UNIVERSITY OF SKÖVDE

DAY 3 – CLASSIFICATION ANALYSIS FOR BIOMARKER DISCOVERY

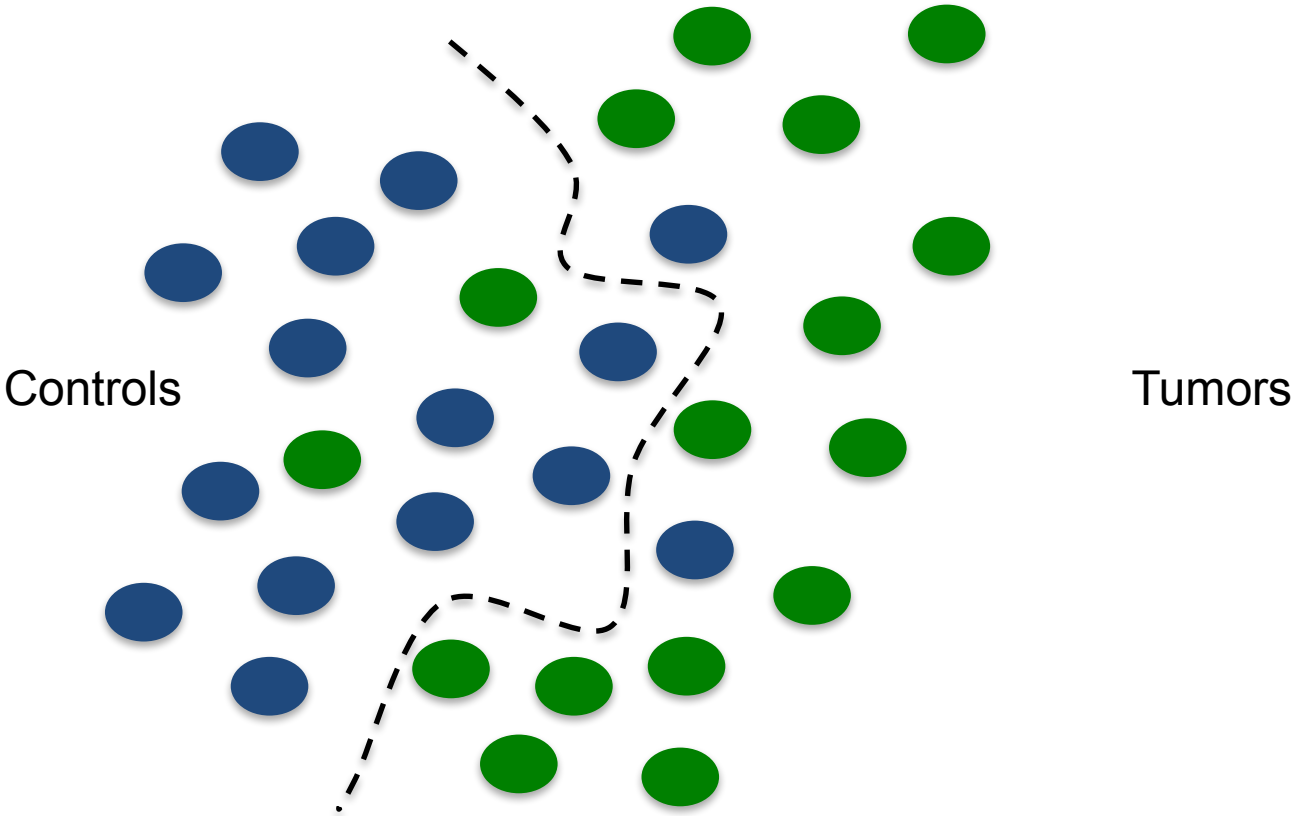
Benjamin Ulfenborg, PhD
Senior lecturer in bioinformatics, University of Skövde
benjamin.ulfenborg@his.se

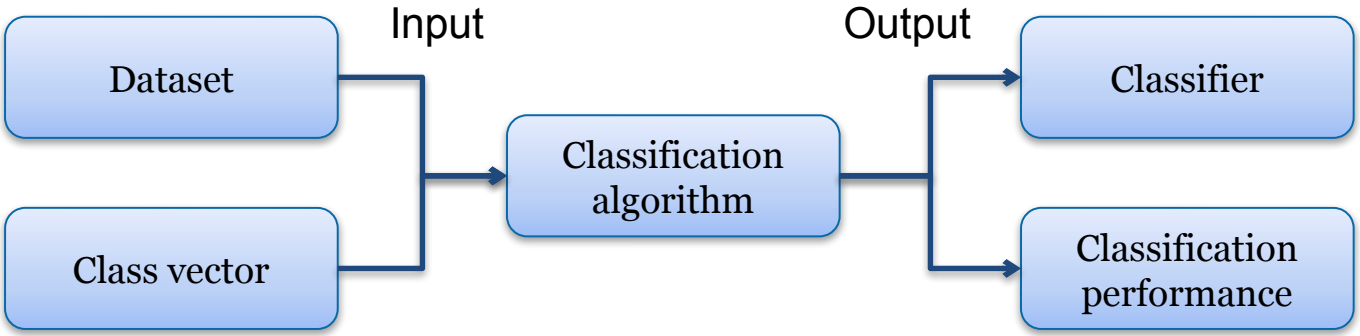
(with small adjustments by Dirk.Repsilber@oru.se)

OUTLINE

UNIVERSITY OF SKÖVDE

- Introduction to data mining
- Example of classification algorithms
- Classification analysis applied to sequencing data


Controls
Tumors



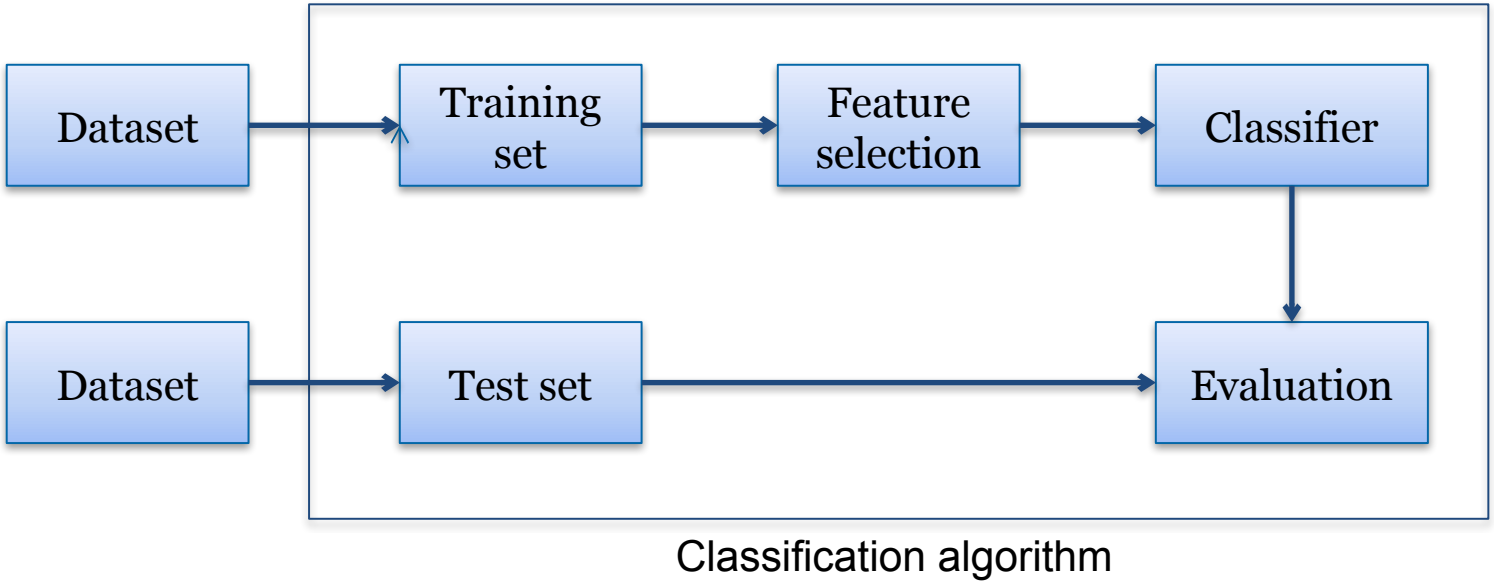
Dataset

Feature	Sample 1	Sample 2	Sample 3	Sample 4
Gene A	47	32	183	145
Gene B	62	39	41	103
Gene C	271	194	61	89
Gene D	73	21	37	71

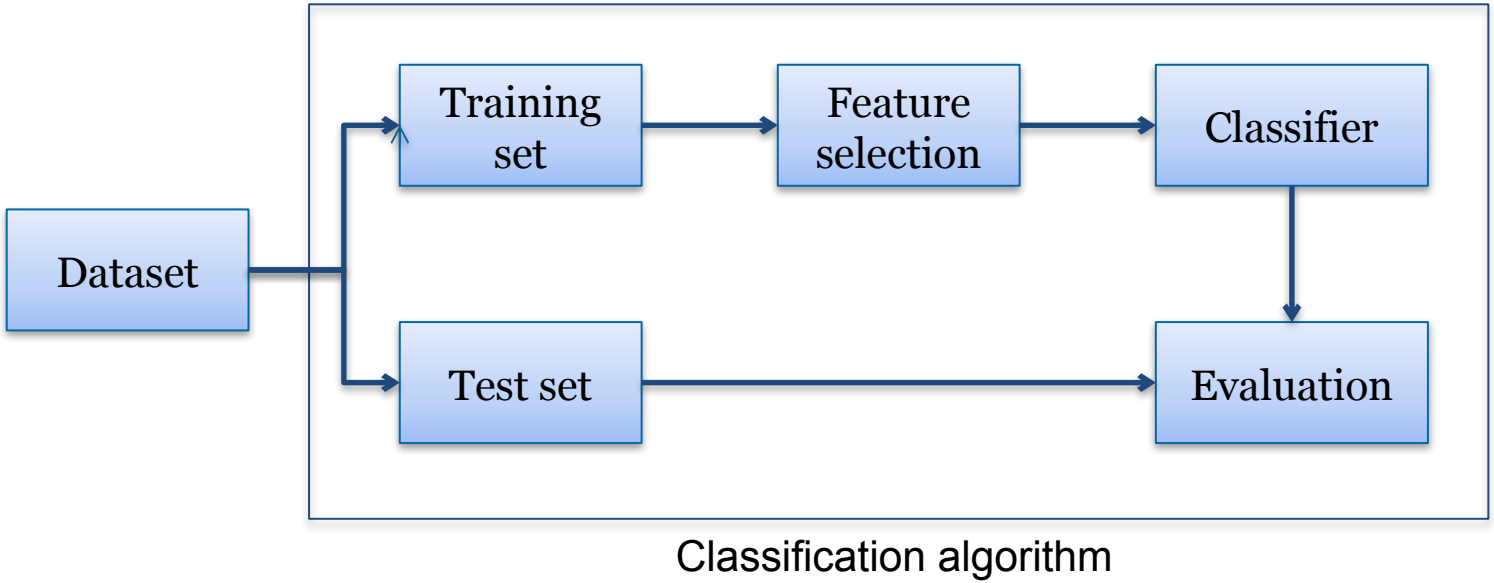
Class vector

Sample	Tissue
Sample 1	Control
Sample 2	Control
Sample 3	Tumor
Sample 4	Tumor

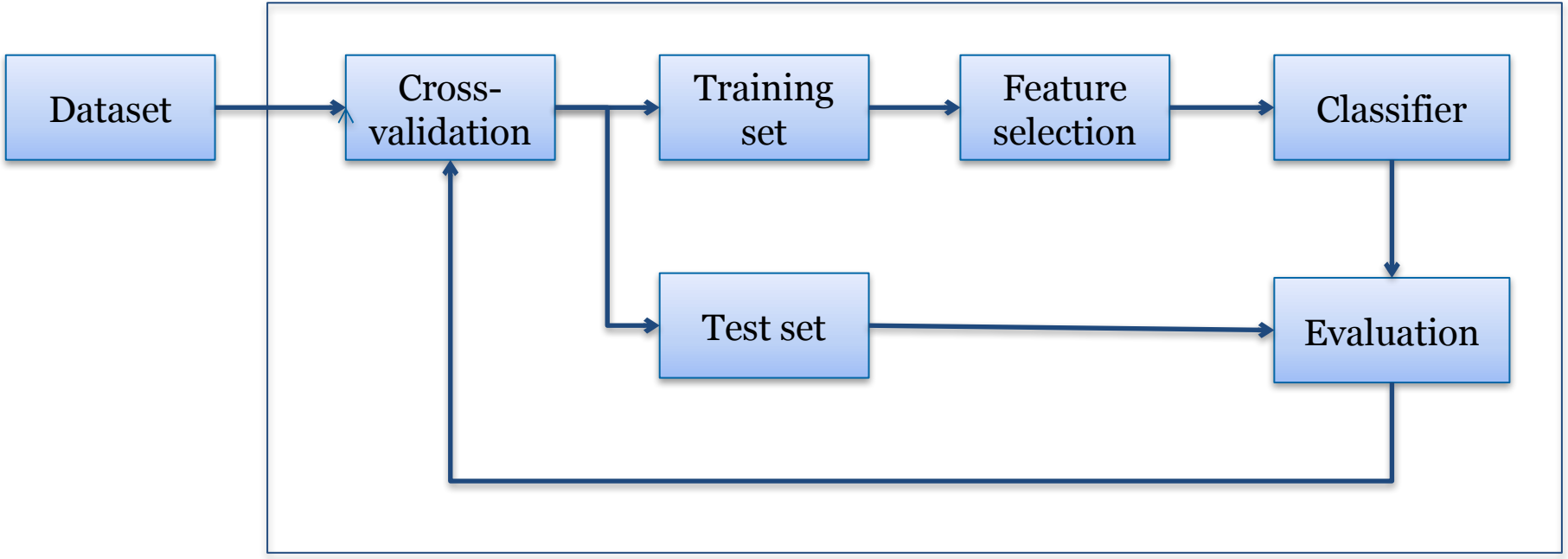
Independent training and test set



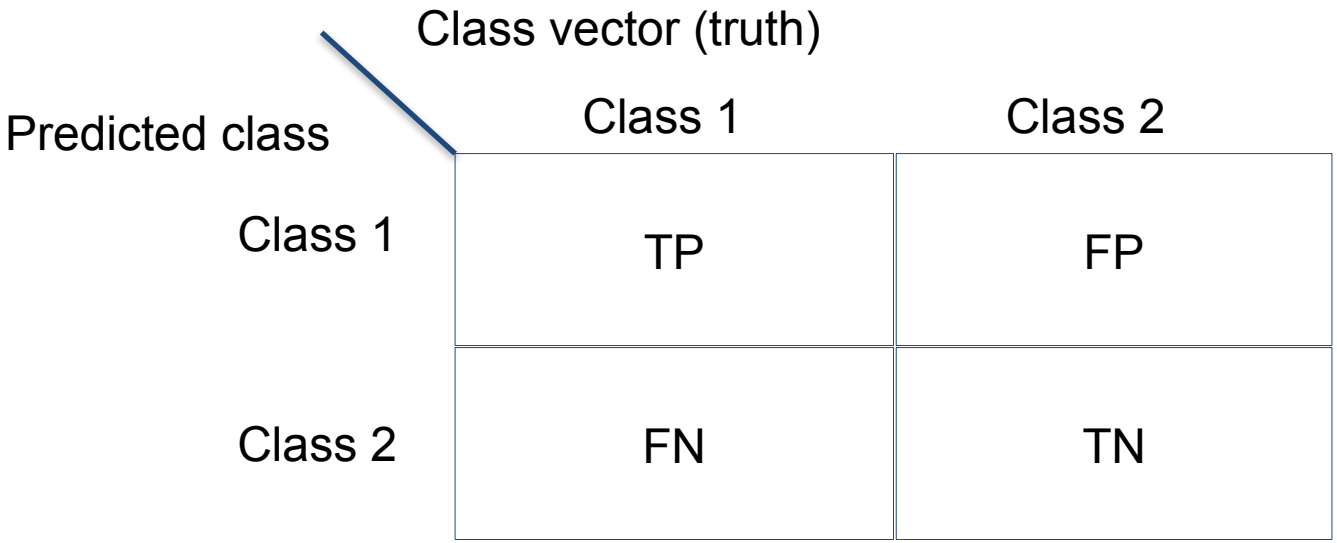
Splitting dataset into training and test set



Cross validation



Confusion matrix



TP: True positive
FP: False positive
FN: False negative
TN: True negative

Class 1: Tumor
Class 2: Control

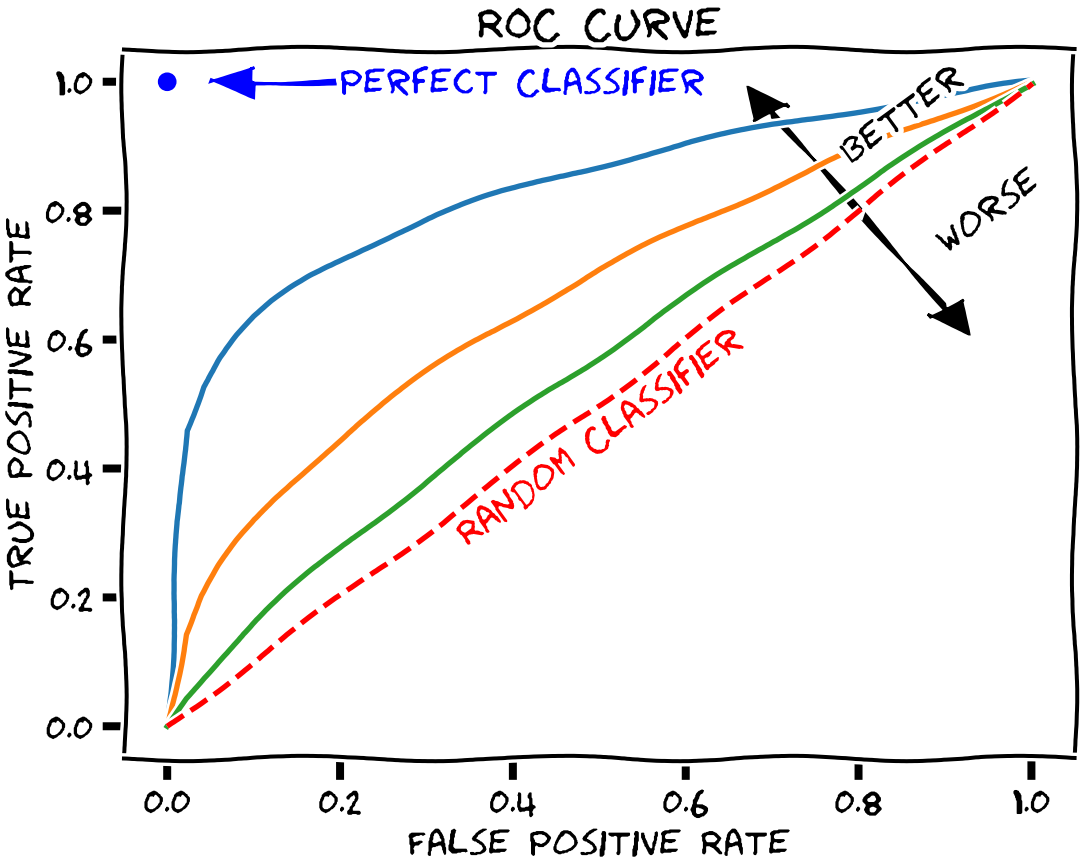
$$SN = \frac{TP}{TP + FN}$$

$$BACC = \frac{SN + SP}{2}$$

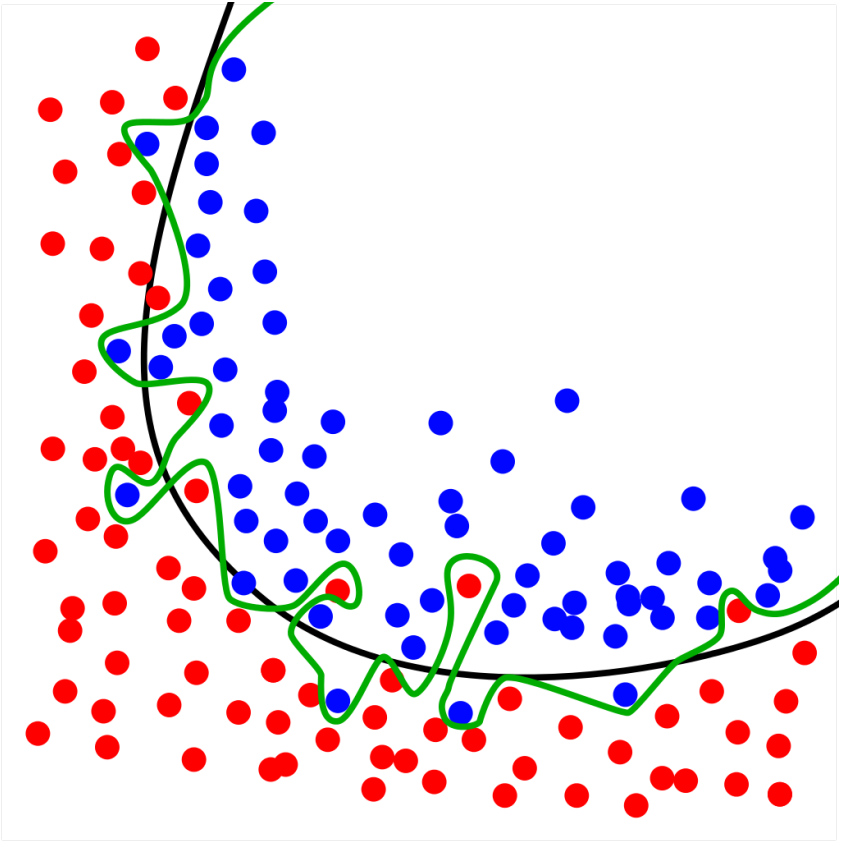
$$SP = \frac{TN}{TN + FP}$$

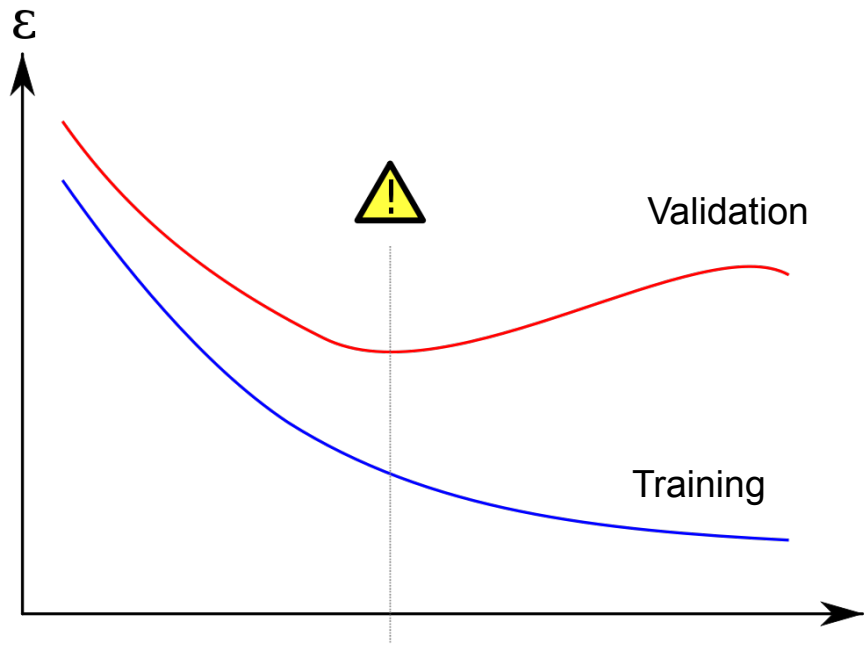
$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

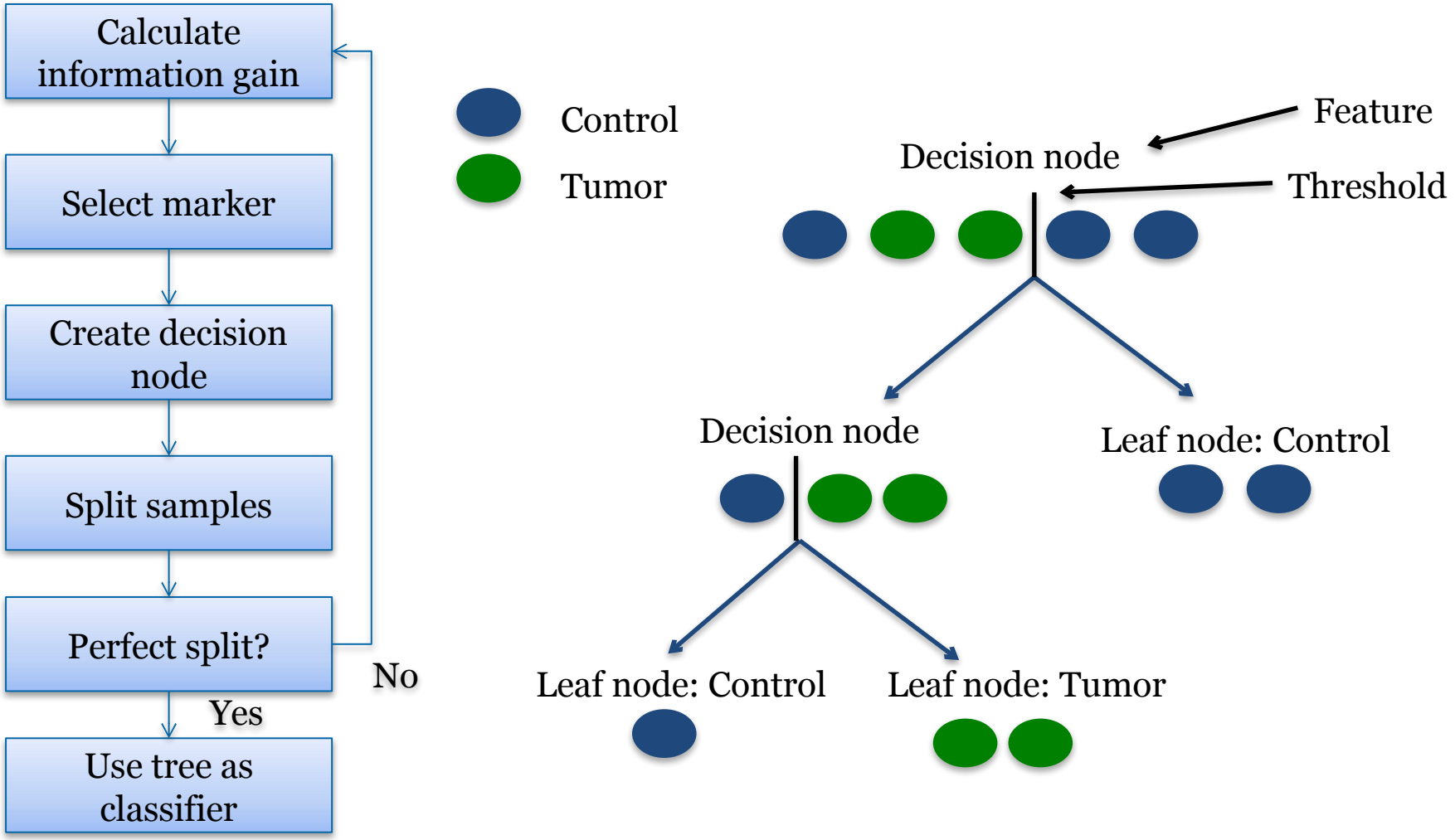
$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$


ROC CURVE
PERFECT CLASSIFIER
BETTER
WORSE
RANDOM CLASSIFIER
TRUE POSITIVE RATE
FALSE POSITIVE RATE
0.0
0.2
0.4
0.6
0.8
1.0

Overfitting





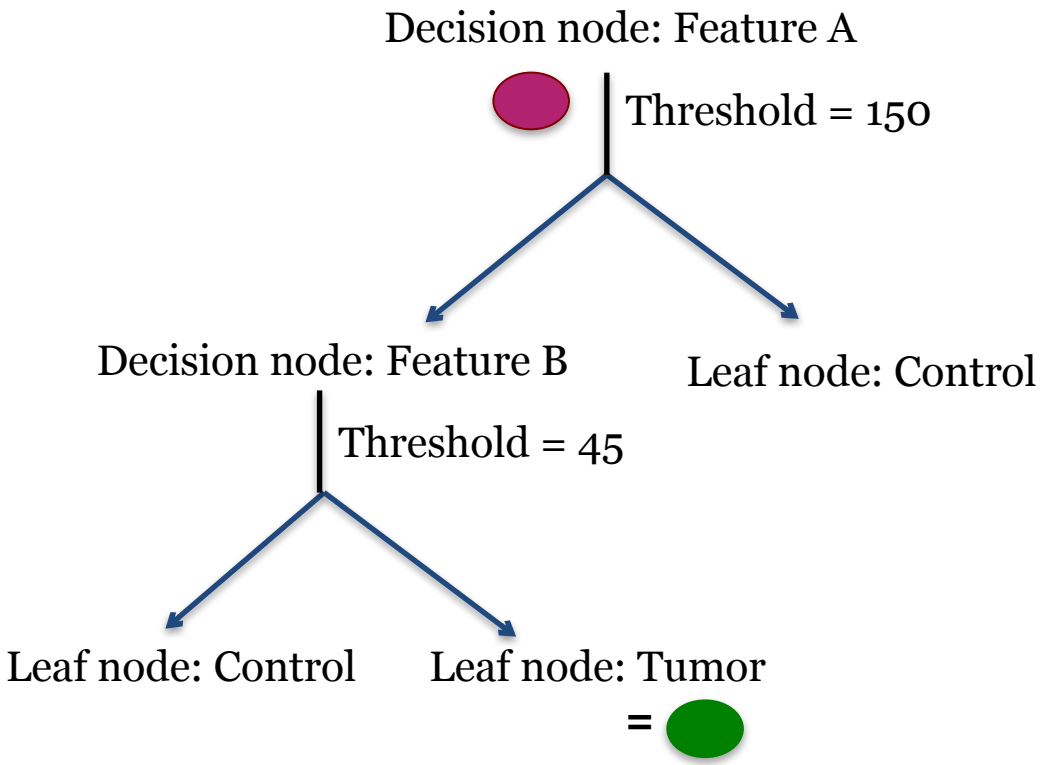

Calculate information gain
Select marker
Create decision node
Split samples
Perfect split?
No
Yes
Use tree as classifier
Control
Tumor
Feature
Threshold
Decision node
Decision node
Leaf node: Control
Leaf node: Control
Leaf node: Tumor

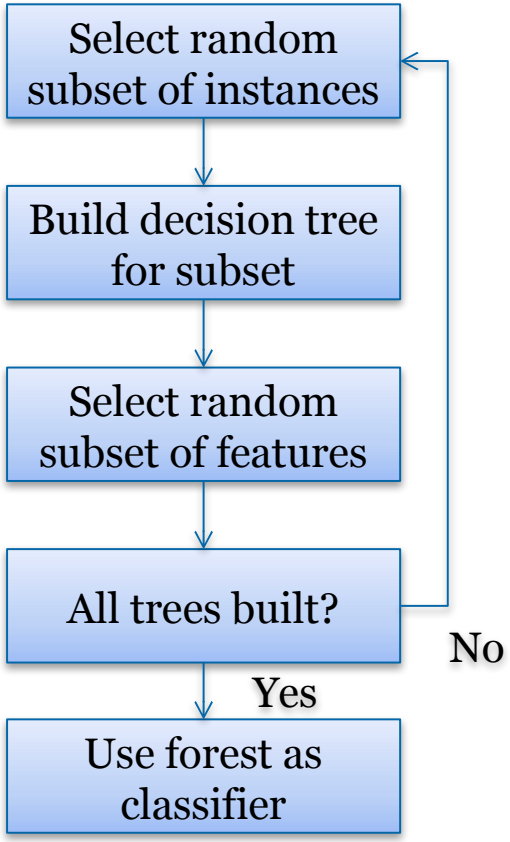
Test instance

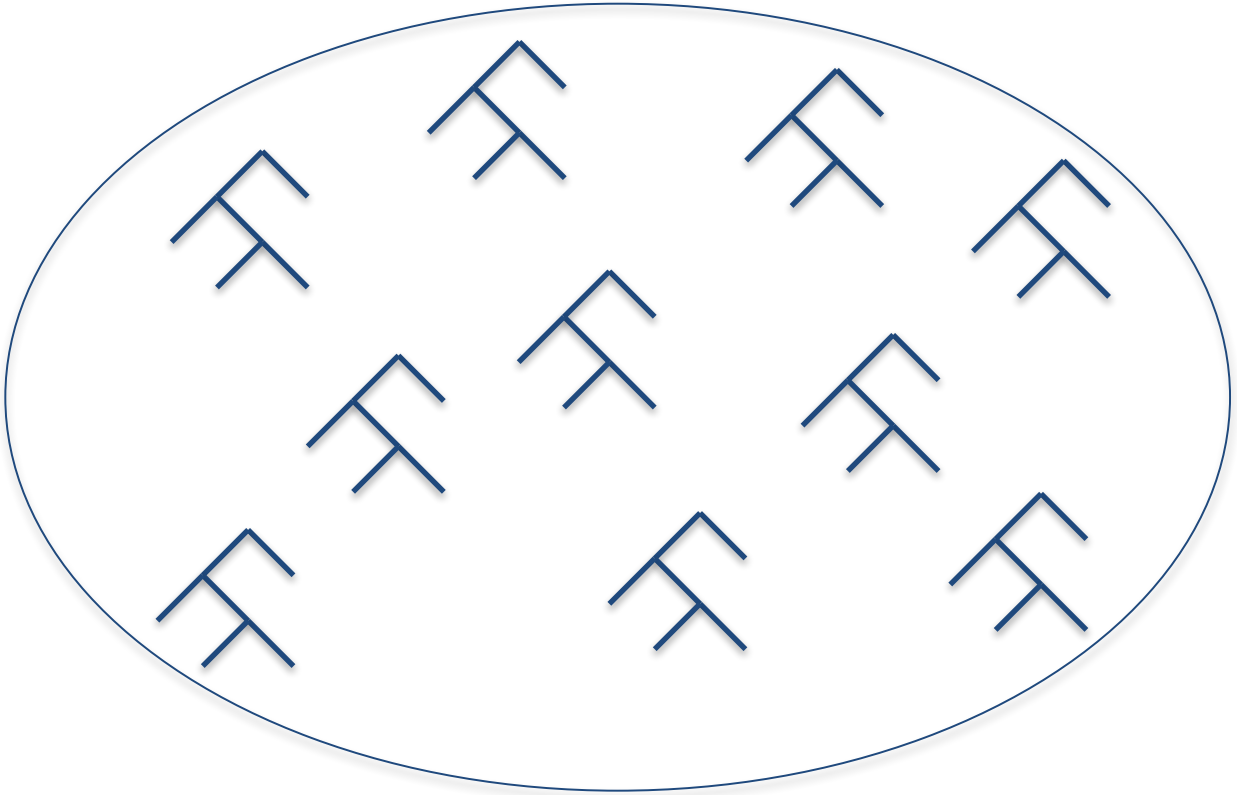
Feature	Value
Feature A	100
Feature B	75

Control

Tumor



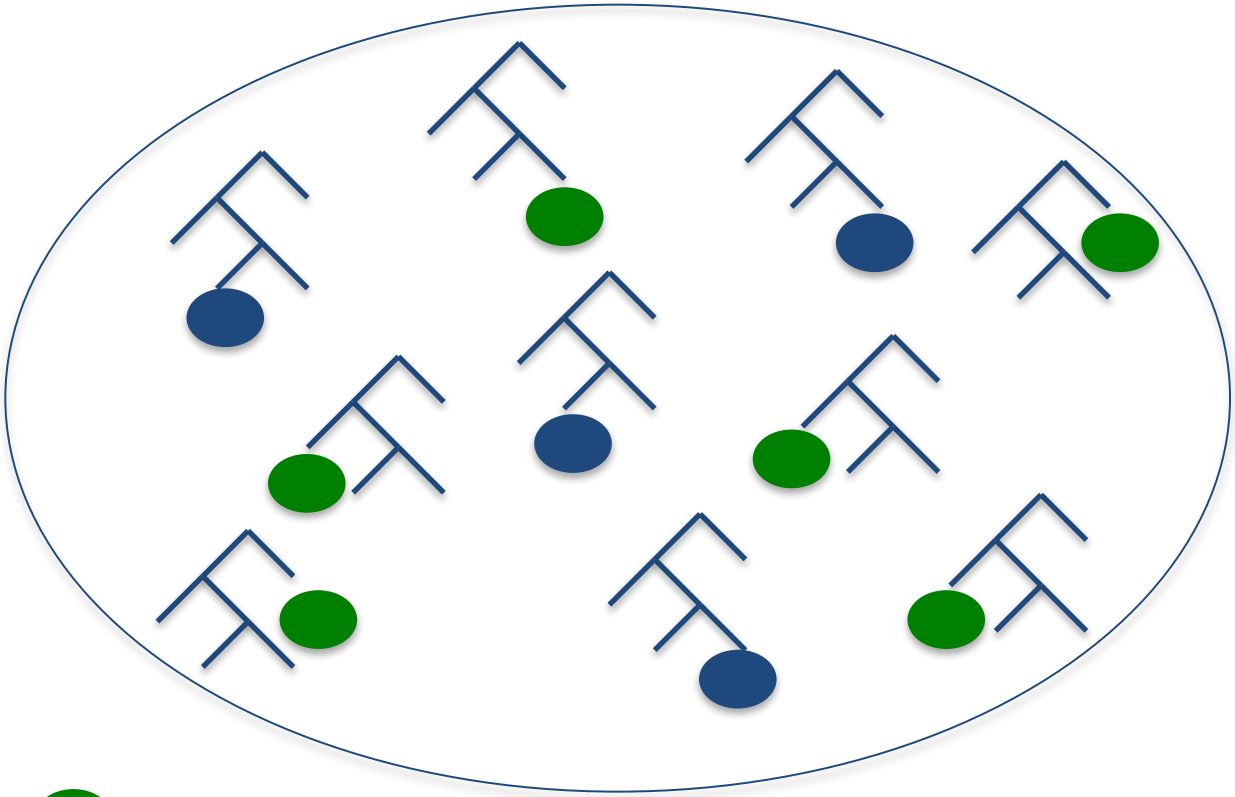




Random forest

Test instance

Feature	Value
Gene A	50
Gene B	30
Gene C	70



Control

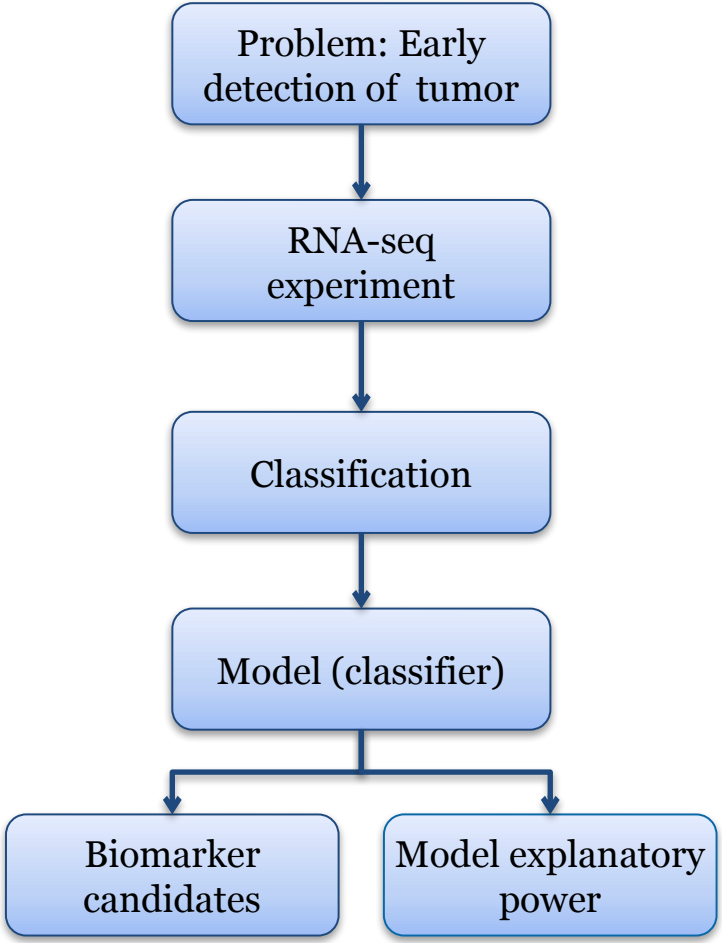
Tumor

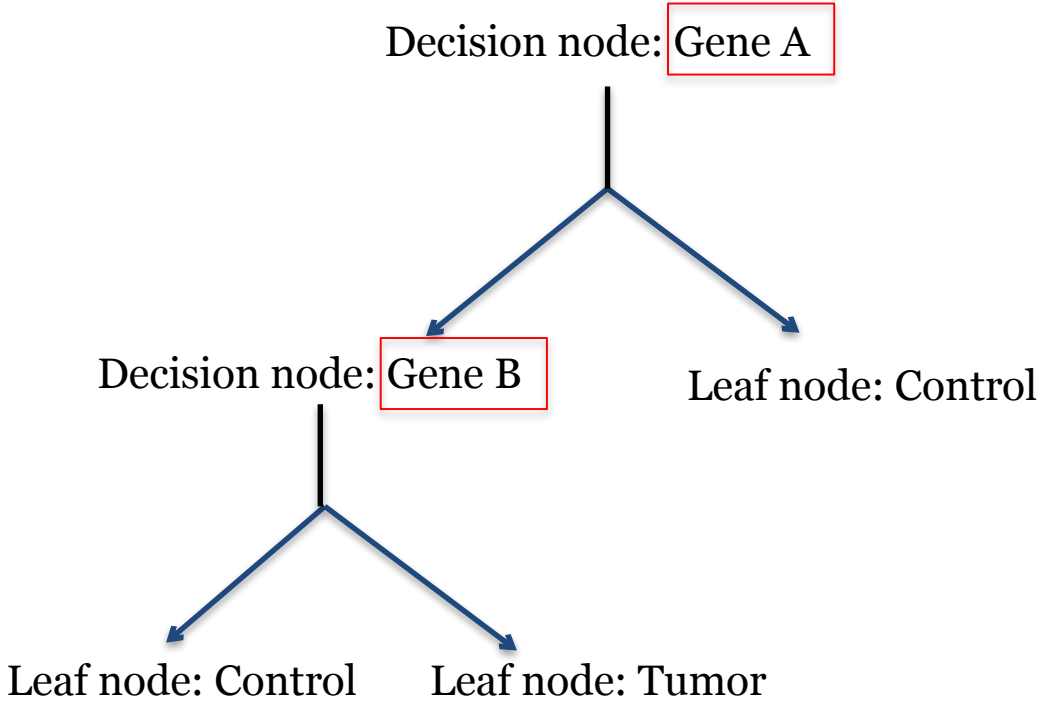
4 votes

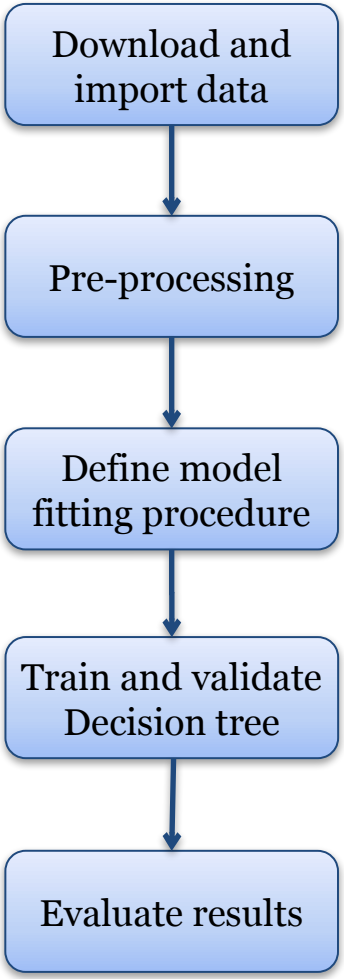
6 votes

=

Random forest


Problem: Early detection of tumor
RNA-seq experiment
Classification
Model (classifier)
Biomarker candidates
Model explanatory power


Decision node: Gene A
Decision node: Gene B
Leaf node: Control
Leaf node: Control
Leaf node: Tumor


Download and import data
Pre-processing
Define model fitting procedure
Train and validate Decision tree
Evaluate results

PREPARE DATA

UNIVERSITY OF SKÖVDE

```
#Continuing after Day 2 slide 20
dds <- DESeqDataSetFromMatrix(as.matrix(countTable),
                              design = ~ 0 + disease + individual,
                              colData = sampleTable)
normCounts <- vst(dds, blind = FALSE)
```

DEFINE CARET FUNCTIONS

UNIVERSITY OF SKÖVDE

```
#Define model fitting procedure
#NB: you will also need the packages statmod and e1071 for this analysis
library(caret)

#New score function
rfSBF$score <- function(x, y){
  sd(x) / mean(x)
}

#New filter function
rfSBF$filter <- function(score, x, y){
  meanLog2CPM <- rowMeans(log2(cpm(countTable) + 1))
  selection <- score > quantile(score, 0.9) & meanLog2CPM > 1
  names(score) %in% names(score)[selection]
}
```

DEFINE CARET FUNCTIONS

UNIVERSITY OF SKÖVDE

```
#Decision tree fit function
dtSBF <- rfSBF
dtSBF$fit <- function(x, y, ...){
  loadNamespace("C50")
  C50::C5.0(x = x, y = y)
}
```

MODEL FITTING

UNIVERSITY OF SKÖVDE

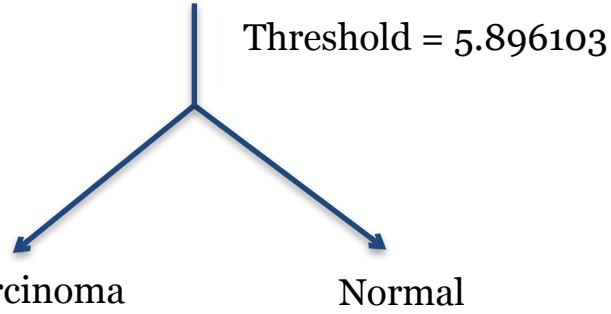
```r
#Create training set
trainingSet <- data.frame(t(assay(normCounts)))

#Train and validate decision tree (C5.0)
dtControl <- sbfControl(
  functions = dtSBF,
  method = "loocv",
  saveDetails = TRUE,
  verbose = FALSE)
dtModel <- sbf(
  trainingSet,
  sampleTable$disease,
  sbfControl = dtControl)

#Show model
summary(dtModel$fit)
```

MODEL FITTING

```
Decision tree:

OTOP2 <= 5.896103: carcinoma (12)
OTOP2 > 5.896103: normal (6)


Evaluation on training data (18 cases):

         Decision Tree
       ----------------
       Size      Errors

          2    0( 0.0%)   <<


        (a)   (b)    <-classified as
       ----  ----
         12          (a): class carcinoma
                6    (b): class normal
```

MODEL FITTING

```
OTOP2 <= 5.896103: carcinoma (12)
OTOP2 > 5.896103: normal (6)
```



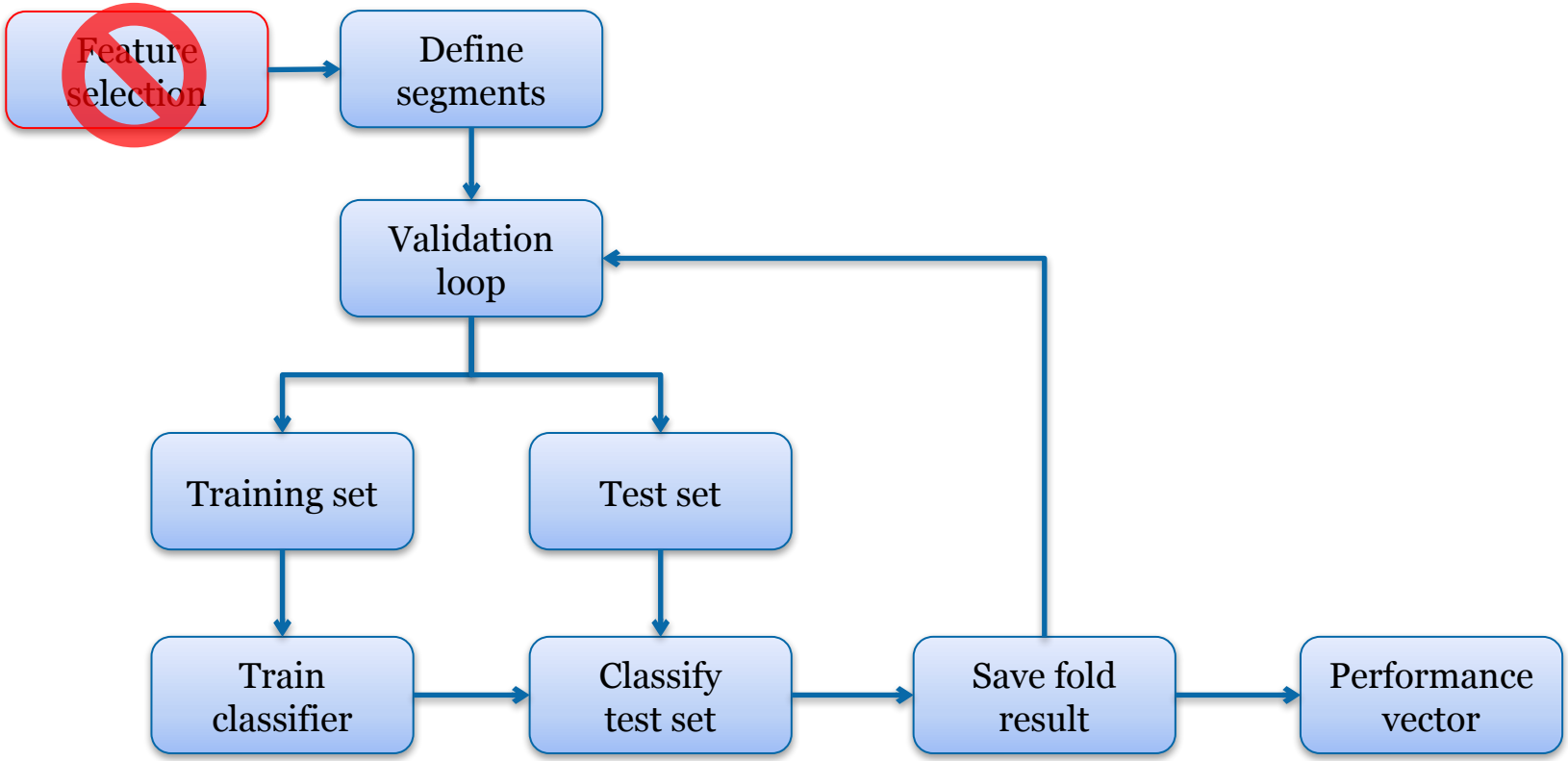
MODEL VALIDATION

```
#Print confusion matrix from validation
confusionMatrix(dtModel$pred$predictions$pred, sampleTable$disease)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  carcinoma normal
  carcinoma        11      0
  normal            1      6


               Accuracy : 0.9444

     95% CI : (0.7271, 0.9986)
    Mcnemar's Test P-Value : 1.000000


    Sensitivity : 0.9167
    Specificity : 1.0000
    Pos Pred Value : 1.0000
    Neg Pred Value : 0.8571
    Prevalence : 0.6667
    Detection Rate : 0.6111
  Detection Prevalence : 0.6111
 Balanced Accuracy : 0.958
 'Positive' Class : carcinoma
```

Predicted class \ Class vector (truth)	Carcinoma	Normal
Carcinoma	TP = 11	FP = 0
Normal	FN = 1	TN = 6

Cross validation with feature selection



Cross validation with feature selection