```
################################################################################
# L5.1. PCA
################################################################################
## clear memory
rm(list = ls())

##------------------------------------------------------------------------------
## L5.1.1. Iris data
##------------------------------------------------------------------------------

## show the data
iris
str(iris)

## plot iris data
x11()
plot(iris[,-5])
## more beautiful
plot(iris[,-5],col = iris[,5],pch=19)

##------------------------------------------------------------------------------
## L5.1.2 PCA
##------------------------------------------------------------------------------
## Let's transform data frame into numerical matrix
## data - numerical data
## classes - type of iris, 1=setosa, 2=versicolor, 3=virginica
Data = as.matrix(iris[,-5])
row.names(Data) = as.character(iris[,5])
classes = as.integer(iris[,5])

## perform PCA
PC = prcomp(Data)
str(PC)

## plot PC1 and PC2 only
plot(PC$x[,1],PC$x[,2],
     col=classes, pch=19)

## plot 3D
library(rgl)
plot3d(PC$x[,1],PC$x[,2],PC$x[,3],
       size = 2,
       col = classes,
       type = "s",
       xlab = "PC1",
       ylab = "PC2",
       zlab = "PC3")



##------------------------------------------------------------------------------
## L5.1.2 PCA for Mice
##------------------------------------------------------------------------------

Mice = read.table("http://edu.sablab.net/data/txt/mice.txt",
                  header=T,sep="\t")
Data=as.matrix(Mice[,-(1:5)])
Data[is.na(Data)] = 0
```

```
PC = prcomp(Data)
str(PC)
## plot PC1 and PC2 only
col=c("#AA002233","#0000AA33")
plot(PC$x[,1],PC$x[,2],col=col[Mice$Sex], pch=19)
##------------------------------------------------------------------------------
## L5.1.3 PCA for ALL
##------------------------------------------------------------------------------

ALL = read.table("http://edu.sablab.net/data/txt/all_data.txt",
                 as.is=T,header=T,sep="\t")
str(ALL) ## see here - experiments are in columns!!!
## Transform to matrix
Data=as.matrix(ALL[,-(1:2)])
## assign colors based on column names
color = colnames(Data)
color[grep("ALL",color)]="red"
color[grep("normal",color)]="blue"
## perform PCA
PC = prcomp(t(Data))
## visualize in 3D
library(rgl)
plot3d(PC$x[,1],PC$x[,2],PC$x[,3],size = 2,col = color,type = "s",
       xlab = "PC1",ylab = "PC2",zlab = "PC3")
## plot PC1 and PC2 only
plot(PC$x[,1],PC$x[,2],col=color,pch=19,cex=2)
text(PC$x[,1],PC$x[,2]+5,colnames(Data),cex=0.6)

PCG = prcomp(Data)
plot(PCG$x[,1],PCG$x[,2],col="#00FF0011",pch=19,cex=2)
plot3d(PCG$x[,1],PCG$x[,2],PCG$x[,3],size = 1,col = "green",type = "s",
       xlab = "PC1",ylab = "PC2",zlab = "PC3")

## check the distribution of the data
source("http://sablab.net/scripts/plotDataPDF.r")
x11()
plotDataPDF(Data,col=color,add.legend=T)
x11()
boxplot(Data,col=color,outline=F,las=2)

################################################################################
# L5.2. Clustering
################################################################################
## clear memory
rm(list = ls())

##------------------------------------------------------------------------------
## L5.2.1 k-means Clustering
##------------------------------------------------------------------------------
Data = as.matrix(iris[,-5])
row.names(Data) = as.character(iris[,5])
classes = as.integer(iris[,5])

## try k-means clustering
clusters = kmeans(x=Data,centers=3,nstart=10)$cluster

## show clusters on PCA
```

```
PC = prcomp(Data)
x11()
plot(PC$x[,1],PC$x[,2],col = classes,pch=clusters)
legend(2,1.4,levels(iris$Species),col=c(1,2,3),pch=19)
legend(-2.5,1.4,c("c1","c2","c3"),col=4,pch=c(1,2,3))

##------------------------------------------------------------------------------
## L5.2.2 Hierarchical clustering
##------------------------------------------------------------------------------

## use heatmap
heatmap(Data)

## use heatmap with colors
color = character(length(classes))
color[classes == 1] = "black"
color[classes == 2] = "red"
color[classes == 3] = "green"
heatmap(Data,RowSideColors=color,scale="none")

## modify the heatmap colors
heatmap(Data,RowSideColors=color,scale="none",
   col = colorRampPalette (c("blue","wheat","red"))(1000))

## For advanced heatmap use:
library(gplots)
heatmap.2(Data,RowSideColors=color,scale="none",trace="none",
          col = colorRampPalette (c("blue","wheat","red"))(1000))

##------------------------------------------------------------------------------
## L5.2.2 Hierarchical clustering: ALL (Task L5.2)
##------------------------------------------------------------------------------
ALL = read.table("http://edu.sablab.net/data/txt/all_data.txt",
                 as.is=T,header=T,sep="\t")
Data=as.matrix(ALL[,-(1:2)])
color = colnames(Data)
color[grep("ALL",color)]="red"
color[grep("normal",color)]="blue"
##---------------------------------
## Task L5.2a. Select top 100 genes

## annotate genes
rownames(Data) = ALL[,1]
## create indexes for normal and ALL columns
idx.norm = grep("normal",colnames(Data))
idx.all = grep("ALL",colnames(Data))

## perform a t-test
pv = double(nrow(Data))
for (i in 1:nrow(Data)){
  pv[i] = t.test(Data[i,idx.all],Data[i,idx.norm])$p.val
}
## select top genes
Top = Data[order(pv),][1:100,]

## make a heatmap
heatmap(Top,ColSideColors=color,
```

```
          col = colorRampPalette (c("blue","wheat","red"))(1000))
          
## scale data first and repeat heatmap
TopSc = t(scale(t(Top)))
heatmap(TopSc,ColSideColors=color,
          col = colorRampPalette (c("blue","white","red"))(1000))

library(gplots)
heatmap.2(TopSc,ColSideColors=color,trace="none",
          col = colorRampPalette (c("blue","white","red"))(1000))

##---------------------------------
## In fact, robust set of signidicant genes, identified by limma
## package of R/Bioconductor is different:
x11()
source("http://sablab.net/scripts/limmaEBS2Class.r")
res = limmaEBS2Class(data = ALL[,-c(1:2)],
                     anno = ALL[,c(1:2)],
                     classes = sub("_.+","",names(ALL[,-c(1:2)])),
                     plotTop=100,
                     col=c("red","blue"))

#############################################################################
# L5.3. Classification
#############################################################################
library(caTools)
classes = sub("_.+","",colnames(Data))
auc = as.double(colAUC(t(Data), classes))
i=which.max(auc)
ALL[i,1:2]
colAUC(Data[i,], classes, plotROC = T)
plot(Data[i,],col=as.factor(classes),pch=19)
##---------------------------------------------------------------------------
## L5.3.1 Iris data
##---------------------------------------------------------------------------

library(caTools)
plot(iris[,-5],col = iris[,5],pch=19)
## let's check which of the parameters is a better predictor
x11()
par(mfcol=c(2,2))
for (ipred in 1:4){
  plot(density(iris[as.integer(iris[,5])==1,ipred]),
       xlim=c(min(iris[,ipred]),max(iris[,ipred])),
       col=1,lwd=2,main=names(iris)[ipred])
  lines(density(iris[as.integer(iris[,5])==2,ipred]),col=2,lwd=2)
  lines(density(iris[as.integer(iris[,5])==3,ipred]),col=3,lwd=2)
}
x11()
par(mfcol=c(2,2))
for (ipred in 1:4){
  cat("\n\n",names(iris)[ipred],"\n")
  print(colAUC(iris[,ipred], iris[,5], plotROC=T))
}


library(e1071)
```

```
model = svm(Species ~ ., data = iris)
svm.res = as.character(predict(model,
                               iris[,-5]))
res = cbind(as.character(iris[,5]), svm.res)

## creat a confusion matrix
ConTab = data.frame(matrix(nr=3,nc=3))
rownames(ConTab) = paste("pred.",levels(iris$Species),sep="")
names(ConTab) = levels(iris$Species)
for (ic in 1:3){
  for (ir in 1:3){
    ConTab[ir,ic] = sum(iris$Species == levels(iris$Species)[ic] & svm.res == levels(iris$
    Species)[ir])
  }
}

##-------------------------------------------------------------------------------
## L5.3.1 Devaux et al. data
##-------------------------------------------------------------------------------
## Devaux Y., et al. Use of circulating microRNAs to diagnose acute myocardial
## infarction. Clin Chem. 2012

MI = read.table("http://edu.sablab.net/data/txt/infarction.txt",
                header=T,sep="\t")

i.c = MI$Type == "ctrl"
i.ns = MI$Type == "nstemi"
i.s = MI$Type == "stemi"
color = character(nrow(MI))
color[i.c]="grey"
color[i.s]="green"
color[i.ns]="blue"
library(rgl)
plot3d(MI[,2:4],type="s",size=1,col=color)
## to visualize overlaping samples we can but some noise
#plot3d(MI[,2:4]+rnorm(3*nrow(MI))*0.1,type="s",size=1,col=color)

model = svm(Type ~ ., data = MI)
svm.res= as.character(predict(model, MI[,-1]))
res = cbind(as.character(MI$Type), svm.res)

## creat a confusion matrix
ConTab = data.frame(matrix(nr=3,nc=3))
rownames(ConTab) = paste("pred.",levels(MI$Type),sep="")
names(ConTab) = levels(MI$Type)
for (ic in 1:3){
  for (ir in 1:3){
    ConTab[ir,ic] = sum(MI$Type == levels(MI$Type)[ic] & svm.res == levels(MI$Type)[ir])
  }
}
ConTab
## crappy results :))
```