|
| 1 2 3 4 5 6 7 8 9 10 | 2 4 8 9 10 6 1 1 7 6 | | 3 5 |
| 1 2 3 4 5 6 7 8 9 10 | 10 3 5 6 1 8 5 2 4 6 | | 7 9 |
| 1 2 3 4 5 6 7 8 9 10 | 10 1 10 5 4 1 10 7 2 2 | | 3 6 8 9 |
| 1 2 3 4 5 6 7 8 9 10 | 1 10 10 4 1 4 10 1 9 9 | | 2 3 5 6 7 8 |
| 1 2 3 4 5 6 7 8 9 10 | 9 6 1 9 2 3 5 10 9 2 | | 4 7 8 |
| 1 2 3 4 5 6 7 8 9 10 | 10 10 8 5 8 7 9 8 3 8 | | 1 2 4 6 |

# Random Forest – random variable selection

- Set the parameters **m** and **s**. These are the same for all nodes and all trees. Specify the number of trees you want to grow.

- For each tree:

  - Use bootstrap to get a training set T of samples. You use only data from this set to build the tree!

  - For each node, first choose **m** genes $x_1,...x_m$ at random.

  - Upon these genes choose the best pair $(x_i, t_i)$.

  - Go on until each node contains elements of one class or consists of **s** elements.

# Random Forest

- Classification rule: First classify a new sample with each tree. In the end form a major vote.
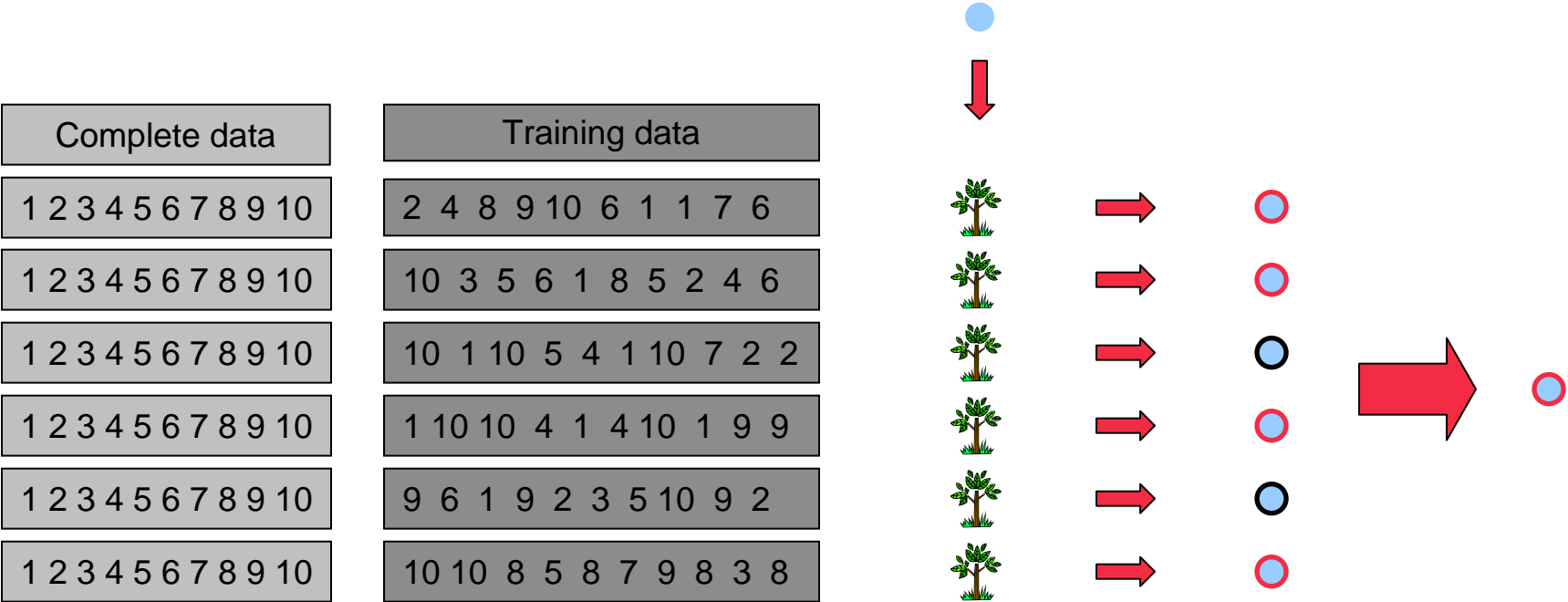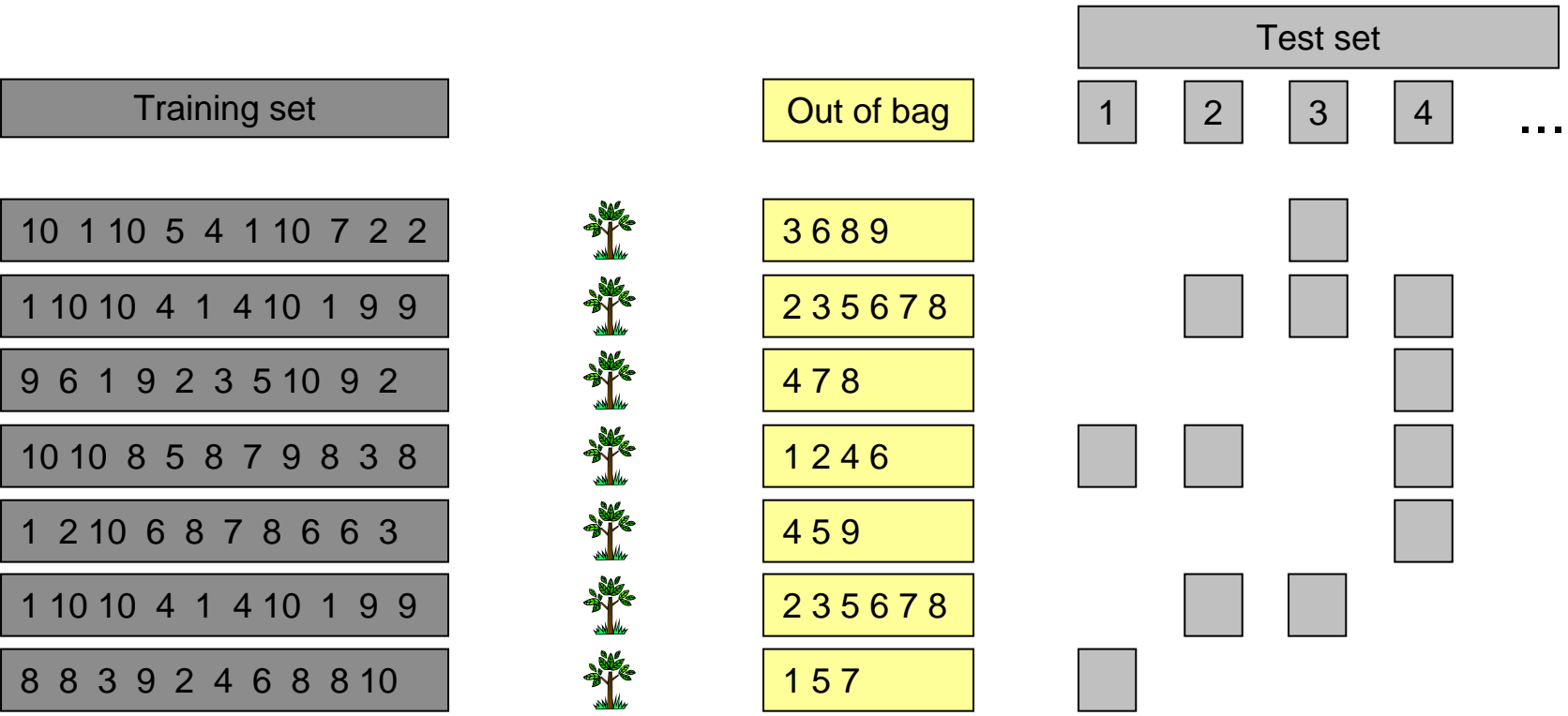
# Estimating the test error

- For each sample x predict the class but use only the trees for which x belongs to the OOB set.

- Good estimate for the test error because the information provided by x was not used for building these trees.

| Training set | | Out of bag | Test set | | | | |
|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | … |
| 10 1 10 5 4 1 10 7 2 2 | 🌳 | 3 6 8 9 | | | ▢ | | |
| 1 10 10 4 1 4 10 1 9 9 | 🌳 | 2 3 5 6 7 8 | | ▢ | | ▢ | |
| 9 6 1 9 2 3 5 10 9 2 | 🌳 | 4 7 8 | | | | ▢ | |
| 10 10 8 5 8 7 9 8 3 8 | 🌳 | 1 2 4 6 | ▢ | ▢ | | ▢ | |
| 1 2 10 6 8 7 8 6 6 3 | 🌳 | 4 5 9 | | | | ▢ | |
| 1 10 10 4 1 4 10 1 9 9 | 🌳 | 2 3 5 6 7 8 | | ▢ | ▢ | | |
| 8 8 3 9 2 4 6 8 8 10 | 🌳 | 1 5 7 | ▢ | | | | |

# Tradeoff correlation - prediction

The forest error rate depends on two things:

- The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate.
- The strength of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

.

Low correlation
Weak individual classifiers

High correlation
Strong individual classifiers

**m**

Small

Large
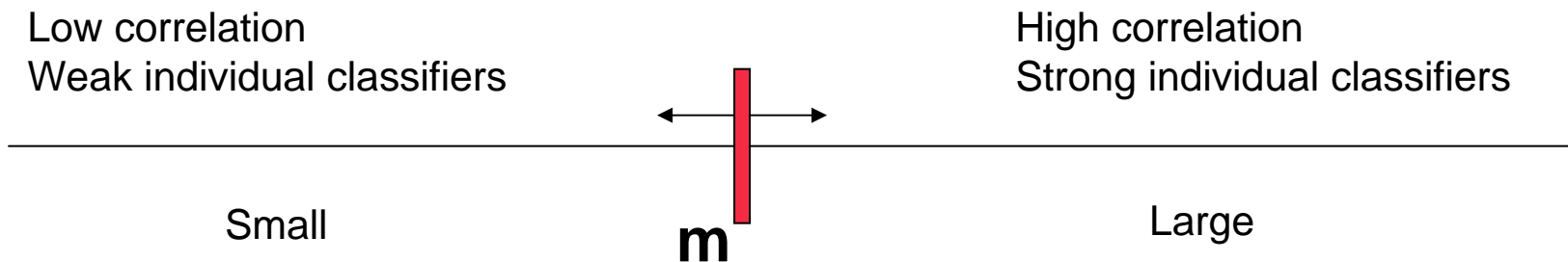
Use the OOB error estimate to find a good parameter m.

# Variable importance

Idea: Change the values of a gene **x** and check whether the OOB error changes dramatically.

- For each gene **x** do the following:
  - For each tree t of a forest permute the values of x for the samples that belong to the out-of-bag set.
  - Estimate the OOB error as before (the samples have new values for the gene x).
  - Compare the OOB error of the samples with the original values to the OOB error of the samples with permuted values. This gives you an importance measure.

## Literature

- S. Dudoit, J. Fridlyand: Classification in microarray experiments

- L. Breiman: Random Forests—Random Features

- L. Breiman: Statistical Modeling: The two Cultures

- J. Oh, M. Laubach, A. Luczak: Estimating Neuronal Variable Importance with Random Forest

- T. Hastie, R. Tibshirani, Jerome Friedman: The Elements of Statistical Learning. Springer 2001.

- R.Tibshirani, T. Hastie, B. Narasimhan, G. Chu: Diagnosis of multiple cancer types by shrunken centroids of gene expression, PNAS, 99(10), 6567–6572, 2002.

dkfz.

Acknowledgement: Florian Markowetz for all the slides (PAM)

**Thank you**