ML4Bioinfor-R

Machine Learning for Bioinformatics using R

Gang Chen
chengang@bgitechsolutions.com

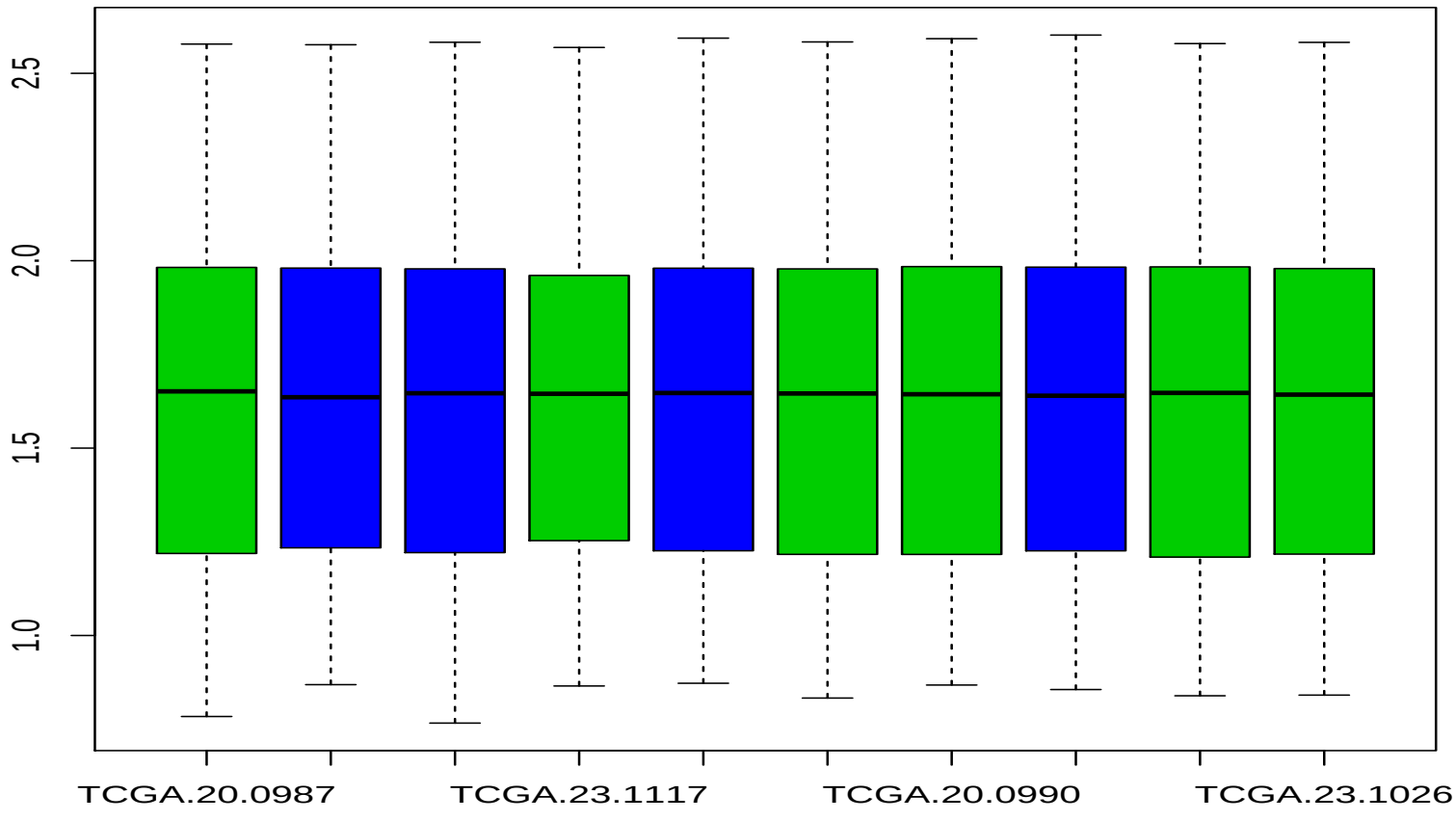
November 22, 2013

华大科技
BGI·tech

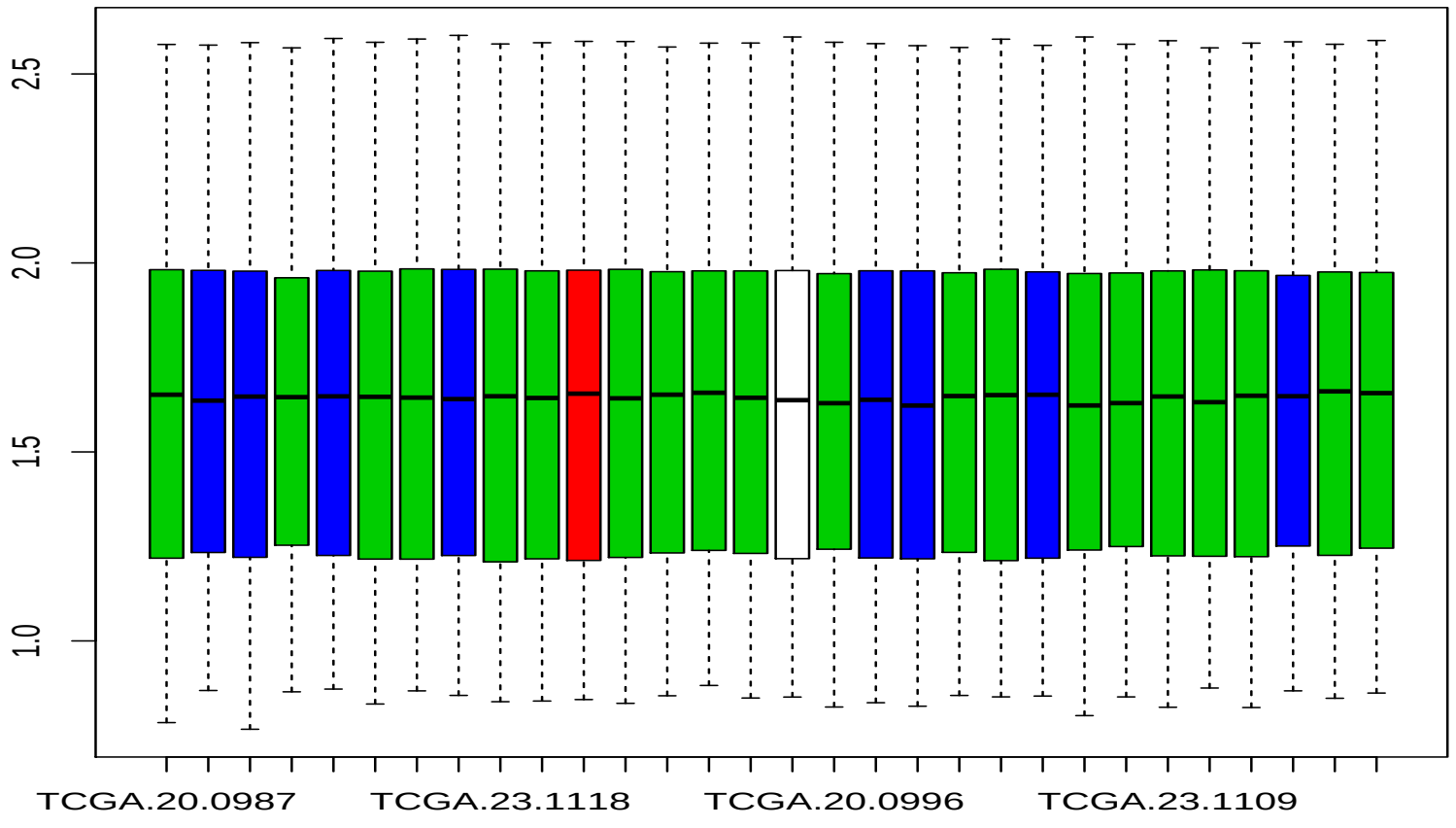
Outline

Next

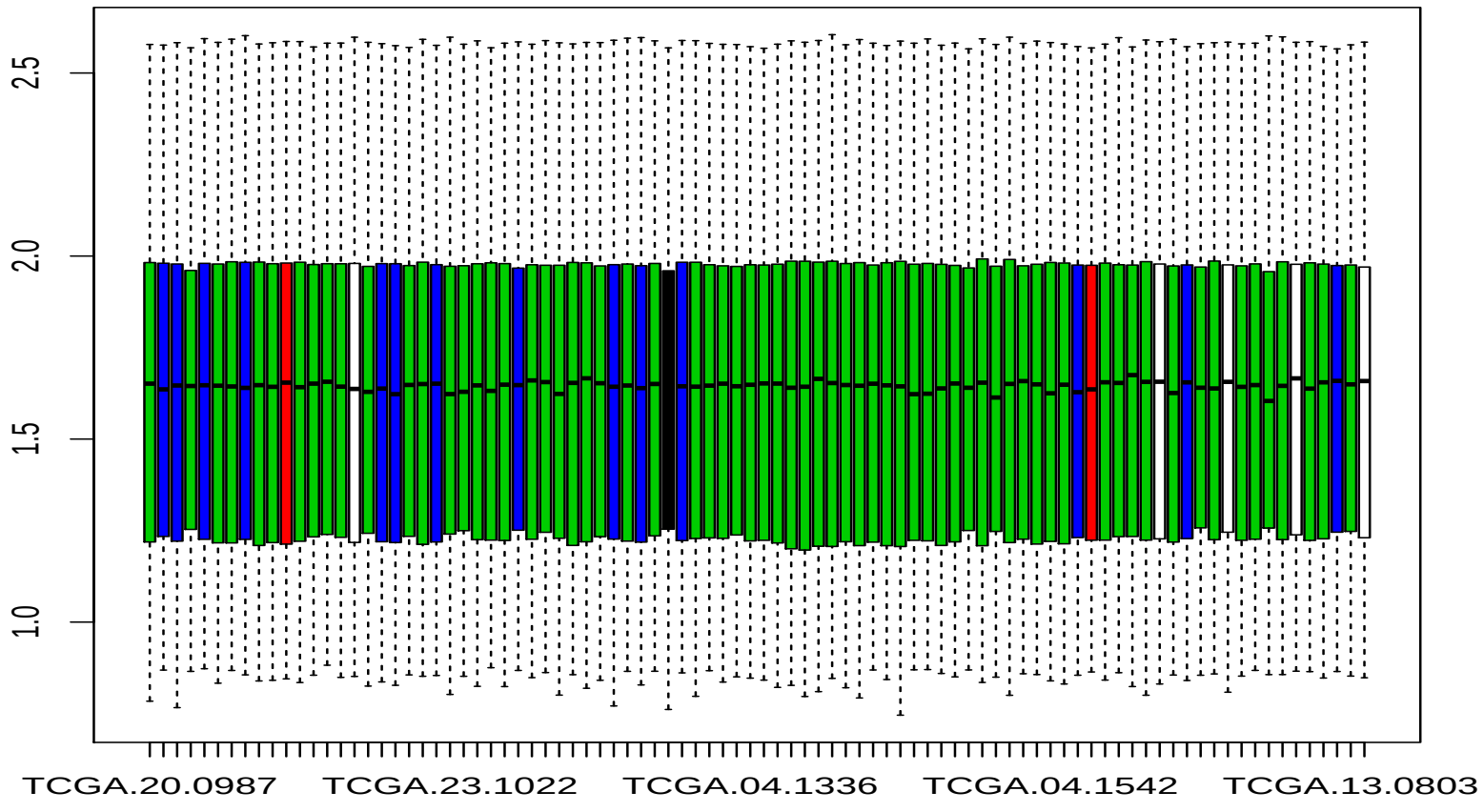
Identification of Biomarker 1



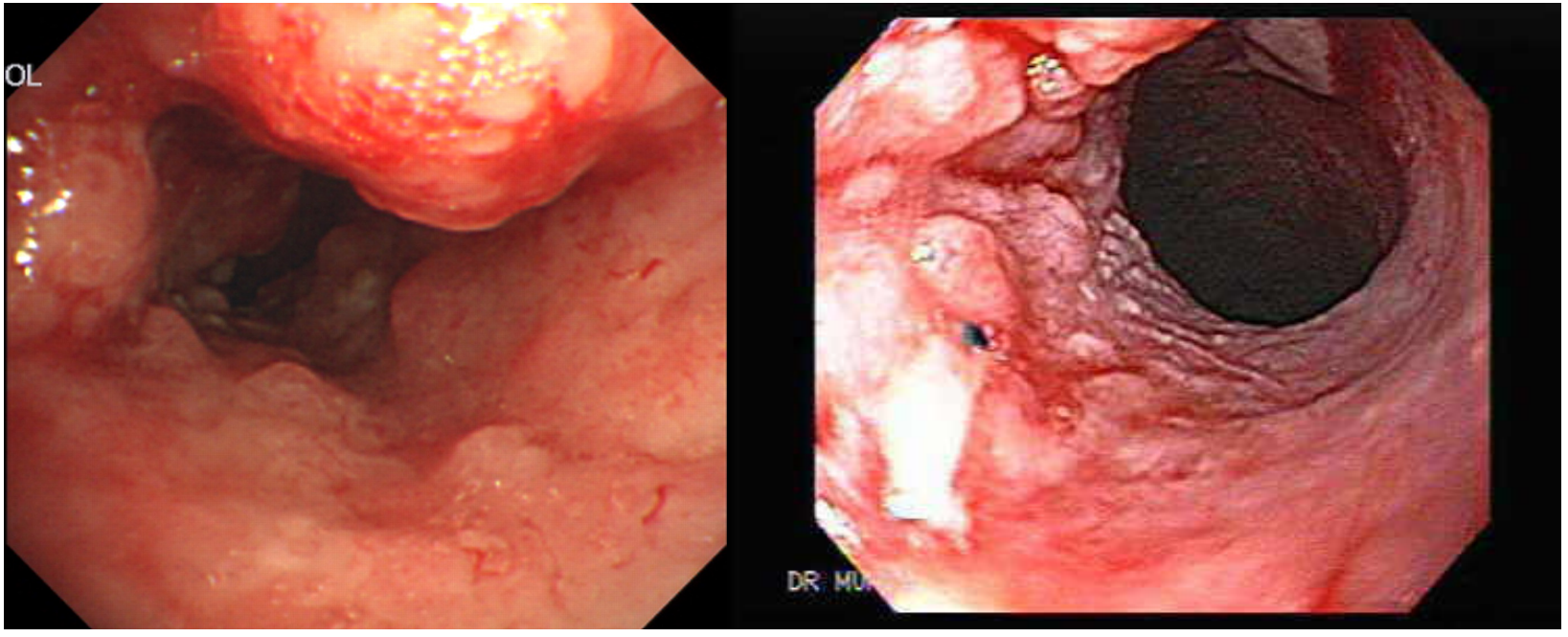
Identification of Biomarker 2



Identification of Biomarker 3

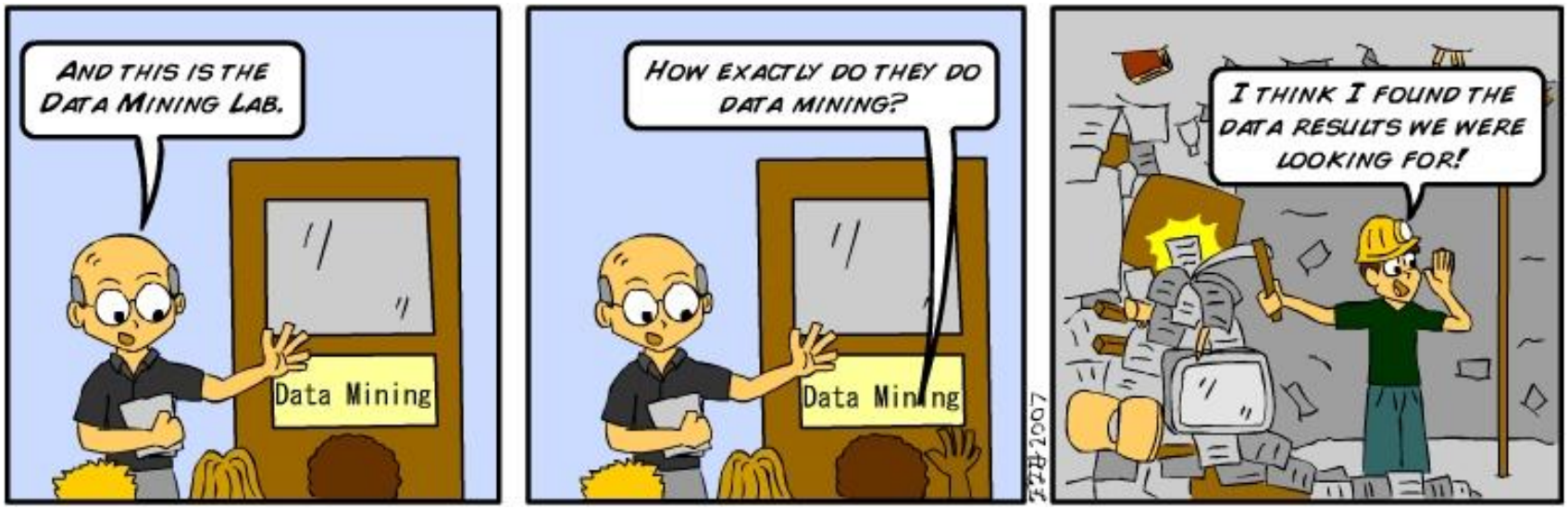


华大科技
BGI tech



How to identify tumor from genomics perspective?

The answer is Data Mining.


AND THIS IS THE DATA MINING LAB.
Data Mining
HOW EXACTLY DO THEY DO DATA MINING?
Data Mining
I THINK I FOUND THE DATA RESULTS WE WERE LOOKING FOR!

What is Data Mining?

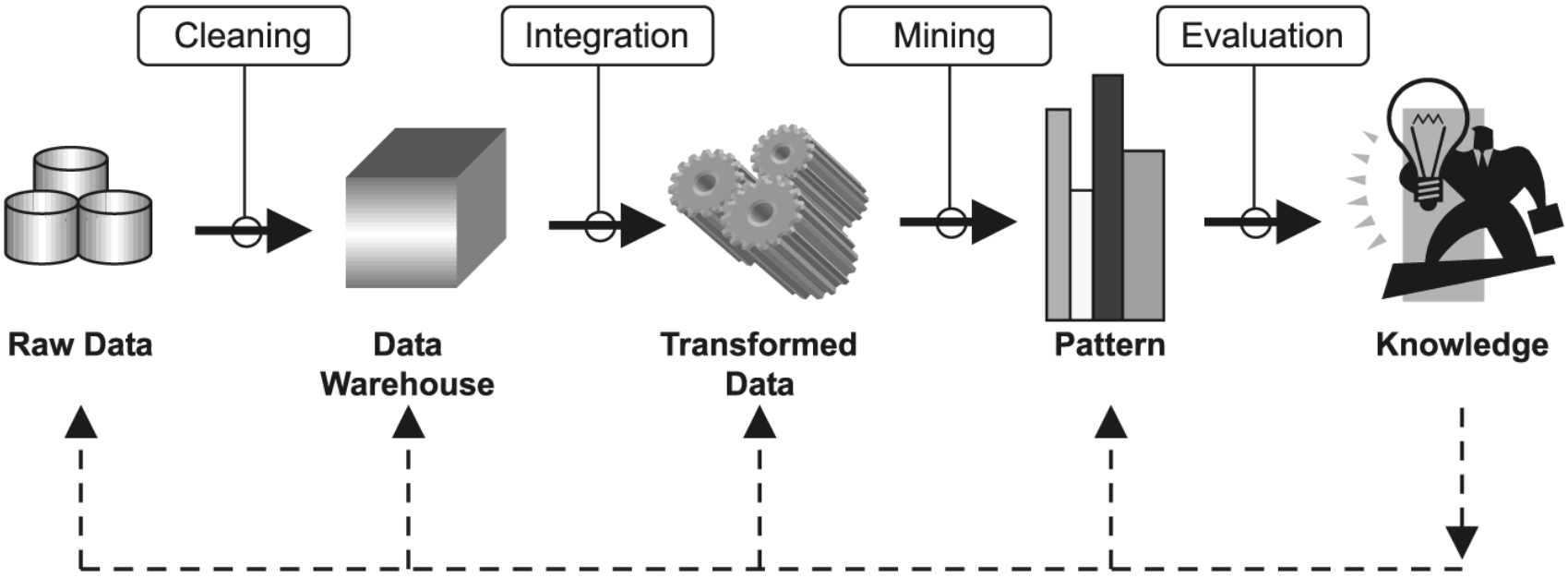
Pang-Ning Tan, Introduction to Data Mining

Data Mining is the process of automatically discovering useful information in large data repositories.

Knowledge Discovery in Databases

Data Mining is an integral part of knowledge discovery in databases(KDD).

Data Mining and KDD



Machine Learning and Bioinformatics

Biological Experiments
Microarray
Sequencing
Mass Spectrum
...

→

Preprocssing
base calling
alignment
variants
...

→

Data Mining
Classification
Regression
Feature Engineering
...

→

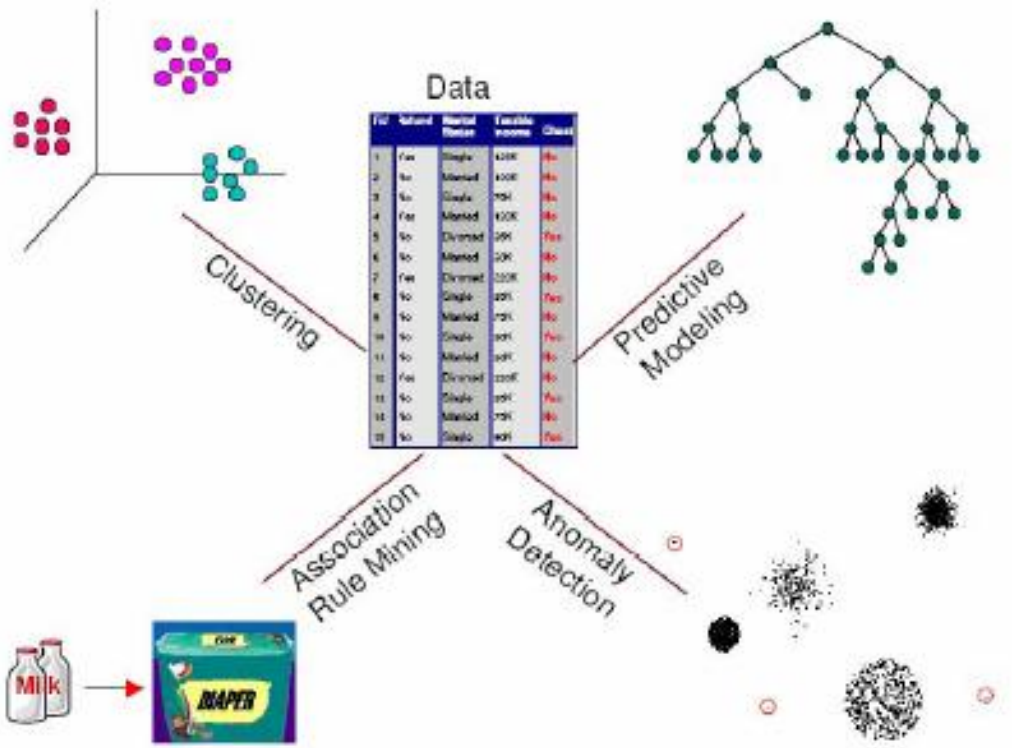
Biological Knowledge

Traditional Data Analysis

Motivations

- Scalability
- High Dimensionality
- Heterogeneous and Complex Data
- Data Ownership and Distribution
- Non-traditional analysis

Data Mining Tasks



Data Mining and Machine Learning

Machine Learning

Machine learning provides the technical basis of data mining.
---Data Mining: Practical Machine Learning Tools and Techniques

Schedule

Schedule

- Introduction
- Unsupervised Learning: Clustering
- Supervised Learning: Classifition
- Feature Learning: Feature Selection
- Discussion

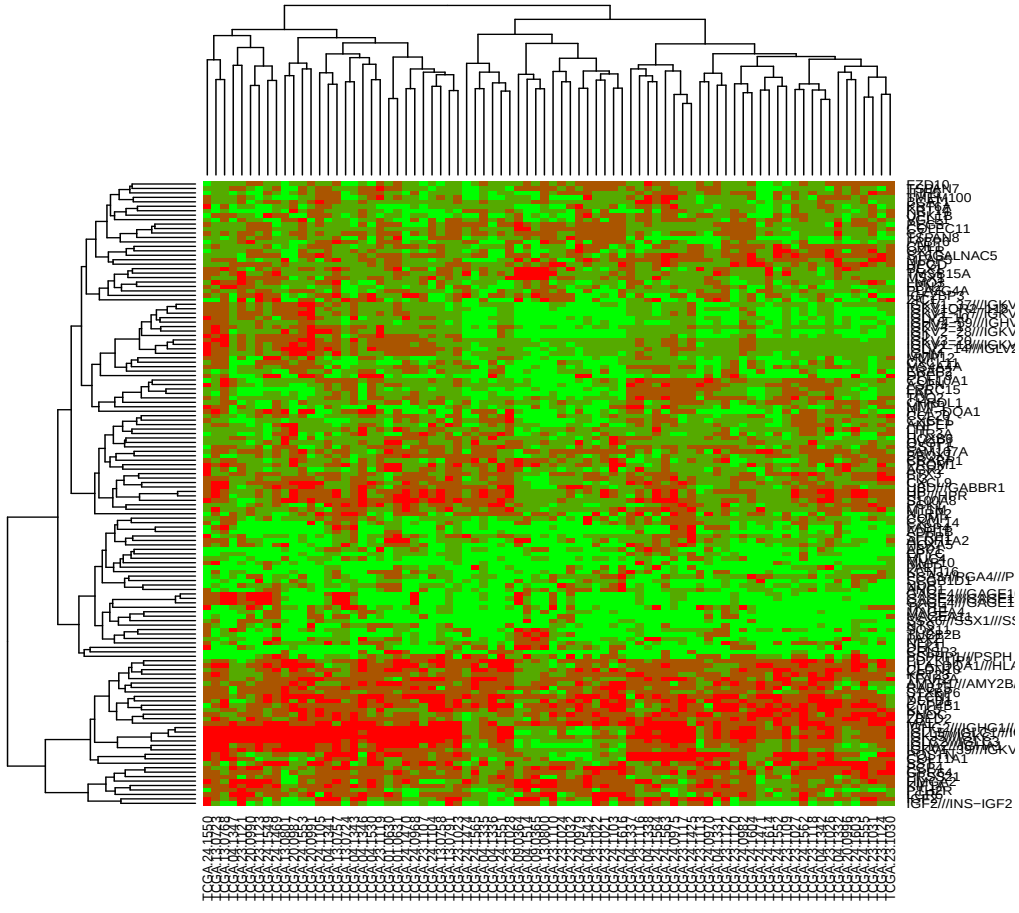
Softwares

Softwares

- R: R is an free platform for data analysis and visuaztion.
- R packages:
  - **e1071** SVM
  - **caret** Feature selection
  - **C50** Decision tree
  - **curatedOvarianData** Data used in this lecture.
- R IDE: Emacs + ESS, Vim + R-Plugin, RStudio for local computer.
- R Studio Server: http://192.168.224.109:8787/

Next

Heat Map



Unsupervised Learning: Clustering

What is clustering?

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).
---Wikipedia

Applications

Applications of Clustering

- Clustering for Understanding
  - Biology
  - Information Retrieval
  - Climate
  - Psychology and medicine
  - business
  - ……
- Clustering for Utility
  - Summarization
  - compression
  - Efficiently finding nearest neighbors
  - ……

Common Clustering Methods

Clustering Methods

- Density-based clustering
- K-means
- Hierarchical Clustering
- Semi-supervised clustering
- ……

Unsupervised Learning in Bioinformatics

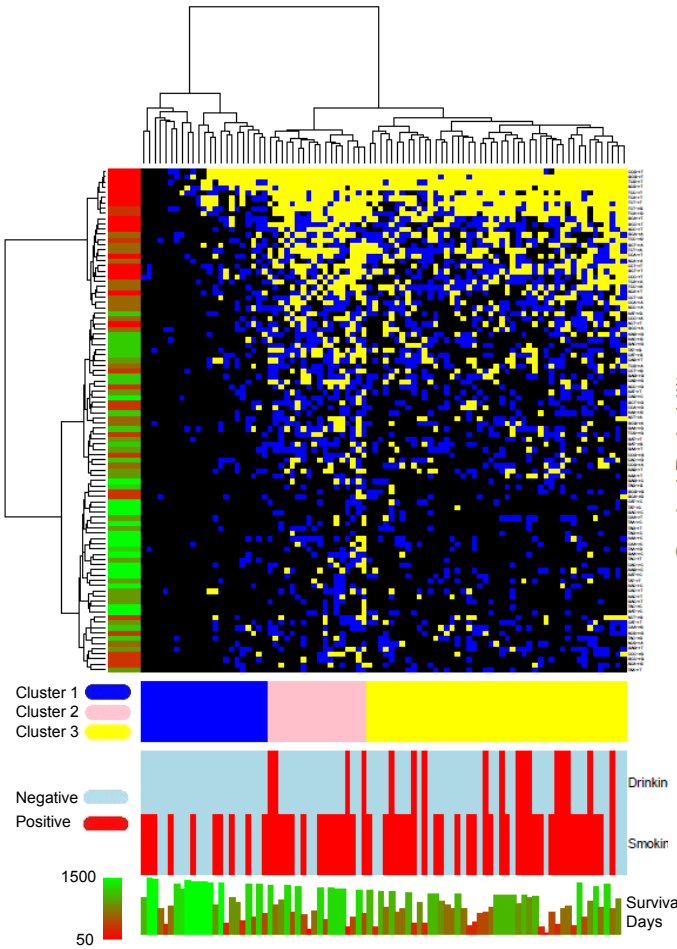
- Discovery of tumor subtypes by clustering gene expression, CNV, miRNA or integrated data.
- Clonal evolution analysis of tumor
- Mutation spectrum clustering
- Pathway or functional annotation based clustering
- Graph clustering for identification of protein functional module or protein complex
- Clustering metagenomic sequences
- Metabolomics
- ……

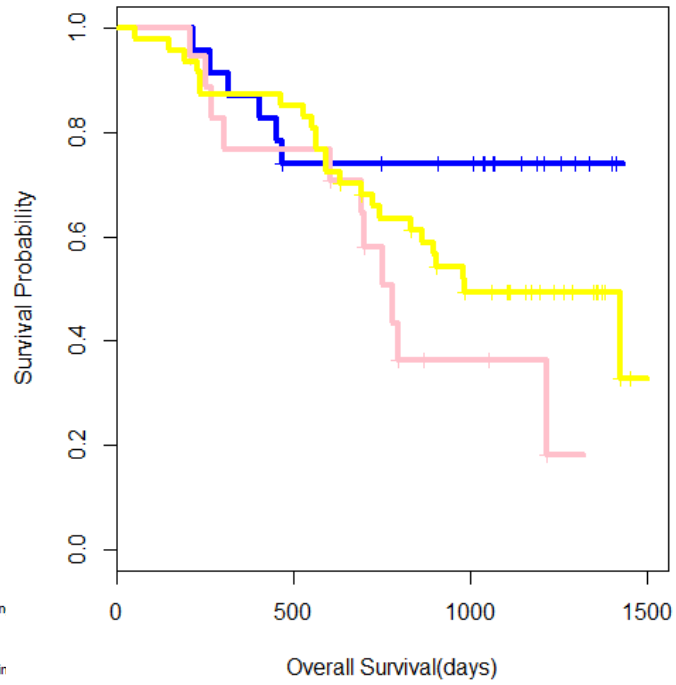
Hierarchical Clustering

Steps

- Calculating distance between individuals
- Combine closest individuals (optional, recaculate distance)
- Visualization and annotation

Clustering in Bioinformatics





Hierarchical Clustering in R

```
help(dist)
help(hclust)
help(heatmap)
```

Distance Calculation

```
dist(x, method = "euclidean",
diag = FALSE,
upper = FALSE,
p = 2)
```

Distance Calculation

```
exprDist = dist(t(subdata))
exprDist
```

```
##              TCGA.20.0987 TCGA.23.1031 TCGA.24.0979 TCGA.23.1117
## TCGA.23.1031        6.765
## TCGA.24.0979        6.495        4.846
## TCGA.23.1117        5.789        5.305        5.726
## TCGA.23.1021        6.097        4.794        4.873        5.458
## TCGA.04.1337        7.009        5.767        6.460        4.511
## TCGA.20.0990        6.429        5.265        5.944        5.618
## TCGA.23.1032        6.970        5.398        5.288        6.379
## TCGA.23.1118        5.517        5.133        4.765        4.597
## TCGA.23.1026        5.392        5.576        5.815        4.420
## TCGA.20.0991        5.388        7.008        7.103        6.001
## TCGA.24.1103        6.648        5.375        4.458        6.125
## TCGA.24.0982        5.699        4.882        5.616        4.821
## TCGA.23.1119        5.628        6.059        6.169        5.151
## TCGA.23.1028        5.135        5.508        5.487        5.069
## TCGA.04.1341        6.312        5.080        6.328        4.921
## TCGA.20.0996        6.647        5.027        5.337        5.342
## TCGA.24.1104        4.447        5.653        5.785        4.994
## TCGA.23.1107        5.415        5.980        6.480        5.255
## TCGA.23.1120        5.548        5.096        4.268        5.123
## TCGA.23.1030        6.042        4.712        5.769        5.278
## TCGA.04.1342        5.516        4.932        5.259        3.878
## TCGA.23.1022        6.616        4.723        4.901        5.594
## TCGA.24.1105        5.625        5.109        6.099        4.709
## TCGA.23.1109        6.094        5.077        5.624        3.991
## TCGA.23.1121        6.466        6.528        7.355        6.291
```

Clustering

```
hclust(d, method = "complete", members = NULL)
```

Clustering

```
exprDist = dist(t(subdata))
exprClust = hclust(exprDist)
exprClust

##
## Call:
## hclust(d = exprDist)
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 80
```

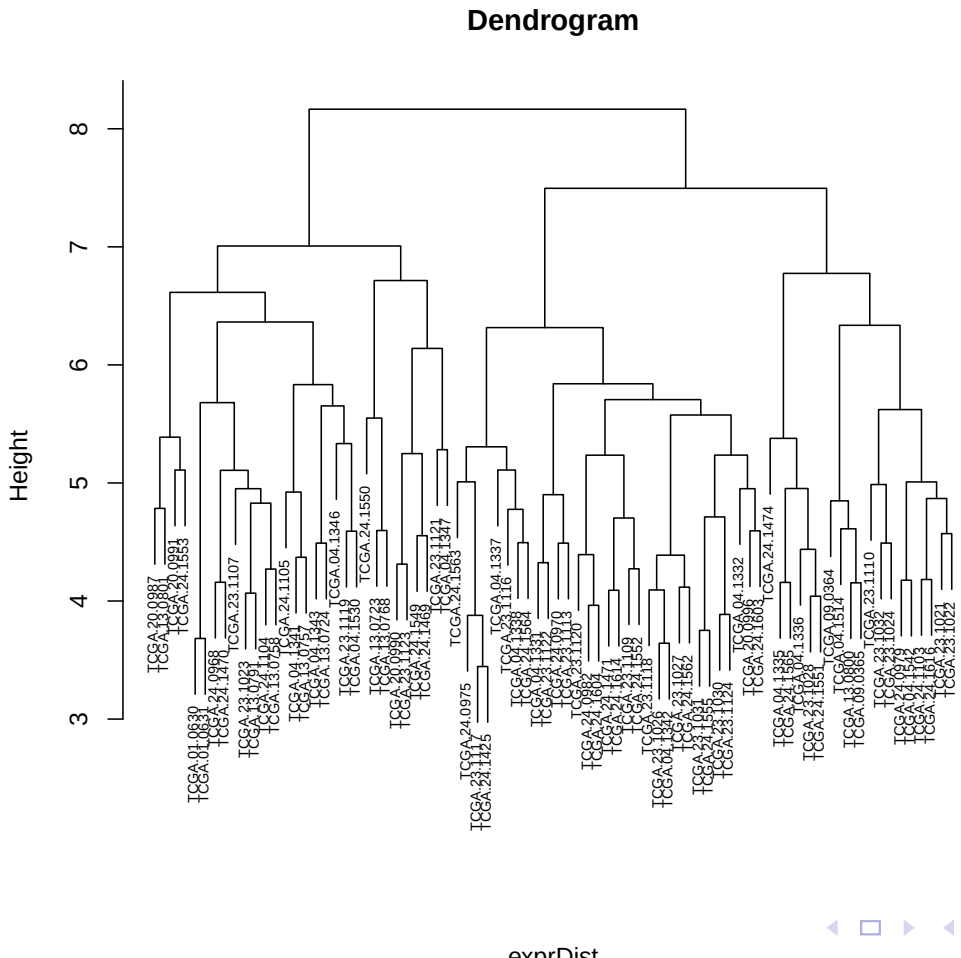
Visualization

Visualization

- Dendrogram: plot.hclust
- Heat Map: heatmap

Dendrogram

```
plot(exprClust, cex = 0.6, main = "Dendrogram")
```

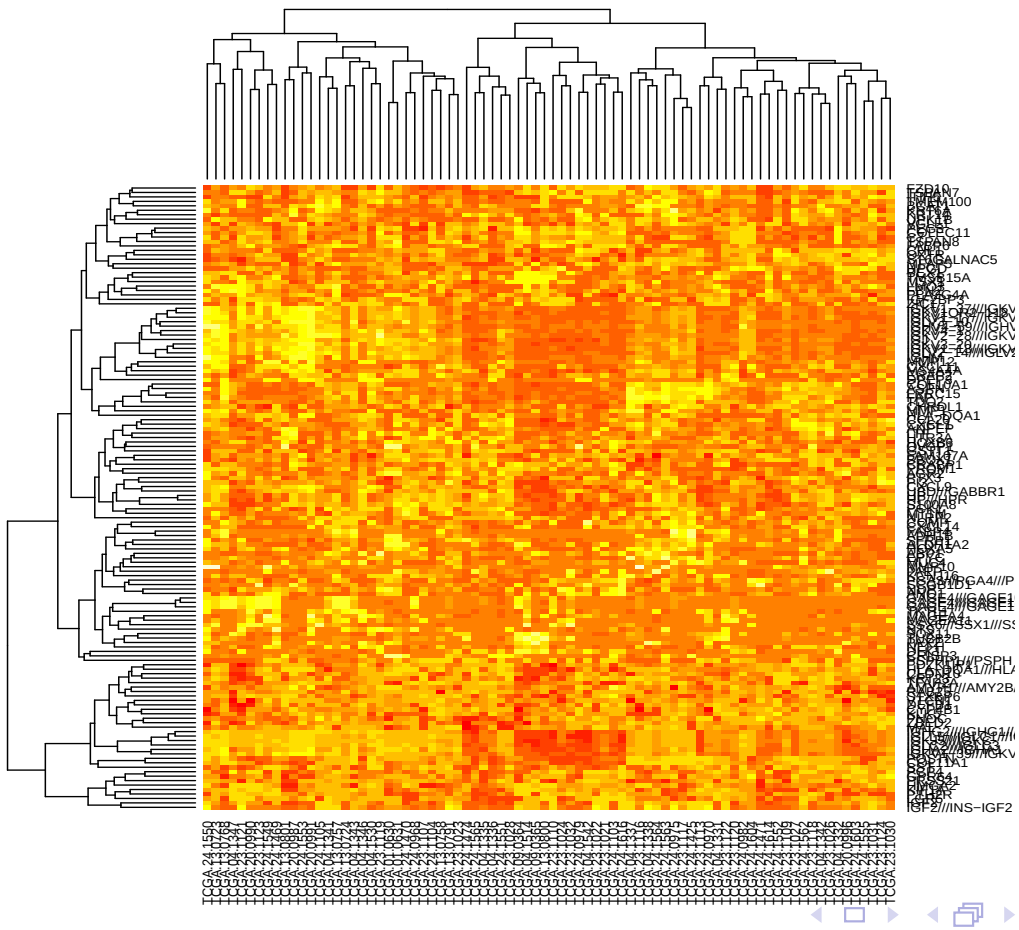


plot.hclust

```
plot(x, labels = NULL,
hang = 0.1,
axes = TRUE,
frame.plot = FALSE,
ann = TRUE,
main = "Cluster Dendrogram",
sub = NULL,
xlab = NULL, ylab = "Height", ...)
```

Heat Map

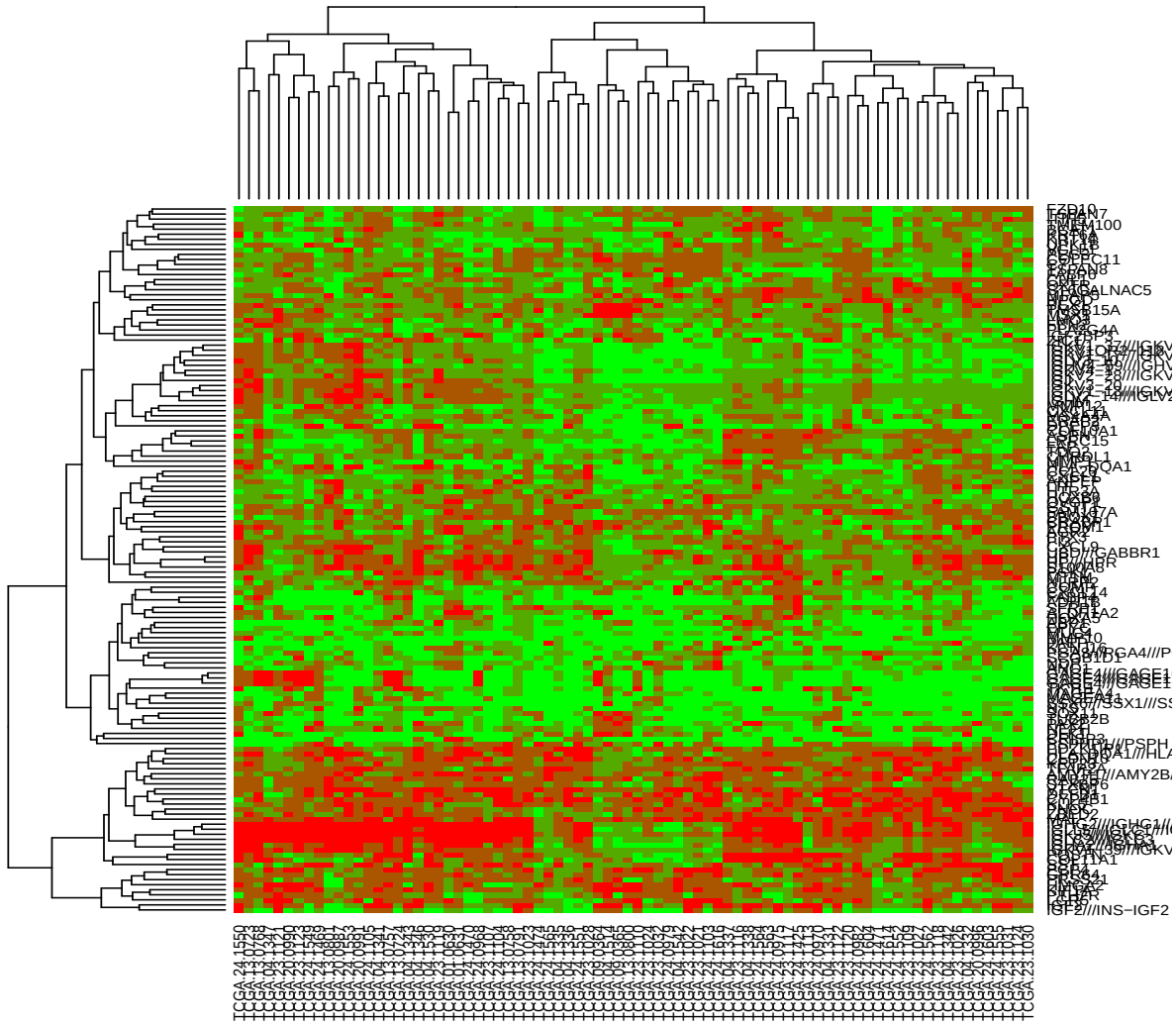
```
heatmap(subdata)
```



heatmap

```
heatmap(x, Rowv = NULL, Colv = if(symm)"Rowv" else NULL,
distfun = dist, hclustfun = hclust,
reorderfun = function(d, w) reorder(d, w),
add.expr, symm = FALSE, revC = identical(Colv, "Rowv"),
scale = c("row", "column", "none"), na.rm = TRUE,
margins = c(5, 5), ColSideColors, RowSideColors,
cexRow = 0.2 + 1/log10(nr), cexCol = 0.2 + 1/log10(nc),
labRow = NULL, labCol = NULL, main = NULL,
xlab = NULL, ylab = NULL,
keep.dendro = FALSE, verbose = getOption("verbose"), ...)
```

How to read Heat Map



Suumary

Clustering

- Clustering is widely used in bioinformatics
- Clustering can be implemented by using built-in functions of R
- Clustering can be visualized as heat map and dendogram

Next

Supervised Learning: Classification

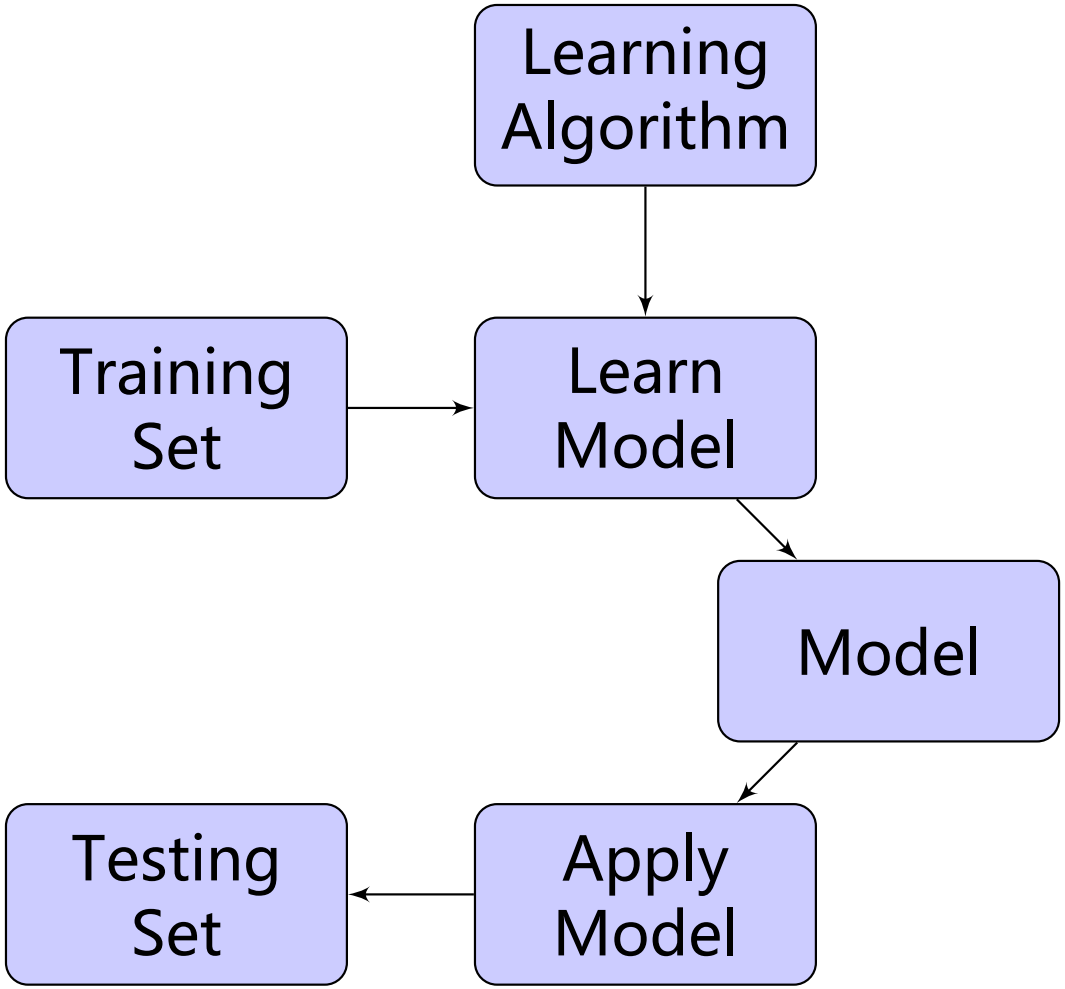
Classification

Assigning objects to one of several predefined categoriies.

Definition

Classification is the task of learning a **target function** f that maps each attribute set x to one of the predefined class labels y.

How to solve a classification problem?



General approach for building a classification model

Evaluation

Confusion Matrix for a 2-class problem

	Prediction=1	Prediction=0
Class=1	f_{11}	f_{10}
Class=0	f_{01}	f_{00}

Evaluation

Accuracy and Error Rate

$$\begin{aligned}\text{Accuracy} &= \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \\ &= \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{00} + f_{01}} \\ \text{ErrorRate} &= \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} \\ &= \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{00} + f_{01}}\end{aligned}$$

Classification in Bioinformatics

Applications

- Classification of diseases, especially cancer.
- Prediction of clinical outcome.
- Prediction of the function of gene or proteins.
- Prediction of the structure of proteins.
- ……

Objective

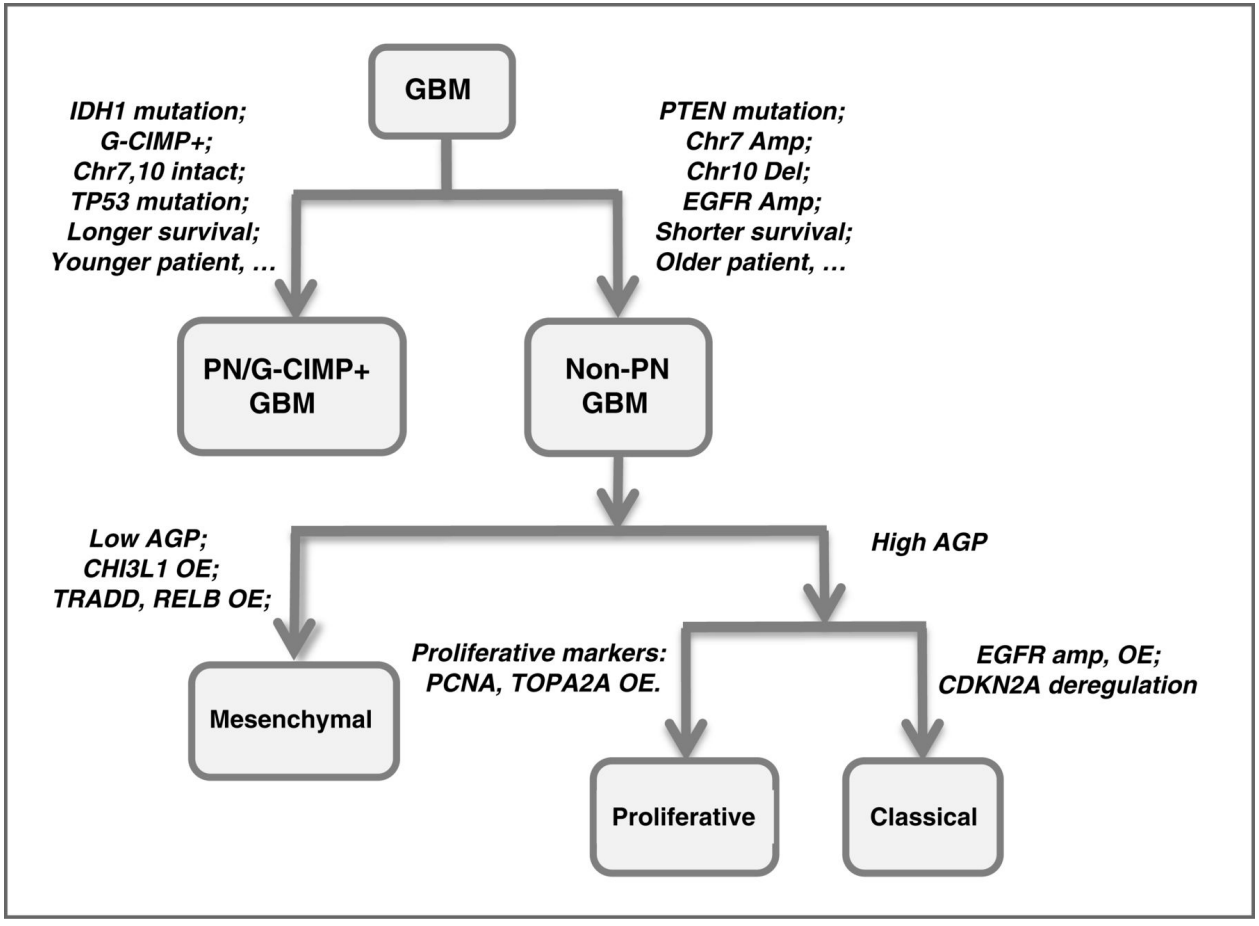
Two Objectives

1. To build accuate classifiers or predictors
2. To derive inferences from the results obtained

Challanges

- data incocnsistency and missing values
- noise
- normalization
- Deimensionality reduction

Decision Tree



Concepts

Types of Nodes

root node :no incoming edges and zero or more outgoing edges.

internal nodes : has exactly one incoming edge and two or more outgoing edges.

leaf or terminal nodes : has exactly one incoming edge and no outgoing edges.

How to build a Decision Tree?

Hunt's Algorithm

Let D_t be the set of training records that are associated with node t and $y = y_1, y_2, \ldots, y_c$ to be the class labels.

1. if all the records in D_t belong to the same class y_t, then t is a leaf node labeled as y_i.
2. if D_i contains records that belong to more than one class, an **attribute test condition** is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition and the records in D_t are distributed to the chirldren based on the outcomes. The algorithm is then recursively applied to each child node.

Algorithm A skeleton decision tree induction algorithm
TreeGrowth(E, F)

if stopping_cond(E, F) = true **then**
    leaf $=$ createNode();
    leaf.label $=$ Classify(E);
    return return leaf
else
    root = createNode();
    roor.test_cond = find_best_split(E, F);
    let V $=$ v|vis a possible outcome of root.test_cond **for** each v $\in$ V
    do **do**
        E_v $=$ e|root.test_cond(e) $=$ vande $\in$ E;
        child $=$ TreeGrowth(E_v, F);
        add child as descendent of root and label the edge
        (root $\rightarrow$ child) as v
    end
end
return root

Overfitting

Reasons

- Noise
- Lack of Representative samples

Occam's Razor

Given two models with the same generalization errors, the simpler model is prefered over the more complex model.

Handling Overfitting in Decision Tree

Prepruning, Early Stopping Rule

Halt tree-growing algorithm before generating a fully grown tree:

- the observed gain falls below a certain threshold.
- the number of records in a node is less than a certain threshold.

Post-pruning

The decision tree is initially grown to its maximum size, then perform pruning.

Decision Tree in R

```
library(C50)
```

C5.0 Algorithm in R

```
library(C50)
C5.0(x, y)
# x is a data frame or a matrix of records

# y is a factor vector of class lables
```

C5.0Control

```
C5.0Control(subset = TRUE,

            bands = 0,
            winnow = FALSE,
            noGlobalPruning = FALSE,
            CF = 0.25,
            minCases = 2,
            fuzzyThreshold = FALSE,
            sample = 0,
            seed = sample.int(4096, size = 1) - 1L,
            earlyStopping = TRUE,
            label = "outcome")
```

Decision Tree on Gene Expression Data

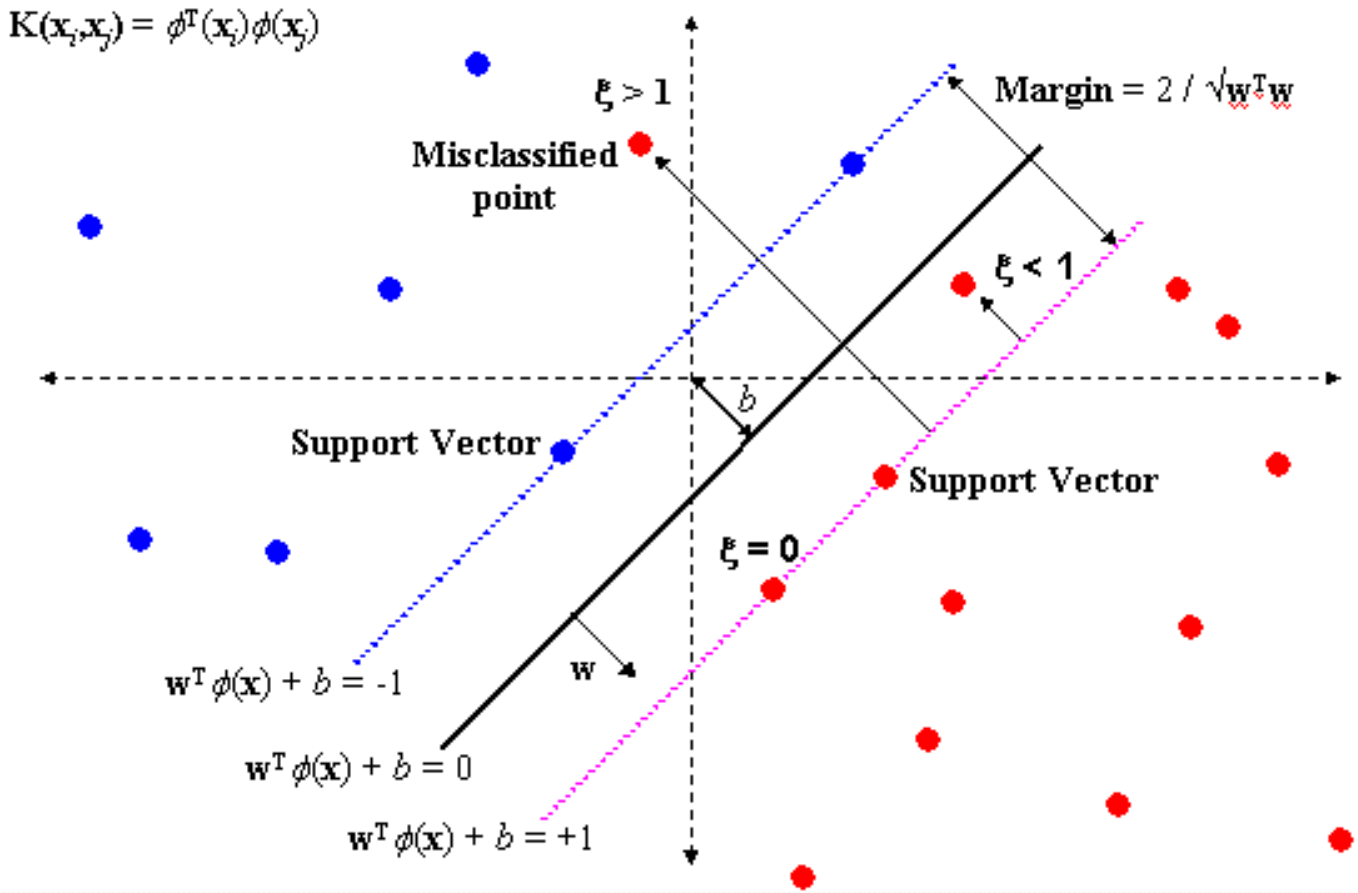
```
library(C50)
sgrade = TCGA_eset$summarygrade
exprData = t(log(exprs(TCGA_eset)[1:200, !is.na(sgrade)]))
sgrade = as.factor(sgrade[!is.na(sgrade)])
dt = C5.0(exprData[1:400, ], sgrade[1:400])
```

```
summary(dt)


##
## Call:
## C5.0.default(x = exprData[1:400, ], y = sgrade[1:400])
##
##
## C5.0 [Release 2.07 GPL Edition]   Fri Nov 22 13:46:47 2013
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 400 cases (201 attributes) from undefined.data
##
## Decision tree:
##
## ACVR1 <= 1.902631: high (65/1)
## ACVR1 > 1.902631:
## :...ABCC4 > 2.132082:
##     :...ABI2 > 1.761186: high (65)
##     :   ABI2 <= 1.761186:
##     :   :...ADAM7 <= 1.048816: low (2)
##     :       ADAM7 > 1.048816: high (7)
##     ABCC4 <= 2.132082:
##     :...ACSBG1 <= 1.065441:
##         :...ADAMTS20 > 1.095798:
##         :   :...ABCA1 <= 2.035188: high (11)
##         :   :   ABCA1 > 2.035188: low (6/1)
##         :   ADAMTS20 <= 1.095798:
##         :   :...ACTR5 > 1.460658: high (82)
##         :       ACTR5 <= 1.460658:
##         :       :...ACBD4 <= 1.408459: high (4)
##         :           ACBD4 > 1.408459: low (2)
```

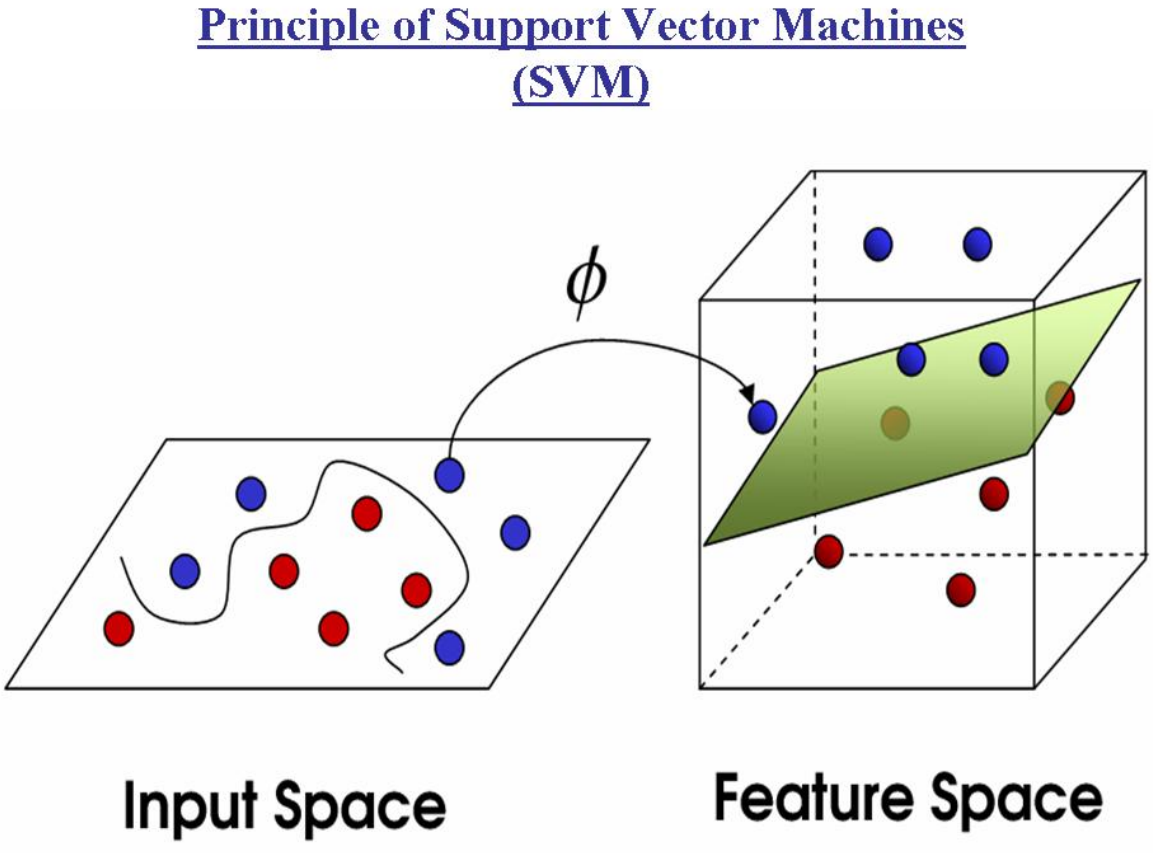
Performance

```
pre = predict(dt, exprData[401:500, ])
table(pre, sgrade[401:500])

##
## pre    high low
##   high   64  10
##   low    21   5
```

SVM



Kernal



SVM in R

```
library(e1071)
```

SVM

```
library(e1071)

## Loading required package: class

model = svm(x = exprData[1:400, ], y = sgrade[1:400], cross =
```

SVM model

```
model


##
## Call:
## svm.default(x = exprData[1:400, ], y = sgrade[1:400], cross = 5)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.005
##
## Number of Support Vectors:  313
```

SVM

```
ret = predict(model, exprData[401:500, ])
table(ret, sgrade[401:500])

##
## ret    high low
##   high   85  15
##   low     0   0
```

Decision Tree and SVM

Decision Tree

- simple
- white-box
- ...

SVM

- complicated
- block-box
- ...

Summary

Classification

- Classification is an important technique for bioinformatics
- Decision tree is a rule-based classification method
- SVM is powerful
- Classification algorithm can be implemented in R easily

Next

华大科技 BGI·tech

```
## Error: argument "n" is missing, with no default
```

Feature Selection

Feature Selection

Feature selection is the process of selecting a subset of relevant features for use in model construction.

Why?

Data always contains many redundant or irrelevant features.

Feature Selection

Filter and Wrapper

- Filter: Chi-Squared, correlation, ...
- Wrapper: Decision tree, SVM, Random Forest...

Feature Selection

```
library(FSelector) library(caret)
```

Filters

Filters

- Chi-squared: discrete attributes
- Consistency: continous and discrete attributes
- Correlation: continous attributes and continous classes
- Entropy: discrete attributes and continous classes

Example

Examples

- SD filter for clustering
- Correlation filter for Classification
- Decision tree based feature selection
- SVM for feature selection

Summary

Summary

- Feature selection is helful to improve performance of classifier
- Feature selection can be used to identify biomarkers
- ...

Next

Future Reading

- Data Mining for Bioinformatics
- Introduction to Machine Learning
- CRAN Task View: Machine Learning & Statistical Learning: http://cran.r-project.org/web/views/MachineLearning.html

Advanced Courses

Advaced Courses

- Machine learning for Big Biological Data
- Implementing high performance machine learning algorithms
- Deep learning for bioinformatics
- Data integration
- Network based machine learning
- ...

Thanks!