R: Advanced Topics

Machine Learning and Big Data

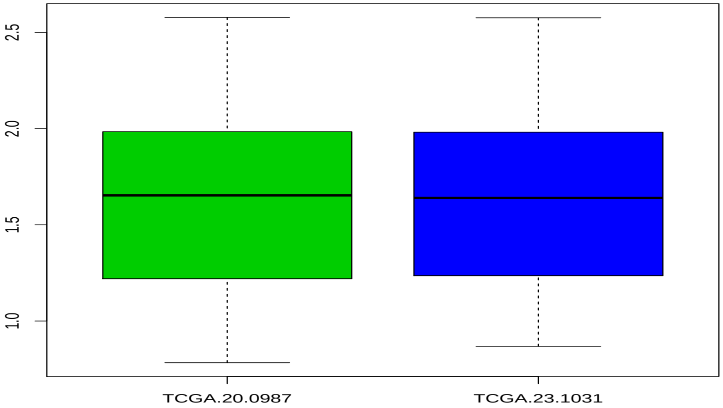
Gang Chen
chengang@genomics.cn

November 14, 2015

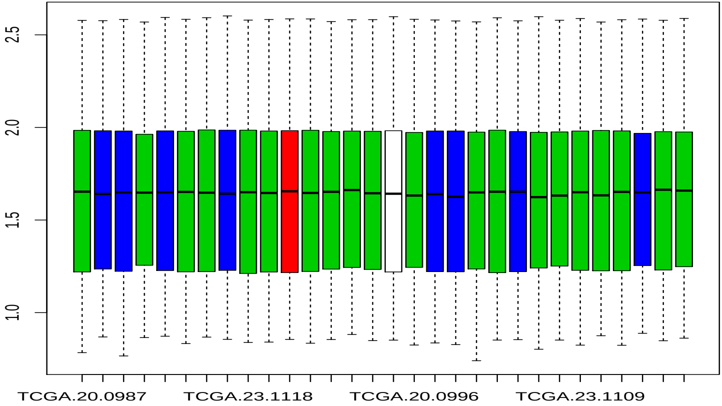
Outline

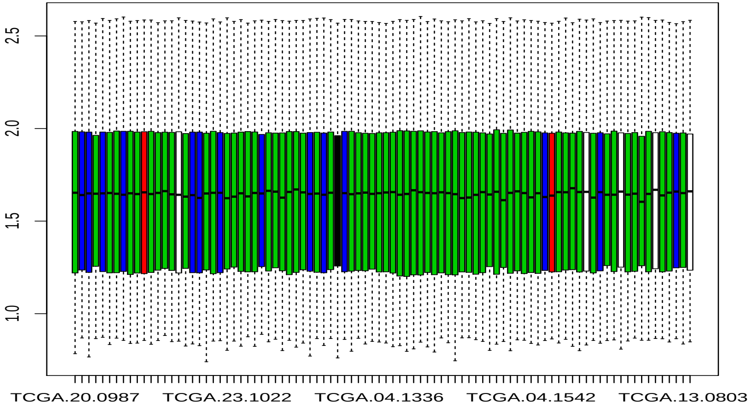
Next

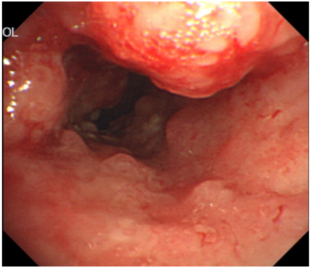
Biomarker for Tumor Stage

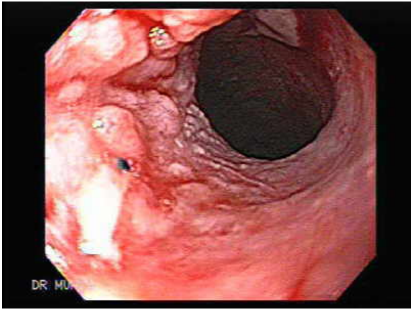


Biomarker for Tumor Stage



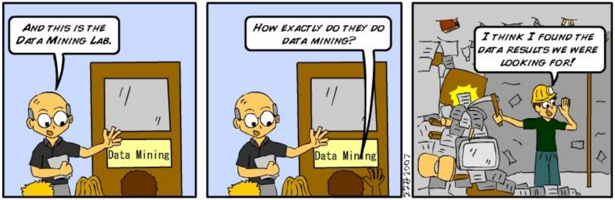

2.5
2.0
1.5
1.0
TCGA.20.0987
TCGA.23.1022
TCGA.04.1336
TCGA.04.1542
TCGA.13.0803





How to identify tumor from genomics perspective?
Is it possible to predict tumor stage?

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?

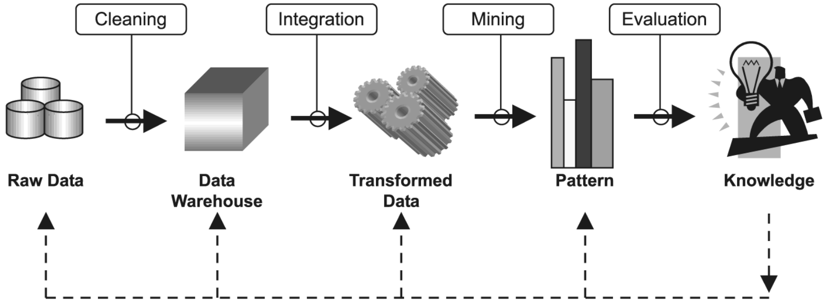
Pang-Ning Tan, Introduction to Data Mining

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
Clustering
Regression
Feature Engineering
...

→

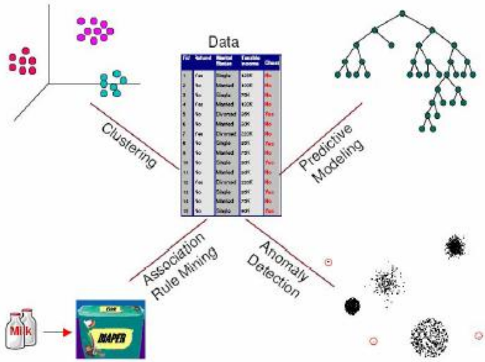
Biological Knowledge

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
- Discussion

Softwares

Softwares

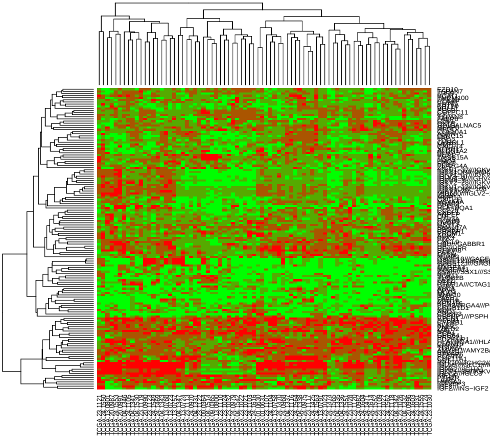
- R: R is an free platform for data analysis and visuaztion.
- R packages:
  - e1071 SVM
  - curatedOvarianData Microarry data of tumor. [Database 2013]
  - mtcars A small dataset for concept demonstration.
- Rgui
- Emacs + ESS
- Vim + R-Plugin
- RStudio
- RStudio in the Cloud

R

- Download from www.r-project.org.
- Installation and start;
- Install e1071 package.

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

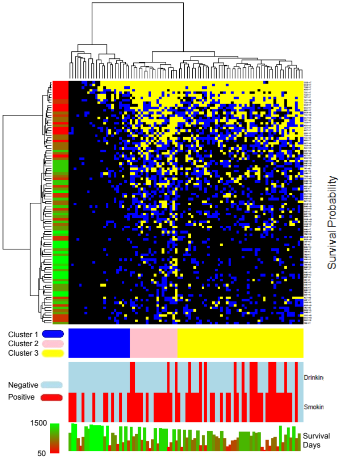
- Discovery of tumor subtypes by clustering gene expression, CNV, miRNA or integrated data.
- Clonal evolution analysis of tumor
- Mutation spectrum clustering
- Pathway or functional annotation based clustering
- Graph clustering for identification of protein functional module or protein complex
- Clustering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

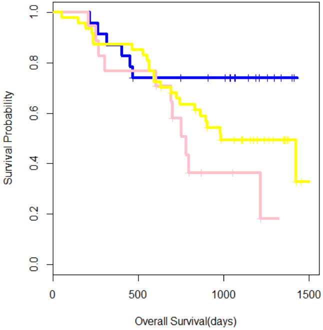
Hierarchical Clustering

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     6.808850
## TCGA.24.0979     6.581221     4.830934
## TCGA.23.1117     5.779257     5.328651     5.726768
## TCGA.23.1021     6.131524     4.801447     4.920307     5.456621
## TCGA.04.1337     7.030940     5.752603     6.475261     4.487964
## TCGA.20.0990     6.465878     5.308989     5.974427     5.647812
## TCGA.23.1032     7.002809     5.420443     5.256359     6.401355
## TCGA.23.1118     5.562177     5.145970     4.799568     4.599808
## TCGA.23.1026     5.388687     5.586002     5.880299     4.388108
## TCGA.20.0991     5.401875     7.061145     7.235305     5.988029
## TCGA.24.1103     6.684103     5.321956     4.508152     6.069577
## TCGA.24.0982     5.690747     4.891739     5.676629     4.789224
## TCGA.23.1119     5.727415     6.252046     6.338474     5.280068
## TCGA.23.1028     5.144723     5.505960     5.534630     5.038784
## TCGA.04.1341     6.449114     5.339794     6.528626     5.084565
## TCGA.20.0996     6.743035     5.049169     5.365110     5.371963
## TCGA.24.1104     4.451846     5.635849     5.834086     4.966425
## TCGA.23.1107     5.398379     6.024515     6.524868     5.269340
## TCGA.23.1120     5.571779     5.116315     4.304913     5.097922
## TCGA.23.1030     6.064855     4.696293     5.837826     5.243088
## TCGA.04.1342     5.574656     4.926885     5.309545     3.867411
## TCGA.23.1022     6.683327     4.765877     4.917100     5.622828
## TCGA.24.1105     5.692072     5.260009     6.253688     4.769659
## TCGA.23.1109     6.134479     5.047752     5.648282     3.975083
## TCGA.23.1121     6.502517     6.605177     7.450118     6.307436
```

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
```

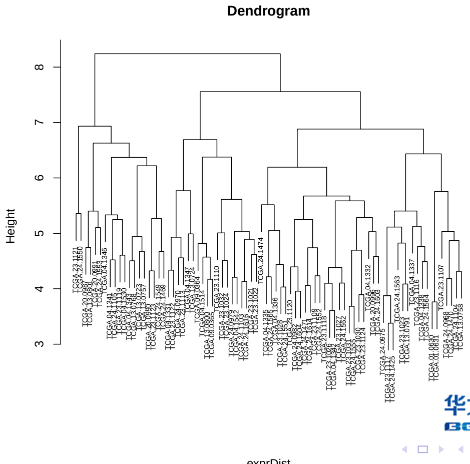
Visualization

Visualization

- Dendrogram: plot.hclust
- Heat Map: heatmap

Dendrogram

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

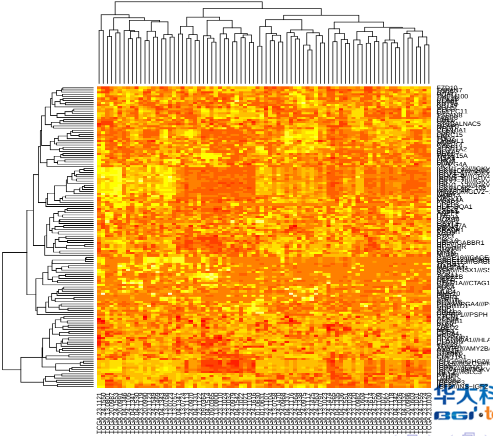


plot.hclust

```
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

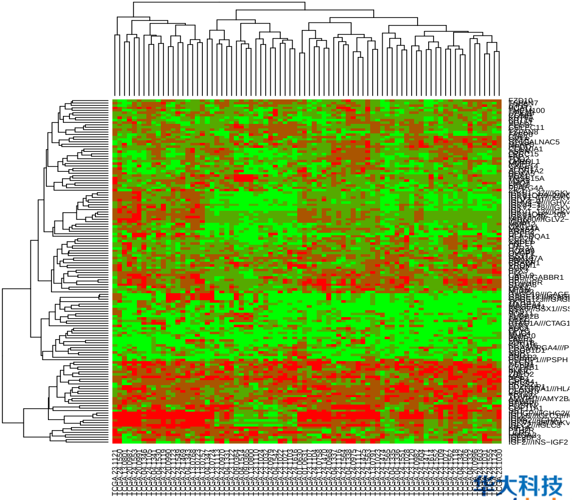
```
heatmap(subdata)
```



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

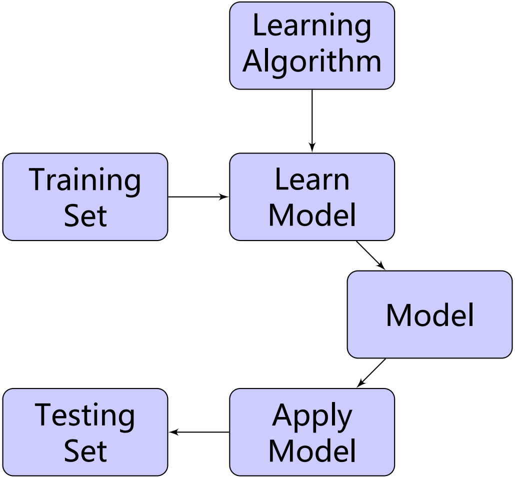
Classification

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

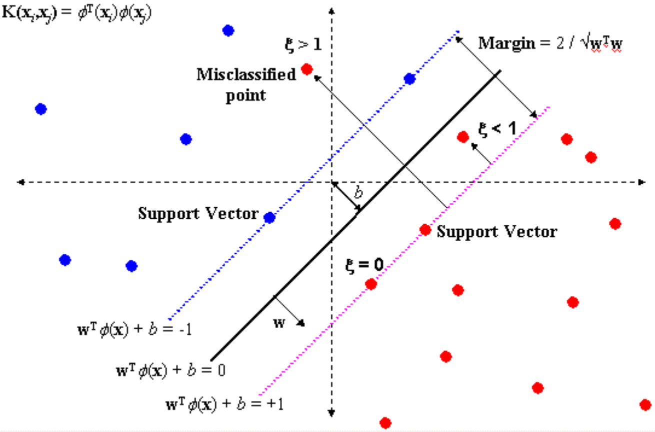
Two Objectives

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

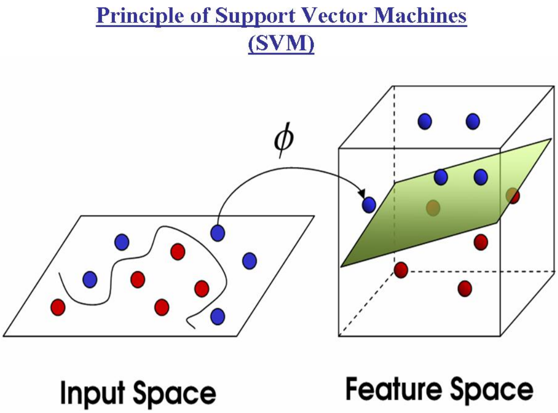
Challanges

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

SVM


K(xi,xj) = φT(xi)φ(xj)
ξ > 1
Margin = 2 / √wTw
Misclassified point
ξ < 1
b
Support Vector
Support Vector
ξ = 0
w
wT φ(x) + b = -1
wT φ(x) + b = 0
wT φ(x) + b = +1

Kernal


Principle of Support Vector Machines
(SVM)
ϕ
Input Space
Feature Space

SVM in R

```
library(e1071)
```

SVM

```
library(e1071)

model = svm(x = exprData[1:400,], y = stages[1:400], cross=5)
```

SVM model

```
model
```

```
##
## Call:
## svm.default(x = exprData[1:400, ], y = stages[1:400], cross = 5)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.001730104
##
## Number of Support Vectors:  200
```

SVM

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

##
## ret  0  1
##   0  0  0
##   1 11 89
```

?

Summary

Classification

- Classification is an important technique for bioinformatics
- SVM is powerful
- Classification algorithm can be implemented in R easily

Next

1. Overview of Machine Learning
2. Unspervised Learning
3. Supervised Learning
4. Future
5. Big Data

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

Next

High Performance Computing

- Task View: High-Performance and Parallel Computing with R
  http://cran.r-project.org/web/views/HighPerformanceComputing.html
  - compiler package and JIT
  - Revolution R Enterprise
  - Rcpp
  - Multi-core
  - GPU
  - MPI

compiler package

These functions provide an interface to a byte code compiler for R.

```
cmpfun(f, options = NULL) compile(e, env = .GlobalEnv,
options = NULL) cmpfile(infile, outfile, ascii = FALSE,
env = .GlobalEnv, verbose = FALSE, options = NULL)
enableJIT(level)
```

see compiler.R for example

Big Data Framework

- Hadoop:
- Spark: SparkR, AMPLab UC BERKELEY http://amplab-extras.github.io/SparkR-pkg/
- Storm:
  - https://github.com/allenday/R-Storm
  - https://github.com/quintona/storm-r