Finding Subclasses of Emphysema

Getting Statistical Summary of Data: To learn more about the data we look at its statistical summary. In addition, a baseline database is created in this step. The code is located in *BiomarkerStat.R*.

```
# read the biomarker data from biomarker.norm Rdata.

#load("/Users/szarei2008/Documents/HarvardResearch/data/analyses/TESRA_AECOPD/biomarkerQC_norm_V10.RData")

#======================================================================

# read proteomic data
proteomic <- read.table("/Users/szarei2008/Documents/HarvardResearch/data/raw/download_11-28-2012/Proteomic/rbm_quest_proteomicdata.txt",sep="\t'
print(paste("The proteomic data file has dimensions ",dim(proteomic)[1]
            ,"rows by",dim(proteomic)[2],"columns."))
```

```
## [1] "The proteomic data file has dimensions  128974 rows by 15 columns."
```

```
#Print the first 20 rows of the data
print(head(proteomic, n = 20))
```

```
##    STUDY.ID PATIENT.ID SAMPLE.ID SAMPLE.NO SAMPLE.COLLECTION.DATE
## 1   NB19751       1000   7474656       101                     NA
## 2   NB19751       1000   7471015       903                     NA
## 3   NB19751       1000   7474656       101                     NA
## 4   NB19751       1000   7471016       904                     NA
## 5   NB19751       1000   7471014       902                     NA
## 6   NB19751       1000   7474656       101                     NA
## 7   NB19751       1000   7471016       904                     NA
## 8   NB19751       1000   7471014       902                     NA
## 9   NB19751       1000   7471015       903                     NA
## 10  NB19751       1000   7471015       903                     NA
## 11  NB19751       1000   7474656       101                     NA
## 12  NB19751       1000   7471016       904                     NA
## 13  NB19751       1000   7471014       902                     NA
## 14  NB19751       1000   7474656       101                     NA
## 15  NB19751       1000   7471015       903                     NA
## 16  NB19751       1000   7474656       101                     NA
## 17  NB19751       1000   7471016       904                     NA
## 18  NB19751       1000   7471014       902                     NA
## 19  NB19751       1000   7471015       903                     NA
## 20  NB19751       1000   7474656       101                     NA
##    PARAMETER.MEASURED SAMPLE.TYPE TEST.RESULT.IN.NUMERIC.FORM  UNIT
## 1                ANG2      PLASMA                     238.125 PG/ML
## 2               CC_16      PLASMA                        6024 PG/ML
## 3               CC_16      PLASMA                        6911 PG/ML
## 4               CC_16      PLASMA                        4772 PG/ML
## 5               CC_16      PLASMA                        6528 PG/ML
## 6                 CRP      PLASMA                        7.94  MG/L
## 7                 CRP      PLASMA                       4.825  MG/L
## 8                 CRP      PLASMA                       5.675  MG/L
## 9                 CRP      PLASMA                        12.3  MG/L
## 10         FIBRINOGEN      PLASMA                             UG/ML
## 11         FIBRINOGEN      PLASMA                     5728.47 UG/ML
## 12         FIBRINOGEN      PLASMA                    7076.965 UG/ML
## 13         FIBRINOGEN      PLASMA                     9480.18 UG/ML
## 14           GLOB_A2M      PLASMA                    1749.895 UG/ML
## 15              GP130      PLASMA                      307120 PG/ML
## 16              GP130      PLASMA                      404332 PG/ML
## 17              GP130      PLASMA                      229828 PG/ML
## 18              GP130      PLASMA                      336260 PG/ML
## 19                RBP      PLASMA                     33812.5  UG/L
## 20                RBP      PLASMA                              UG/L
##    TEST.RESULT.IN.TEXT.FORM ANALYTICAL.METHOD ASSAY.SITE ANALYST.NAME
## 1                                                  QUEST           NA
## 2                                                  QUEST           NA
## 3                                                  QUEST           NA
## 4                                                  QUEST           NA
## 5                                                  QUEST           NA
## 6                                                  QUEST           NA
## 7                                                  QUEST           NA
## 8                                                  QUEST           NA
## 9                                                  QUEST           NA
## 10                      ALQ                        QUEST           NA
## 11                                                 QUEST           NA
## 12                                                 QUEST           NA
## 13                                                 QUEST           NA
## 14                                                 QUEST           NA
## 15                                                 QUEST           NA
## 16                                                 QUEST           NA
## 17                                                 QUEST           NA
## 18                                                 QUEST           NA
## 19                                                 QUEST           NA
## 20                      BLQ                        QUEST           NA
##    SAMPLE.REMARK                      SAMPLE.COMMENT
## 1             NA
## 2             NA
## 3             NA
## 4             NA
## 5             NA
## 6             NA
## 7             NA
## 8             NA
## 9             NA
## 10            NA UPPER LIMIT OF QUANTIFICATION=20000
## 11            NA
## 12            NA
## 13            NA
## 14            NA
## 15            NA
## 16            NA
## 17            NA
## 18            NA
## 19            NA
## 20            NA  LOWER LIMIT OF QUANTIFICATION=5500
```

```
#-------------
#Selecting the Baseline SAMPLE.NO only.
baselines<-subset(proteomic,SAMPLE.NO==101)

#To view the list of available biomarker in data
biomarkerlist<-unique(baselines$PARAMETER.MEASURED)
print(biomarkerlist)
```

```
##   [1] "ANG2"               "CC_16"              "CRP"
##   [4] "FIBRINOGEN"         "GLOB_A2M"           "GP130"
##   [7] "RBP"                "RETINOL"            "SURFACTANT_D"
##  [10] "TGFB1"              "TGFB2"              "TIMP1"
##  [13] "TIMP2"              "AMPHIREGULIN"       "ALPHA1_ANTI_TRPSN"
##  [16] "AFP"                "AGRP"               "APOA1"
##  [19] "APOC3"              "APOH"               "AXL"
##  [22] "BDNF"               "BMP6"               "BTC"
##  [25] "C3"                 "CA125"              "CA19_9"
##  [28] "CALCT"              "CD40"               "CD40_LIG"
##  [31] "CEA"                "CGA"                "CKMB"
##  [34] "CNTF"               "EGF"                "EGFR"
##  [37] "ENA78"              "EN_RAGE"            "EOT1"
##  [40] "EOT2"               "ERP"                "EPIREGULIN"
##  [43] "ESEL"               "ET1_Z"              "FABP"
##  [46] "FACTOR_VII"         "FAS"                "FAS_LIG"
##  [49] "FGF4"               "FGFB"               "G_CSF"
##  [52] "GROWTH_HORMONE"     "GM_CSF"             "GROA"
##  [55] "HPT"                "HB_EGF"             "HCC4"
##  [58] "HGF"                "I_309"              "ICAM1"
##  [61] "IFNG"               "IGA"                "IGE"
##  [64] "IGF1"               "IGM"                "IL1A"
##  [67] "IL1B"               "IL10"               "IL12P40"
##  [70] "IL12P70"            "IL13"               "IL15"
##  [73] "IL16"               "IL18"               "IL1RA"
##  [76] "IL2"                "IL2RA"              "IL25"
##  [79] "IL3"                "IL4"                "IL5"
##  [82] "IL6"                "IL6R"               "IL7"
##  [85] "IL8"                "INSL"               "IP10"
##  [88] "TGFB1_LAP"          "LEP"                "LYMT"
##  [91] "MCP1"               "MCP2"               "MCP3"
##  [94] "MCP4"               "M_CSF"              "MDC"
##  [97] "MICA"               "MIG"                "MIP1A"
## [100] "MIP1B"              "MIP3A"              "MMP1"
## [103] "MMP10"              "MMP2"               "MMP3"
## [106] "MMP7"               "MMP9"               "MMP9_TOTAL"
## [109] "MPIF1"              "MPO"                "NGFB"
## [112] "NRCAM"              "OPG"                "PAP"
## [115] "PAPP_A"             "PARC"               "PDGF_BB"
## [118] "PLGF"               "PSA"                "S_RAGE"
## [121] "S100B"              "SCF"                "SGOT"
## [124] "SODS_1"             "SORTILIN"           "TECK"
## [127] "TF"                 "TGFA"               "TGFB3"
## [130] "TS1"                "TENSCIN_C"          "TNFR1"
## [133] "TNFA"               "TNFB"               "TP"
## [136] "TRAIL_R3"           "TSH"                "VITDBP"
## [139] "VEGF"               "HMGB1"
```

```
#--------------
#for keeping all statistical summary together for all biomarkers.

# If you want to create histograms change the variable below to TRUE
createPlots=FALSE

alltogether<-NULL
for(i in 1:length(biomarkerlist)){
  #Statistical Summary of the Biomarker
  biomarkername<- as.character(biomarkerlist[i])
  #Select the specific biomarkers
  selectbiomarker<-subset(baselines,PARAMETER.MEASURED==biomarkername)


  #Print the statistical summary of that biomarker
  statsummary<-summary(as.numeric(selectbiomarker$TEST.RESULT.IN.NUMERIC.FORM),na.rm=TRUE)
  ALQcount<-sum(selectbiomarker$TEST.RESULT.IN.TEXT.FORM=="ALQ")
  BLQcount<-sum(selectbiomarker$TEST.RESULT.IN.TEXT.FORM=="BLQ")
  QNScount<-sum(selectbiomarker$TEST.RESULT.IN.TEXT.FORM=="QNS")
  Total<-nrow(selectbiomarker)
  Site1<-paste(unique(selectbiomarker$ASSAY.SITE),sep="")
  Site2<-matrix(c(Site1),1,length(Site1))
  Site3<-apply(format(Site2), 1, paste, collapse=" ")
  #Combine the statssummary and outliers together

  combined<-cbind(Biomarker.Names=biomarkername,rbind(statsummary),ALQcount,BLQcount,QNScount,Total,Site3)
  #To avoid repetition of titles.
  alltogether<-rbind(alltogether,combined)
  #Write the statistical summary in a csv file

  #Plot Histograms
  if(createPlots && !is.na(statsummary[1])){
    par(mfrow=c(1,1))
    png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/Histograms/",biomarkername,".Histogram.png",sep=""), width=480, hei
    hist(as.numeric(selectbiomarker$TEST.RESULT.IN.NUMERIC.FORM),xlab=paste('Biomarker',biomarkername),
         main='Biomarker Histogram',col='red')
    dev.off()
  }
}

#write.table(alltogether,file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/statsummary.csv",row.names=FALSE,sep=',')
#========================================================================

# Table that consists of all baseline (101) entry. #Selecting the Baseline SAMPLE.NO only.
#baselines<-subset(proteomic,SAMPLE.NO==101)
write.table(baselines, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/baseline.csv",row.names=FALSE, sep=',')
#========================================================================
### DATA MODIFICATION
#Delet the biomarkers that have more than 90% BLQ
BLQRatio1<-as.numeric(alltogether[,"BLQcount"])/as.numeric(alltogether[,"Total"])
BLQRatio<-cbind(alltogether,BLQRatio1)
write.table(BLQRatio,file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/statsummary.csv",row.names=FALSE,sep=',')
BLQfinal<-subset(BLQRatio,BLQRatio1<=0.9)
write.table(BLQfinal,file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BLQfinal.csv",row.names=FALSE,sep=',')
#-----------------------------
### Fill out a numeric value in (TEST.RESULT.IN.NUMERIC.FORM) for those that were missing  due to BLQ, ALQ, QNS
```

Fixing missing data

Repairing missed data: In the first step we replace BLQ, ALQ, and QNS by a numarical value using the provided numbers in the corresponding comment cell, we call this file "baselinesRepairReplace". In addition, We also created another file in which we replaced BLQ, ALQ, and QNS by NA, the file name is"baselinesRepairRemoved.csv". The code is in the file *BiomarkerRepair.R*.

```
## We start using the baselline table which was a subset of proteomic data who had SAMPLE.NO==101 (Baseline data)
## The code for creating baseline.csv is located in BiomarkerStat.R file.
baselines <- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/baseline.csv",sep=",",header=T,stringsAsFactors = FALSE)
for(i in 1:nrow(baselines)){
  if (baselines[i, "TEST.RESULT.IN.TEXT.FORM"]=="BLQ"){
    comment<-baselines[i,"SAMPLE.COMMENT"]
    #Split the comment right at the equal sign
    commentSplited<-strsplit(comment,"=")
    #Since the split result is stored as a list we first need to unlist them, next we will convert the result into a matrix of 2 elements.
    commentmatrix<-matrix(unlist(commentSplited),ncol=2)
    #print(commentmatrix)
    baselines[i,"TEST.RESULT.IN.NUMERIC.FORM"]<-as.numeric(commentmatrix[1,2])/2
    #print( baselines[i,"TEST.RESULT.IN.NUMERIC.FORM"])
  }
  if (baselines[i, "TEST.RESULT.IN.TEXT.FORM"]=="ALQ"){
    comment2<-baselines[i,"SAMPLE.COMMENT"]
    #Split the comment right at the equal sign
    commentSplited2<-strsplit(comment2,"=")
    #Since the split result is stored as a list we first need to unlist them, next we will convert the result into a matrix of 2 elements.
    commentmatrix2<-matrix(unlist(commentSplited2),ncol=2)
    baselines[i,"TEST.RESULT.IN.NUMERIC.FORM"]<-as.numeric(commentmatrix2[1,2])
  }
}
#Removing the entire row that had QNS in TEST.RESULT.IN.TEXT.FORM
baselines<-subset(baselines,TEST.RESULT.IN.TEXT.FORM!="QNS")

# We are removing the Biomarkers that have more than 90% BLQ which on BiomarkerStat table we saw only IL12P70 had this criteria.
baselines<-subset(baselines,PARAMETER.MEASURED!="IL12P70")

write.table(baselines, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/baselinesRepairReplace.csv",row.names=FALSE, sep=',')

# We also want to have a second dataset where all BLQ, ALQ and QNS are all removed which we call the file RepairRemoved
baselines<-subset(baselines,TEST.RESULT.IN.TEXT.FORM!="ALQ")
baselines<-subset(baselines,TEST.RESULT.IN.TEXT.FORM!="BLQ")
```

```
write.table(baselines, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/baselinesRepairRemoved.csv",row.names=FALSE, sep=',')
```

Reformatting The Data: Next we format the data in a way that each row is one patient and each column is one biomarker. This is done with *removed* data file. The code is in ***BioPersonRemoved.R*** file.

```
baselines <- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/baselinesRepairRemoved.csv",sep=",",header=T,stringsAsFactors = FA
#Find the unique values of patient ID
personlist<-unique(baselines$PATIENT.ID)

#Find the unique values of biomarkers (PARAMETER.MEASURED)
biomarkerlist<-unique(baselines$PARAMETER.MEASURED)

#Make a new table where rows consistes of each patient ID and columns are the corresponding biomarkerlist
Biomarkerperperson=matrix(nrow=length(personlist),ncol=length(biomarkerlist))
rownames(Biomarkerperperson)<-personlist
colnames(Biomarkerperperson)<-biomarkerlist
for (i in 1:nrow(baselines)) {
  id<-baselines[i,"PATIENT.ID"]
  bio<-baselines[i, "PARAMETER.MEASURED"]
  result<-baselines[i,"TEST.RESULT.IN.NUMERIC.FORM"]
  Biomarkerperperson[as.character(id),bio]<-result
}
#Adding a column to a table: Adding patient ID column to the table Biomarkerperperson
Biomarkerperperson<-cbind(PATIENT.ID=personlist,Biomarkerperperson)
write.table(Biomarkerperperson, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemoved.csv",row.names=F, sep=',')
```

Filling all of NAs: After removing BLQ, ALQ, and QNS data we have many missing data. Therefore we are going to use K-nearest neighbor averaging imputation method to fill out the missing values. The code is locare in file ***Imputation.R***

```
# In this program we are going to use K-nearest neighbor averaging imputation method to fill out the missing values.
# We used the BioPersonRemoved table where all the BLQ, ALQ and QNS rows were removed. Therefore in this data set which is a matrix we have cells
# which corresponds to the biomarker test results that were originallt either ALQ, BLQ or QNS.

library(impute)
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemoved.csv",sep=",",header=T,stringsAsFactors = FALSE)

#-----------------------------------------------------------
# To find the columns with more than 50% of missing data we run the following code.( remember i=1 is the patient Id so we start with i=2:140)
# Since the total number of rows is 461, to find columns with more than 50% null we set t > 230
# We removed the following columns [1] "TGFB2" , "GM_CSF", "IL1B", "TF", "EPIREGULIN", "TGFB3" , "NGFB", "S100B", "TGFA", "LYMT"

BioPersonLessNa<-BioPerson[,1]
BioPersonLessNaColName<-colnames(BioPerson)[1]

for (i in 2:140){
  t<-sum(as.integer(is.na(BioPerson[,i])))
  if (t>230){

    print(colnames(BioPerson)[i])

  }
  else{
    BioPersonLessNa<-cbind(BioPersonLessNa,BioPerson[,i])
    BioPersonLessNaColName<-cbind(BioPersonLessNaColName,colnames(BioPerson)[i])
  }
}
```

```
## [1] "TGFB2"
## [1] "GM_CSF"
## [1] "IL1B"
## [1] "TF"
## [1] "EPIREGULIN"
## [1] "TGFB3"
## [1] "NGFB"
## [1] "S100B"
## [1] "TGFA"
## [1] "LYMT"
```

```
colnames(BioPersonLessNa)<-BioPersonLessNaColName
BioPersonLessNa<-BioPersonLessNa[,(-1)]
BioPersonImputed<-impute.knn(BioPersonLessNa,k = 10, rowmax = 0.5, colmax = 0.8, maxp = 1500, rng.seed=362436069)
```

```
## Warning in knnimp(x, k, maxmiss = rowmax, maxp = maxp): 4 rows with more than 50 % entries missing;
##  mean imputation used for these rows
```

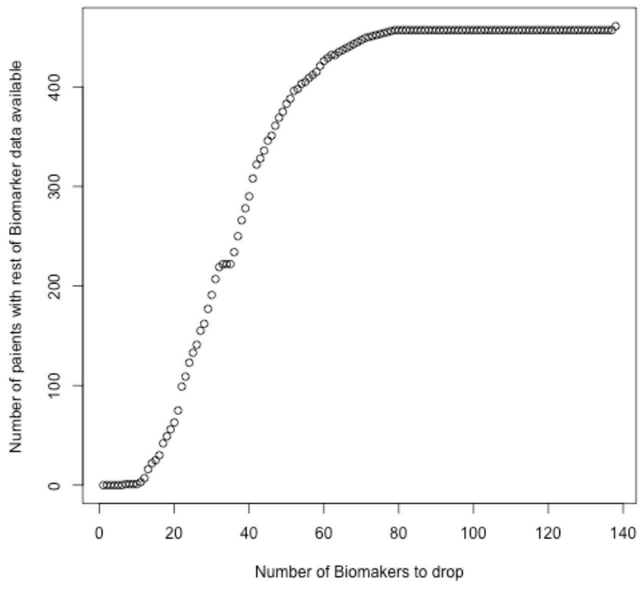
```
sum(as.numeric(is.na(BioPersonImputed)))
```

```
## [1] 0
```

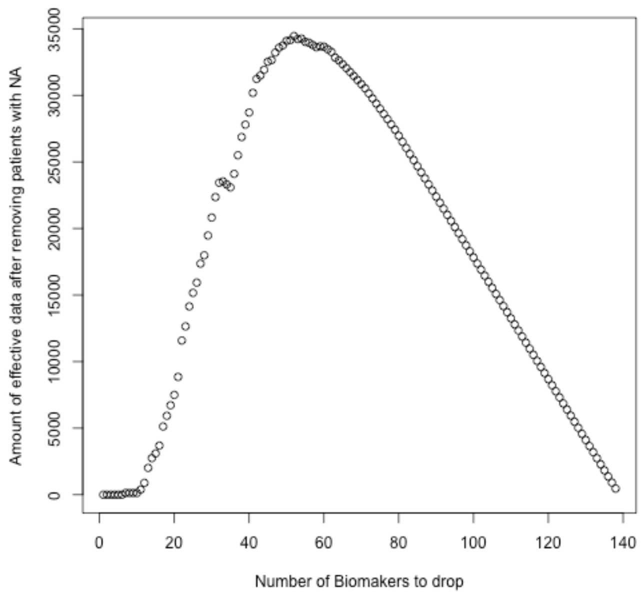
```
BioPersonKnnImputed<-cbind(PATIENT.ID=BioPerson[,1],BioPersonImputed$data)
write.table(BioPersonKnnImputed, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonKnnImputed.csv",row.names=F, sep=',')
```

Removing all of NAs: We tried different methods to fix the missing data but the best method turned out to be just removing the missing data using the following simple optimization technique. The code is locare in file ***BioPersonReduced.R***

```
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemoved.csv",sep=",",header=T,stringsAsFactors = FALSE)
fullmarkers<-matrix(ncol = 1,nrow = 138,data = 0)
dataRemained<-matrix(ncol = 1,nrow = 138,data = 0)
for(i in 1:138){
  #Counting the columns number of NAs in each column and sorting them
  nas<-sort(colSums(is.na(BioPerson)))
  #Choozing the last i column (The last i columns are the i hogest NAs since we have sorted them)
  lastNas<-nas[140-i+1:140]
  #Using set difference we find the remainig names of biomarkers
  remainingNames<- setdiff(colnames(BioPerson),names(lastNas))
  #Finally choosing for all of the patients we choose the wanted biomarkers
  BioPersonLessBio<-BioPerson[,remainingNames]
  #print(paste("removing",i,"Biomarkers resulted in ",sum(rowSums(is.na(BioPersonLessBio))==0),"patients with full Biomarkerts"))
  #claculating how many of patients have full biomaker information for the remaining biomarkers
  fullmarkers[i]=sum(rowSums(is.na(BioPersonLessBio))==0)
  #calculating the amount of data meaning the number of patients with full biomarkers times number of biomarkers.
  dataRemained[i]=(139-i)*fullmarkers[i]
}

#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/SelectionOfNumberOfBioMarkersToRemoveBaesdOnPatientCount.png",sep=""),
plot(1:138,fullmarkers, xlab="Number of Biomakers to drop", ylab="Number of paients with rest of Biomarker data available")
```



```
#dev.off()



#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/SelectionOfNumberOfBioMarkersToRemoveBaesdOnEffectiveData.png",sep="",
plot(1:138,dataRemained, xlab="Number of Biomakers to drop", ylab="Amount of effective data after removing patients with NA")
```



```
#dev.off()


#selecting number of biomarkers to drop based on number of patirnts to keep. Either use this or the next
NumPatientNeeded<-300
NumBioDrop<-which.max(fullmarkers>=NumPatientNeeded)

#selecting number of biomarkers to drop based on the maxim data available. Either use this or the one before
NumBioDrop<-which.max(dataRemained)


#Now taking out the data like above
nas<-sort(colSums(is.na(BioPerson)))
lastNas<-nas[140-NumBioDrop+1:140]
removedBioMarkers<-na.omit(names(lastNas))
remainingNames<- setdiff(colnames(BioPerson),names(lastNas))
#If decided to keep the following two biomarkers uncomment the following two lines
#wantedBios<-c("CC_16","IL6")
#remainingNames<-union(remainingNames,wantedBios)
BioPersonLessBio<-BioPerson[,remainingNames]

completePatients<-(rowSums(is.na(BioPersonLessBio))==0)
BioPersonLessBioLessPerson<-BioPersonLessBio[completePatients,]
removedPatients<-BioPersonLessBio[(rowSums(is.na(BioPersonLessBio))>0),1]


write.table(BioPersonLessBioLessPerson, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonReduced.csv",row.names=F, sep=',')

#saving the reduced data

TheFileName<-"/Users/szarei2008/Documents/HarvardResearch/SZscripts/DetailsOfReducedBioMarkersAndPatients.csv"
write.table("The biomarkers that are removed from the data:", file=TheFileName,row.names=F, sep=',',col.names = F)
#NOTE: The following lines have APPEND=TRUE
write.table((removedBioMarkers), file=TheFileName,row.names=F, sep=',',append = T,col.names = F)
write.table(" The list of psatients that are removed from data", file=TheFileName,row.names=F, sep=',',append = T,col.names = F)
write.table((removedPatients), file=TheFileName,row.names=F, sep=',',append = T,col.names = F)
```

Data Correlation

We study the correlation between the biomarks on each other. This code is in ***correlationRemoved.R***

```
##   FINDING CORRELATION of BIOMARKERS
# This code reads the table from BioPersonReplaced.R final table which has fill out a value for BQL, AQL and QNS.
BioData <- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemoved.csv",sep=",",header=T,stringsAsFactors = FALSE)

# Removed the first column from the table that consisted of patient's ID.
BioData<- BioData[,-(1)]
#Remove the entire rows with NA value
#BioData<-na.omit(BioData)
#correlation<-cor(BioData,method ="pearson",use = "pairwise.complete.obs")
#correlation<-cor(BioData,method ="pearson",use = "complete.obs")
correlation<-matrix(ncol=ncol(BioData),nrow=ncol(BioData))
#Assiging the row and col names which are the biomarkers names.
```

```
rownames(correlation)<-colnames(BioData)
colnames(correlation)<-colnames(BioData)
print(paste("Dimension of Correlation matrix is", dim(correlation)))
```

```
## [1] "Dimension of Correlation matrix is 139"
## [2] "Dimension of Correlation matrix is 139"
```

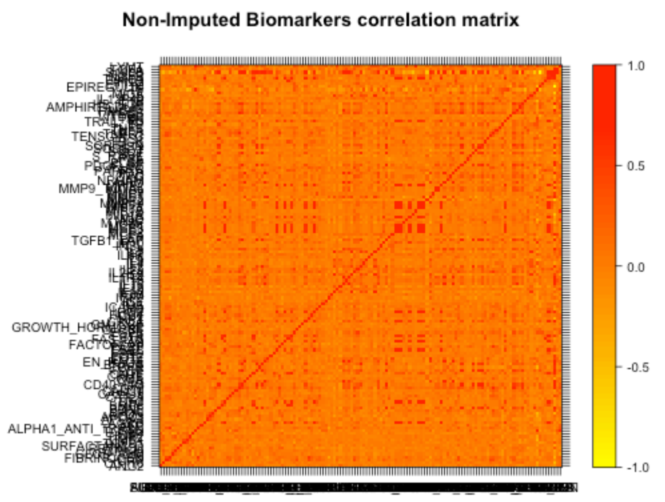
```
n<-ncol(correlation)
# Creating an empty vector for storing the upper triangular entries of correlation matrix.
# If n=139 then we will have 139*139 -(139 (where correlations are 1)/2( since there are two triangles.))
HistVector<-matrix(nrow=1,ncol=((n*n)-n)/2)
n<-1

for(i in 1:139){
  for(j in i:139){
    # The correlation value of 1 in diagonal of correlation matrix ( the correlation of a biomarker to itself is 1)
    if(i==j){
      correlation[i,j]<-1
    }else{
      # First select two biomarkers (two different columns)
      twoBio<-cbind(BioData[,i],BioData[,j])

      # Next remove the NA values.
      twoBio<-na.omit(twoBio)

      # Now find the correlation between the two biomarkers.
      twoBioCorr<-cor(twoBio,method ="pearson",use = "pairwise.complete.obs")
      if(!is.na(twoBioCorr[1,2])){
        correlation[i,j]<-twoBioCorr[1,2]
        correlation[j,i]<-twoBioCorr[1,2]
      }else{
        correlation[i,j]<-0
        correlation[j,i]<-0
      }
      HistVector[1,n]<-correlation[i,j]
      n<-n+1
    }
  }
}

names<-rownames(correlation)
report<-NULL
for(i in 1:139){
  for(j in i:139){
    if (i!=j && correlation[i,j]>=0.75){
      #print(paste(names[i],names[j],correlation[i,j]))
      report<-rbind(report, cbind(names[i],names[j],correlation[i,j]))
    }
  }
}

print(report)
```

```
##       [,1]         [,2]         [,3]
##  [1,] "TGFB2"      "TF"         "0.989595593708496"
##  [2,] "AGRP"       "BTC"        "0.801713161505351"
##  [3,] "AGRP"       "FGF4"       "0.907891493448374"
##  [4,] "AGRP"       "MCP3"       "0.753351370553741"
##  [5,] "AGRP"       "TGFB3"      "0.953581104480492"
##  [6,] "BDNF"       "TGFB1_LAP"  "0.803371014328322"
##  [7,] "BDNF"       "PDGF_BB"    "0.893400895268033"
##  [8,] "BMP6"       "FAS_LIG"    "0.779074409289335"
##  [9,] "BTC"        "MCP2"       "0.788300696839662"
## [10,] "BTC"        "MCP3"       "0.797942709942511"
## [11,] "BTC"        "MCP4"       "0.765421504037117"
## [12,] "BTC"        "MMP1"       "0.755580515773025"
## [13,] "BTC"        "MPIF1"      "0.777716457984009"
## [14,] "BTC"        "TGFB3"      "0.968702621256955"
## [15,] "CD40_LIG"   "EGF"        "0.889140971254271"
## [16,] "CD40_LIG"   "TGFB1_LAP"  "0.826444719589759"
## [17,] "CD40_LIG"   "PDGF_BB"    "0.767268400759363"
## [18,] "CD40_LIG"   "SORTILIN"   "0.757777912428501"
## [19,] "CD40_LIG"   "S100B"      "0.912875553960919"
## [20,] "CEA"        "EN_RAGE"    "0.866193402088768"
## [21,] "CEA"        "MPO"        "0.811391144558121"
## [22,] "CEA"        "S100B"      "0.959283806461622"
## [23,] "EGF"        "TGFB1_LAP"  "0.814263603748915"
## [24,] "EGF"        "PDGF_BB"    "0.770091141675113"
## [25,] "EGF"        "S100B"      "0.830096397499127"
## [26,] "ENA78"      "SCF"        "0.957209759735659"
## [27,] "EN_RAGE"    "IL16"       "0.830073652510075"
## [28,] "EN_RAGE"    "MPO"        "0.950266264289719"
## [29,] "EN_RAGE"    "PAP"        "0.812657500736488"
## [30,] "EN_RAGE"    "S100B"      "0.961757054839278"
## [31,] "FAS"        "HGF"        "0.774508585579792"
## [32,] "FAS"        "MCP3"       "0.774660762926056"
## [33,] "FAS"        "MCP4"       "0.755205440309823"
## [34,] "FAS"        "MIP3A"      "0.752900506520837"
## [35,] "FAS"        "TGFB3"      "0.805335792561563"
## [36,] "FGF4"       "TGFB3"      "0.89144493546199"
## [37,] "GROA"       "S100B"      "0.804388853003351"
## [38,] "HGF"        "MCP2"       "0.898851890879923"
## [39,] "HGF"        "MCP3"       "0.941598093686159"
## [40,] "HGF"        "MCP4"       "0.935183257374354"
## [41,] "HGF"        "MIP3A"      "0.922811855188398"
## [42,] "HGF"        "MMP1"       "0.845907455425453"
## [43,] "HGF"        "MMP10"      "0.817396214511788"
## [44,] "HGF"        "MPIF1"      "0.906204806801347"
## [45,] "HGF"        "TGFB3"      "0.811802873589721"
## [46,] "IL16"       "IL1RA"      "0.906647199403688"
## [47,] "IL16"       "MPO"        "0.905780081927645"
## [48,] "IL16"       "PAP"        "0.878944652174836"
## [49,] "IL16"       "S100B"      "0.960315103755883"
## [50,] "IL1RA"      "MPO"        "0.750119630717687"
## [51,] "IL1RA"      "PAP"        "0.797136728455231"
## [52,] "IL1RA"      "S100B"      "0.93762574658399"
## [53,] "IL8"        "S100B"      "0.945943225773316"
## [54,] "TGFB1_LAP"  "PDGF_BB"    "0.818162540459141"
## [55,] "TGFB1_LAP"  "SORTILIN"   "0.758048496048479"
## [56,] "TGFB1_LAP"  "S100B"      "0.826839624201823"
## [57,] "MCP2"       "MCP3"       "0.949044080739078"
## [58,] "MCP2"       "MCP4"       "0.953303175828903"
## [59,] "MCP2"       "MIG"        "0.762247366286839"
## [60,] "MCP2"       "MIP3A"      "0.926925913879674"
## [61,] "MCP2"       "MMP1"       "0.824531483016075"
## [62,] "MCP2"       "MMP10"      "0.782685269343327"
## [63,] "MCP2"       "MPIF1"      "0.920135108183934"
## [64,] "MCP2"       "TGFB3"      "0.816705527919665"
## [65,] "MCP3"       "MCP4"       "0.972220692067461"
## [66,] "MCP3"       "MIP3A"      "0.957517116242213"
## [67,] "MCP3"       "MMP1"       "0.875318344434787"
## [68,] "MCP3"       "MMP10"      "0.833449925515848"
## [69,] "MCP3"       "MPIF1"      "0.959708674420747"
## [70,] "MCP3"       "TGFB3"      "0.858119327136872"
## [71,] "MCP4"       "MIP3A"      "0.949125486514896"
## [72,] "MCP4"       "MMP1"       "0.852914628715042"
## [73,] "MCP4"       "MMP10"      "0.816834993026653"
## [74,] "MCP4"       "MPIF1"      "0.936597749168772"
## [75,] "MCP4"       "TGFB3"      "0.83909444709571"
## [76,] "MIP3A"      "MMP1"       "0.841068666782024"
## [77,] "MIP3A"      "MMP10"      "0.807484782437263"
## [78,] "MIP3A"      "MPIF1"      "0.922178941837309"
## [79,] "MIP3A"      "TGFB3"      "0.825738929898881"
## [80,] "MMP1"       "MMP10"      "0.875365489141119"
## [81,] "MMP1"       "MPIF1"      "0.860516232151409"
## [82,] "MMP1"       "TGFB3"      "0.866924591970921"
## [83,] "MMP10"      "MPIF1"      "0.818307282803146"
## [84,] "MMP10"      "TGFB3"      "0.860020470079578"
## [85,] "MMP9"       "MMP9_TOTAL" "0.788871394903933"
## [86,] "MMP9_TOTAL" "S100B"      "0.801690552675162"
## [87,] "MPIF1"      "TGFB3"      "0.845801607135179"
## [88,] "MPO"        "PAP"        "0.856080762101637"
## [89,] "MPO"        "S100B"      "0.964225738672935"
## [90,] "PAP"        "S100B"      "0.952421530665685"
## [91,] "PDGF_BB"    "S100B"      "0.819415053350741"
## [92,] "TS1"        "S100B"      "0.832112660083611"
## [93,] "HB_EGF"     "S100B"      "0.810908947863482"
## [94,] "TGFB3"      "S100B"      "0.957084735859685"
## [95,] "NGFB"       "S100B"      "0.910486287020597"
## [96,] "S100B"      "TGFA"       "0.917014069605095"
```

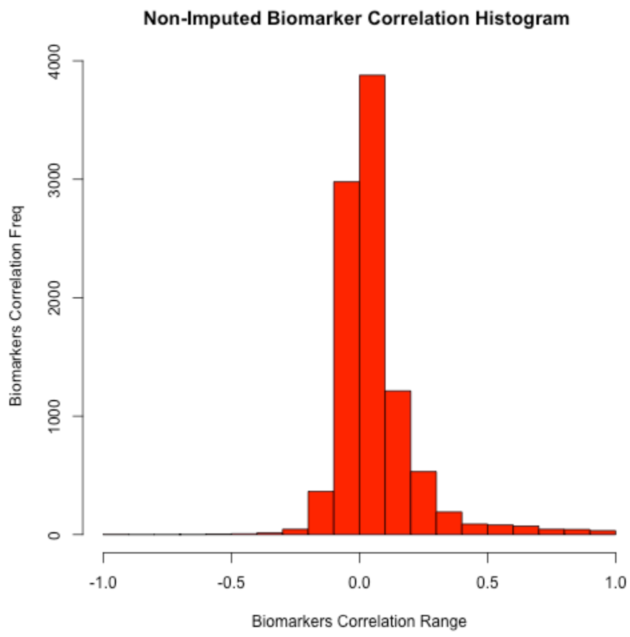
```
write.table(correlation, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/correlationRemoved.csv",row.names=F, sep=',')
write.table(report, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/correlation75Removed.csv",row.names=F, sep=',')

#------------------
# Draw a correlation matrix
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.1.3
```

```
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/correlationmatrixRemoved.png",sep=""), width=1400, height=1400, bg='tı
rgb.palette <- colorRampPalette(c("yellow", "red"), space = "rgb")
levelplot(correlation, main="Non-Imputed Biomarkers correlation matrix", xlab="", ylab="", col.regions=rgb.palette(220), cuts=200, at=seq(-1,1,0.
```



```
#dev.off()

#--------------------
#Draw a Histogram
#Plot Histograms
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HistoCorrRemoved.png",sep=""), width=480, height=480, bg='transparent
hist(as.numeric(HistVector),xlab=paste('Biomarkers Correlation Range'),ylab=paste('Biomarkers Correlation Freq'),
     main='Non-Imputed Biomarker Correlation Histogram ',col='red')
```



```
#dev.off()
#------
write.table(HistVector, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/HistVecCorrRemoved.csv",row.names=F, sep=',')
```

Principle Component Analysis

Normalizing Data:

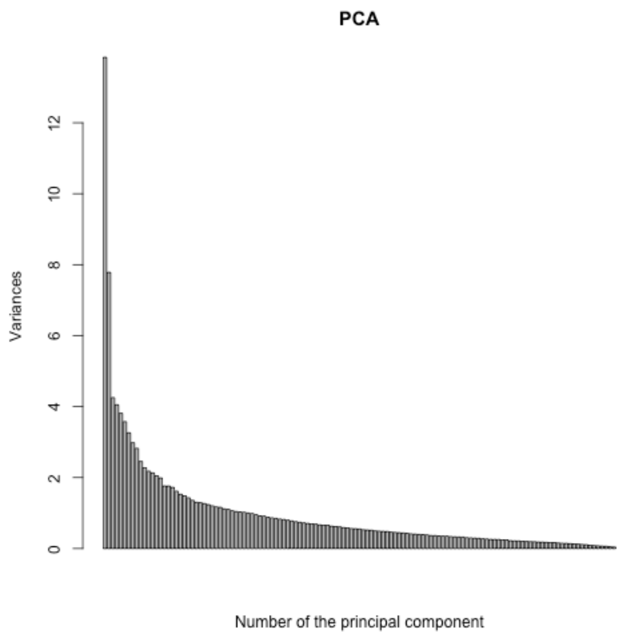
It is recommended to normalize the data before PCA. Here, Empirical Normal Quantile Transformation (ENQT) is used for nomalization. The code is located in

BioPersonKnnImputedENQT.R

```
# Empirical Normal Quantile transformation Normalization= ENQT
# In this program we transform the data we obtained after KNN imputation.
library(multic)
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonKnnImputed.csv",sep=",",header=T,stringsAsFactors = FALSE)
# We get rid of the NA values.
#BioPerson<-na.omit(BioPerson)
# We start with second column since the first one is patient ID
for (i in 2:ncol(BioPerson)){
  normalizedcol<-tRank(BioPerson[,i])
  BioPerson[,i]<-matrix(normalizedcol,ncol=1)
}
write.table(BioPerson, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonKnnImputedENQT.csv",row.names=F, sep=',')
```

Principle Component Analysis PCA is not used in the final analysis but it was done during the analysis. The code is located in ***PCAImputedENQTBioPerson.R***

```
# In this program we are applying the Principal Component analysis on the imputed data that were also transformed using the ENQT method.
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonKnnImputedENQT.csv",sep=",",header=T,stringsAsFactors = FA
BioPersonPatientID<-BioPerson[,1]
BioPerson<-BioPerson[,(-1)]

# Calculate the PCA of the data
PCA<-prcomp(BioPerson,scale=F)

# Write the corresponding std ( or eigen values), Proportion of Variance and Cumulative Proportion to find the required number of Principal compo

SummaryPCA<-summary(PCA)
write.table(SummaryPCA$importance, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/SummaryPCAKnnImputedENQT.csv",row.names=T, sep=','

#Plot Screeplot
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/ScreeplotImputedEnQTBioPerson.png",sep=""), width=480, height=480, bg=
screeplot(PCA,npcs = 129, xlab='Number of the principal component' )
```



```
#dev.off()
#-----------

# Each principal is a linear combinations of all the biomarkers and the corresponding coefficients are stored in the PCA rotation attribution.
#Therefore to get the corresponding equations of the principal we perform a matrix multiply.
PCARotation<-PCA$rotation
PCAEighty<-PCARotation[,1:53]
PCAPerson<-as.matrix(BioPerson) %*% PCAEighty
PCAPerson<-cbind(PATIENT.ID=BioPersonPatientID,PCAPerson)
write.table(PCAPerson, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/PCAPersonKnnImputedENQT.csv",row.names=F, sep=',')
write.table(cbind(Biomarker.Names=rownames(PCARotation),PCARotation), file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/PCAEquationsKn
ImportantBiomarkers<-NULL
for (i in 1:53){
  ImportantBiomarkers<-c(ImportantBiomarkers, names(which.max(PCARotation[,i])))
  }
BioPersonImportant<-cbind(PATIENT.ID=BioPersonPatientID)
for (i in 1:53){
  BioPersonImportant<-cbind(BioPersonImportant,BioPerson[,ImportantBiomarkers[i]])
}
colnames(BioPersonImportant)[2:54]<-ImportantBiomarkers
write.table(BioPersonImportant, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPerson53KnnImputedENQT.csv",row.names=F, sep=',')

print(ImportantBiomarkers)
```

```
##  [1] "VEGF"           "BDNF"           "BMP6"           "LEP"
##  [5] "MMP2"           "IL3"            "GP130"          "APOH"
##  [9] "INSL"           "CRP"            "EN_RAGE"        "MMP9"
## [13] "MMP3"           "NRCAM"          "FGFB"           "AFP"
## [17] "VITDBP"         "TIMP1"          "FABP"           "MIG"
## [21] "CKMB"           "MCP1"           "G_CSF"          "IGA"
## [25] "CEA"            "CKMB"           "CALCT"          "IL4"
## [29] "GLOB_A2M"       "HCC4"           "ESEL"           "EOT2"
## [33] "IL2"            "IL10"           "GROWTH_HORMONE" "ET1_Z"
## [37] "S_RAGE"         "IL10"           "ERP"            "IL18"
## [41] "IL13"           "IGE"            "IL6"            "ET1_Z"
## [45] "IL4"            "ANG2"           "CKMB"           "GP130"
## [49] "FAS_LIG"        "GROWTH_HORMONE" "IL15"           "NRCAM"
## [53] "TNFB"
```

```
#=======================
# To find out the total correlations among each Biomarkers and their corresponding first 53 Principal component.
for (i in 1:129){
  if(max(abs(PCARotation[,i]))>0.5){
    print (i)
  }
}
```

```
## [1] 124
## [1] 125
## [1] 127
## [1] 128
## [1] 129
```

Clustering - KMEAN

Normalizing Data: It is recommended to normalize the data before clustering. Here, Empirical Normal Quantile Transformation (ENQT) is used for nomalization. The code is located in ***BioPersonRemovedENQT.R***

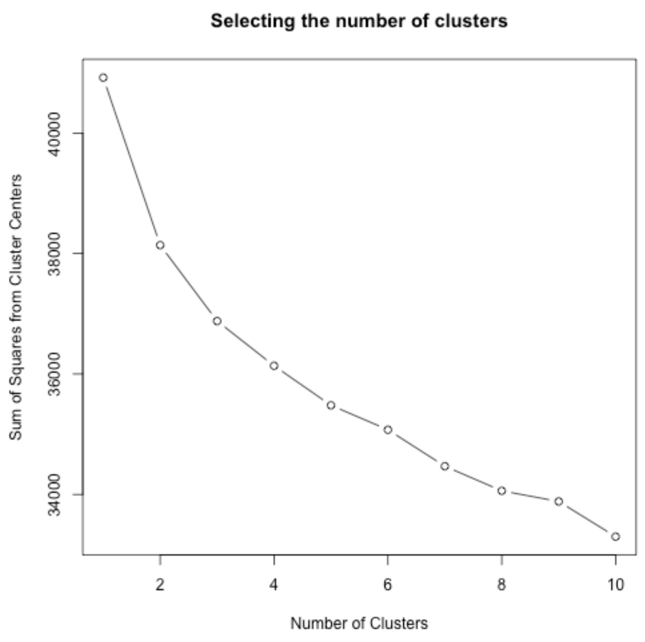
```
# Empirical Normal Quantile transformation Normalization= ENQT
library(multic)
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemoved.csv",sep=",",header=T,stringsAsFactors = FALSE)
# We get rid of the NA values.
BioPerson<-na.omit(BioPerson)
# We start with second column since the first one is patient ID
for (i in 2:ncol(BioPerson)){
  normalizedcol<-tRank(BioPerson[,i])
  BioPerson[,i]<-matrix(normalizedcol,ncol=1)
}
write.table(BioPerson, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemovedENQT.csv",row.names=F, sep=',')
```

Clustering by KMEANS In this section the data is clustered with KMEANs methods. The data in BioPersonReplaced is used since the data in BioPersonRemoved is not usable. As it is shown in the plot, this method is not usefull. The code is located in ***clusteringReplacedENQT.R***.

```
BioData<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonReplacedENQT.csv",sep=",",header=T,stringsAsFactors = FALSE)
#removing patient IDs
PatientIDs<-BioData[,1]
BioData<- BioData[,-(1)]
#deciding number of clusters
wss <- (nrow(BioData)-1)*sum(apply(BioData,2,var))
for (i in 2:10) wss[i] <- sum(kmeans(BioData, centers=i)$withinss)
print(wss)
```

```
##  [1] 40921.00 38139.22 36879.26 36136.72 35479.96 35074.16 34468.70
##  [8] 34060.15 33883.69 33299.55
```

```
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/clusterCountsReplaced.png",sep=""), width=480, height=480, bg='transpa
plot(1:10, wss, type="b", main="Selecting the number of clusters", xlab="Number of Clusters", ylab="Sum of Squares from Cluster Centers")
```



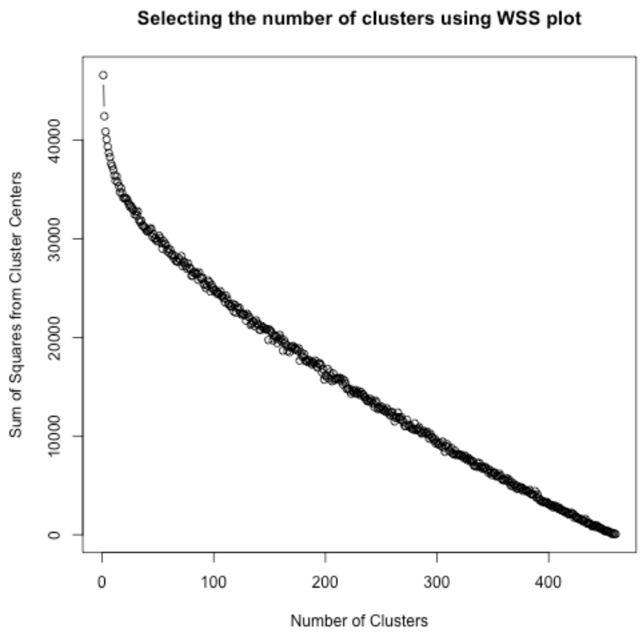
```
#dev.off()
```

PCA clustering by KMEANS This time principle components of data are clustered rather than the data itself. The code is located in ***clusterPCAPersonKnnImputedENQT.R***.

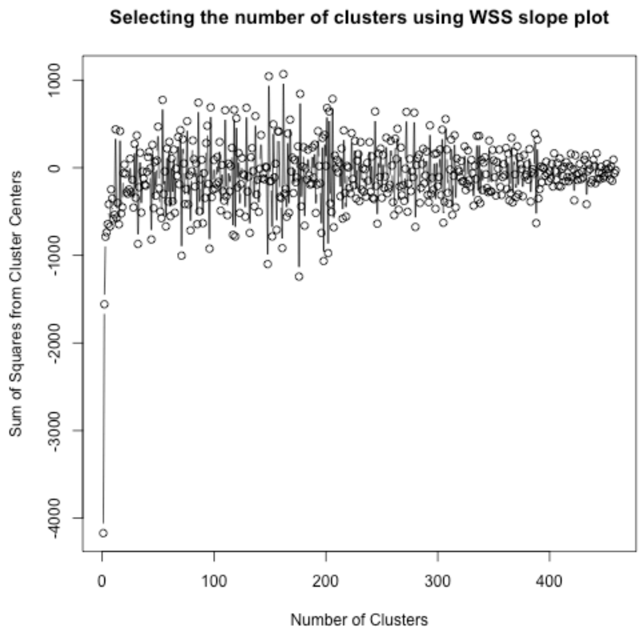
```
PCAData<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/PCAPersonKnnImputedENQT.csv",sep=",",header=T,stringsAsFactors = FALS
#removing patient IDs
PatientIDs<-PCAData[,1]
PCAData<- PCAData[,-(1)]
#deciding number of clusters using withinss which is within cluster sum of squares distances.
wss <- (nrow(PCAData)-1)*sum(apply(PCAData,2,var))
for (i in 2:460) wss[i] <- sum(kmeans(PCAData, centers=i,algorithm="Lloyd")$withinss)
print(wss)
```

```
##   [1] 46606.46648 42436.11558 40878.06498 40090.30120 39352.03708
##   [6] 38709.72349 38293.01531 37622.94519 37375.51266 37024.38636
##  [11] 36441.06619 35904.00195 36342.44431 35769.65799 35367.07340
##  [16] 34721.77664 35139.84919 34617.42266 34167.43983 34105.28487
##  [21] 34141.31809 34072.98838 33793.80718 33496.23303 33248.64028
##  [26] 33352.26392 33073.31254 32936.32285 32482.03495 32755.07769
##  [31] 32411.16633 32780.35934 31910.90387 31688.93212 31860.10237
##  [36] 31348.64280 31307.07480 31184.75265 30990.26467 30785.31395
##  [41] 30738.47366 30861.02350 31043.09604 31036.83441 30217.79946
##  [46] 30480.56169 30019.15502 30085.57849 29766.92180 29850.06334
##  [51] 30318.80670 30084.31427 29583.99343 29008.43350 29783.05591
##  [56] 29482.90742 29441.58866 28771.96538 28991.41868 28606.39291
##  [61] 28901.47576 28348.31025 28453.09145 28069.63358 28276.12785
##  [66] 27761.34655 27753.45475 27659.85264 28012.30698 27823.90746
##  [71] 28252.60657 27248.74245 27574.61626 27624.37858 27209.18831
##  [76] 26967.48771 27498.86027 27081.42602 26996.67867 26282.03149
##  [81] 26242.90097 26554.46887 26652.64054 26438.66711 26488.70808
##  [86] 25844.93980 26588.33513 26091.48005 25961.14833 26051.52959
##  [91] 25418.30999 25356.07545 25011.88138 25293.79696 25772.66767
##  [96] 25589.75106 24665.85568 25354.98507 25019.41981 24771.54630
## [101] 24858.86787 24619.80899 24429.38407 24552.50234 24391.12683
## [106] 24684.89972 24195.55461 24226.87342 23960.29404 23584.76541
## [111] 24241.30899 23914.11267 23372.13438 23418.81489 23149.66978
## [116] 23170.28964 23398.22661 22632.13603 23293.55815 22509.29965
## [121] 23074.02994 22850.07550 23018.35075 22502.83776 22315.57246
## [126] 22425.93073 22353.78417 22267.49980 21708.59893 22393.14155
## [131] 22122.32407 22022.23606 21278.20008 21397.42236 21622.03844
## [136] 21584.05659 21144.97622 21734.65155 21429.01966 21293.06070
## [141] 20788.94958 20777.55564 21150.59115 21029.94494 20964.35374
## [146] 20832.35618 20842.30068 20847.07541 19748.10265 20792.12456
## [151] 20636.40795 20486.50345 19702.04288 20198.41172 20087.99784
## [156] 20134.64869 19425.95966 19841.84078 20255.33534 19923.90733
## [161] 19578.37077 18661.09205 19729.74329 19433.34188 19193.37651
## [166] 18634.12770 19023.74990 18509.76085 19058.50857 19016.43851
## [171] 19132.11085 18842.47150 18930.62710 18766.12928 18653.81472
## [176] 18897.16022 17655.07594 18498.47909 18057.97016 18071.80085
## [181] 18306.87798 17646.00984 17640.78521 17507.21678 17712.89883
## [186] 17626.72885 17435.00701 17259.16618 17492.02842 17380.96649
## [191] 17645.30485 17129.14053 16937.13653 17356.04343 17374.96803
## [196] 17196.58439 16453.11250 16794.13713 15729.85108 16104.34348
## [201] 16125.14884 16808.44174 15833.86499 16476.67985 16069.43603
## [206] 15564.82703 16350.46909 15670.89647 15810.85135 15805.65955
## [211] 15896.93584 15843.48857 15896.16989 15570.75461 15790.63087
## [216] 15210.10829 15636.54738 15348.20398 14790.77458 14810.48656
## [221] 14812.58248 14581.61110 14287.89987 14683.07096 14553.86106
## [226] 14352.62181 14659.81128 14309.18187 14390.90031 14546.65547
## [231] 14164.93182 14457.91058 14207.51503 14199.05508 13925.36668
## [236] 13907.73961 13576.95469 13632.96784 13486.03141 13778.77337
## [241] 13504.78595 13634.44480 13291.06378 12934.15896 13576.32518
## [246] 13368.19862 12729.85848 12891.23758 12785.58998 13003.85137
## [251] 12659.29438 12805.02771 12576.27553 12474.77426 12817.57557
## [256] 12525.23772 12291.06349 12097.17813 12404.16469 12428.79776
## [261] 12036.56988 11495.95192 11827.59373 11928.12386 12373.63001
## [266] 11895.00082 11683.37374 11914.05604 11602.36836 11404.40014
## [271] 11025.71026 11015.36231 11652.71838 11150.28236 10970.15085
## [276] 11070.41553 10803.73666 10762.36787 10656.72150 11284.94952
## [281] 10609.24137 10638.95812 10751.06376 10897.81261 10496.38776
## [286] 10267.70631 10307.54130 10524.80174 10346.24268 10259.39225
## [291] 10445.88440 10137.02648  9744.34339 10227.55886  9929.21695
## [296]  9813.73061  9563.15291  9719.84530  9252.12500  9415.50323
## [301]  9205.05708  9100.92267  9309.53188  8998.06123  9433.19558
## [306]  8805.68406  8417.14898  8972.72178  9180.68808  8894.98789
## [311]  8487.34704  8465.86401  8787.30280  8248.69643  8155.95908
## [316]  8526.81148  8074.29260  8180.64895  8159.82449  8116.91689
## [321]  8008.97459  8086.05296  7928.30247  7773.68013  8017.02379
## [326]  7762.04947  7578.48899  7775.41458  7631.06586  7413.15145
## [331]  7503.98657  7441.81363  7008.48801  7028.64744  6900.42881
## [336]  7262.87617  6895.34241  7258.88414  6854.19039  6941.91190
## [341]  6687.35208  6744.77980  6863.80679  6789.58748  6557.76125
## [346]  6280.03883  6589.59782  6322.87904  6049.44759  6252.89730
## [351]  6336.48270  6228.45901  5911.13358  5975.64243  5965.32743
## [356]  6171.86583  5953.01793  5642.66589  5536.07815  5578.05753
## [361]  5494.53015  5720.88622  5479.36078  5639.76105  5265.72100
## [366]  4974.87827  5317.81214  4921.88312  5099.38125  5024.02017
## [371]  4686.19908  4931.41196  4629.93682  4749.08428  4789.95379
## [376]  4777.11640  4451.27304  4595.64993  4494.11771  4414.27701
## [381]  4453.42859  4318.34107  4532.48925  4145.88182  4096.21768
## [386]  4077.71430  4011.67071  4400.36172  3769.07836  4088.30837
## [391]  3738.14470  3494.47013  3508.29946  3341.81226  3344.99452
## [396]  3338.32407  3246.63388  3316.03014  3249.29408  3093.21414
## [401]  3054.03025  2935.30971  2983.26273  2769.02224  2933.69052
## [406]  2894.60801  2757.59765  2811.99020  2604.92577  2712.33302
## [411]  2496.67909  2484.13022  2388.70554  2400.99912  2247.87293
## [416]  2312.04832  2204.93353  2050.46943  2069.24011  2227.85606
## [421]  2117.32760  2100.42390  1728.76402  1871.33638  1791.19484
## [426]  1652.31368  1588.01982  1634.33020  1504.00401  1460.37997
## [431]  1350.31029  1441.96215  1583.38029  1166.99256  1126.77261
## [436]  1167.37120  1154.32757   980.74546   839.82356   930.56830
## [441]   951.34432   775.25648   940.41161   815.97205   759.46100
## [446]   630.10767   641.66511   581.39286   464.47262   370.27384
## [451]   354.29571   396.51316   332.00907   319.62329   188.25753
## [456]   202.68594    52.22507   145.38285    82.31539    52.49683
```

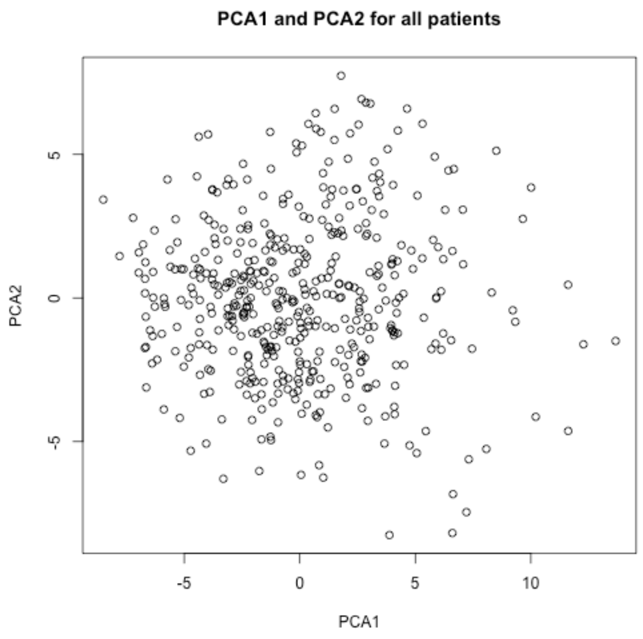
```
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/clusterCountsPCAImputed.png",sep=""), width=480, height=480, bg='trans
plot(1:460, wss, type="b", main="Selecting the number of clusters using WSS plot", xlab="Number of Clusters", ylab="Sum of Squares from Cluster (
```



```
#dev.off()

slopewss<- wss[2:460]-wss[1:459]
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/SlopeclusterCountsPCAImputed.png",sep=""), width=480, height=480, bg=
plot(1:459, slopewss, type="b", main="Selecting the number of clusters using WSS slope plot", xlab="Number of Clusters", ylab="Sum of Squares fro
```

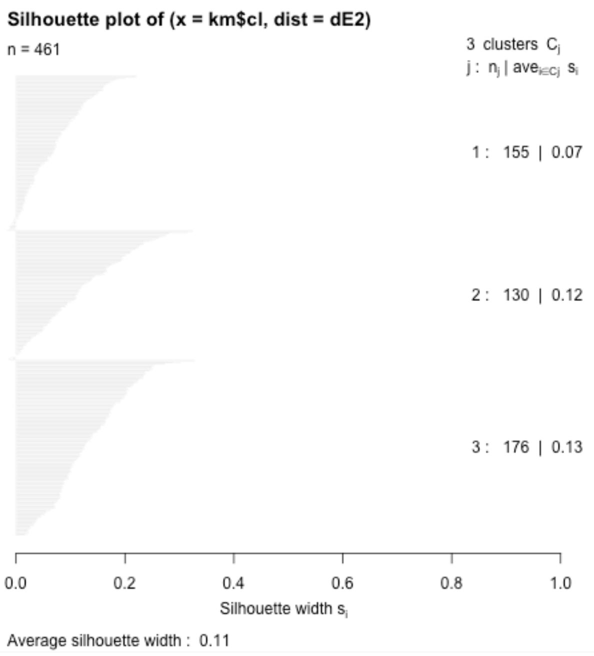


```
#dev.off()

#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/ClusterVisualPCA1PCA2.png",sep=""), width=480, height=480, bg='transpa
plot(PCAData[,1], PCAData[,2], main="PCA1 and PCA2 for all patients", xlab="PCA1", ylab="PCA2")
```

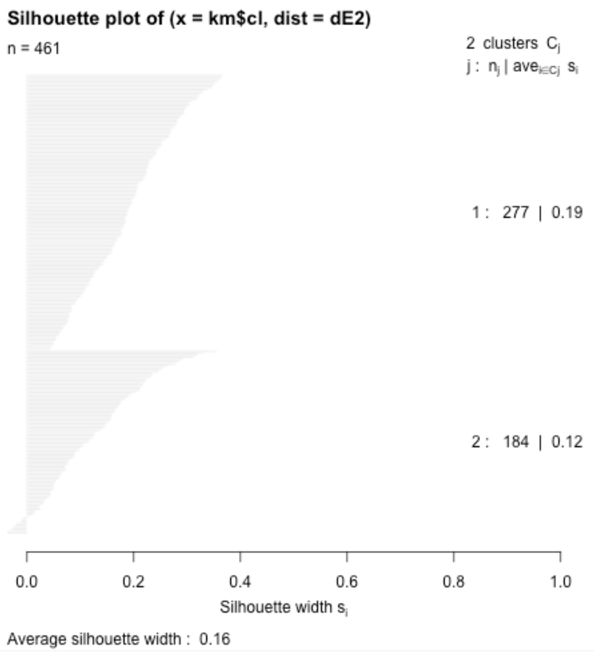


```
#dev.off()
#=======
#Plotting the clusters
library(cluster)
library(tools)
library(HSAUR)
km    <- kmeans(PCAData,3)
dissE <- daisy(PCAData)
```

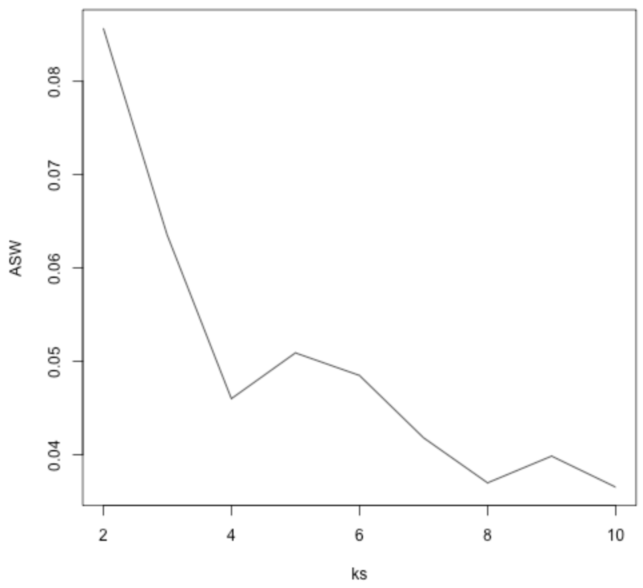
```
dE2   <- dissE^2
sk2   <- silhouette(km$cl, dE2)
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/ClusterVisualSilhouette3clusterPCA.png",sep=""), width=480, height=48

plot(sk2)
```



```
#dev.off()


#Plotting the clusters
library(cluster)
library(tools)
library(HSAUR)
km    <- kmeans(PCAData,2)
dissE <- daisy(PCAData)
dE2   <- dissE^2
sk2   <- silhouette(km$cl, dE2)
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/ClusterVisualSilhouette2clusterPCA.png",sep=""), width=480, height=48

plot(sk2)
```



```
#dev.off()


library(fpc)
d <- dist(PCAData)
#cluster.stats(d, km$cluster)
ks <- 2:10
ASW <- sapply(ks, FUN=function(k) {
  cluster.stats(d, kmeans(PCAData, centers=k, nstart=5)$cluster)$avg.silwidth
})
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/ClusterVisualSilhouettePCA.png",sep=""), width=480, height=480, bg='tı

plot(ks, ASW, type="l")
```



```
#dev.off()
```

Factor Analysis

Factor Analysis In this step we perform factor analysis. The code is located ***FactorAnalysisReduced.R***.

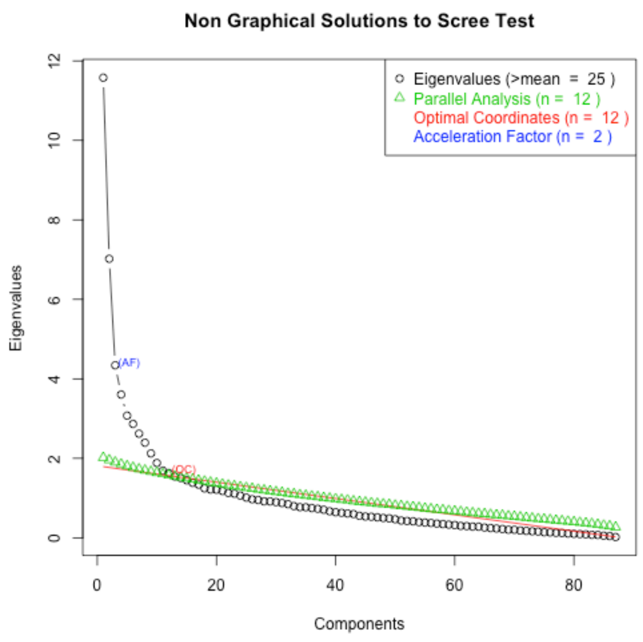
```
# In this program we run Factor analysis on our imputed data.
#For p = 130 (Number of variables in our data), the variance-covariance matrix Î£ contains
#(p*(p+1))/2=(130Ã—129)/2=8385
#p(m+1)=140(m+1)=8385 ==> m=65 Alos look at the results of the principal components analysis which in our case 53 PCA explained 80% variation in
# To look at the mathematical formulas please see the following websites:
# https://onlinecourses.science.psu.edu/stat505/node/79
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonReduced.csv",sep=",",header=T,stringsAsFactors = FALSE)
library (psych)
library( GPArotation)

#removing patient IDs
PatientIDs<-BioPerson[,1]
BioPerson<- BioPerson[,-(1)]
#calculate the correlation matrix
corMat <- cor(BioPerson)
names<-colnames(BioPerson)
report<-NULL
for(i in 1:ncol(BioPerson)){
  for(j in i:ncol(BioPerson)){
    if (i!=j && abs(corMat[i,j])>=0.75){
      #print(paste(names[i],names[j],correlation[i,j]))
      report<-rbind(report, cbind(names[i],names[j],corMat[i,j]))
    }
  }
}

print(report)
```

```
##       [,1]        [,2]         [,3]
##  [1,] "BDNF"      "TGFB1_LAP"  "0.808807520148811"
##  [2,] "BDNF"      "PDGF_BB"    "0.893912398968315"
##  [3,] "BMP6"       "FAS_LIG"    "0.806856902089423"
##  [4,] "CD40_LIG"   "EGF"        "0.889267889274508"
##  [5,] "CD40_LIG"   "TGFB1_LAP"  "0.820892430589892"
##  [6,] "CD40_LIG"   "PDGF_BB"    "0.770606572375412"
##  [7,] "CD40_LIG"   "SORTILIN"   "0.757428404418639"
##  [8,] "EGF"        "TGFB1_LAP"  "0.804013151616195"
##  [9,] "EGF"        "PDGF_BB"    "0.776907217390693"
## [10,] "ENA78"      "SCF"        "0.953804674156216"
## [11,] "IL16"       "IL1RA"      "0.880787312400816"
## [12,] "TGFB1_LAP"  "PDGF_BB"    "0.810722872098753"
## [13,] "TGFB1_LAP"  "SORTILIN"   "0.755683872368537"
## [14,] "MCP2"       "MCP4"       "0.75420322663316"
## [15,] "MMP10"      "MMP9_TOTAL" "0.780023267887322"
## [16,] "MMP9"       "MMP9_TOTAL" "0.81122089987677"
```

```
# Determine Number of Factors to Extract

library(nFactors)
```

```
## Loading required package: MASS
## Loading required package: boot
##
## Attaching package: 'boot'
##
## The following object is masked from 'package:psych':
##
##     logit
##
## The following object is masked from 'package:lattice':
##
##     melanoma
##
##
## Attaching package: 'nFactors'
##
## The following object is masked from 'package:lattice':
##
##     parallel
```

```
ev <- eigen(cor(BioPerson)) # get eigenvalues
ap <- parallel(subject=nrow(BioPerson),var=ncol(BioPerson),
               rep=100,cent=.05)
nS <- nScree(x=ev$values, aparallel=ap$eigen$qevpea)

#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/ScreePlotFactoranalysisBioPersonReduced.png",sep=""), width=480, heig
plotnScree(nS)
```



```
#dev.off()


#use fa() to conduct an orthogonal principal-axis exploratory factor analysis
#save the solution to an R variable
factors<-12
```

```
solution <- fa(r = corMat, nfactors = factors, rotate = "varimax", fm = "pa", max.iter = 10000)
loadingfactor<-as.table(solution$loadings)

#loadingfactortable<-cbind(Biomarke=rownames(loadingfactor),loadingfactor)

write.table(corMat, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/CorrelationMatrixBioPersonReduced.csv",row.names=FALSE, sep=',')
write.table(cbind(Biomarke=rownames(loadingfactor),loadingfactor), file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/LoadinFactorReduce

# We want to creat a table that lists the biomarkes within each factors that have loadingfactors more than 0.5.
#(leaving a soace between each group) so three rows one writes the name and second rows their corresponding loading factor and third a space.
highloadingfactor<-matrix(data='', nrow=(3*factors)+3,ncol = 1+max(colSums(loadingfactor>.5)))
BioNames<-dimnames(loadingfactor)[[1]]
PANames<-dimnames(loadingfactor)[[2]]

for(i in 1:factors){
  indexes<-loadingfactor[,i]>.5
  if(sum(indexes)>0){
    data<-sort(loadingfactor[indexes,i],decreasing = T)
    highloadingfactor[3*i-2+3,2:(length(data)+1)]<-BioNames[indexes]
    highloadingfactor[3*i-1+3,2:(length(data)+1)]<-loadingfactor[indexes,i]
    highloadingfactor[3*i-2+3,1]<-PANames[i]
  }

}
#Print the biomarkers that appears in more than one factor

highloadingfactor[1,1]<-"the biomarkers that appears in more than one factor"
data<-BioNames[rowSums(loadingfactor>0.5)>1]
if (length(data)>0){
  highloadingfactor[2,1:length(data)]<-data
}

write.table(highloadingfactor, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/HighLoadinFactorReduced.csv",row.names=FALSE,col.names
```

Demogeaphic Data

To analyze demogeaohic data we need to append it to BioPerson dara. The code is located in ***CombinBioPersonRemovedWtDemo.R***.

```
demoext<- read.csv("/Users/szarei2008/Documents/HarvardResearch/data/raw/download_11-28-2012/Clinical/demoext-1.csv",sep=",",header=T,stringsAsFa
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemoved.csv",sep=",",header=T,stringsAsFactors = FALSE)
#Since we had zero in the demoext table first we change them to NA.
demoext[demoext==0]<-NA

#The following are the variable of interests in demoext table.
demoextImportants<-c("AGE","HGTCM", "WGTKG","BMI","TOBANPY","EXACERB","BDLCOC","BDLCOPC","BFEV1A","BFEV1FVC","BFEV1PPA","BFVCA","BFVCPPA","BP151'
                     "BLP151","BLRA9101","BRA9101","BLVOL1","B6MWT","BBODE","BMMRC","BTLCPP","BRVPP","BFRCPP","BIC","BSGRQ","SEX","RACE")

#Only selct the columns that we have specified above.
#demoext[,demoextImportants]
#In here we are attaching a null matrix with the size specified below to original Bioperson table so we can
# add all new above 28 variables in form of columns to Bioperson table. (Original BioPerson size is 461:140 and new table BiopersonDemo is 461:1
BiopersonDemo<-cbind(BioPerson,matrix(data=NA,nrow=nrow(BioPerson),ncol=length(demoextImportants)))

#Adding column names of the important variables.
colnames(BiopersonDemo)[141:168]<-demoextImportants
# For every row of BiopersonDemo we fillout the 8 columns with their corresponding value from the demoext table.
# We use the subset command to pull out the specific patient ID from demoext that is the same value to BioPersonDemo row i column patientID.
#Next we say from that row that subset pulled out, we want the 28 most important variables listed above as demoextImportants
# Under SEX column we replaced female by value 0 and  male by 1.
#Under RACE column : Black = 1, Caucation =2 , Oriental=3, other=4

for (i in 1:nrow(BioPerson)){
  eachpersondata<-subset(demoext,PT==BiopersonDemo[i,"PATIENT.ID"])

  if(eachpersondata[1,"SEX"]=="MALE"){
    eachpersondata[1,"SEX"]<-1
  }
  else {
    eachpersondata[1,"SEX"]<-0
  }

  if(eachpersondata[1,"RACE"]=="BLACK"){
    eachpersondata[1,"RACE"]<-1
  }

  if(eachpersondata[1,"RACE"]=="CAUCASIAN"){
    eachpersondata[1,"RACE"]<-2
  }

  if(eachpersondata[1,"RACE"]=="ORIENTAL"){
    eachpersondata[1,"RACE"]<-3
  }
  if(eachpersondata[1,"RACE"]=="OTHER"){
    eachpersondata[1,"RACE"]<-4
  }

  BiopersonDemo[i,141:168]<- eachpersondata[1,demoextImportants]
}

write.table(BiopersonDemo, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemovedDemoext.csv",row.names=FALSE, sep=',')
```

Correlation: The correlation of biomarkers with the demographic data is analyzed below. The code is located in ***correlationBioPersonRemovedDemo.R***.

```
BioPersonRemovedDemoext<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonRemovedDemoext.csv",sep=",",header=T,strings
#Find the correlation between each biomarker with those important variable from the table demoextImportants which were 28 columns from demoext ta
correlation<-cor(BioPersonRemovedDemoext[,2:140],BioPersonRemovedDemoext[,141:166],method="pearson",use="pairwise.complete.obs")

#Total = 26 * 139 = 3614 combinations

rowname<-rownames(correlation)
colname<-colnames(correlation)
report<-NULL
for(i in 1:139){
  for(j in 1:26){
    if (abs(correlation[i,j])>=0.3){
      #print(paste(names[i],names[j],correlation[i,j]))
      report<-rbind(report, cbind(rowname[i],colname[j],correlation[i,j]))
    }
  }
}

print(report)
```

```
##       [,1]         [,2]        [,3]
##  [1,] "APOA1"      "WGTKG"     "-0.368852015693308"
##  [2,] "FABP"       "AGE"       "0.31096416164726"
##  [3,] "INSL"       "WGTKG"     "0.342460142219343"
##  [4,] "INSL"       "BMI"       "0.352043040505307"
##  [5,] "LEP"        "WGTKG"     "0.404877352310211"
##  [6,] "LEP"        "BMI"       "0.635466782727167"
##  [7,] "LEP"        "BFVCA"     "-0.34297026692392"
##  [8,] "LEP"        "BLVOL1"    "-0.392103498826149"
##  [9,] "OPG"        "AGE"       "0.304318956894352"
## [10,] "TF"         "EXACERB"   "0.850261582722961"
## [11,] "TF"         "BP151"     "0.378217488636363"
## [12,] "TF"         "BLP151"    "0.359350986636472"
## [13,] "TF"         "BLRA9101" "-0.341848089180906"
## [14,] "TF"         "BRA9101"   "-0.401437966265873"
## [15,] "TF"         "BRVPP"     "-0.317383493906417"
## [16,] "EPIREGULIN" "HGTCM"     "0.304876331494061"
## [17,] "EPIREGULIN" "WGTKG"     "0.350944102498571"
## [18,] "S100B"      "BMI"       "-0.315241136670697"
## [19,] "S100B"      "EXACERB"   "-0.427550860671566"
## [20,] "S100B"      "BFEV1A"    "0.376880569283704"
## [21,] "S100B"      "BFEV1PPA" "0.419005222365932"
## [22,] "S100B"      "BFVCA"     "0.36310447490924"
## [23,] "S100B"      "B6MWT"     "0.532788951257585"
## [24,] "S100B"      "BBODE"     "-0.701541756963285"
## [25,] "S100B"      "BTLCPP"    "0.59120111013022"
## [26,] "S100B"      "BRVPP"     "0.482845264257135"
## [27,] "S100B"      "BIC"       "0.713091384511969"
```

P-Value analysis In addition to the correlation, it is important to look at the p-values as well. The code is located in ***PValuecorrelationBioPersonReducedDemo***.

```
BioPersonRemovedDemoext<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonReducedDemoext.csv",sep=",",header=T,strings
#Find the correlation between each biomarker with those important variable from the table demoextImportants which were 28 columns from demoext ta
#28 demograhics and one for the patient ID
biomarkers<-ncol(BioPersonRemovedDemoext)-28-1
#only last 26 because we are igonoring sex and race
ktests<-matrix(ncol = biomarkers,nrow = 28)
ptests<-matrix(ncol = biomarkers,nrow = 28)
ptestCor<-matrix(ncol = biomarkers,nrow = 28)
bothtests<-matrix(ncol = biomarkers,nrow = 28)
for(i in 1:biomarkers){
  for(j in 1:28){
    x<-BioPersonRemovedDemoext[,1+i]
    y<-BioPersonRemovedDemoext[,biomarkers+1+j]
    ktest<-cor.test(x,y,alternative = "two.sided", method = "kendall")
    ptest<-cor.test(x,y,alternative = "two.sided", method = "pearson")
    ktests[j,i]<-ktest$p.value
    ptests[j,i]<-ptest$p.value
    ptestCor[j,i]<-ptest$estimate
    if(ktests[j,i]< .05 & ptests[j,i]<.05){
      bothtests[j,i]<-"Sig"
    }else{
      bothtests[j,i]<-"NS"
    }
  }
}

rownames(ptests)<-colnames(BioPersonRemovedDemoext)[(biomarkers+2):(biomarkers+1+28)]
colnames(ptests)<-colnames(BioPersonRemovedDemoext)[(2):(biomarkers+1)]
rownames(ktests)<-rownames(ptests)
```

```
colnames(ktests)<-colnames(ptests)
rownames(bothtests)<-rownames(ptests)
colnames(bothtests)<-colnames(ptests)
rownames(ptestCor)<-rownames(ptests)
colnames(ptestCor)<-colnames(ptests)

write.table(cbind(Demo=rownames(ptests),ptests), file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/PvalueBioReducedDemoPearson.csv",row
write.table(cbind(Demo=rownames(ptests),ptestCor), file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/CorOfPvalueBioReducedDemoPearson.c
write.table(cbind(Demo=rownames(ptests),ktests), file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/PvalueBioReducedDemoKendall.csv",row
write.table(cbind(Demo=rownames(ptests),bothtests), file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/PvalueBioReducedDemo2Pearson_Kend
```

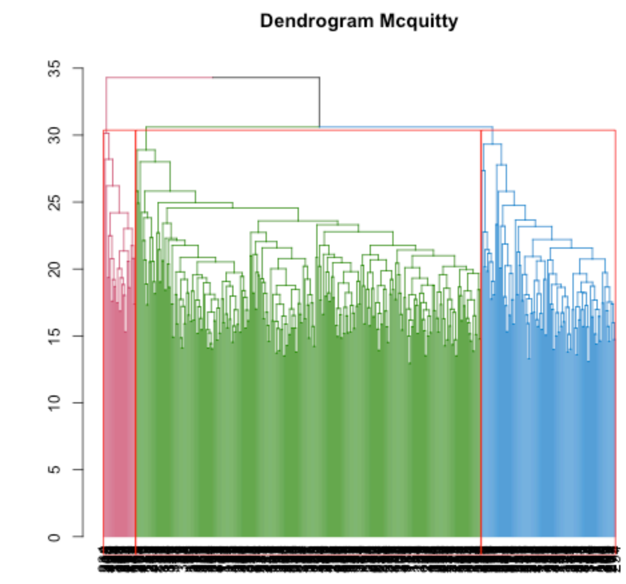
Hierarchical Clustering

Performing hierarchical clustering

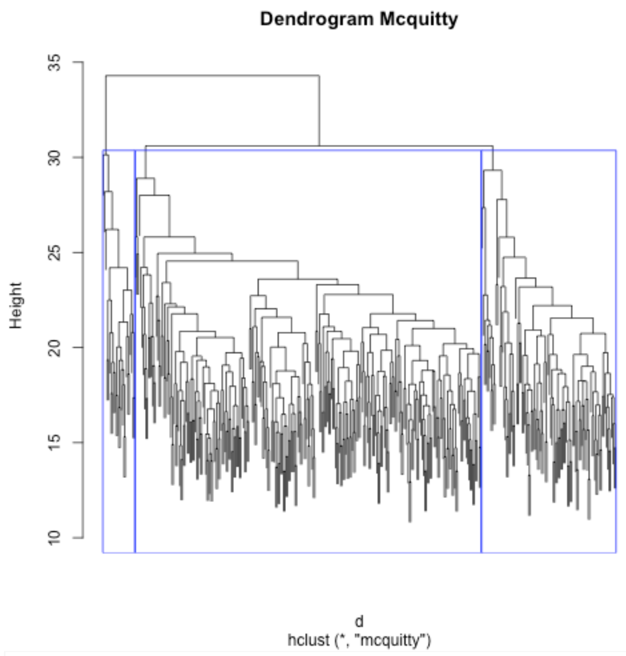
```
BioPerson<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonReduced.csv",sep=",",header=T,stringsAsFactors = FALSE)
#removing patient IDs
PatientIDs<-BioPerson[,1]
BioPerson<- BioPerson[,-(1)]
#==================================================================================================================================================
#Hierarchical clustering, Method:mcquitty, Distance: canberra
library(dendextend)
```

```
##
## Welcome to dendextend version 1.0.1
##
## Type ?dendextend to access the overall documentation and
## browseVignettes(package = 'dendextend') for the package vignette.
## You can execute a demo of the package via: demo(dendextend)
##
## More information is available on the dendextend project web-site:
## https://github.com/talgalili/dendextend/
##
## Contact: <tal.galili@gmail.com>
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
##
##                      To suppress the this message use:
##                      suppressPackageStartupMessages(library(dendextend))
##
##
## Attaching package: 'dendextend'
##
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(colorspace)

d<-dist(BioPerson, method = "canberra")
hierclust<-hclust(d,method="mcquitty")
dend <- as.dendrogram(hierclust,leaflab="none")
# order it the closest we can to the order of the observations:
#dend <- rotate(dend,1:396)
# Color the branches based on the clusters:
dend <- color_branches(dend, k=3, labels=FALSE) #, groupLabels=iris_species)
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/DendogramMcquittyReducedColor.png",sep=""), width=480, height=480, bg=
plot(dend,horiz =F, main="Dendrogram Mcquitty")
#plot(hierclust,labels=F, main="Dendrogram Mcquitty")
rect.hclust(hierclust,k=3,border="red")
```



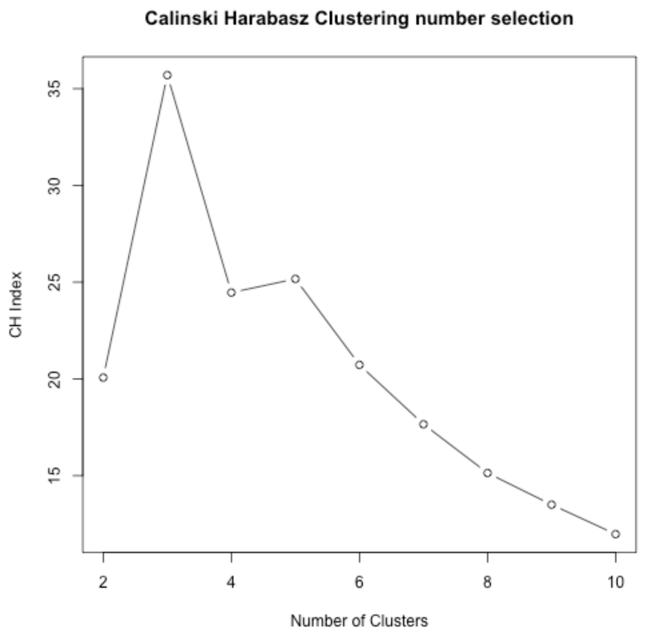
```
#dev.off()
#------------
#Plot Black and white dendogram
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/DendogramMcquittyReducedBW.png",sep=""), width=480, height=480, bg='wl
plot(hierclust,labels=F, main="Dendrogram Mcquitty")
rect.hclust(hierclust,k=3,border="blue")
```



```
#dev.off()

#==========================================================================================================================================

# NbClust package and finding cluster number methods
#==========================================================================================================================================

library(NbClust)
#Using ch as index, ch ==> Calinski Harabasz 1974
nb <- NbClust(BioPerson, diss=NULL, distance = "canberra", min.nc=2, max.nc=10, method = "mcquitty", index = "ch", alphaBeale = 0.1)
print(paste("best number of clusters: ", nb$Best.nc[1]))
```

```
## [1] "best number of clusters:  3"
```

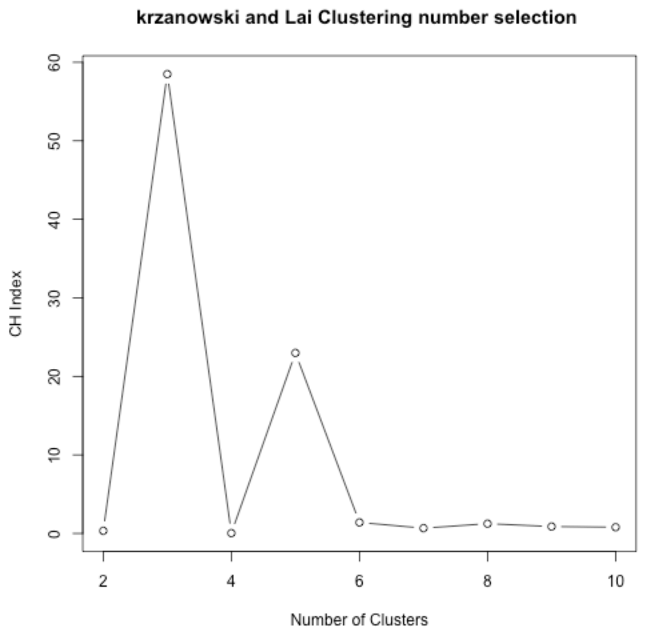
```
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/NBClustReducedChMethod.png",sep=""), width=480, height=480, bg='transp
plot(2:10,nb$All.index,type = "b", main = "Calinski Harabasz Clustering number selection", xlab = "Number of Clusters", ylab = "CH Index")
```



```
#dev.off()
#---------
#Using kl as index, kl ==> krzanowski and Lai 1988
nb <- NbClust(BioPerson, diss=NULL, distance = "canberra", min.nc=2, max.nc=10, method = "mcquitty", index = "kl", alphaBeale = 0.1)
print(paste("best number of clusters: ", nb$Best.nc[1]))
```

```
## [1] "best number of clusters:  3"
```

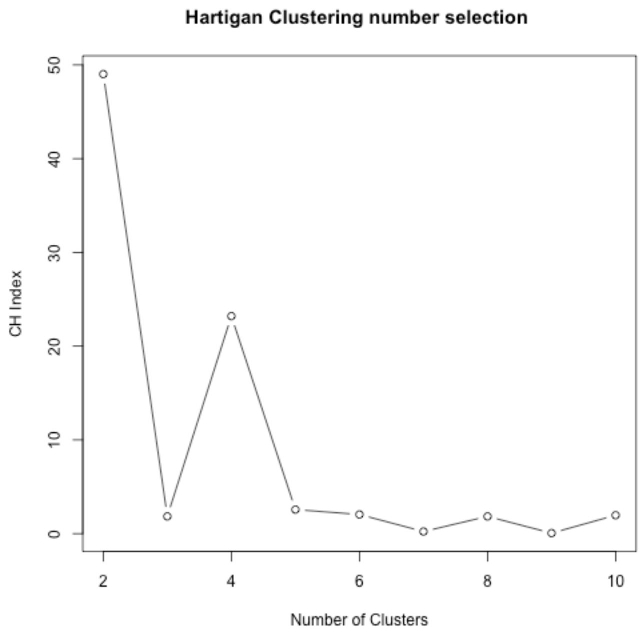
```
#par(mfrow=c(1,1))
#png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/NBClustReducedKlMethod.png",sep=""), width=480, height=480, bg='transp
plot(2:10,nb$All.index,type = "b", main = "krzanowski and Lai Clustering number selection ", xlab = "Number of Clusters", ylab = "CH Index")
```



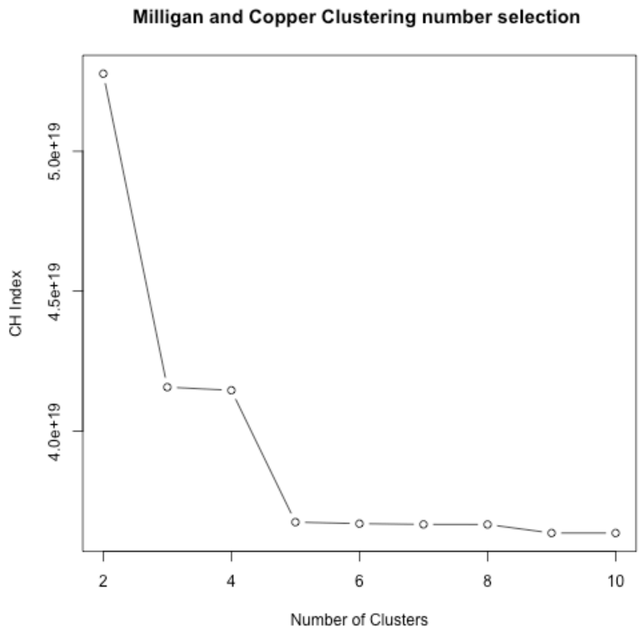
```
#dev.off()

methods=c("hartigan", "trcovw", "rubin","cindex","duda", "pseudot2","beale")
```

```
methodsName=c("Hartigan","Milligan and Copper","Friedman and Rubin","Huber and Levin", "Duda and Hart","Duda and Hart","Beale")
for (i in 1:length(methods)){
  print(methods[i])
  nb <- NbClust(BioPerson, diss=NULL, distance = "canberra", min.nc=2, max.nc=10, method = "mcquitty", index = methods[i], alphaBeale = 0.1)
  print(paste("best number of clusters: ", nb$Best.nc[1]))
  #par(mfrow=c(1,1))
  #png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/NBClustReduced",methods[i],"Method.png",sep=""), width=480, height=4
  plot(2:10,nb$All.index,type = "b", main = paste(methodsName[i],"Clustering number selection "), xlab = "Number of Clusters", ylab = "CH Index")
  #dev.off()
}
```

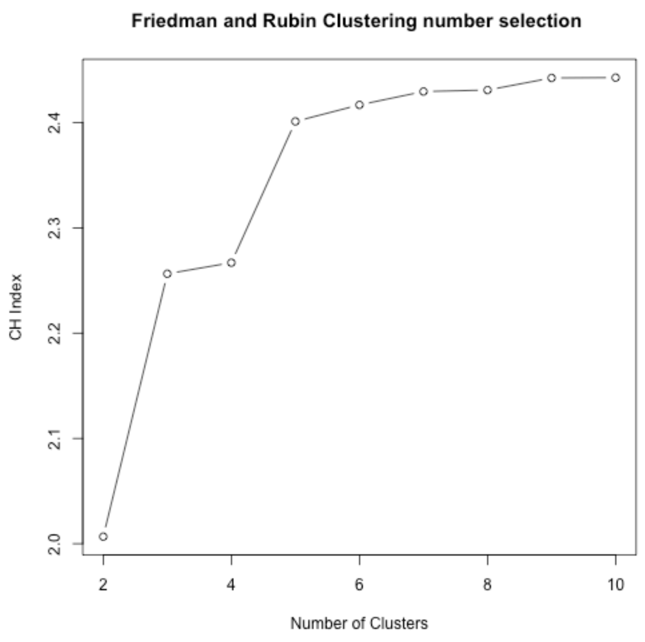
```
## [1] "hartigan"
## [1] "best number of clusters:  3"
```



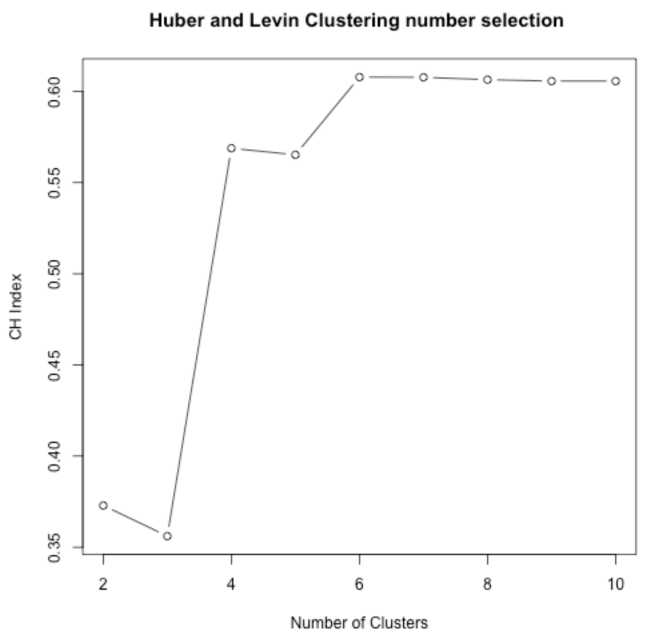
```
## [1] "trcovw"
## [1] "best number of clusters:  3"
```



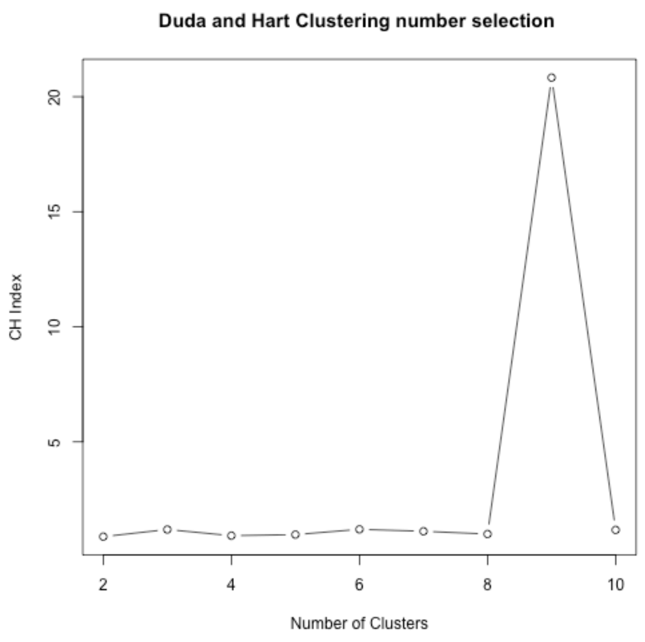
```
## [1] "rubin"
## [1] "best number of clusters:  3"
```



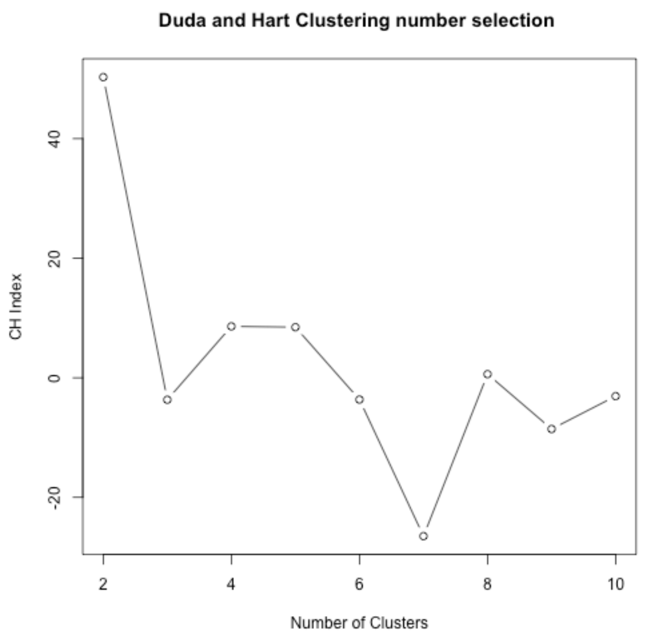
```
## [1] "cindex"
## [1] "best number of clusters:  3"
```



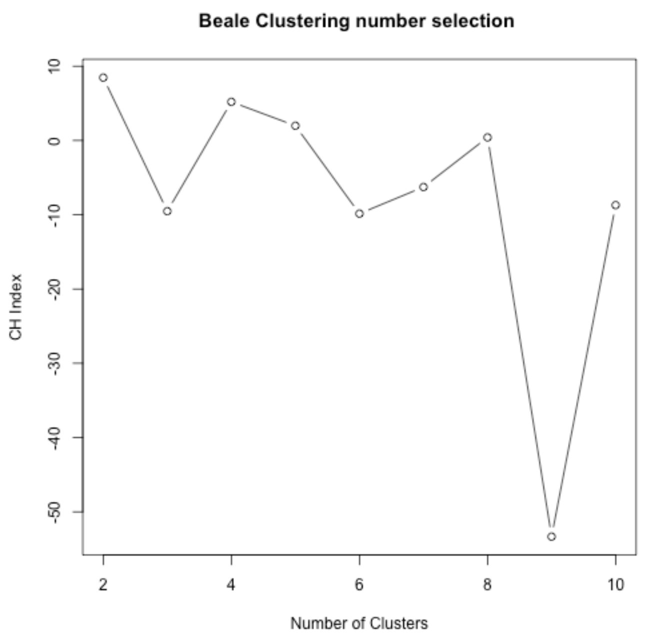
```
## [1] "duda"
## [1] "best number of clusters:  3"
```



```
## [1] "pseudot2"
## [1] "best number of clusters:  3"
```



```
## [1] "beale"
## [1] "best number of clusters:  3"
```



```
#Using hartigan as index, hartigan ==> Hartigan 1975

#Using trcovw as index, hartigan ==> Milligan and Copper 1975

#Using rubin as index, hartigan ==> Friedman and Rubin 1967 : In the comments of the plot you can call this Fiedman as well. In other paper he me

#Using cindex as index, hartigan ==> Huber and Levin 1976

#Using duda as index, hartigan ==> duda and hart 1973

#Using pseudot2 as index, hartigan ==> duda and hart 1973

#Using beale as index, hartigan ==> Beale 1969

#=====================================================================================================================================
# Pull out the clustring result
#=====================================================================================================================================
clusters<-cutree(hierclust,k=3)
BioPersonDemo<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonReducedDemoext.csv",sep=",",header=T,stringsAsFactors
BioPersonDemoClust<-cbind(BioPersonDemo,clusters=clusters)

VarNames<-colnames(BioPersonDemo)
for(i in 1:3){
  BioPersonDemoClusteri<-subset(BioPersonDemoClust,clusters==i)
  alltogether<-NULL
  for(j in 2:ncol(BioPersonDemo)){
    statsummary<-summary(na.omit(BioPersonDemoClusteri[,j]))
    StatSD<-sd(na.omit(BioPersonDemoClusteri[,j]))
    combined<-cbind(Variable.Name=VarNames[j],rbind(statsummary),SD=StatSD)
    alltogether<-rbind(alltogether,combined)
  }
  write.table(alltogether,file=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HierClusterBioPersonReduced__",i,".csv",sep=""),row.n
}
write.table(BioPersonDemoClust,file=("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonDemoClustReduced.csv"),row.names=FALSE,sep='


sum(clusters==1)
```

```
## [1] 267
```

```
sum(clusters==2)
```

```
## [1] 104
```

```
sum(clusters==3)
```

```
## [1] 25
```

Combining the data from hierarchical clustering with the factor analysis.

```
factorsData<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HighLoadinFactorReduced12.csv",sep=",",header=F,stringsAsFactors
cluster1Data<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HierClusterBioPersonReduced__1.csv",sep=",",header=T,stringsAsFa
cluster2Data<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HierClusterBioPersonReduced__2.csv",sep=",",header=T,stringsAsFa
cluster3Data<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HierClusterBioPersonReduced__3.csv",sep=",",header=T,stringsAsFa
rownames(cluster1Data)<-cluster1Data[,1]
cluster1Data<-cluster1Data[,(-1)]
rownames(cluster2Data)<-cluster2Data[,1]
cluster2Data<-cluster2Data[,(-1)]
rownames(cluster3Data)<-cluster3Data[,1]
cluster3Data<-cluster3Data[,(-1)]
colnames(cluster1Data)<-c("C1.Min",  "C1.1st.Qu.",     "C1.Median",       "C1.Mean",   "C1.3rd.Qu.",       "C1.Max",           "C1.SD")
colnames(cluster2Data)<-c("C2.Min",  "C2.1st.Qu.",     "C2.Median",       "C2.Mean",   "C2.3rd.Qu.",       "C2.Max",           "C2.SD")
colnames(cluster3Data)<-c("C3.Min",  "C3.1st.Qu.",     "C3.Median",       "C3.Mean",   "C3.3rd.Qu.",       "C3.Max",           "C3.SD")

nfactors<-11

#for each factor get the list of biomarkers
alltogether<-NULL
for(i in 1:nfactors){
  namerow<-3*i+1
  datarow<-3*i+2
  biomarkers<-factorsData[namerow,2:ncol(factorsData)]
  biomarkersLoading<-factorsData[datarow,2:ncol(factorsData)]
  if(biomarkers[ncol(factorsData)-1]==""){
    lastindex<-which.max(biomarkers=="")-1
    biomarkers<-biomarkers[1:lastindex]
    biomarkersLoading<-biomarkersLoading[1:lastindex]
  }
  names(biomarkersLoading)<-biomarkers
  biomarkersLoading<-sort(biomarkersLoading,decreasing = T)
  biomarkers<-names(biomarkersLoading)
  #now for each biomarkers in current factor we need to get the data from each cluster
  for(j in 1:length(biomarkers)){
    names(biomarkersLoading)[j]<-"Loading.Factor"
    onerow<-cbind(Factor=factorsData[namerow,1],Biomarker=biomarkers[j],biomarkersLoading[j],cluster1Data[biomarkers[j],],cluster2Data[biomarkers
    alltogether<-rbind(alltogether,onerow)
  }

}

write.table(alltogether,file=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HierClusterBioPersonReducedFactorTogether.csv",sep=""),


#addind demographic data
demostart<-nrow(cluster1Data)-28+1
varnames<-rownames(cluster1Data)
alltogether<-NULL
#i :0:25
for(i in 0:25){
rownumber<-i+demostart
onerow<-cbind(Demog.Info=varnames[rownumber],cluster1Data[varnames[rownumber],],cluster2Data[varnames[rownumber],],cluster3Data[varnames[rownumbe
alltogether<-rbind(alltogether,onerow)
}
emptyline<-" "
write.table(emptyline,file=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HierClusterBioPersonReducedFactorTogether.csv",sep=""),ap

write.table(alltogether,file=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/HierClusterBioPersonReducedFactorTogether.csv",sep=""),
```

t-Test Next we run Anova and tTest on the clusters. The code is in ***AnovaTukey.R***.

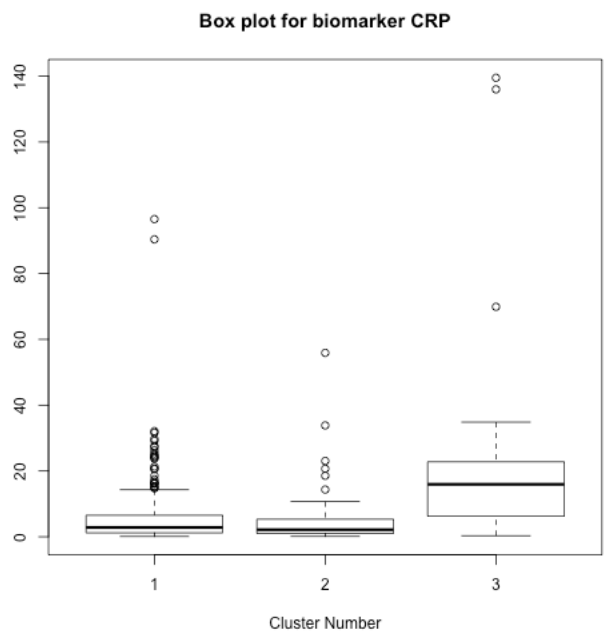
```
# In this program we run t stat test on three clusters for each biomarker value
DemoCluster<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonDemoClustReduced.csv",sep=",",header=T,stringsAsFactors

variablenames<-colnames(DemoCluster)
#excluding patient ID, Sex, race and clusters columns
variablenames<-variablenames[2:(length(variablenames)-3)]

#In here we are sorting the data based on clusrer number on the last column. key of sorting is "cluster"
require(data.table)
```

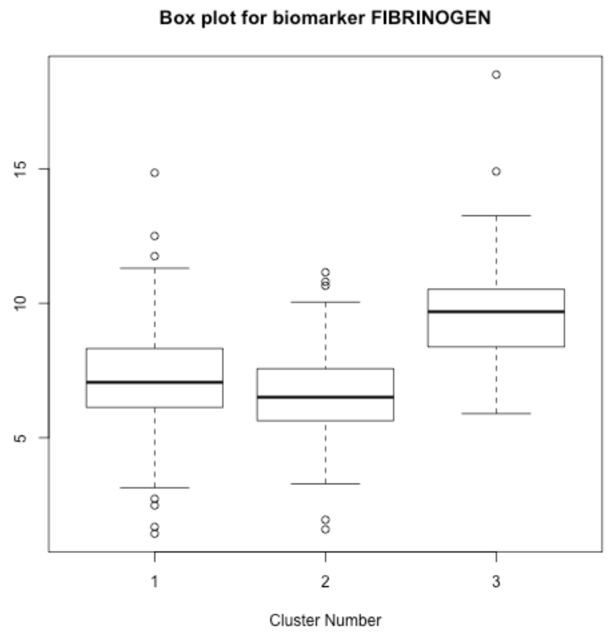
```
## Loading required package: data.table
##
## Attaching package: 'data.table'
##
## The following object is masked from 'package:dendextend':
##
##     set
```

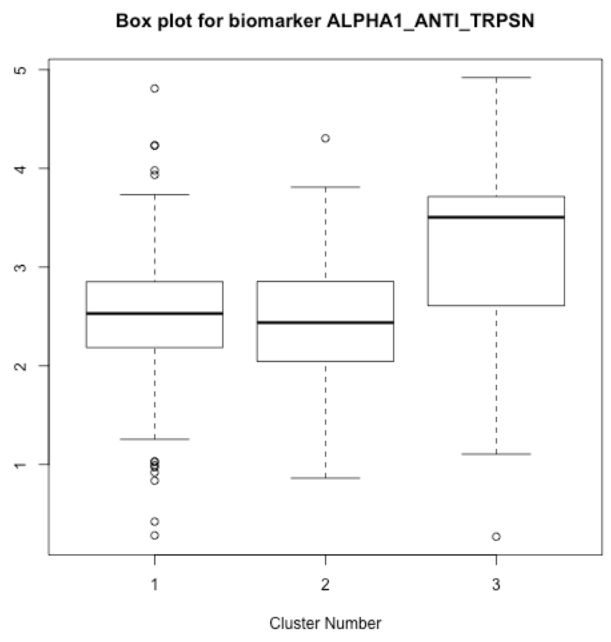
```
data<-data.table(DemoCluster,key="clusters")
data<-as.data.frame(data)
report<-NULL
for (i in 1:length(variablenames)){
  biomarkername<-variablenames[i]
  #choose biomarker and clusters columns.
  biocluster<-data[,c(biomarkername,"clusters")]
  colnames(biocluster)<-c("biomarker","clusters")
  biocluster$clusters<-factor(biocluster$clusters)
  aov.out<-aov(biomarker~clusters, data=biocluster)
  anovaP<-anova(aov.out)[1,5]
  #TukeyHSD(aov.out)
  tukey<-TukeyHSD(aov.out)
  oneRow<-cbind(Variable=biomarkername, Anova.P=anovaP,t(tukey$clusters[,4]))
  report<-rbind(report,oneRow)
  par(mfrow=c(1,1))
  png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/AnovaTukeyPlots/AnovaTukeyPlot_",biomarkername,"_.png",sep=""), widtl
  plot(tukey)
  title(xlab=paste("for",biomarkername), line=4)
  dev.off()
}

write.table(report, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/AnovaTukeyReportReduced.csv",row.names=FALSE, sep=',')
```

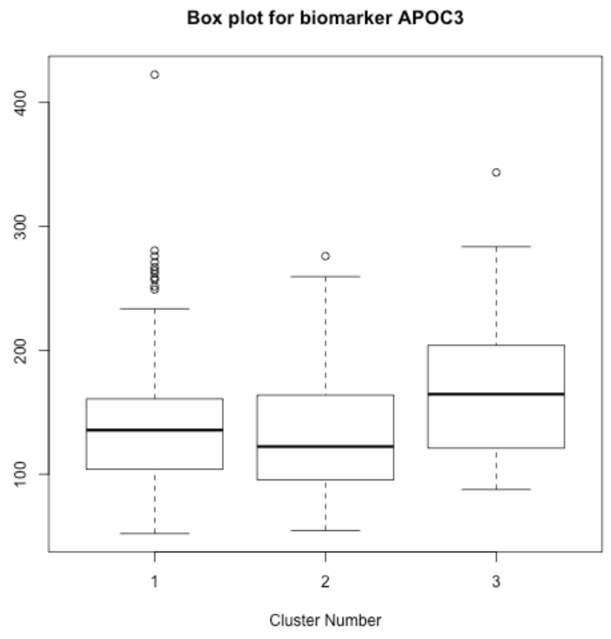
Cluster 3 vs Cluster 1 and 2: Next we compare cluster 3 vs cluster 1 and 2 and also plot the box plots. The code is in ***AnovaTukey3Vs1and2.R***.

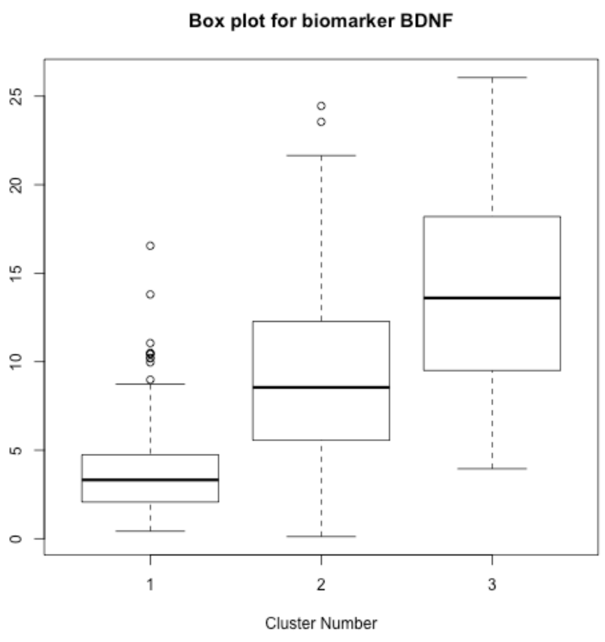
```
# In this program we run t stat test on three clusters for each biomarker value
DemoCluster<- read.table("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BioPersonDemoClustReduced.csv",sep=",",header=T,stringsAsFactors

#renaming all of the clsuter 2s with cluster 1
#indexes=DemoCluster[,"clusters"]==2
#DemoCluster[indexes,"clusters"]<-1

variablenames<-colnames(DemoCluster)
#excluding patient ID, Sex, race and clusters columns
variablenames<-variablenames[2:(length(variablenames)-3)]

keepOnlyDesiredData<-TRUE
#In here we are sorting the data based on clusrer number on the last column. key of sorting is "cluster"
require(data.table)
data<-data.table(DemoCluster,key="clusters")
data<-as.data.frame(data)
report<-NULL
for (i in 1:length(variablenames)){
  biomarkername<-variablenames[i]
  #choose biomarker and clusters columns.
  biocluster<-data[,c(biomarkername,"clusters")]
  colnames(biocluster)<-c("biomarker","clusters")
  biocluster$clusters<-factor(biocluster$clusters)
  aov.out<-aov(biomarker~clusters, data=biocluster)
  anovaP<-anova(aov.out)[1,5]
  if(anovaP>0.05 && keepOnlyDesiredData){
    next
  }
  #TukeyHSD(aov.out)
  tukey<-TukeyHSD(aov.out)
  oneRow<-cbind(Variable=biomarkername, Anova.P=anovaP,t(tukey$clusters[,4]))
  report<-rbind(report,oneRow)
  if(tukey$clusters[1,4]<=0.05 && tukey$clusters[2,4]<=0.05 && tukey$clusters[3,4]<=0.05){
    #par(mfrow=c(1,1))
    #png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BoxPlot3Clusters/BoxPlot_",biomarkername,"_.png",sep=""), width=4
    boxplot(biomarker~clusters, data=biocluster, main=paste("Box plot for biomarker", biomarkername),xlab="Cluster Number")
    #dev.off()
  }

  if(tukey$clusters[1,4]>0.05 && tukey$clusters[2,4]<=0.05 && tukey$clusters[3,4]<=0.05){
    #par(mfrow=c(1,1))
    #png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/BoxPlot3rdCluster/BoxPlot_",biomarkername,"_.png",sep=""), width=
    boxplot(biomarker~clusters, data=biocluster, main=paste("Box plot for biomarker", biomarkername),xlab="Cluster Number")
    #dev.off()
  }


  #par(mfrow=c(1,1))
  #png(filename=paste("/Users/szarei2008/Documents/HarvardResearch/SZscripts/AnovaTukeyPlots/AnovaTukeyPlot_",biomarkername,"_.png",sep=""), wid
  #plot(tukey)
  #title(xlab=paste("for",biomarkername), line=4)
  #dev.off()
}
```

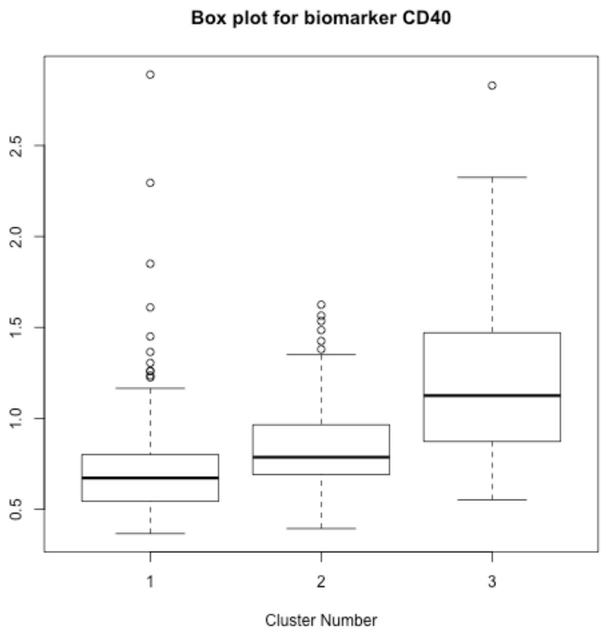

Box plot for biomarker CRP
0
20
40
60
80
100
120
140
1
2
3
Cluster Number

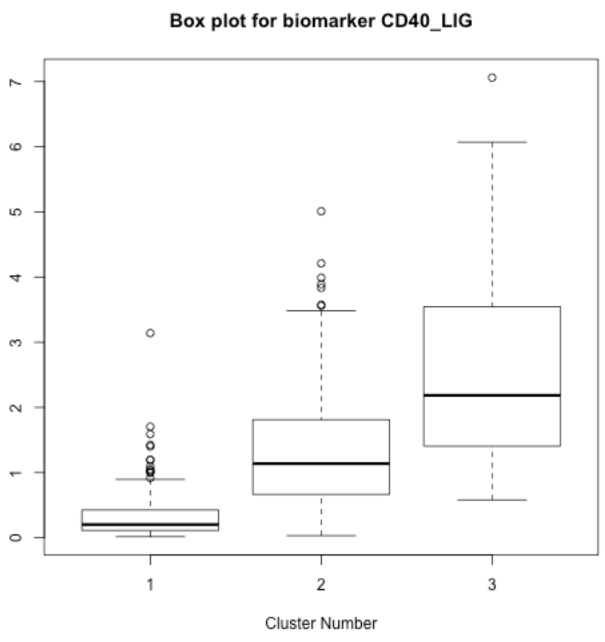

Box plot for biomarker FIBRINOGEN
5
10
15
1
2
3
Cluster Number

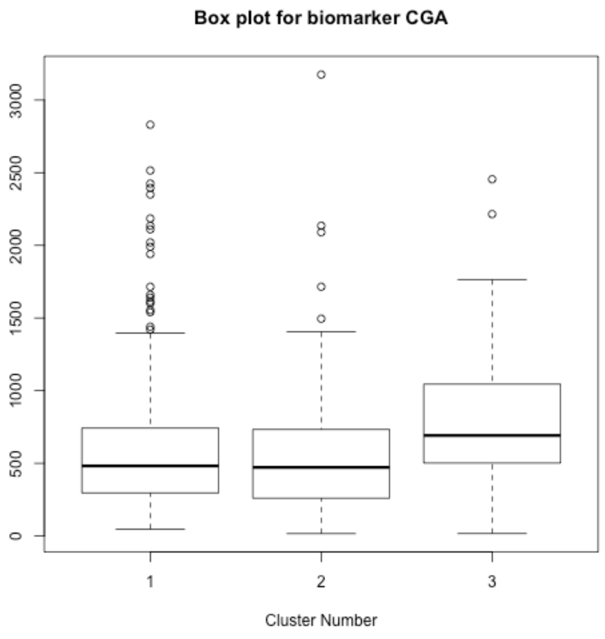

Box plot for biomarker ALPHA1_ANTI_TRPSN
1
2
3
4
5
1
2
3
Cluster Number

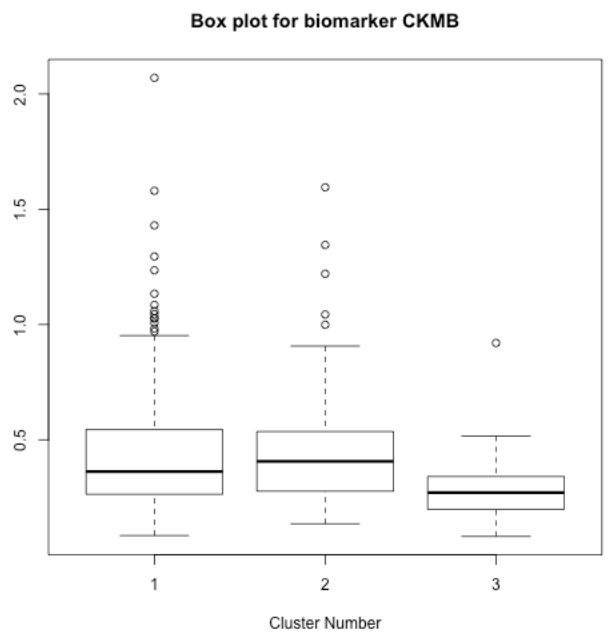

Box plot for biomarker APOC3
100
200
300
400
1
2
3
Cluster Number

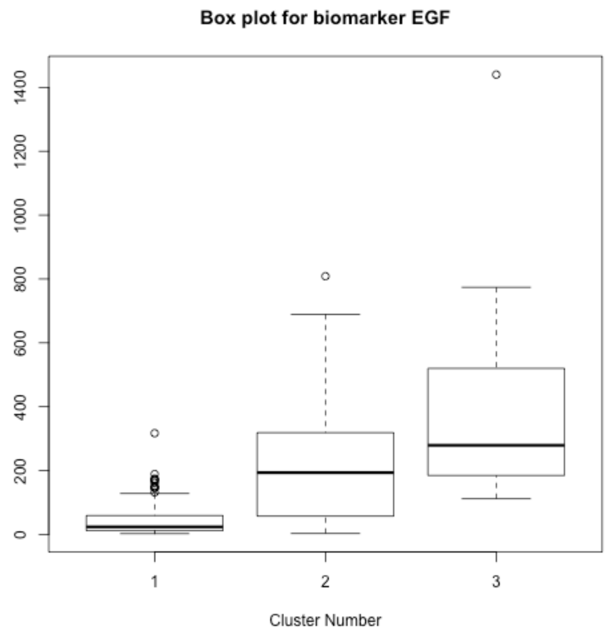

Box plot for biomarker BDNF
0
5
10
15
20
25
1
2
3
Cluster Number

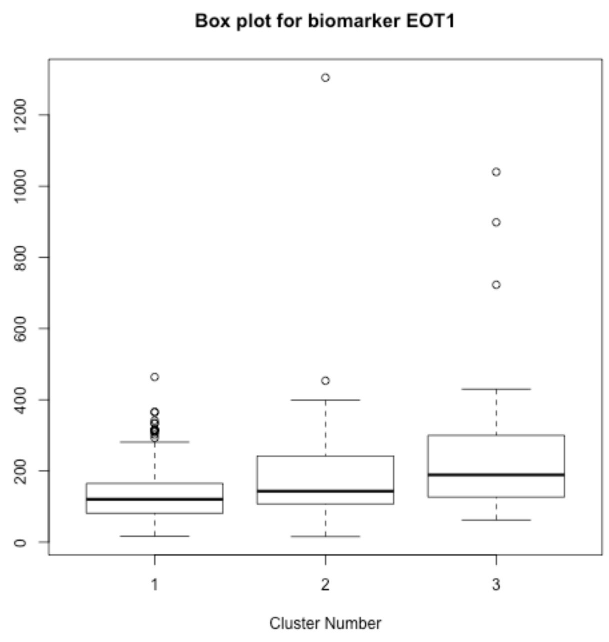

Box plot for biomarker CD40
0.5
1.0
1.5
2.0
2.5
1
2
3
Cluster Number

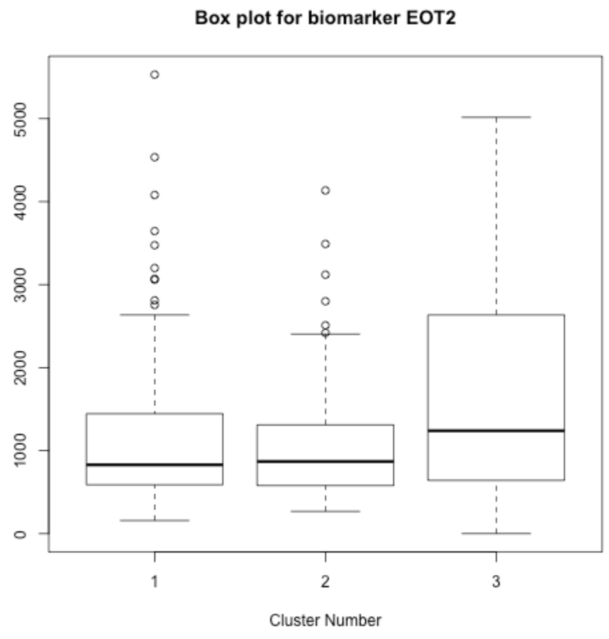

Box plot for biomarker CD40_LIG
0
1
2
3
4
5
6
7
1
2
3
Cluster Number

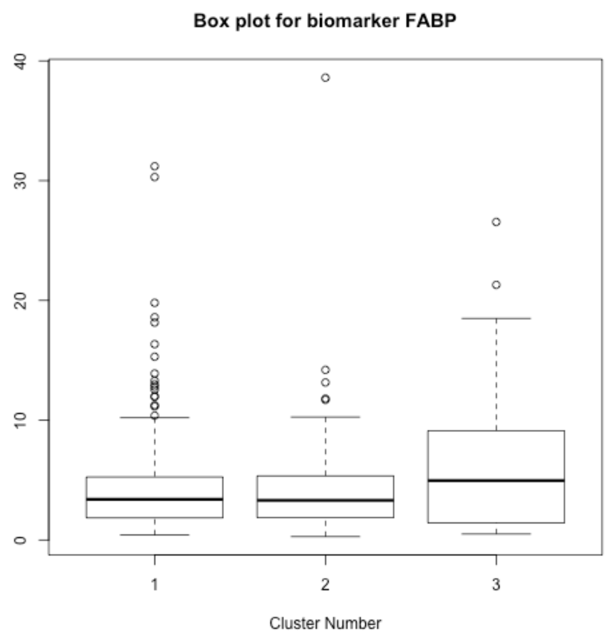

Box plot for biomarker CGA
0
500
1000
1500
2000
2500
3000
1
2
3
Cluster Number

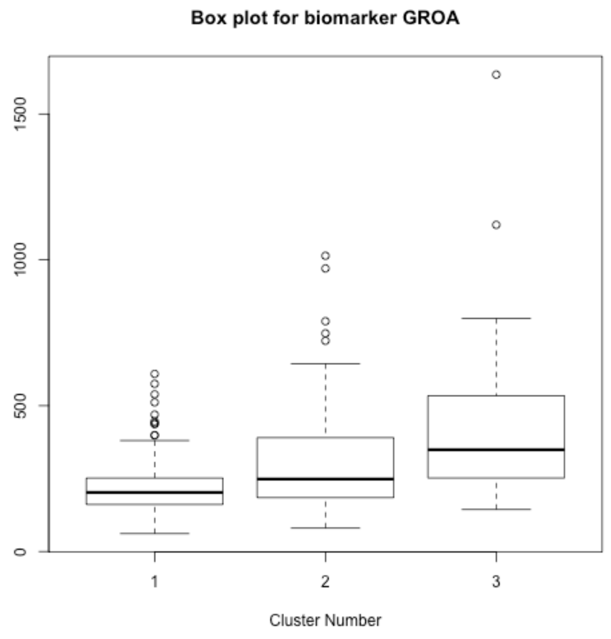

Box plot for biomarker CKMB
2.0
1.5
1.0
0.5
1
2
3
Cluster Number

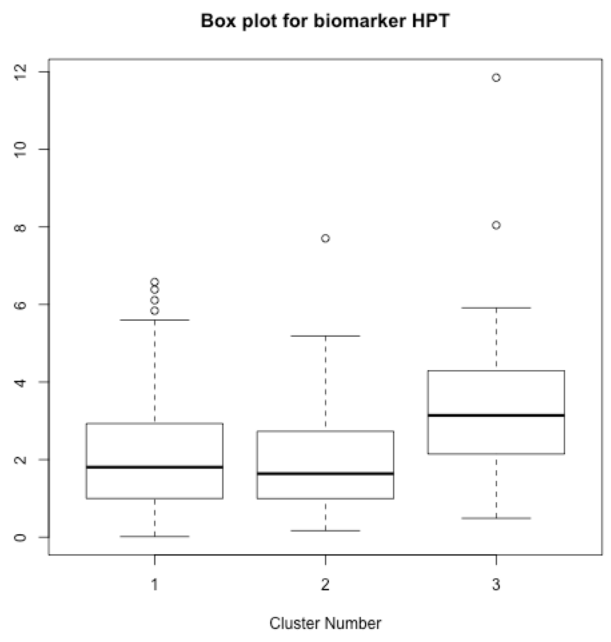

Box plot for biomarker EGF
1400
1200
1000
800
600
400
200
0
1
2
3
Cluster Number

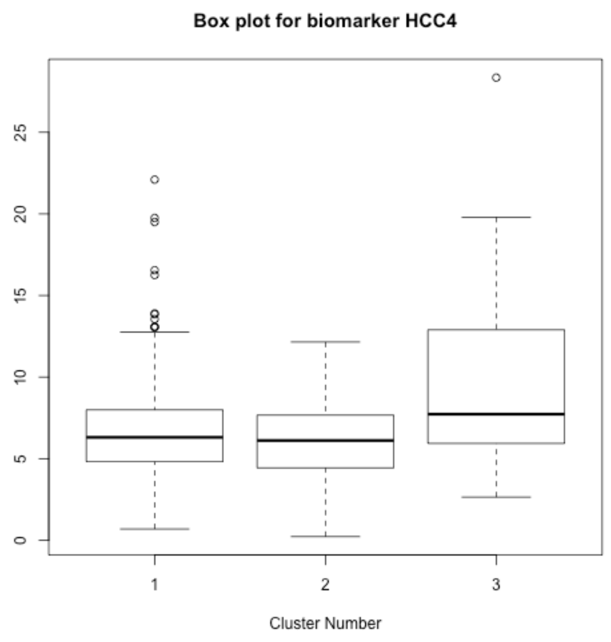

Box plot for biomarker EOT1
0
200
400
600
800
1000
1200
1
2
3
Cluster Number

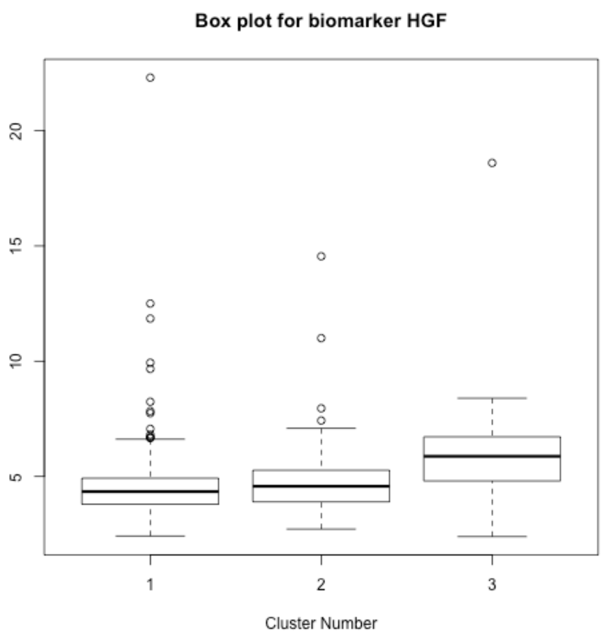

Box plot for biomarker EOT2
0
1000
2000
3000
4000
5000
1
2
3
Cluster Number

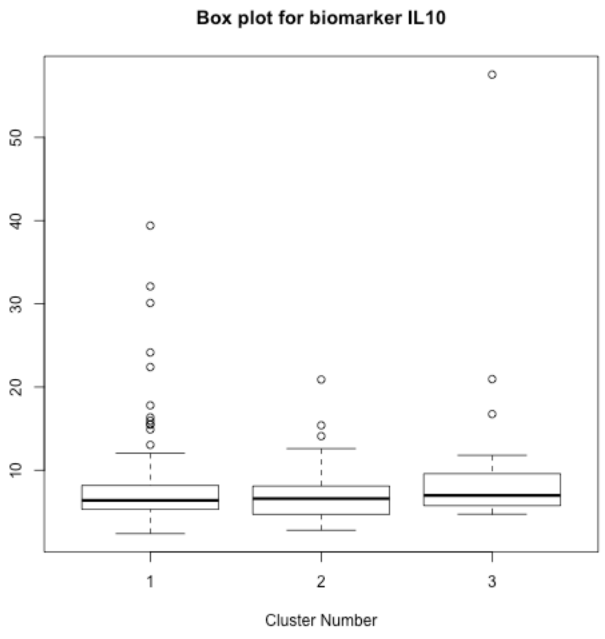

Box plot for biomarker FABP
0
10
20
30
40
1
2
3
Cluster Number

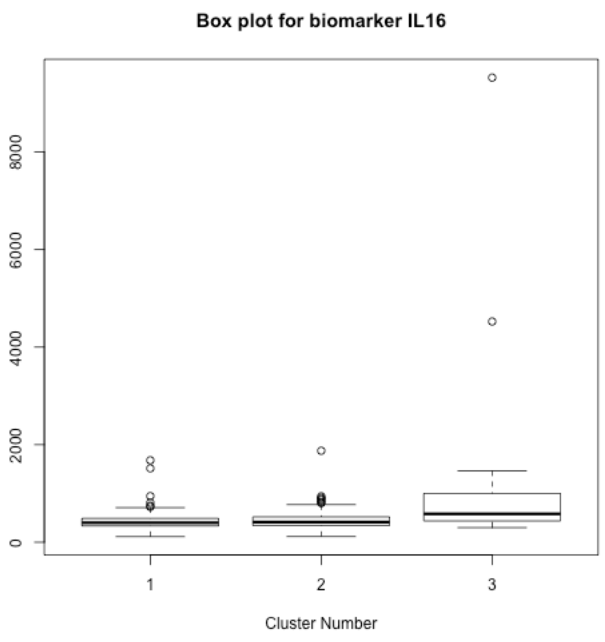

Box plot for biomarker GROA
0
500
1000
1500
1
2
3
Cluster Number

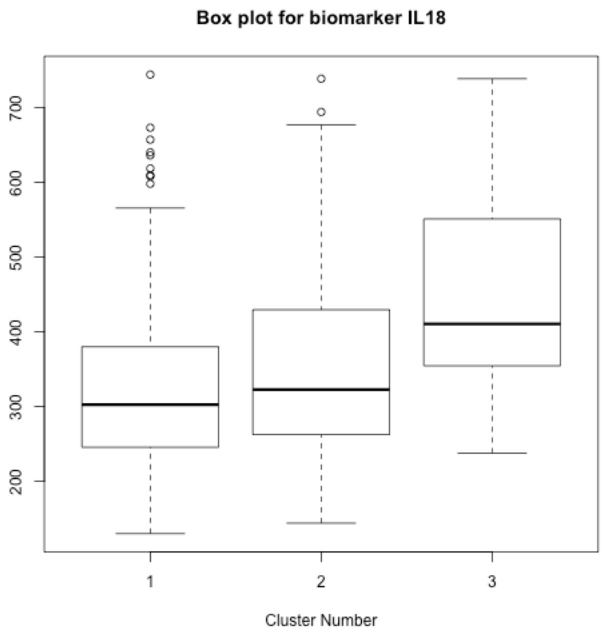

Box plot for biomarker HPT
0
2
4
6
8
10
12
1
2
3
Cluster Number

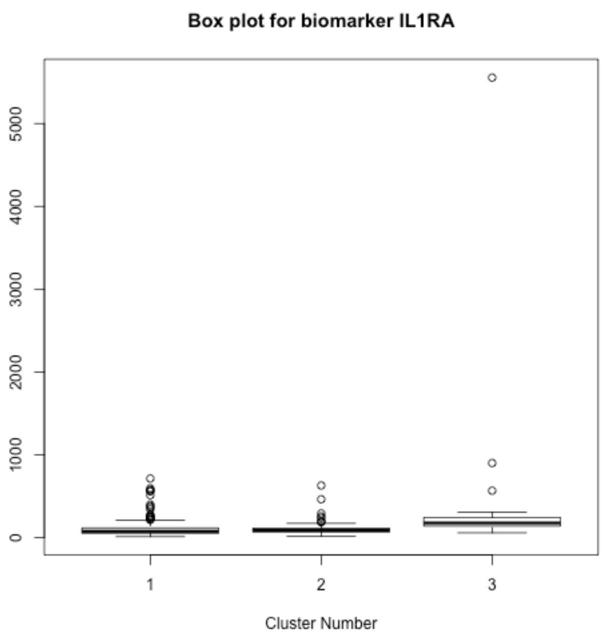

Box plot for biomarker HCC4
0
5
10
15
20
25
1
2
3
Cluster Number

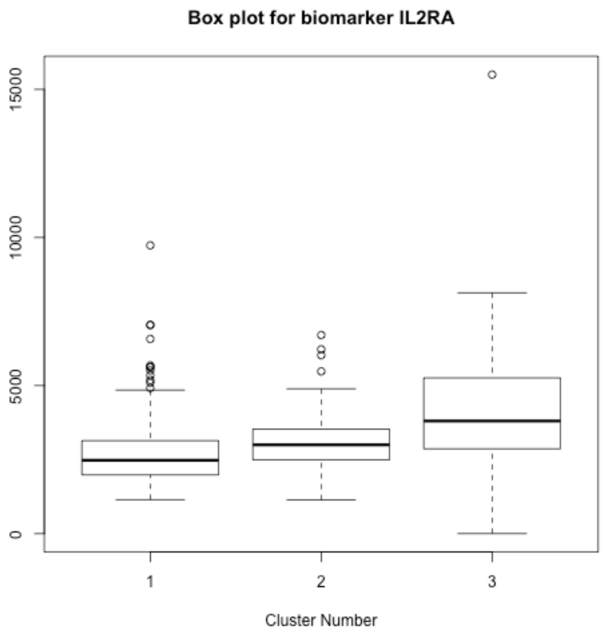

Box plot for biomarker HGF
20
15
10
5
1
2
3
Cluster Number

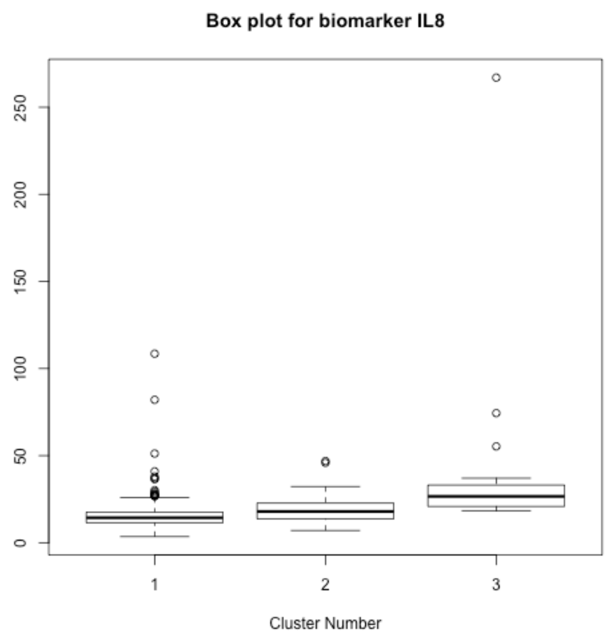

Box plot for biomarker IL10
50
40
30
20
10
1
2
3
Cluster Number

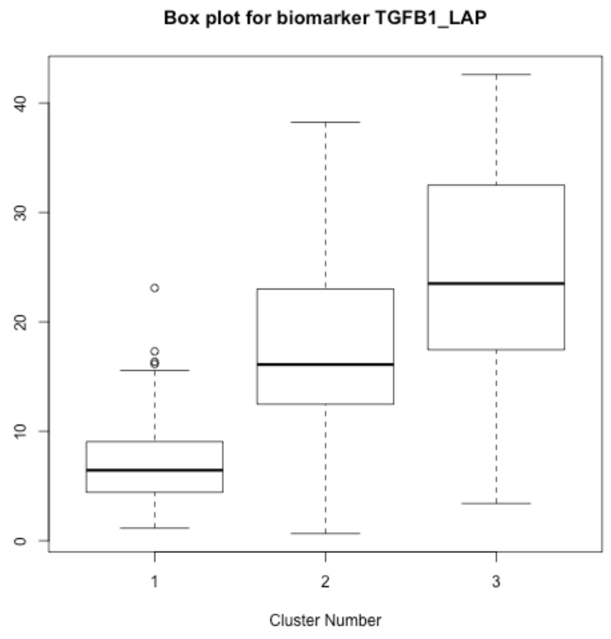

Box plot for biomarker IL16
0
2000
4000
6000
8000
1
2
3
Cluster Number

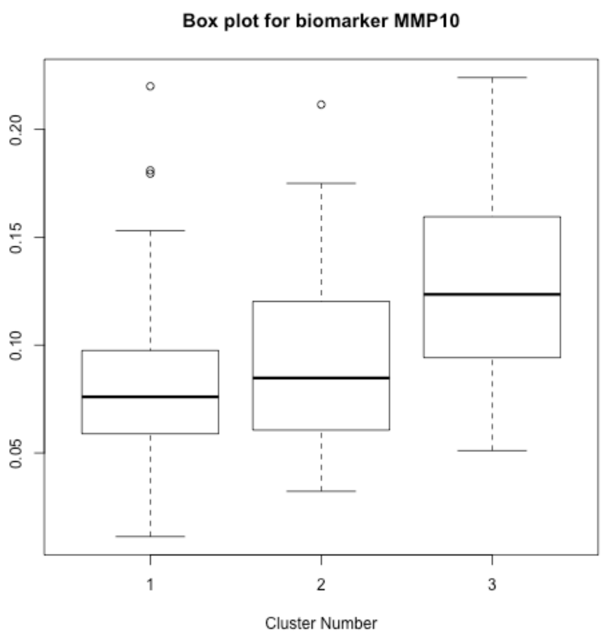

Box plot for biomarker IL18
200
300
400
500
600
700
1
2
3
Cluster Number

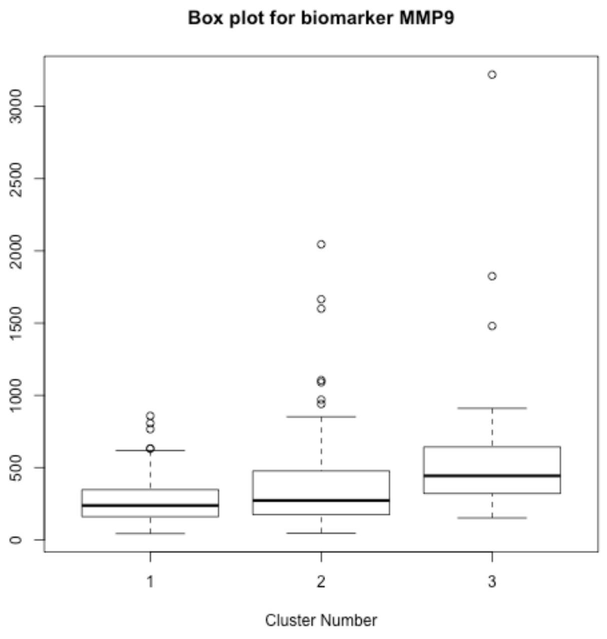

Box plot for biomarker IL1RA
0
1000
2000
3000
4000
5000
1
2
3
Cluster Number

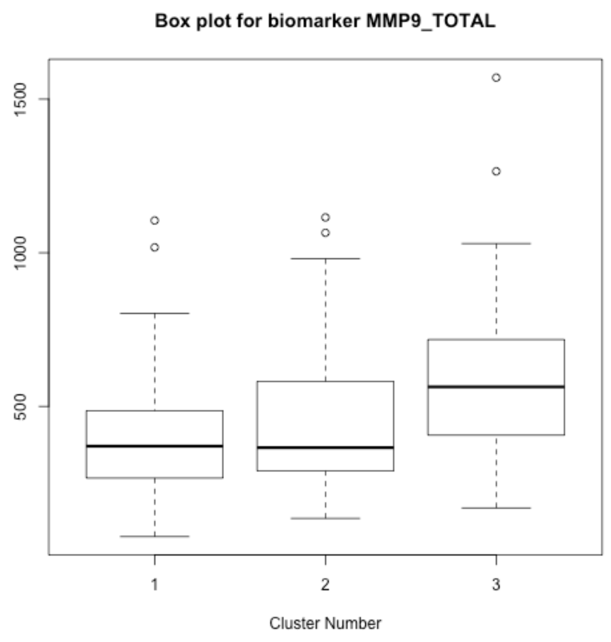

Box plot for biomarker IL2RA
0
5000
10000
15000
1
2
3
Cluster Number

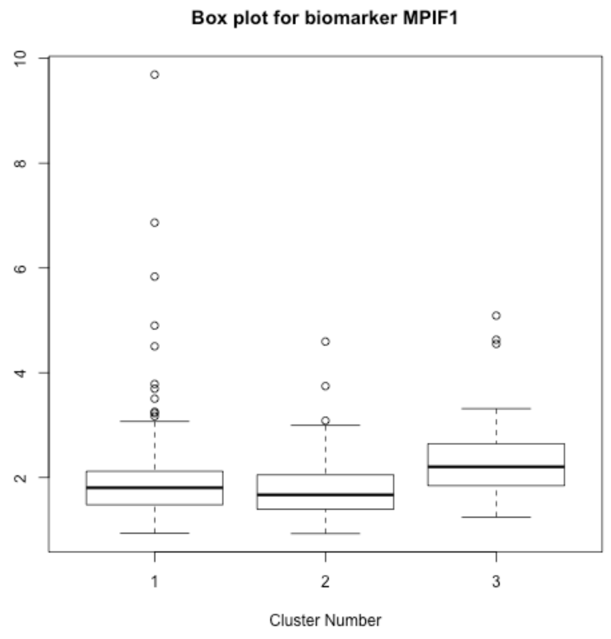

Box plot for biomarker IL8
0
50
100
150
200
250
1
2
3
Cluster Number

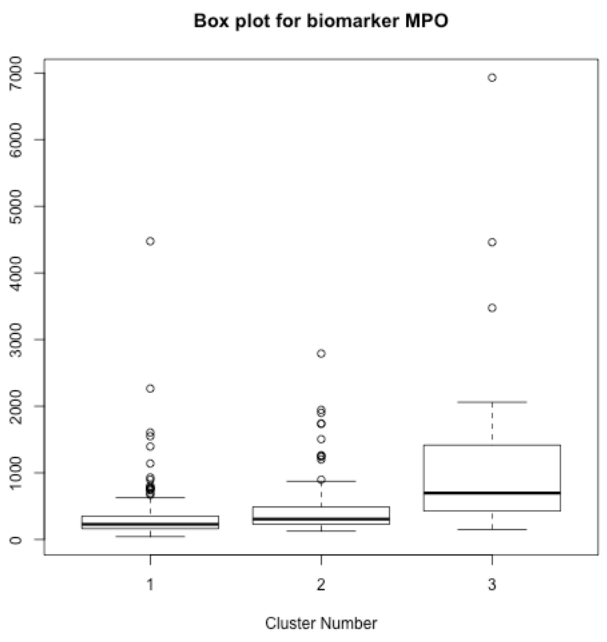

Box plot for biomarker TGFB1_LAP
0
10
20
30
40
1
2
3
Cluster Number

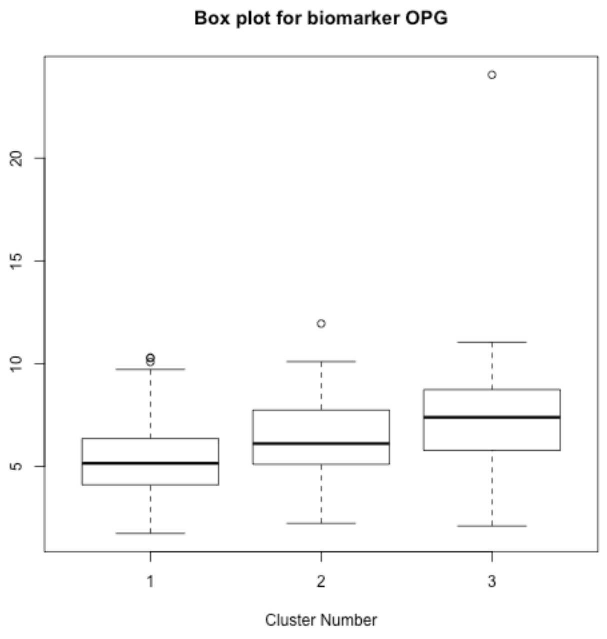

Box plot for biomarker MMP10
0.05
0.10
0.15
0.20
1
2
3
Cluster Number

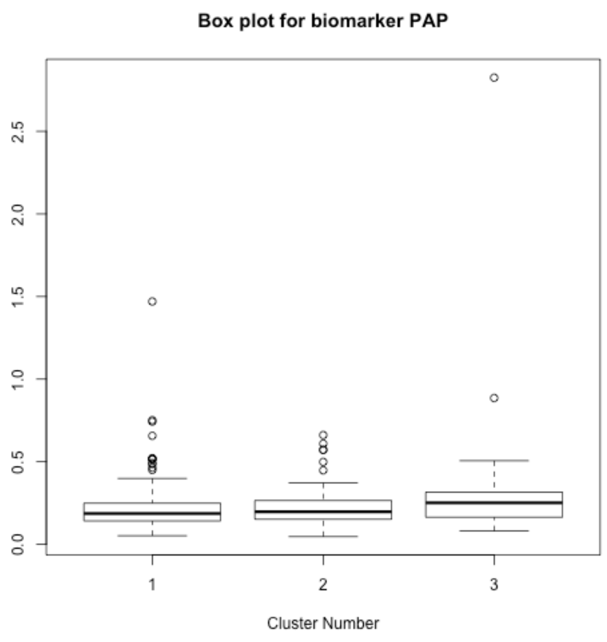

Box plot for biomarker MMP9
0
500
1000
1500
2000
2500
3000
1
2
3
Cluster Number

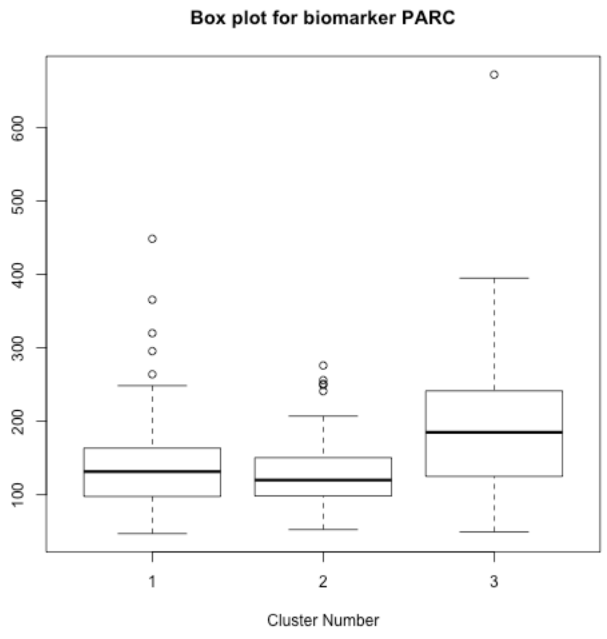

Box plot for biomarker MMP9_TOTAL
1500
1000
500
1
2
3
Cluster Number

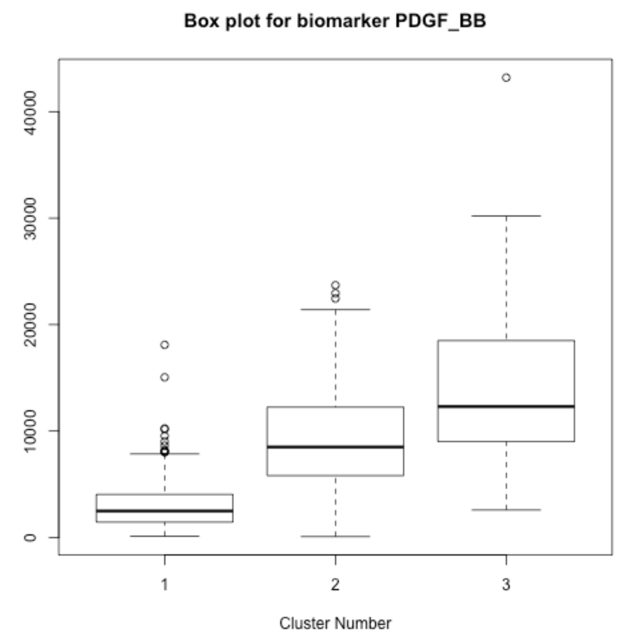

Box plot for biomarker MPIF1
10
8
6
4
2
1
2
3
Cluster Number

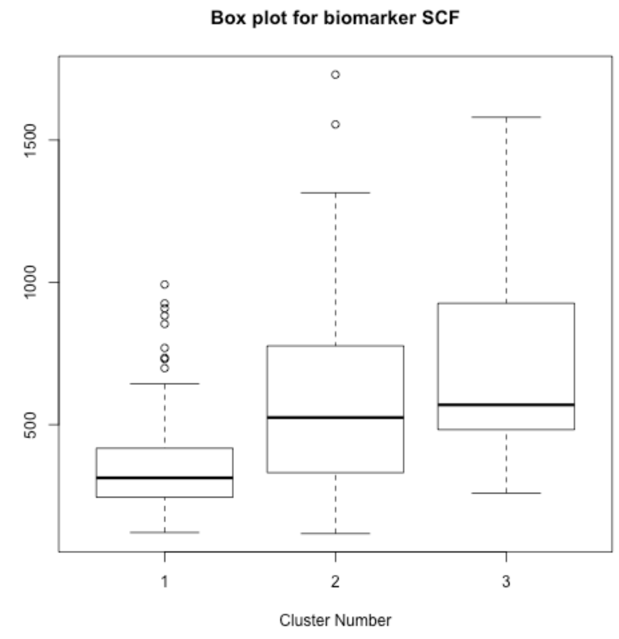

Box plot for biomarker MPO
0
1000
2000
3000
4000
5000
6000
7000
1
2
3
Cluster Number

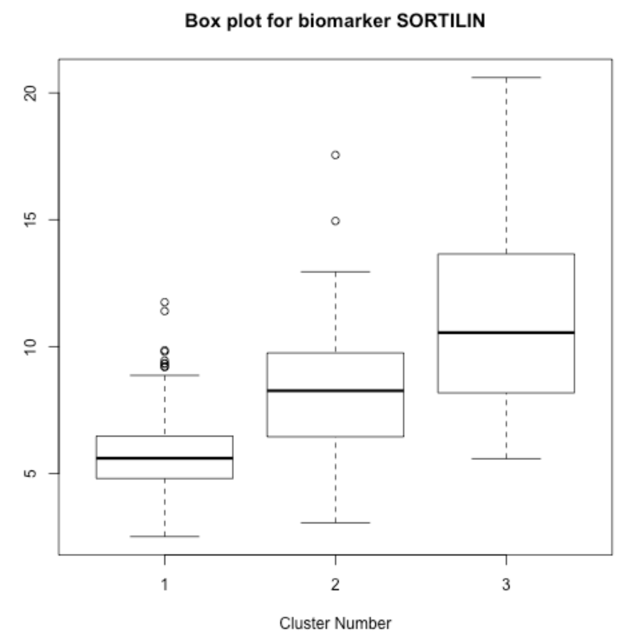

Box plot for biomarker OPG
5
10
15
20
1
2
3
Cluster Number

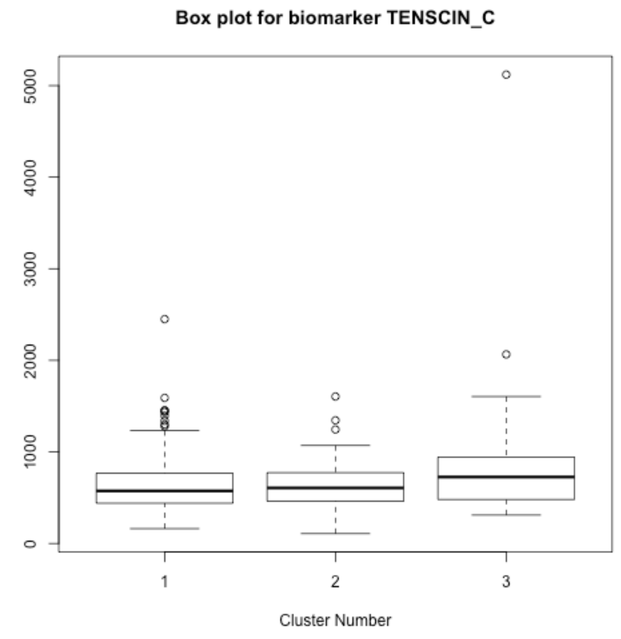

Box plot for biomarker PAP
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
Cluster Number

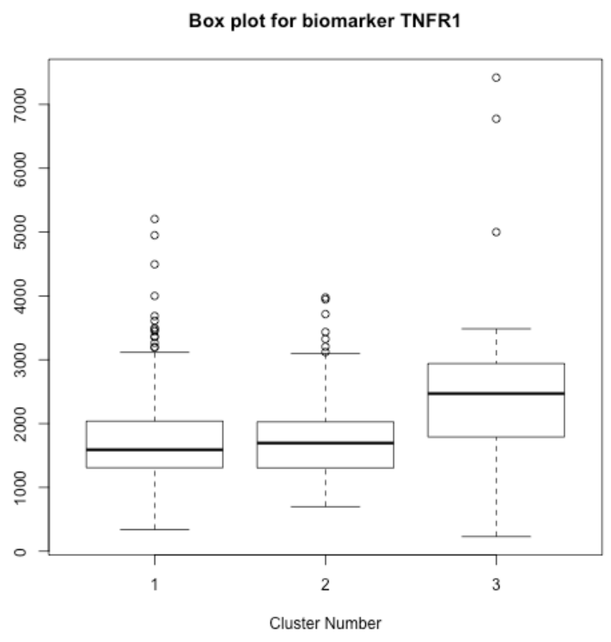

Box plot for biomarker PARC
100
200
300
400
500
600
1
2
3
Cluster Number

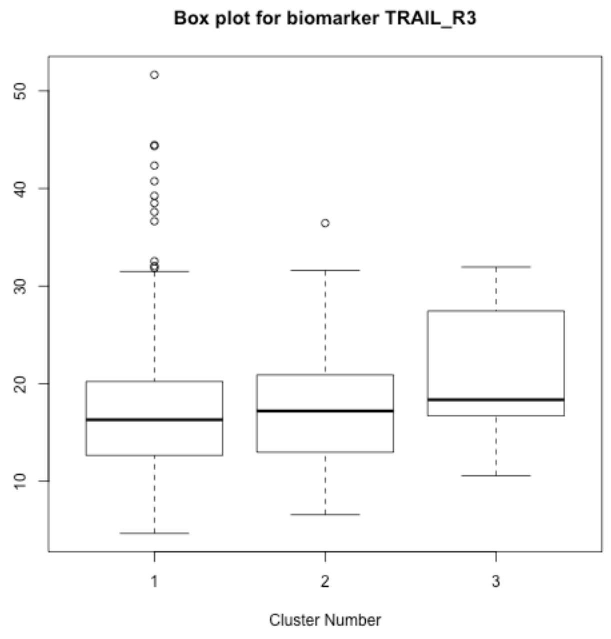

Box plot for biomarker PDGF_BB
0
10000
20000
30000
40000
1
2
3
Cluster Number

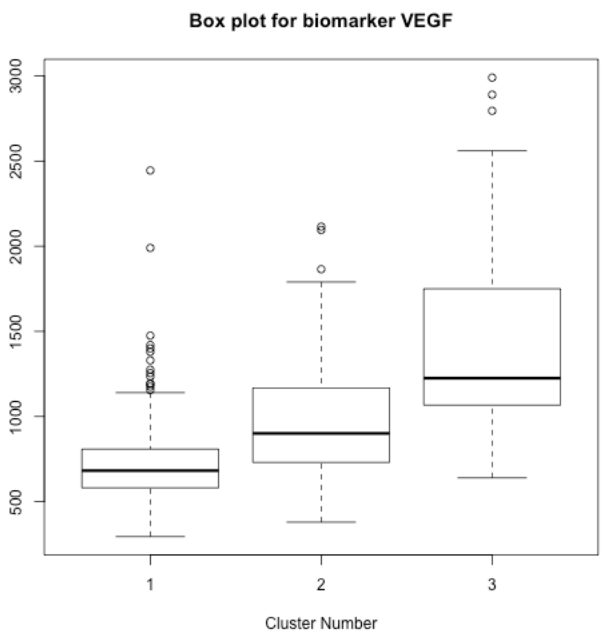

Box plot for biomarker SCF
500
1000
1500
1
2
3
Cluster Number

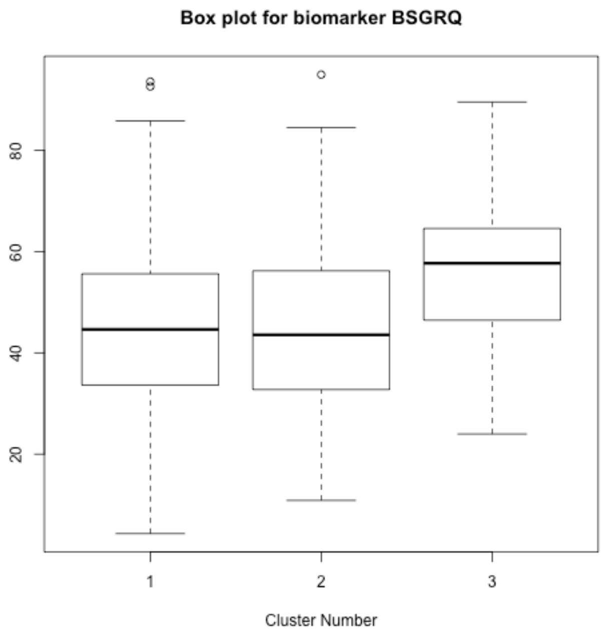

Box plot for biomarker SORTILIN
20
15
10
5
1
2
3
Cluster Number


Box plot for biomarker TENSCIN_C
5000
4000
3000
2000
1000
0
1
2
3
Cluster Number


Box plot for biomarker TNFR1
0
1000
2000
3000
4000
5000
6000
7000
1
2
3
Cluster Number


Box plot for biomarker TRAIL_R3
10
20
30
40
50
1
2
3
Cluster Number





```
write.table(report, file="/Users/szarei2008/Documents/HarvardResearch/SZscripts/AnovaTukeyReportReduced3vs1nad2.csv",row.names=FALSE, sep=',')
```