PTCL Gene Expression based stratification

Integration of Transcriptional and Mutational Data Improves the Stratification of Peripheral T-Cell Lymphoma series

true
true

2019/02/07

Contents

Built with R version:
3.5.0

Libraries

Load necessary libraries

```
library(HiDimDA)
library(affy)
```

```
## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min
```

```
## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
library(ComplexHeatmap)
```

```
## Loading required package: grid

## ========================================
## ComplexHeatmap version 1.18.1
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://bioconductor.org/packages/ComplexHeatmap/
##
## If you use it in published research, please cite:
## Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##   genomic data. Bioinformatics 2016.
## ========================================
```

```
library(plot3D)
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(circlize)
```

```
## ========================================
## circlize version 0.4.4
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: http://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
## ========================================
```

```
library(AnnotationDbi)
```

```
## Loading required package: stats4
```

```
## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:gplots':
##
##     space

## The following object is masked from 'package:base':
##
##     expand.grid
```

```
library(limma)
```

```
##
## Attaching package: 'limma'

## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
```

```
library(lattice)
library(org.Hs.eg.db)
```

```
##
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:AnnotationDbi':
##
##     select
```

```
library(RColorBrewer)
library(AnnotationDbi)
library(rglwidget)
```

```
## The functions in the rglwidget package have been moved to rgl.
```

```
library(survival)
library(VennDiagram)
```

```
## Loading required package: futile.logger
```

```
library(org.Hs.eg.db)
library(GenomicRanges)
```

```
## Loading required package: GenomeInfoDb
```

```
library(GenomicFeatures)
library(rtracklayer)
library(biomaRt)
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:S4Vectors':
##
##     expand

## Loading required package: foreach

## Loaded glmnet 2.0-16
```

```
library(survival)
library(Hmisc)
```

```
## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:AnnotationDbi':
##
##     contents

## The following object is masked from 'package:Biobase':
##
##     contents

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(ConsensusClusterPlus)
library(pheatmap)
library(ggplot2)
library(heatmap.plus)
library(rgl)
library("tmod")
library(SQDA)
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(e1071)
```

```
##
## Attaching package: 'e1071'

## The following object is masked from 'package:Hmisc':
##
##     impute
```

```
set1 = c(brewer.pal(9,"Set1"), brewer.pal(8, "Dark2"))

violinJitter <- function(x, magnitude=1){
  d <- density(x)
  data.frame(x=x, y=runif(length(x),-magnitude/2, magnitude/2) * approxfun(d$x, d$y)(x))
}

rotatedLabel <- function(x0 = seq_along(labels), y0 = rep(par("usr")[3], length(labels)), labels, pos =
  w <- strwidth(labels, units="user", cex=cex)
  h <- strheight(labels, units="user",cex=cex)
  u <- par('usr')
  p <- par('plt')
  f <- par("fin")
  xpd <- par("xpd")
  par(xpd=NA)
  text(x=x0 + ifelse(pos==1, -1,1) * w/2*cos(srt/360*2*base::pi), y = y0 + ifelse(pos==1, -1,1) * w/2 *
  par(xpd=xpd)
}


avefc = function (y, log=TRUE, replace= FALSE) {
     if (log) y = 2^y
   if (replace) y = y + (1-min(y))
   m = apply(y,1,mean)
     y.n = y/m
     y.n2 = y.n
     y.n2 [y.n2 < 1] = 1/ (y.n2 [y.n2 < 1])
     ave.fc = apply (y.n2, 1, mean)
     return(ave.fc)
     }
```

Ensembl Library

For gene convertion from array to HUGO

```
ensembl = useMart( "ensembl", dataset = "hsapiens_gene_ensembl" )
```

Database and Metrics

Gene Expression Data

Upload or generate GEP normalized matrix

```
### choice 1: import processed matrix
data.dir="./"

setwd(data.dir)
load (file.path(data.dir,"/Rmd.files/541_PTCL_batch_adjusted_geo.id.Rdata"))

geneExpr = adj.data[,-c(477:541)] ### remove normal T-cell samples

# import batch and re-order accordingly
load(file.path(data.dir,"/Rmd.files/PTCL.batch.Rdata"))
batch = batch [order(batch$nameNEW),]
batch.series = as.vector(batch$center)

### end of choice 1

### choice 2: generate your own affy object and custom data

# download CEL files from GEO series GSE6338, GSE19067, GSE19069, GSE40160, GSE58445, GSE65823 and EBI
# GSM368580.CEL, GSM368582.CEL, GSM368584.CEL, GSM368586.CEL, GSM368589.CEL, GSM368591.CEL, GSM368594.C
### celfiles <- dir("~/Documents/DATI/PTCL.nos/GSE6338-GSE19067-GSE19069-GSE40160-GSE58445-GSE65823-ETA
### library(affy)
### gset = justRMA(celfile.path = "/Users/emagene/Documents/DATI/PTCL.nos/GSE6338-GSE19067-GSE19069-GSE
### geneExpr = exprs(gset)
### batch adjustment
### library(sva)
### # import batch and re-order accordingly
### load("./Rmd.files/PTCL.batch.Rdata")
### batch = batch [order(rownames(batch)),]
### batch.series = as.vector(batch$center)
### geneExprNEW = geneExpr [ , order(colnames(geneExpr)) ]
### geneExprNEW = geneExprNEW[grep("AFFX",rownames(geneExprNEW), invert=TRUE),]
### # check order correspondence and, if correct, adjust data
### if (all(colnames(geneExprNEW) == rownames(batch))) {
###   adj.data = ComBat (geneExprNEW, batch.series, mod = NULL, par.prior = TRUE, prior.plots = TRUE)
### } else {
###   cat("Error: colnames and batch did not correspond")
### }
### geneExpr = adj.data
### colnames(geneExpr) = as.vector(batch$nameNEW)
### end of choice 2
```

Patients Data

Upload patients information with mutational data

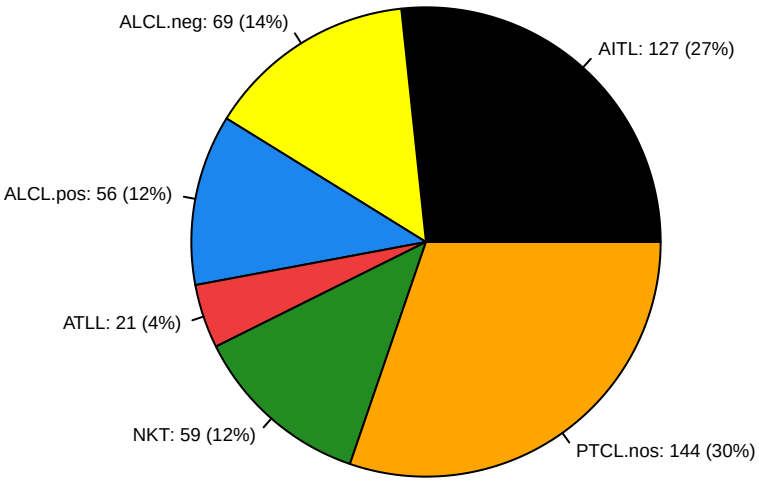
```
pts.info.data <- read.table("./Rmd.files/541_paz_info_MUT.txt", sep="\t", header=TRUE, check.names=FALSE
pts.info.data<- pts.info.data[1:476,]
# customize colors for categories
levels(as.factor(pts.info.data$final.molec))
```

```
## [1] "AITL"     "ALCL.neg" "ALCL.pos" "ATLL"     "NKT"      "PTCL.nos"
```

```
# "AITL"     "ALCL.neg" "ALCL.pos" "ATLL"     "NKT"      "PTCL.nos"

colorz = c("black", "yellow","dodgerblue2","brown2","forestgreen", "orange")
temp = split (  pts.info.data$sample.nameNEW, pts.info.data$final.molec )
colorx = colnames(geneExpr)
for (i in 1:length(colorz)) colorx [ which(colorx %in% unlist(temp[i])) ] = colorz[i]
library(gplots)
colorx = col2hex(colorx)
```

Pie Chart with Percentages

```
slices <- table(pts.info.data$final.molec)
lbls <- names(table(pts.info.data$final.molec))
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, ": ", slices, " (", pct, "%)", sep="" ) # add percents to labels

par(mfrow=c(1,1), mar=c(3,3,3,3), xpd=F)
pie(slices,labels = lbls, init.angle = 0, col=colorz, main="", cex=0.6, radius=0.8)
```



PCA

```
# apply variational filter

afc2 = avefc(geneExpr, log=TRUE, replace=FALSE)
data541exprs.vf = geneExpr [afc2 >= 2, ]
dim(data541exprs.vf )
```

```
## [1] 1868  476
```

```
# retry PCA on shorted gene list
data541m = t(as.matrix(data541exprs.vf))
pca<-prcomp(data541m,scale=T)
mfrow3d(nr = 1, nc = 1, sharedMouse = T)
plot3d(pca$x,rgl.use=F,col=colorx,size=0.6,type="s")
rglwidget()
```

Heatmap of hierarchical clustering on the main PTCL entities

```
mat_heat = as.matrix(data541exprs.vf[,-c(477:541)])
base_mean = rowMeans(mat_heat)
mat_scaled = t(apply(mat_heat, 1, scale))
types = pts.info.data$final.molec[-c(477:541)]
color.annot = col2hex(colorz); names (color.annot)= names(temp)
ha = HeatmapAnnotation(df = data.frame(type = types) , col = list(type = c( color.annot ) ) )
ha@anno_list[[1]]@color_mapping@colors = col2hex(colorz)
names(ha@anno_list[[1]]@color_mapping@colors) = names(temp)
ht = Heatmap(mat_scaled, name = "expression", km = 7, clustering_method_columns = "ward.D", col = colorl
```

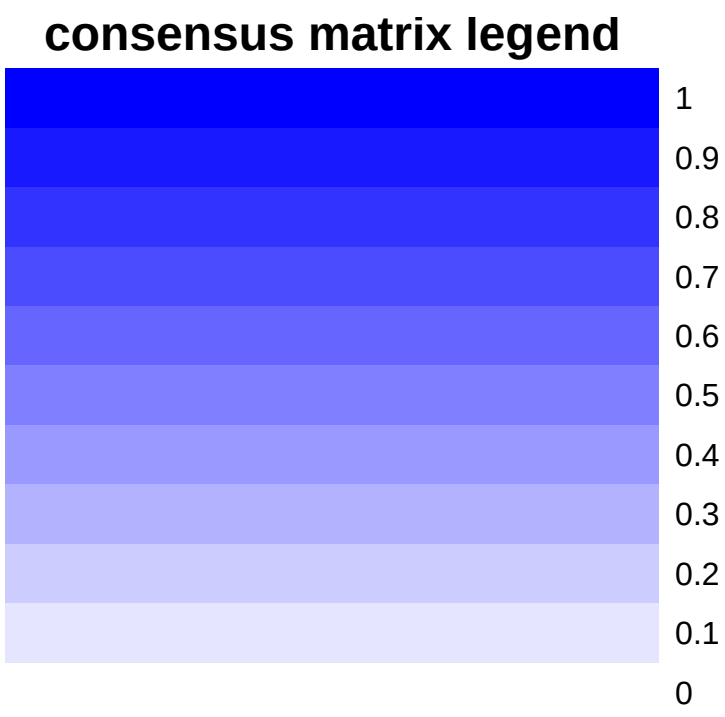
Consensus Clustering and LOOCV on the entire data set

```
pts.info.data_sel<-pts.info.data[pts.info.data$final.molec %in% c("AITL","ALCL.neg","NKT","PTCL.nos"),]
mat_clust<- mat_heat[,colnames(mat_heat) %in% pts.info.data_sel$sample.nameNEW]
title=tempdir()
mat_clust<- data.matrix(mat_clust)
```

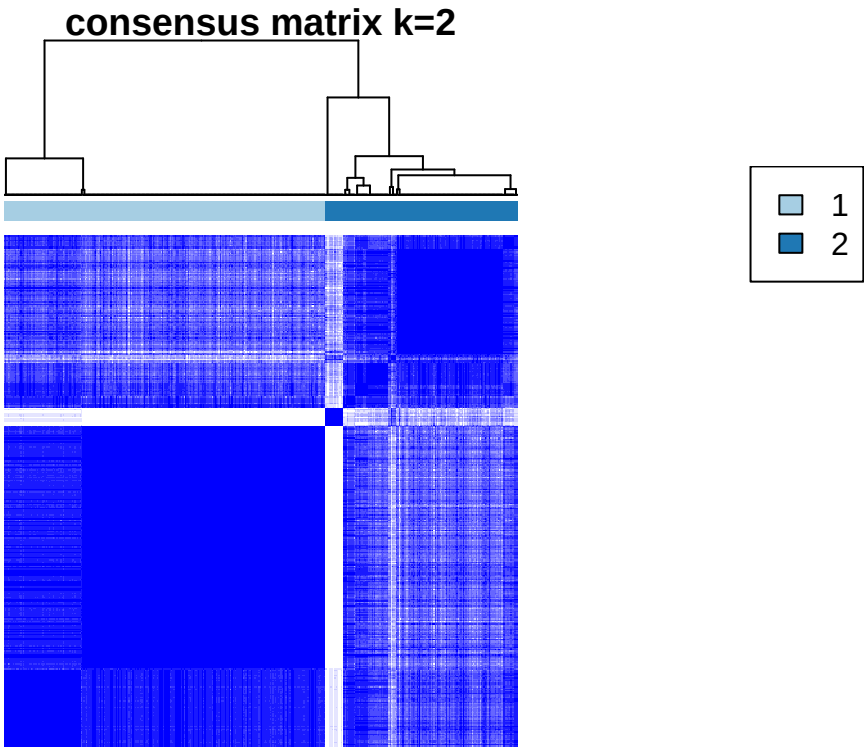
```
mat_clust = sweep(mat_clust,1, apply(mat_clust,1,median,na.rm=T))
results_clust = ConsensusClusterPlus(mat_clust,maxK=8,
                                     pFeature=1,
                                     title=title,
                                     clusterAlg="hc",
                                     innerLinkage="ward.D2",
                                     finalLinkage="ward.D2",
                                     distance="euclidean",
                                     seed=123456789)
```

```
## end fraction
```

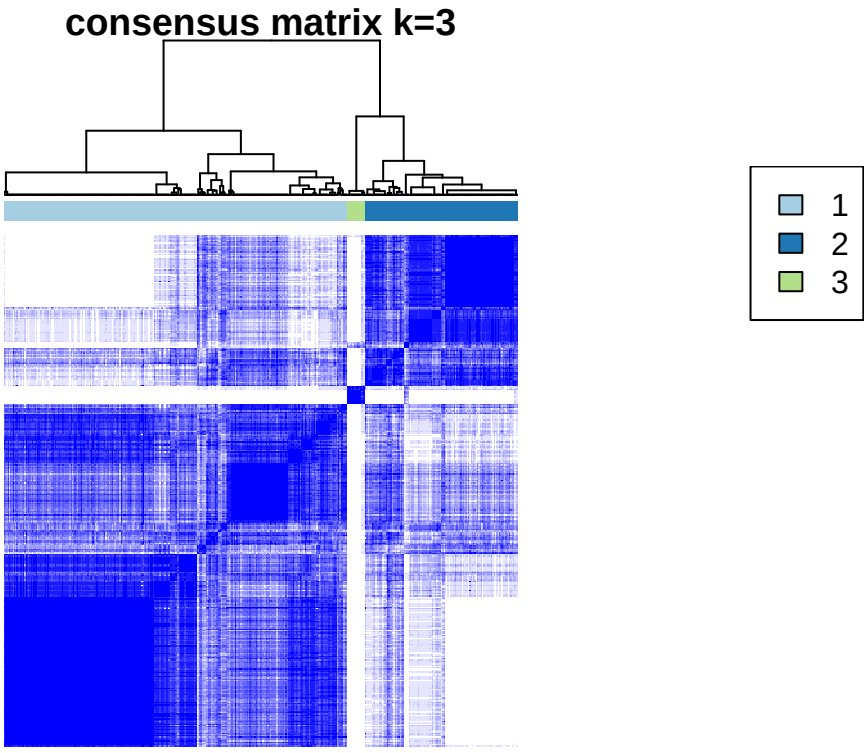
```
## clustered
```



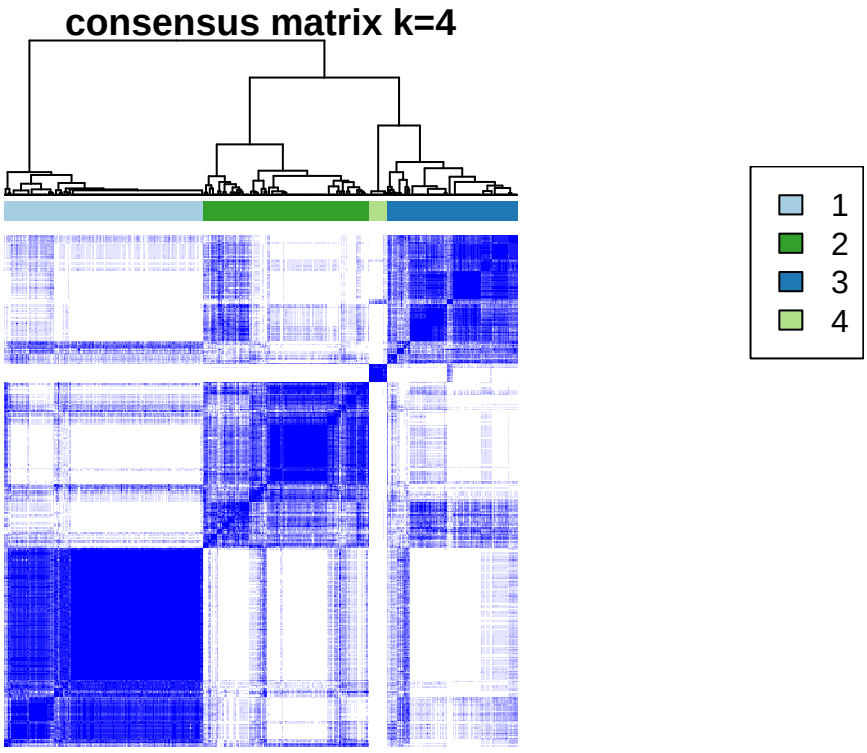

```
## clustered
```



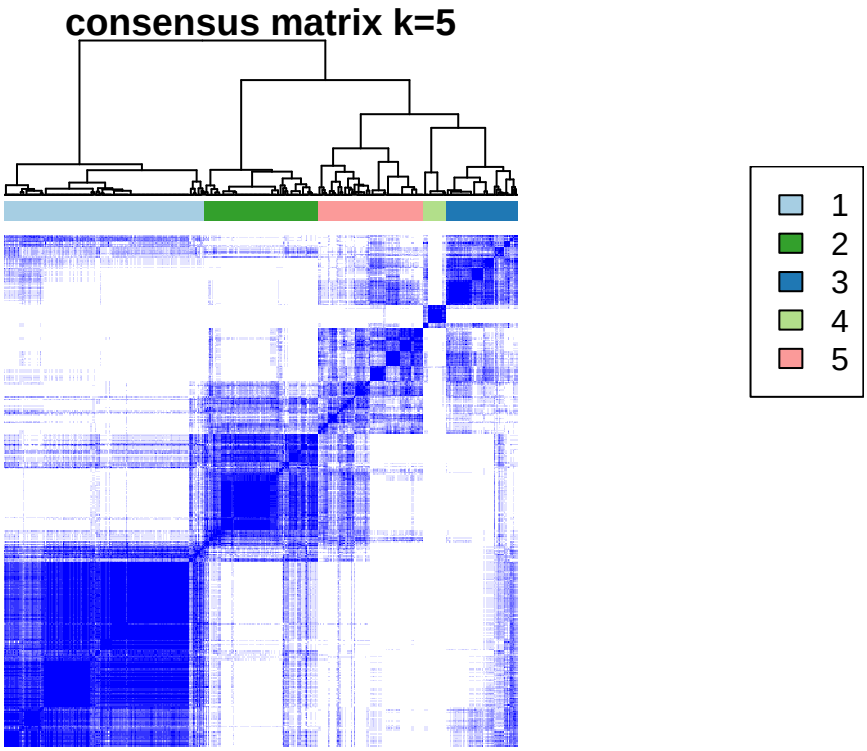

```
## clustered
```



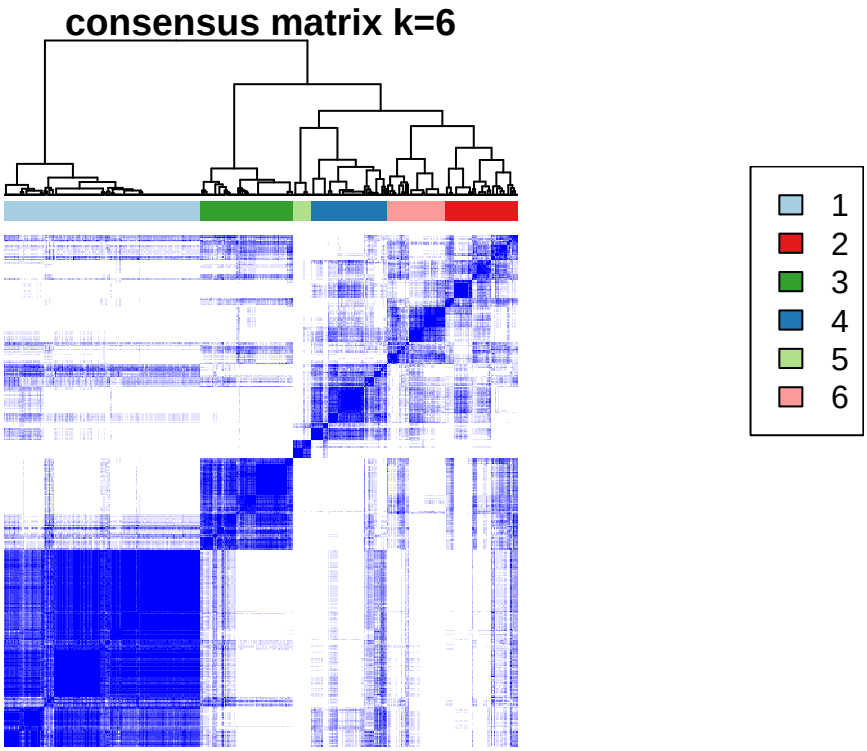

```
## clustered
```



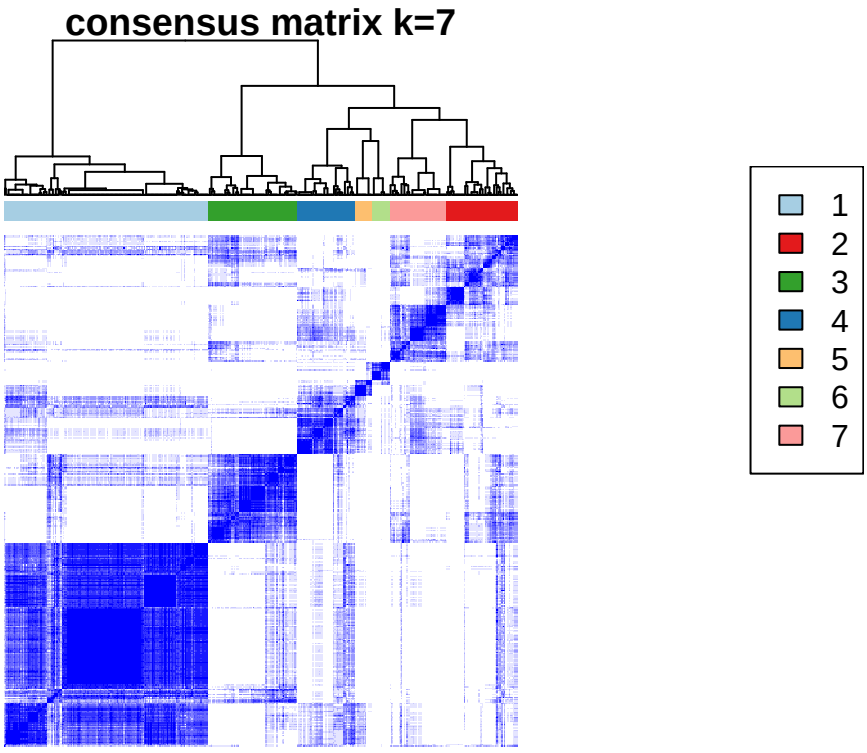

```
## clustered
```

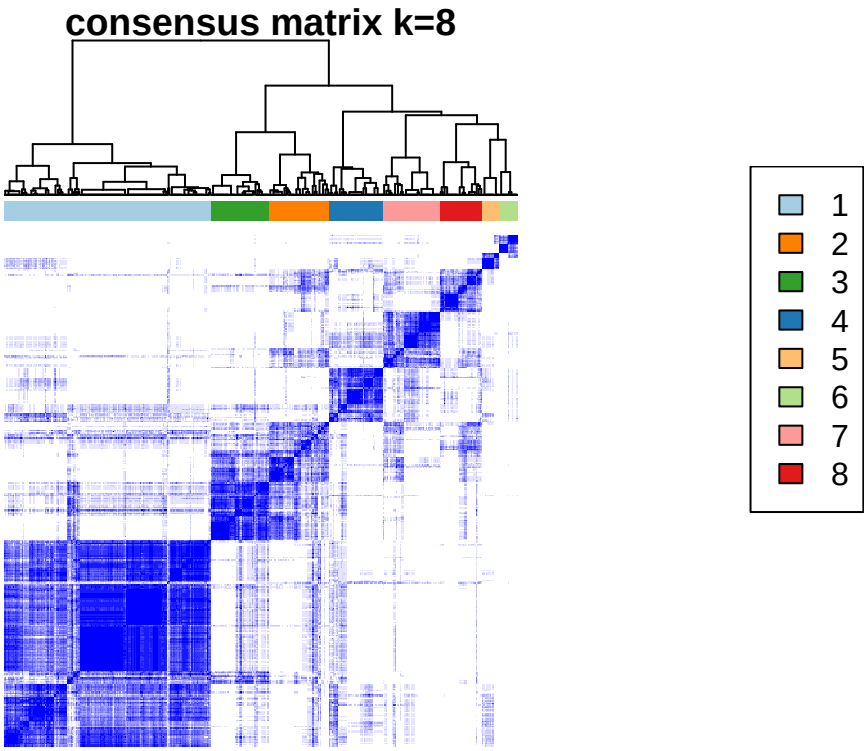


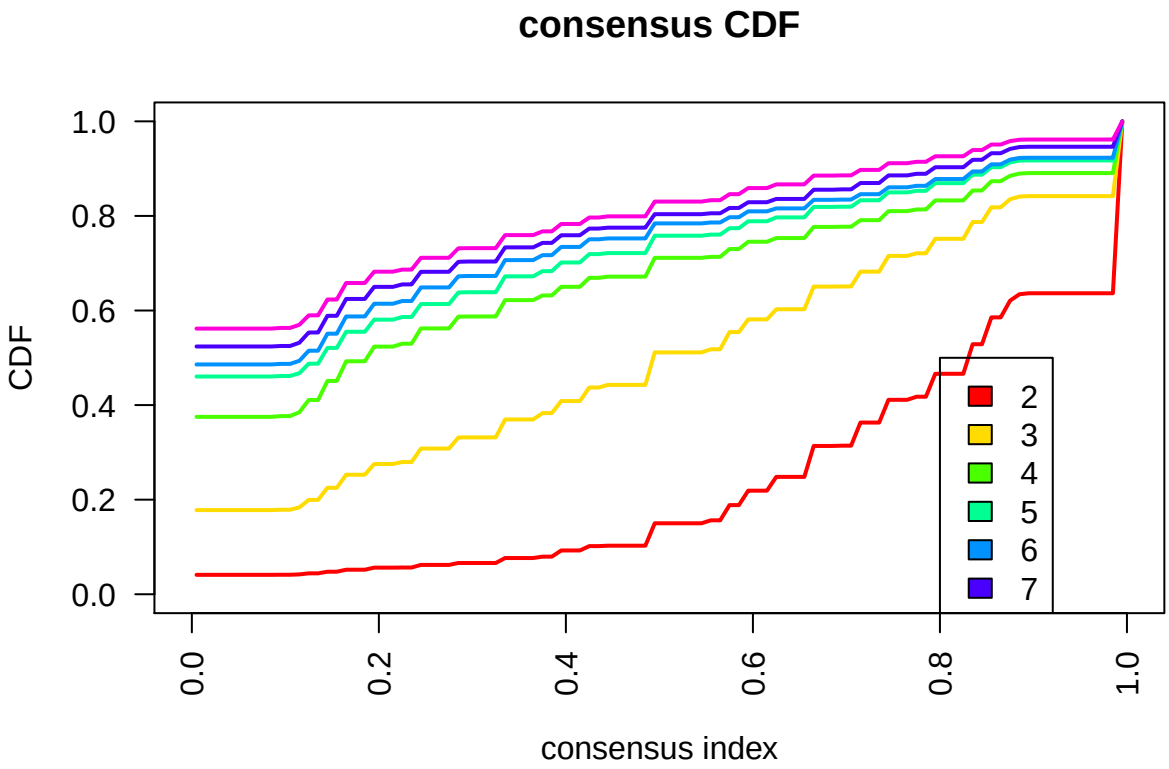

```
## clustered
```

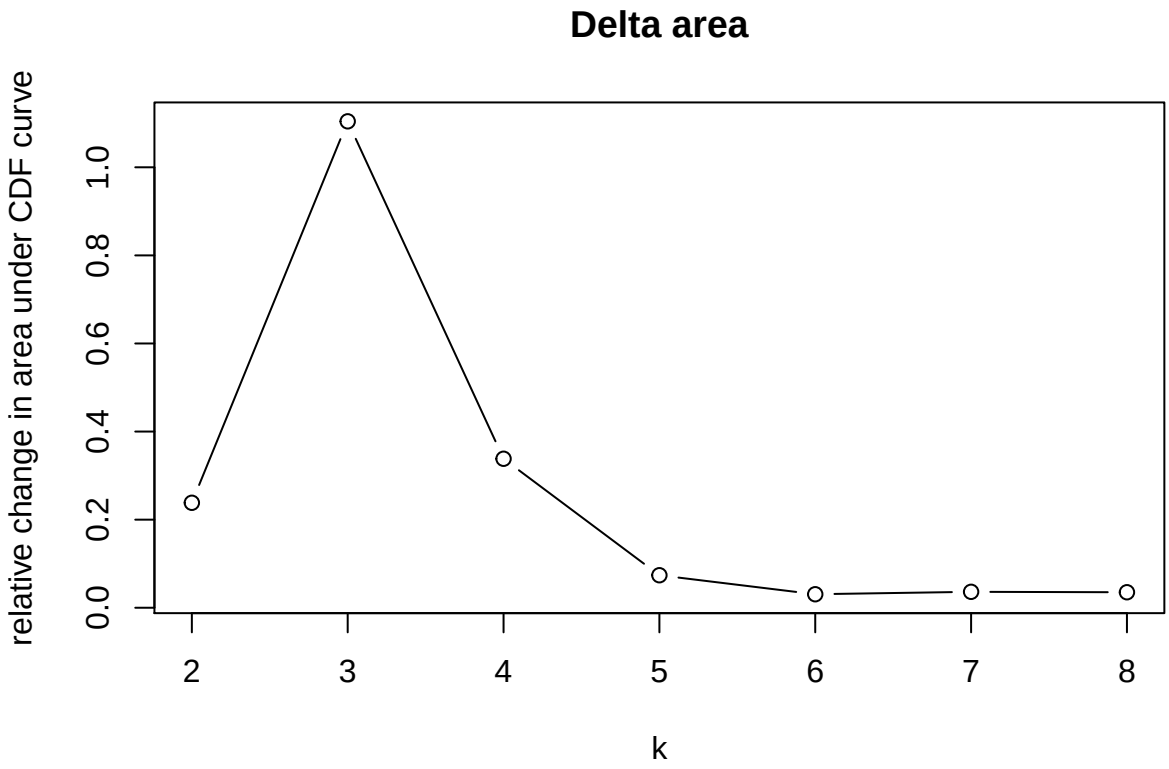


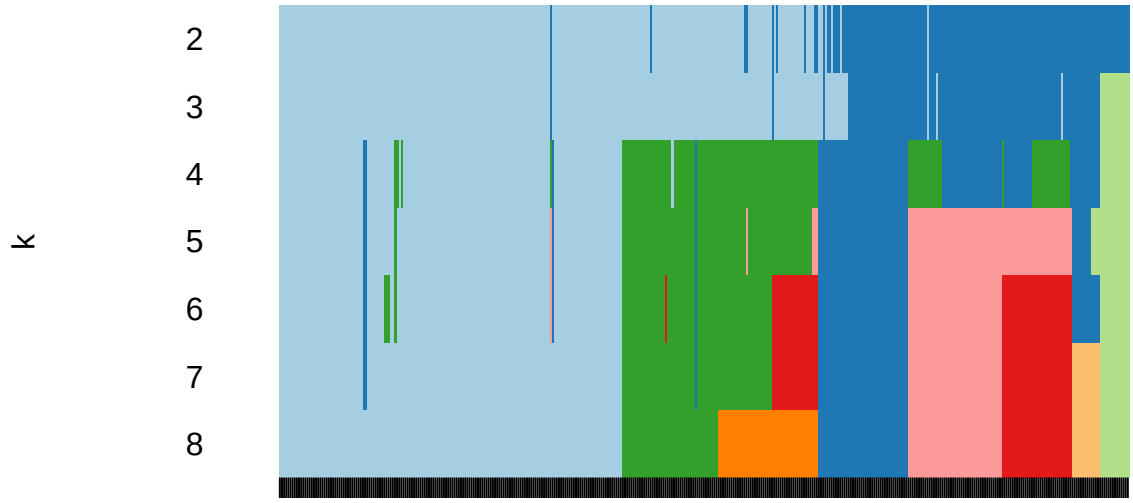

```
## clustered
```


consensus matrix k=7
1
2
3
4
5
6
7


consensus matrix k=8
1
2
3
4
5
6
7
8


consensus CDF
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
consensus index
2
3
4
5
6
7


Delta area
relative change in area under CDF curve
0.0
0.2
0.4
0.6
0.8
1.0
2
3
4
5
6
7
8
k




```
kk<- as.data.frame((results_clust[[8]]$consensusClass)) ##### 6 clusters considering the orioginla 6 di
kk$geo.id<- rownames(kk)
colnames(kk)[1]<- "cluster"
table(kk$cluster)
```

```
##
##   1   2   3   4   5   6   7   8
## 161  47  45  42  13  14  44  33
```

```
kk$cluster[kk$cluster==1]<-"AITL"
kk$cluster[kk$cluster==2]<- "ALCL.neg"
kk$cluster[kk$cluster==3]<-"PTCL.nos"
kk$cluster[kk$cluster==4]<- "PTCL.nos"
kk$cluster[kk$cluster==5] <- "unclassified"
kk$cluster[kk$cluster==6] <-"NKT"

kk2<- kk[kk$cluster!="unclassified",]
pts.info.data2<- pts.info.data_sel[,c(1,6)]
colnames(pts.info.data2)[1]<- "geo.id"
conf_mat_hist<- merge(kk2, pts.info.data2, by="geo.id")
table(conf_mat_hist$cluster, conf_mat_hist$final.molec  )
```

```
##
##            AITL ALCL.neg NKT PTCL.nos
##   7           2        3  36        3
```

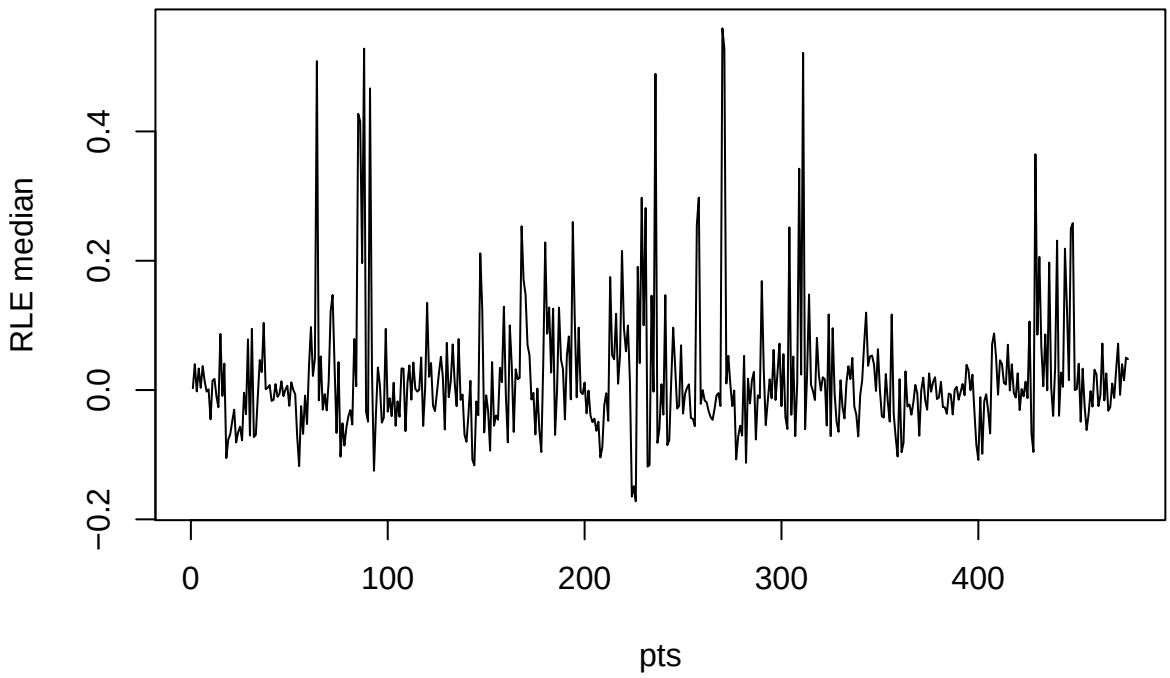
```
##   8           0       28   1        4
##   AITL       99        6   0       56
##   ALCL.neg    3       20  17        7
##   NKT         6        2   3        3
##   PTCL.nos   15        8   2       62
```

```
u <- union(conf_mat_hist$cluster, conf_mat_hist$final.molec )
t <- table(factor(conf_mat_hist$cluster, u), factor(conf_mat_hist$final.molec, u))
confusionMatrix(t)
```

```
## Confusion Matrix and Statistics
##
##
##            AITL ALCL.neg PTCL.nos NKT  7  8
##   AITL       99        6       56   0  0  0
##   ALCL.neg    3       20        7  17  0  0
##   PTCL.nos   15        8       62   2  0  0
##   NKT         6        2        3   3  0  0
##   7           2        3        3  36  0  0
##   8           0       28        4   1  0  0
##
## Overall Statistics
##
##                Accuracy : 0.4767
##                  95% CI : (0.4259, 0.5278)
##     No Information Rate : 0.3497
##     P-Value [Acc > NIR] : 1.963e-07
##
##                   Kappa : 0.3109
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: AITL Class: ALCL.neg Class: PTCL.nos
## Sensitivity               0.7920         0.29851          0.4593
## Specificity               0.7625         0.91536          0.9004
## Pos Pred Value            0.6149         0.42553          0.7126
## Neg Pred Value            0.8844         0.86136          0.7559
## Prevalence                0.3238         0.17358          0.3497
## Detection Rate            0.2565         0.05181          0.1606
## Detection Prevalence      0.4171         0.12176          0.2254
## Balanced Accuracy         0.7772         0.60693          0.6798
##                      Class: NKT Class: 7 Class: 8
## Sensitivity            0.050847       NA       NA
## Specificity            0.966361    0.886  0.91451
## Pos Pred Value         0.214286       NA       NA
## Neg Pred Value         0.849462       NA       NA
## Prevalence             0.152850    0.000  0.00000
## Detection Rate         0.007772    0.000  0.00000
## Detection Prevalence   0.036269    0.114  0.08549
## Balanced Accuracy      0.508604       NA       NA
```

Check relative log expression after batch correction

```
rle.custom = function (a, logged2 = TRUE, file = NULL, colorbox= NULL, labels=NULL , legend = NULL ) {
    a.m <- apply(a,1,median)
if (logged2) {
    for (i in 1:dim(a)[2]) {
         a [,i] <-  a [,i] - a.m
    }
    } else {
        for (i in 1:dim(a)[2]) {
         a [,i] <-  log (a [,i] / a.m )
    }
    }
   # png(file,10240,3840)
   # par(mar=c(10,4,6,2))
  #  boxplot (a, ylim= c(-5,5), outline=F, col=colorbox, xlab="pts", names=labels, las=2, cex.axis = 1.
  #  legend("bottomright",legend = c(levels(as.factor(pts.info.data$final.molec))),
  #    fill = colorz, # 6:1 reorders so legend order matches graph
  #    title = "Legend",
  #    cex = 5)
  #  dev.off()

    a.c = apply(a, 2, stats::quantile)
    return(a.c)
}

#rle.medians = rle.custom(geneExpr, colorbox=colorx, file="./RLE.541.png", labels=pts.info.data$sample.
#plot(rle.medians[3,], type="l", xlab="pts", ylab="RLE median" )
rle.medians = rle.custom(geneExpr, colorbox=colorx, file="./RLE.541.png", labels=pts.info.data$sample.na
plot(rle.medians[3,], type="l", xlab="pts", ylab="RLE median" )
```



Build Final Gene Expression Matrix

Define design file and filter geneExpr for patients included in design data frame and

```
design <- pts.info.data[,c(1:2,6:8,14:17)]
rownames(design)<- design[,1]
design<- design[,-c(1:2)]
design<-na.omit(design) ### select onyl patients with all mutations data available (n=53)
design$age<- as.numeric(as.character(design$age))
design$age<- design$age - median(design$age)
design[design == "WT"] <- 0
design[design == "MUT"] <- 1
design$final.molec[design$final.molec=="AITL"] <- 0
design$final.molec[design$final.molec=="PTCL.nos"] <- 1
design$gender[design$gender=="M"] <- 1
design$gender[design$gender=="F"] <- 0
design$offset <- rep(1, nrow(design))
design<-design[,c(8,1:7)]

all(pts.info.data$sample.nameNEW == colnames(geneExpr)) ## check correspondence
```

```
## [1] TRUE
```

```
# geneExpr = geneExpr [ , order (pts.info.data$geo.id)] ### do only to set correspondence in case of cu
# colnames(geneExpr) = pts.info.data$sample.nameNEW [ order (pts.info.data$geo.id)]

geneExpr2<- (geneExpr[, rownames(design)])
geneExpr2<- data.matrix(geneExpr2, rownames.force = NA)
design<- data.matrix(design, rownames.force = NA)
```

Generation of simplified signature

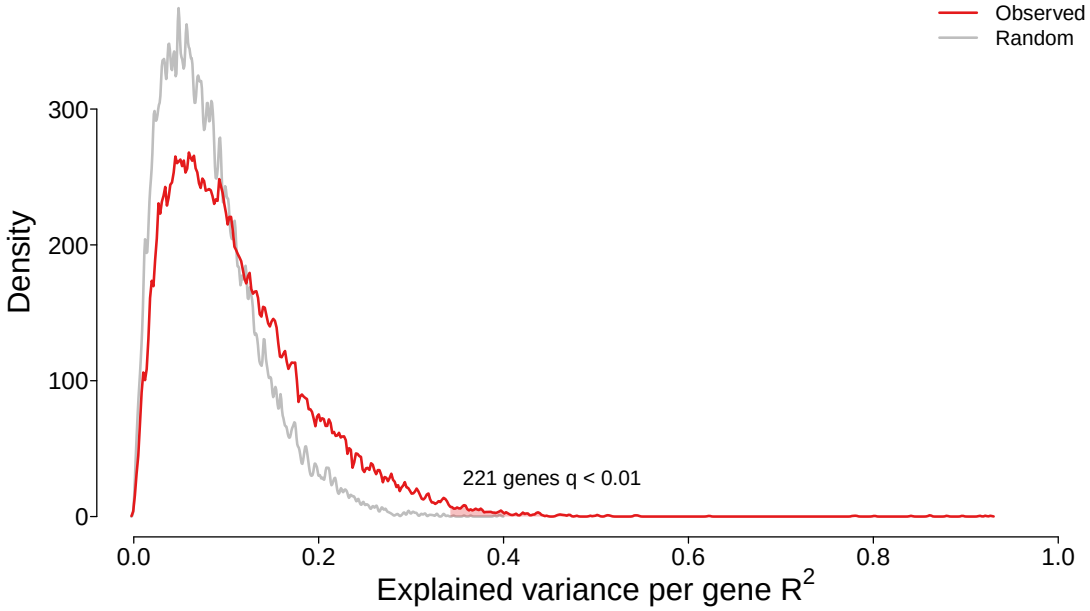
Model fitting

We use the lmFit function from the limma package. This comes with a whole series of powerful and reliable tests.

```
glm = lmFit(geneExpr2[,rownames(design)], design = design )
glm = eBayes(glm)
F.stat <- classifyTestsF(glm[,-1],fstat.only=TRUE)
glm$F <- as.vector(F.stat)
df1 <- attr(F.stat,"df1")
df2 <- attr(F.stat,"df2")
if(df2[1] > 1e6){
  glm$F.p.value <- pchisq(df1*glm$F,df1,lower.tail=FALSE)
}else
  glm$F.p.value <- pf(glm$F,df1,df2,lower.tail=FALSE)

set.seed(12345678)
rlm <- lmFit(geneExpr[,rownames(design)], apply(design, 2, sample))
rlm <- eBayes(rlm)
F.stat <- classifyTestsF(rlm[,-1],fstat.only=TRUE)
rlm$F <- as.vector(F.stat)
df1 <- attr(F.stat,"df1")
df2 <- attr(F.stat,"df2")
if(df2[1] > 1e6){
  rlm$F.p.value <- pchisq(df1*rlm$F,df1,lower.tail=FALSE)
}else
  rlm$F.p.value <- pf(rlm$F,df1,df2,lower.tail=FALSE)
F.stat <- classifyTestsF(glm[,2:5],fstat.only=TRUE)
df1 <- attr(F.stat,"df1")
df2 <- attr(F.stat,"df2")
F.p.value <- pchisq(df1*F.stat,df1,lower.tail=FALSE)
R.stat <- classifyTestsF(rlm[,2:5],fstat.only=TRUE)
Rall = 1 - 1/(1 + glm$F * (ncol(design)-1)/(nrow(design)-ncol(design)))
Rgenetics = 1 - 1/(1 + F.stat * 4/(nrow(design)-ncol(design)))
Pgenetics = 1 - 1/(1 + R.stat * 4/(nrow(design)-ncol(design)))
names(Rgenetics) <- names(Pgenetics) <- names(Rall) <-  rownames(geneExpr)
```

Check Differentially Expressed Genes

```
par(bty="n", mgp = c(2,.33,0), mar=c(3,2.5,1,1)+.1, las=1, tcl=-.25, xpd=NA)
d <- density(Pgenetics,bw=1e-3)
f <- 40 #nrow(gexpr)/512

par(mfrow=c(1,1))
par(mar=c(8,5,5,5), xpd=F)
plot(d$x, d$y * f, col='grey', xlab=expression(paste("Explained variance per gene ", R^2)), main="", lwd
title(ylab="Density", line=2.5, cex.lab=1.5)
d <- density(Rgenetics, bw=1e-3)
r <- min(Rgenetics[p.adjust(F.p.value,"BH")<0.01]) ######## threshold to select 412 genes
x0 <- which(d$x>r)
polygon(d$x[c(x0[1],x0)], c(0,d$y[x0])* f, col=paste(set1[1],"44",sep=""), border=NA)
lines(d$x, d$y* f, col=set1[1], lwd=2)
text(d$x[x0[1]], d$y[x0[1]]*f +20, pos=4, paste(sum(Rgenetics > r), "genes q < 0.01"))
legend("topright", bty="n", col=c(set1[1], "grey"), lty=1, c("Observed","Random"), lwd=2)
```




```
glmPrediction <- glm$coefficients %*% t(design)
rlmPrediction <- rlm$coefficients %*% t(design)
```

Print signficiant genes

```
kk<-as.data.frame((p.adjust(F.p.value,"BH")<0.01))
kk$gene<- rownames(kk)
colnames(kk)[1]<-"code"
```

```
kk2<-kk[kk$code=="TRUE",]
### sort(kk2$gene) ##### if you want to print the entire list of differentially expressed genes

kk_tab<-as.data.frame((p.adjust(F.p.value,"BH")))
kk_tab$gene<- rownames(kk_tab)
kk_tab2<- as.data.frame(kk_tab[kk_tab$`(p.adjust(F.p.value, "BH"))`<0.01,])
geneannotation1 <- getBM( attributes = c("ensembl_transcript_id", "entrezgene", "external_gene_name"), 

geneannotation2<-unique( geneannotation1[,c(2,3)])
geneannotation2$gene<- paste0(geneannotation2$entrezgene, "_at")
require(plyr)
```

```
## Loading required package: plyr
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     is.discrete, summarize
```

```
## The following object is masked from 'package:IRanges':
##
##     desc
```

```
## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```
tab_save<- join(kk_tab2, geneannotation2, by="gene")
tab_save<-tab_save[,c(4,2,1)]
colnames(tab_save)<- c("gene","entrezgene","q.value")
# write.table(tab_save, "/Volumes/GoogleDrive/My Drive/PTCL/AJH/table_differentially_expressed_gene.txt
tab_save$external_gene_name[tab_save$external_gene_name %in%  sort(unique(geneannotation1$external_gene_
```

```
## NULL
```

Calculation of significant effects per covariate

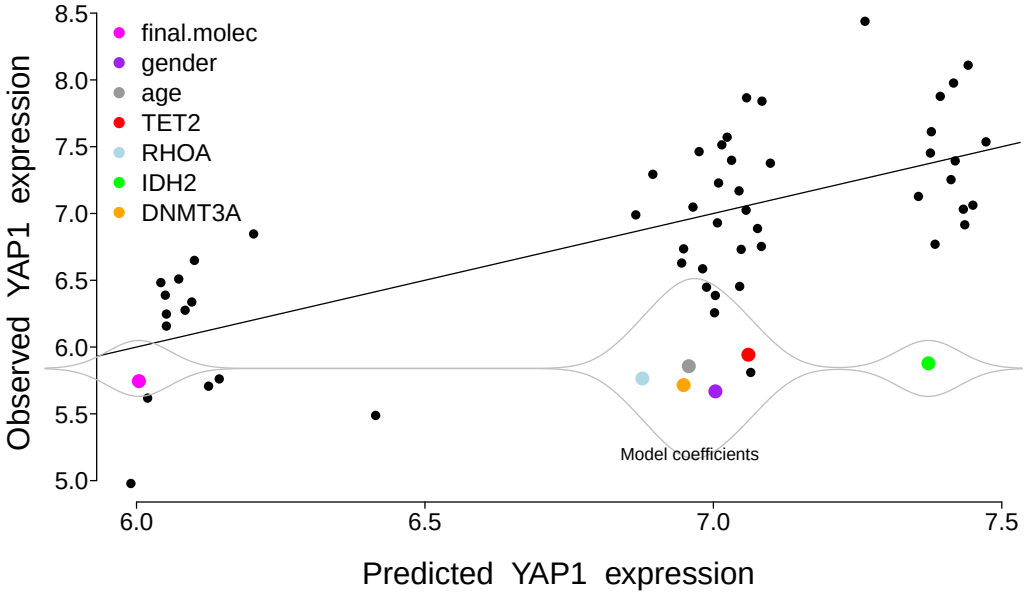
Extract the list of differentially expressed genes by mutation

```
### customize colors in colMutations
# colMutations = c(brewer.pal(8,"Set1")[-6], rev(brewer.pal(8,"Dark2")), brewer.pal(7,"Set2"))[c(1:12,1
# o <- order(apply(col2rgb(colMutations),2,rgb2hsv)[1,])
# colMutations <- colMutations[rev(o)][(4*1:19 +15) %% 19 + 1][1:7]
colMutations = col2hex(c("magenta", "purple","gray60","red","lightblue","green","orange"))
names(colMutations) <- colnames(design)[-1]

gene_code<- kk2$gene
```

```
tab=NULL
tab2_list=list()
for(i in (1:length(kk2$gene)))
{
  gene_single<- gene_code[i]
  y <- glm$coefficients[gene_single,-1]+glm$coefficients[gene_single,1]
  w <- glm$p.value[gene_single,-1] < 0.01
  int<-c(gene_single, as.character(w))
  tab<- rbind(tab, int)
  tab2_list[[i]]<-glm$p.value[gene_single,-1]
}
rownames(tab)<-seq(1:nrow(tab))
colnames(tab)<- c("gene",colnames(design)[-1])
tab23<- do.call("rbind", tab2_list)
rownames(tab23)<- kk2$gene

# Write to disk a file with all significant genes with p value for each

# write.table((tab23), "/Volumes/GoogleDrive/My Drive/PTCL/AJH/table_differentially_expressed_gene_true
```

Example of single gene extraction

```
 # temp_name = unique(getBM( attributes = c("ensembl_transcript_id", "entrezgene", "external_gene_name"
 # mart = ensembl)$external_gene_name)
# setwd("./Rmd.files/figure_paper/all_figures_from_rmd/")
  #pdf("Figure_2b.pdf", width = 10, height = 7)
  par(mfrow=c(1,1))
  par(mar=c(10,8,5,5), xpd=F)
  par(bty="n", mgp = c(1.5,.33,0),las=1, tcl=-.25, xpd=F)
  temp_name<- "YAP1"
  plot(glmPrediction[gene_single,], geneExpr[gene_single,rownames(design)], ylab="", xlab="",
       pch=16, cex=1, cex.axis=1.2, cex.lab=1.5)
  title(ylab=(paste("Observed ",temp_name, " expression")), line=2.5, cex.lab=1.5)
   title( xlab=(paste("Predicted ",temp_name, " expression")), line=2.5, cex.lab=1.5)
  abline(0,1)
  u <- par("usr")
  par(xpd=NA)
  y <- glm$coefficients[gene_single,-1]+glm$coefficients[gene_single,1]
  u <- par("usr")
  x0 <- rep(u[3]+1,ncol(design)-1)
  y0 <- u[4] + 0.05*(u[4]-u[3]) - rank(-y)/length(y) * (u[4]-u[3])/1.2
  d <- density(y)
  lines(d$x, d$y/5+1+u[3], col="grey")
  lines(d$x, -d$y/5+1+u[3], col="grey")
points(x=y, y=x0+violinJitter(y, magnitude=0.25)$y, col=colMutations, pch=16, cex=1.5)
  text(x=glm$coefficients[gene_single,1], y= 5.2, "Model coefficients", cex=0.8)
legend("topleft",names(colMutations), col = colMutations, bty= "n", cex = 1.2, pch = 16)
```

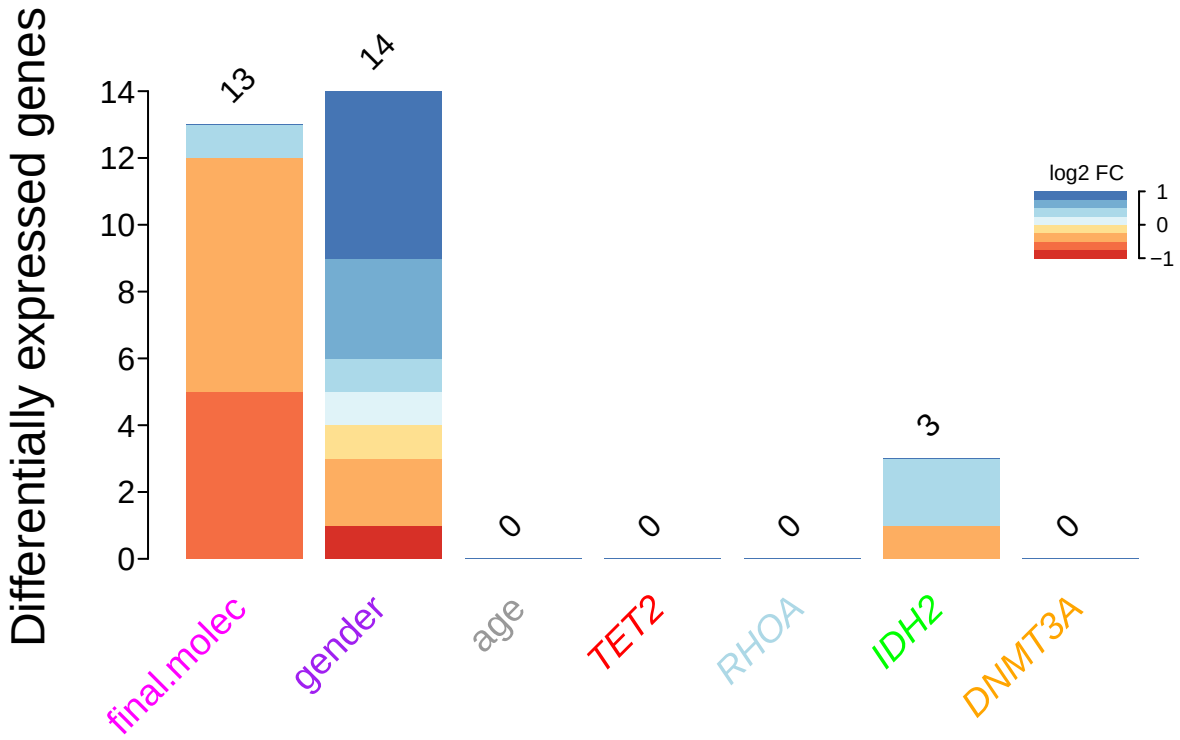



```
#dev.off()
```

Plot significant effects per covariate (q<0.01)

```
testResults <- decideTests(glm, method="hierarchical",adjust.method="BH", p.value=0.01)[,-1]
significantGenes <- sapply(1:ncol(testResults), function(j){
  c <- glm$coefficients[testResults[,j]!=0,j+1]
  table(cut(c, breaks=c(-5,seq(-1.5,1.5,l=7),5)))
})

colnames(significantGenes) <- colnames(testResults)
rownames(tab)<-c(1:nrow(tab))
tab2<- as.data.frame(tab)
tab2$gene<-as.character(as.character(tab2$gene))
tab2$final.molec<-as.character(as.character(tab2$final.molec))
tab2$TET2<-as.character(as.character(tab2$TET2))
tab2$RHOA<-as.character(as.character(tab2$RHOA))
tab2$IDH2<-as.character(as.character(tab2$IDH2))
tab2$DNMT3A<-as.character(as.character(tab2$DNMT3A))



par(mfrow=c(1,1))
```

```
par(bty="n", mgp = c(2.5,.33,0), mar=c(5,5.5,5,0)+.1, las=2, tcl=-.25)
b <- barplot(significantGenes, las=2, ylab = "Differentially expressed genes", col=brewer.pal(8,"RdYlBu
rotatedLabel(x0=b-0.1, y0=rep(-0.5, ncol(significantGenes)), labels=colnames(significantGenes), cex=1.2
rotatedLabel(b-0.1, colSums(significantGenes), colSums(significantGenes), pos=3, cex=, srt=45)#dev.off(
clip(0,30,0,1000)
x0 <- 7.5
image(x=x0+c(0,0.8), y=par("usr")[4]+seq(-1,1,l=9) -4, z=matrix(1:8, ncol=8), col=brewer.pal(8,"RdYlBu"
text(x=x0+1.1, y=par("usr")[4]+c(-1,0,1) -4, format(seq(-1,1,l=3),2), cex=0.66)
lines(x=rep(x0+0.9, 2), y=par("usr")[4]+c(-1,1) -4)
segments(x0+0.9,par("usr")[4] + 1-4,x0+0.95,par("usr")[4] + 1-4)
segments(x0+0.9,par("usr")[4] + 0-4,x0+0.95,par("usr")[4] + 0-4)
segments(x0+0.9,par("usr")[4] + -1-4,x0+0.95,par("usr")[4] + -1-4)
text(x0 + 0.45, par("usr")[4] + 1.5-4, "log2 FC", cex=.66)
```



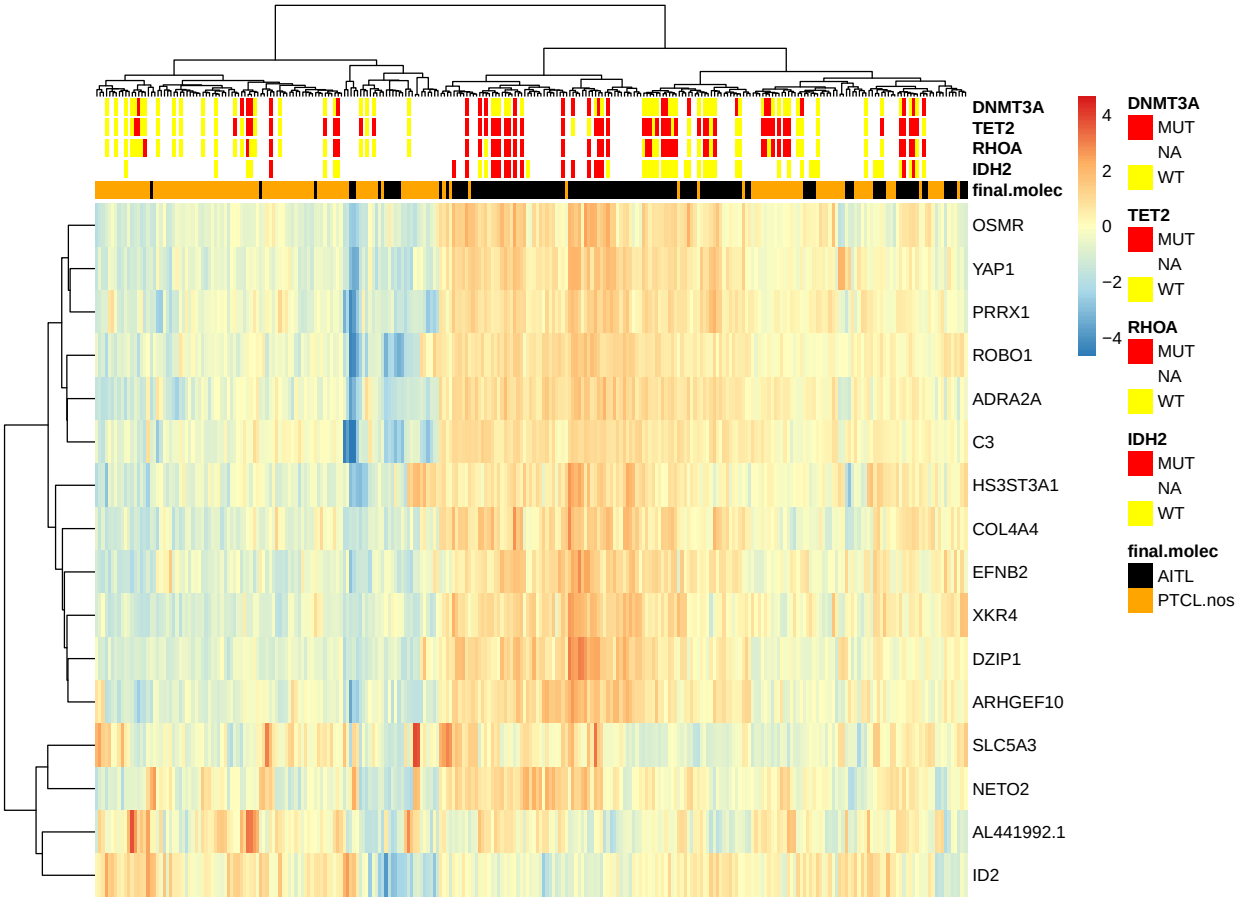
Print the list of differently expressed genes using the Ensembl annotation

```
select_hist<- pts.info.data[pts.info.data$final.molec == "AITL" |  pts.info.data$final.molec == "PTCL.n
gene<- as.data.frame(testResults)
sig_genes<- gene[gene$final.molec!= 0 |gene$IDH2 != 0 | gene$TET2 != 0 | gene$DNMT3A != 0 | gene$RHOA !=
list_genes<-sort(rownames(sig_genes)) ##### list of signficiant genes
geneannotation1 <- getBM( attributes = c("ensembl_transcript_id", "entrezgene", "external_gene_name"), f
sort(unique(geneannotation1$external_gene_name))
```

```
##  [1] "ADRA2A"     "AL441992.1" "ARHGEF10"   "C3"         "COL4A4"
##  [6] "DZIP1"      "EFNB2"      "HS3ST3A1"   "ID2"        "NETO2"
## [11] "OSMR"       "PRRX1"      "ROBO1"      "SLC5A3"     "XKR4"
## [16] "YAP1"
```

Generate a heatmap with AITL, PTCL-NOS with the extracted differentially expressed genes.

```
gep<- geneExpr[,select_hist$sample.nameNEW]
mat<- gep[list_genes,]

rownames(mat) = c(unique(geneannotation1$external_gene_name))

mycol= c("red","white","yellow")
mylabel = select_hist[,c("sample.nameNEW","final.molec","IDH2","RHOA","TET2","DNMT3A")]
rownames(mylabel) = mylabel$sample.nameNEW
mylabel$sample.nameNEW = NULL
mylabel.nocol = mylabel
mylabel.col = mylabel
mylabel.col[is.na(mylabel.col)]<-0
#head(mylabel.col)
mylabel.col$final.molec[mylabel.col$final.molec == "AITL"] = "black"; mylabel.col$final.molec[mylabel.co
for (a in 2:5) mylabel.col[,a] = factor(mylabel.col[,a], levels = levels(as.factor(mylabel.col[,a])), la

mat  <- mat - rowMeans(mat)
par(mfrow=c(1,1))
pheatmap(mat, annotation_col = mylabel.nocol, annotation_colors = list(final.molec = c(AITL = "black", I
                                    IDH2 = c(MUT=mycol[1],"NA"=mycol[2],WT=mycol[3]),
                                    RHOA = c(MUT=mycol[1],"NA"=mycol[2],WT=mycol[3]),
                                    TET2 = c(MUT=mycol[1],"NA"=mycol[2],WT=mycol[3]),
                                    DNMT3A = c(MUT=mycol[1],"NA"=mycol[2],WT=mycol[3]) ) , show_colnames =
         border_color= NA, color = colorRampPalette(rev(brewer.pal(n = 5 , name = "RdYlBu")))(100), scal
```




```
#pheatmap::pheatmap(test, filename="test.pdf")
```

LOOCV on AILT, PTCLnos based on 16-gene model

```
y = t(mat)
cl.orig = c()
for (u in 1:nrow(y)) cl.orig [u] = unlist(strsplit(rownames(y)[u],"\\."))[1]

perm.mother = rownames(y)
perm.son = combn (perm.mother, length(perm.mother)-1)

output2 <- cbind(perm.mother, NA)

for (i in 1:length(perm.mother)) {
  train <- y [ perm.son[,i], ]
  test <- y [ ! ( rownames(y) %in% perm.son[,i]) , ]
  cl <- cl.orig [which(rownames(y)%in%perm.son[,i])]
  # z <- lda(train, cl)
    z <- Dlda(train, factor(cl))
```

```
  p <- predict(z,test)$class
  output2  [ setdiff(1:271, which( rownames(y) %in% perm.son[,i]) ) , 2  ] = as.character(p)
#  output  [ output[,1] == rownames(test) , 3  ] = z$scaling [1,1]
#  output  [ output[,1] == rownames(test) , 4  ] = z$scaling [2,1]
#  output  [ output[,1] == rownames(test) , 5  ] = z$scaling [3,1]
}
output<- output2
colnames(output) = c("true","LOOCV.predicted")
output = as.data.frame(output)
output$true.class = cl.orig

output$LOOCV.predicted<- as.character(output$LOOCV.predicted)
output$LOOCV.predicted[output$LOOCV.predicted==0]<-"AITL"
output$LOOCV.predicted[output$LOOCV.predicted==1]<-"PTCL"
confusionMatrix(table(output$true.class, output$LOOCV.predicted  ))
```

```
## Confusion Matrix and Statistics
##
##
##        AITL PTCL
##   AITL  109   18
##   PTCL   13  131
##
##                Accuracy : 0.8856
##                  95% CI : (0.8416, 0.9209)
##     No Information Rate : 0.5498
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7698
##  Mcnemar's Test P-Value : 0.4725
##
##             Sensitivity : 0.8934
##             Specificity : 0.8792
##          Pos Pred Value : 0.8583
##          Neg Pred Value : 0.9097
##              Prevalence : 0.4502
##          Detection Rate : 0.4022
##    Detection Prevalence : 0.4686
##       Balanced Accuracy : 0.8863
##
##        'Positive' Class : AITL
##
```

```
colnames(output)[1]<-"sample"
design2<- as.data.frame(design)
design2$sample<- rownames(design2)
dd<- merge(output,design2, by="sample")

table(dd$LOOCV.predicted, dd$true.class)
```

```
##
##        AITL PTCL
```

```
##   AITL   38    1
##   PTCL    1   13
```

```
  #             AITL ALCL-ALKneg PTCL-NOS
  # AITL           4           2        7
  # ALCL-ALKneg    0           4        2
  # PTCL-NOS       7           2        2
table(dd$LOOCV.predicted, dd$RHOA)
```

```
##
##         0  1
##   AITL 11 28
##   PTCL  7  7
```

```
  #             0 1
  # AITL        9 4
  # ALCL-ALKneg 6 0
  # PTCL-NOS    9 2
table(dd$LOOCV.predicted, dd$TET2)
```

```
##
##         0  1
##   AITL  7 32
##   PTCL  6  8
```

```
  #             0 1
  # AITL        8 5
  # ALCL-ALKneg 6 0
  # PTCL-NOS    9 2
table(dd$LOOCV.predicted, dd$IDH2)
```

```
##
##         0  1
##   AITL 25 14
##   PTCL 13  1
```

```
  #            0  1
  # AITL        11  2
  # ALCL-ALKneg  6  0
```

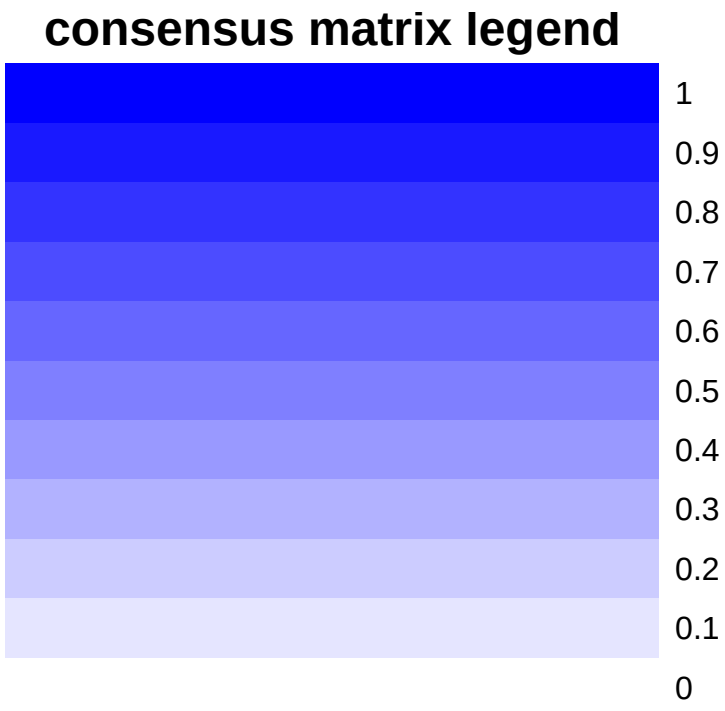
Extracting the most significant clusters based on 19-gene signature

Analyze sample stratification based on the extracted differentially expressed genes betwee AILT and PTCL-nos and the ALCL ALK-negative 3-gene model.

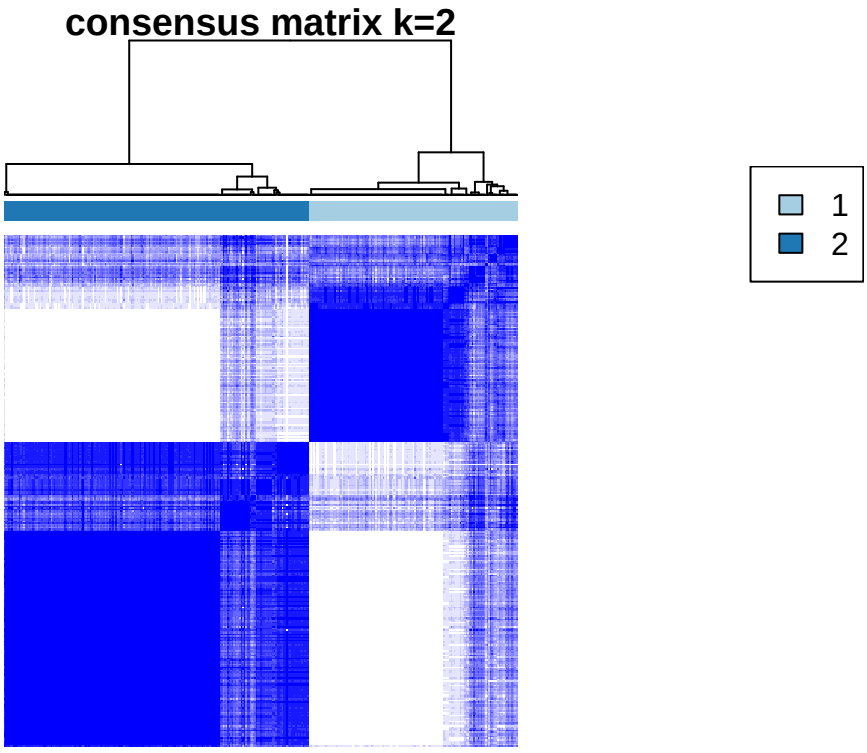
```
select_hist<- pts.info.data[pts.info.data$final.molec == "AITL" | pts.info.data$final.molec == "PTCL.nos
# Add three classifier genes for ALCL ALK-neg [Agnelli et al, Blood, 2012]
# Check on array
anaplastic_gene<- c("TNFRSF8","BATF3","TMOD1")
geneannotation2 <- getBM( attributes = c("entrezgene", "external_gene_name"), filters = "external_gene_
```

```
anaplastic_gene_ARRAY<- paste0(geneannotation2$entrezgene, "_at")

# Append 16-gene model to 3-gene model
list_genes_all<- c(list_genes, anaplastic_gene_ARRAY)

# Redo consensus cluster analysis
gep<- geneExpr[,select_hist$sample.nameNEW]
mat<- gep[list_genes_all,]
title=tempdir()
d<- data.matrix(mat)
d = sweep(d,1, apply(d,1,median,na.rm=T))
results = ConsensusClusterPlus(d,maxK=8,
                               pFeature=1,
                               title=title,
                               clusterAlg="hc",
                               innerLinkage="ward.D2",
                               finalLinkage="ward.D2",
                               distance="euclidean",
                               seed=123456789)
```

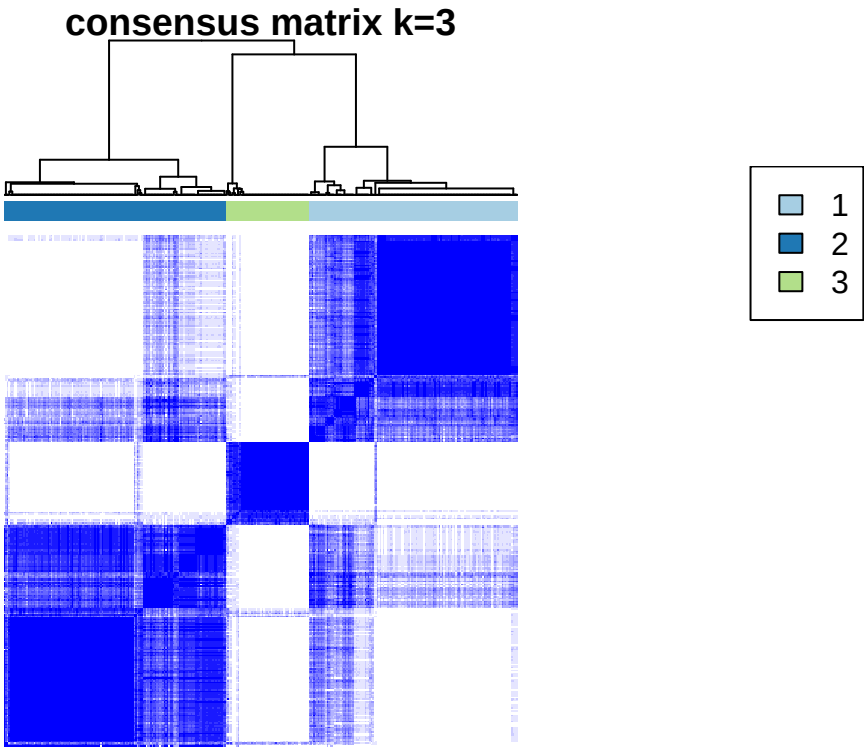
```
## end fraction

## clustered
```



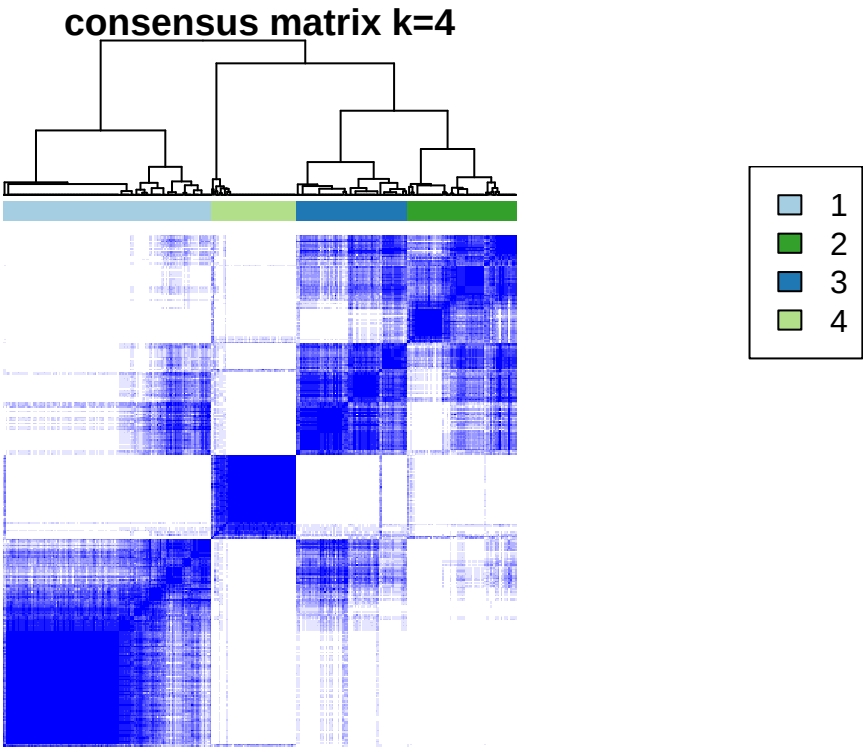

```
## clustered
```



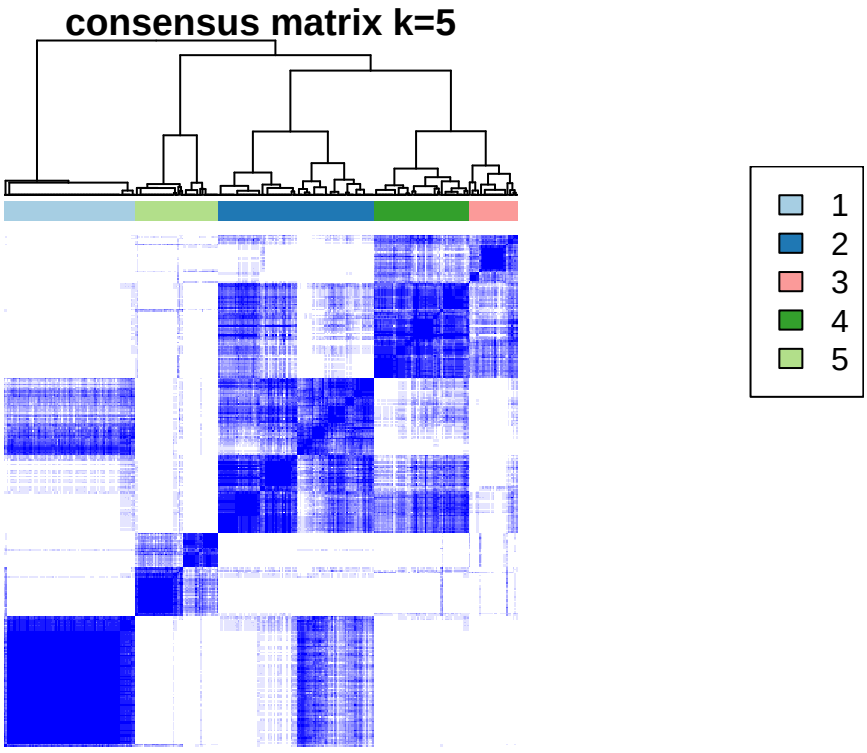

```
## clustered
```



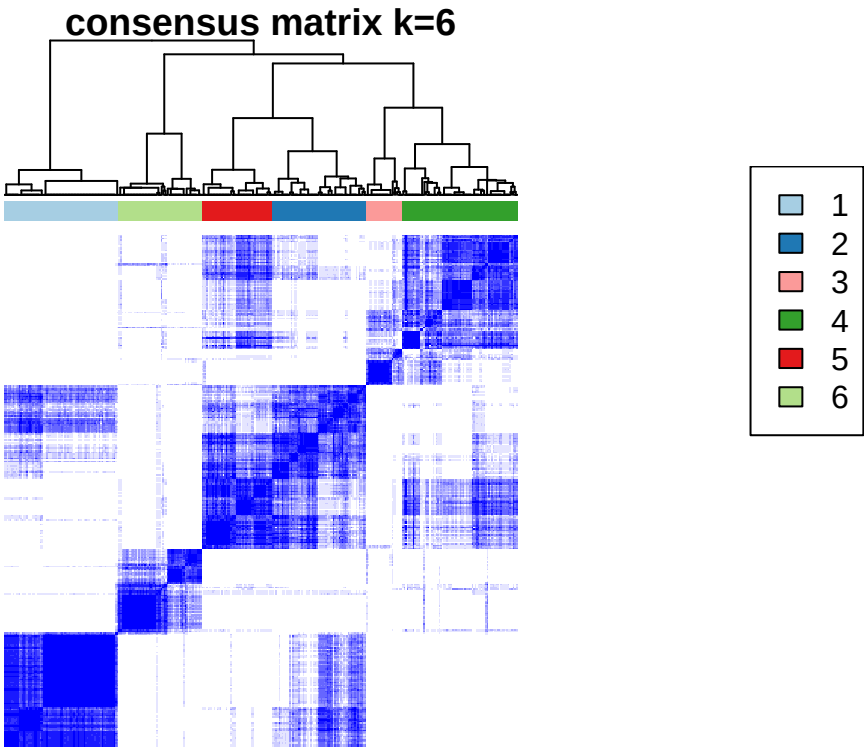

```
## clustered
```



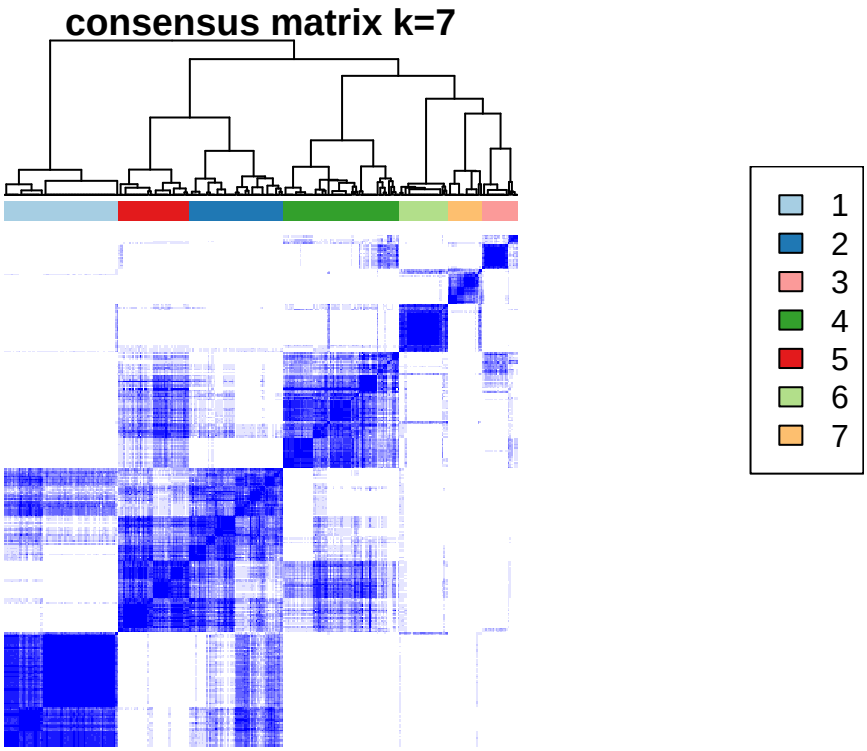

```
## clustered
```

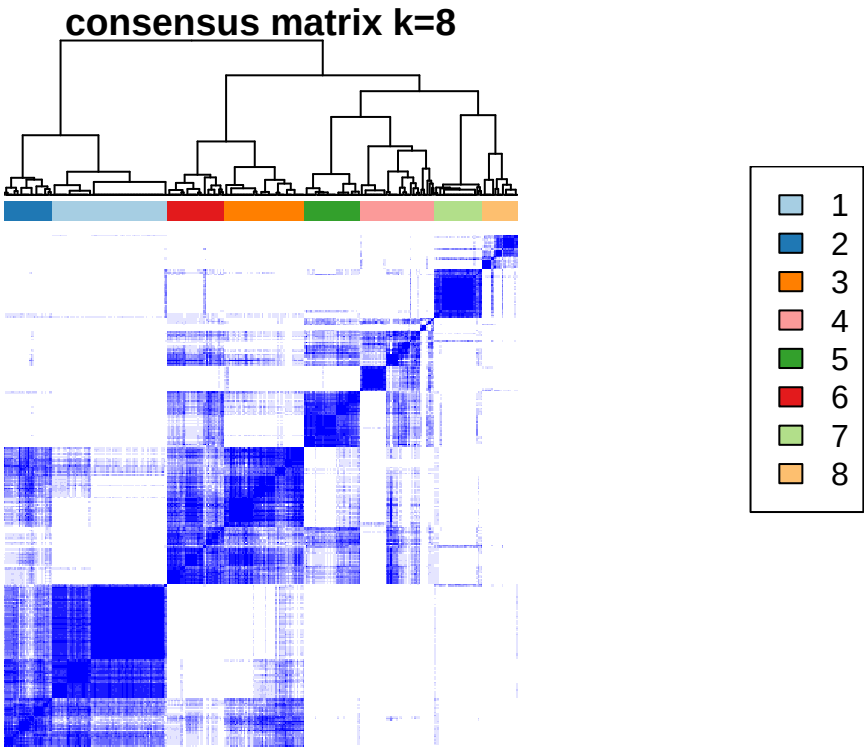


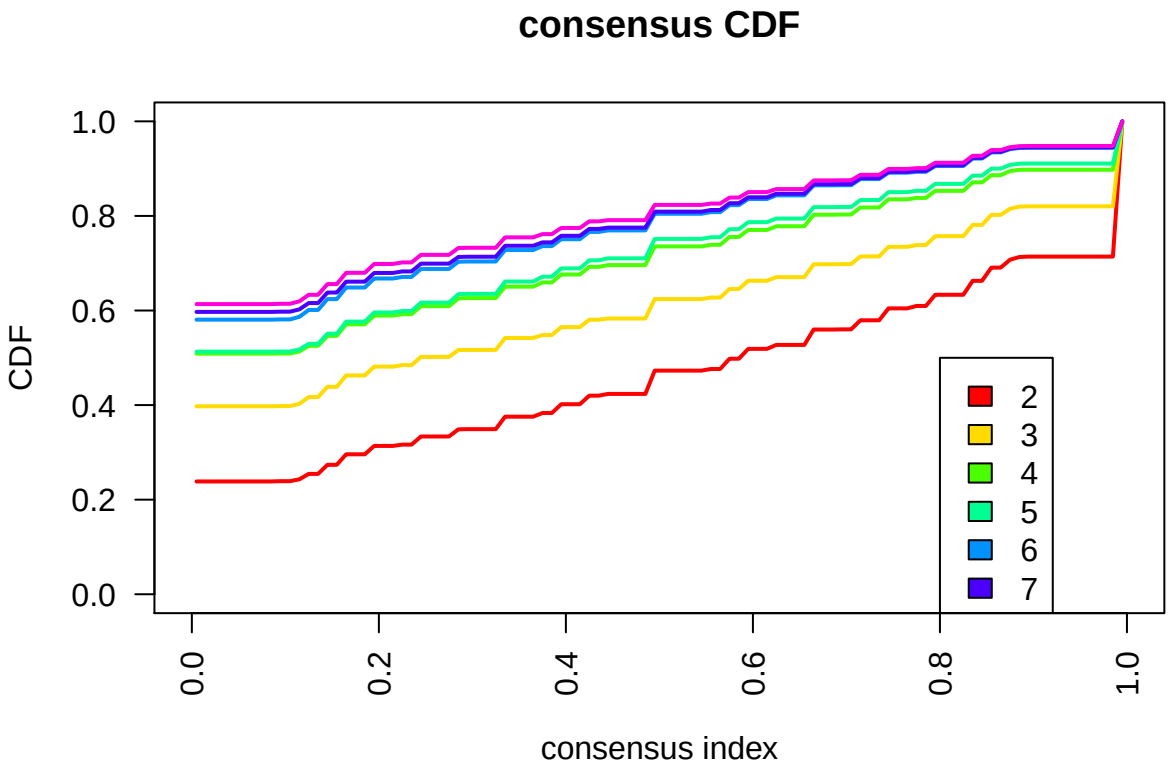

```
## clustered
```

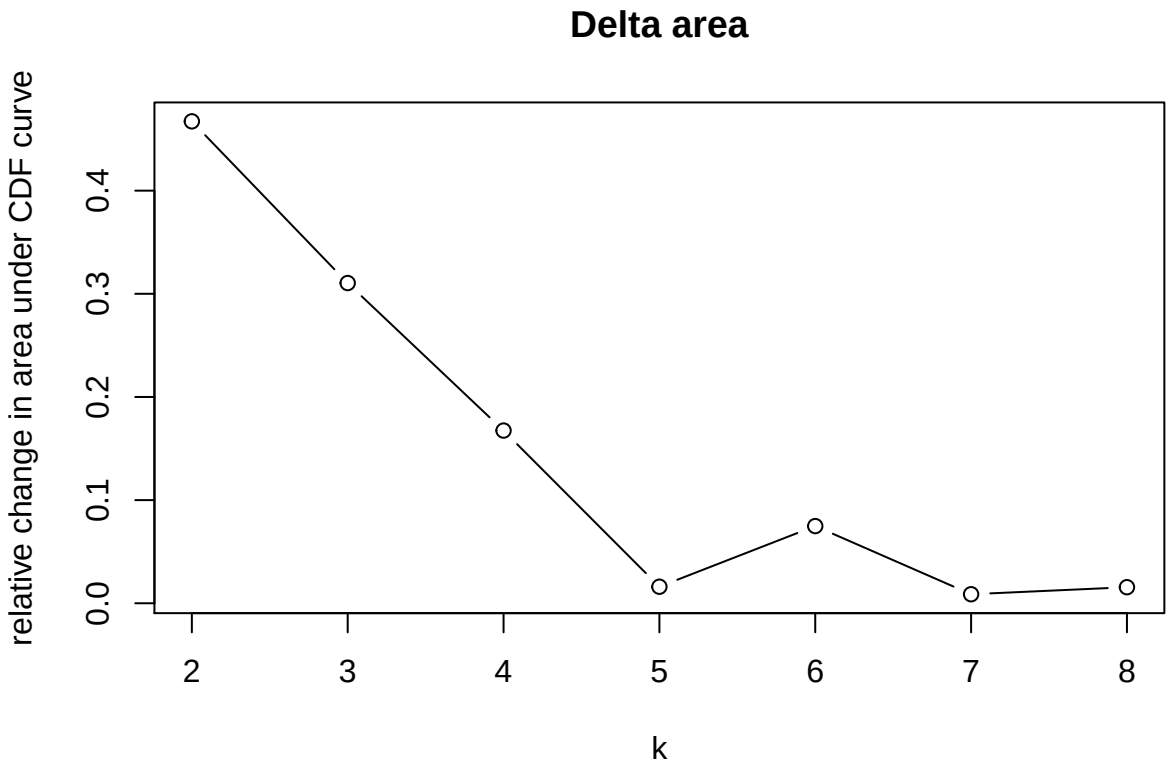


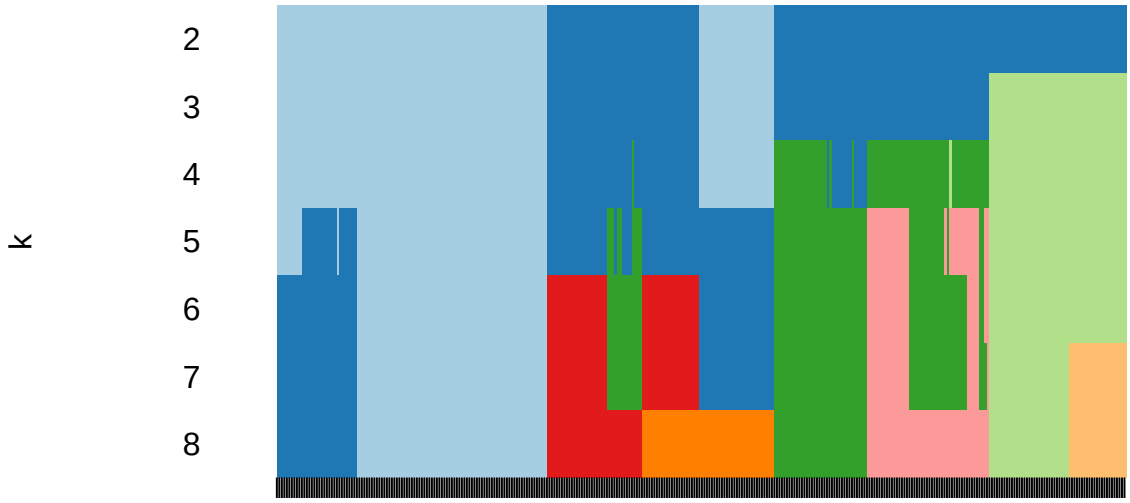

```
## clustered
```


consensus matrix k=7
1
2
3
4
5
6
7


consensus matrix k=8
1
2
3
4
5
6
7
8


consensus CDF
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
consensus index
2
3
4
5
6
7


Delta area
relative change in area under CDF curve
0.0
0.1
0.2
0.3
0.4
2
3
4
5
6
7
8
k



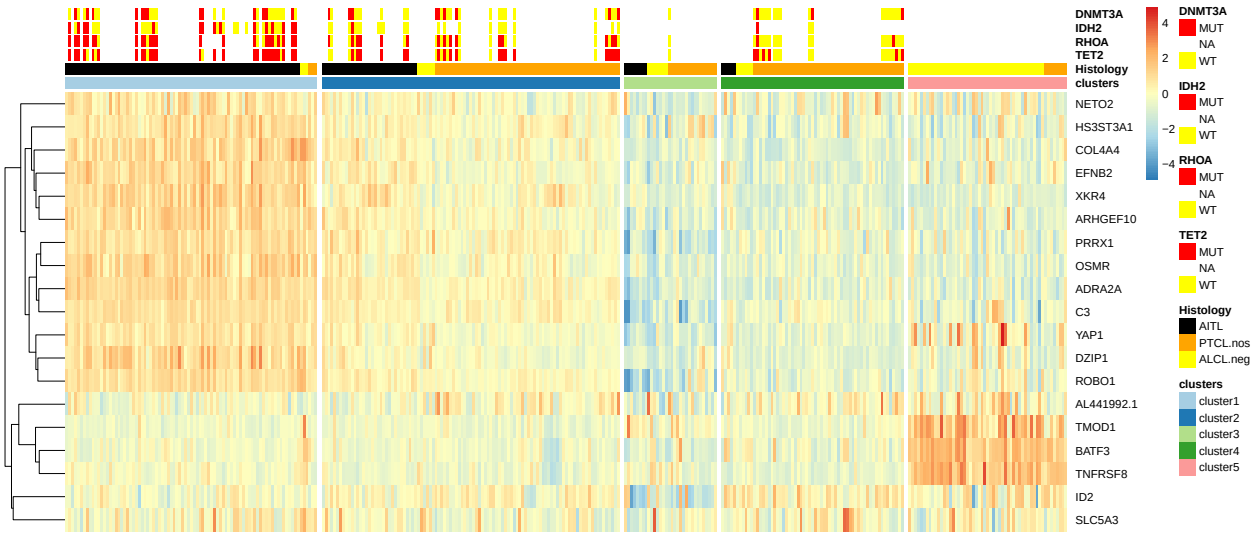

```
kk<- as.data.frame((results[[5]]$consensusClass)) ##### 5 significant cluster
kk$geo.id<- rownames(kk)
colnames(kk)[1]<- "cluster"
table(kk$cluster)
```

```
##
##   1   2   3   4   5
##  87 103  32  63  55
```

Plot heatmap AITL, PTCL-NOS, ALCL-neg and the 19-gene model

```
heat<- merge(t(mat), kk, by.x = 0, by.y="geo.id")
heat2<- merge(heat, pts.info.data, by.x = 1, by.y="sample.nameNEW")
heat2<- heat2[order(heat2$cluster),]
mycol= c("red","white","yellow")
mylabel = heat2[,c("Row.names","cluster","final.molec","TET2","RHOA","IDH2","DNMT3A")]
colnames(mylabel)<- c("sample.names","clusters","Histology","TET2","RHOA","IDH2","DNMT3A")
rownames(mylabel) = mylabel$sample.names
mylabel$sample.names  = NULL
mylabel.nocol = mylabel
mylabel.col = mylabel
mylabel.col[is.na(mylabel.col)]<-0
#head(mylabel.col)
```

```
mylabel.col$Histology[mylabel.col$Histology == "AITL"] = "black"; mylabel.col$Histology[mylabel.col$Hist
for (a in c(3:6)) mylabel.col[,a] = factor(mylabel.col[,a], levels = levels(as.factor(mylabel.col[,a]))
mycol_plus<- c(brewer.pal(11,"Paired"),brewer.pal(6,"Set2"))
for (a in 1) mylabel.col[,a] = factor(mylabel.col[,a], levels = levels(as.factor(mylabel.col[,a])), labe
mylabel.nocol$clusters<-as.numeric(as.character(mylabel.nocol$clusters))
mylabel.nocol$clusters<-as.character(paste("cluster",mylabel.nocol$clusters, sep=""))

# pdf("Figure_3.pdf", width = 20, height = 10)
par(mfrow=c(1,1))
par(mar=c(5,5,5,5), xpd=F)
mat3<- t(data.matrix(heat2[,2:20]))
colnames(mat3)<-heat2$Row.names
mat3= mat3[order(rownames(mat3)),]
temp_name = getBM( attributes = c("ensembl_transcript_id", "entrezgene", "external_gene_name"), filters
temp_name = temp_name[!duplicated(temp_name[,1]),]
rownames(mat3) = c(temp_name [,2] )
mat3  <- mat3 - rowMeans(mat3)
num_clust<- as.numeric(table(mylabel.nocol$clusters))
num<- c(num_clust[1], sum(num_clust[1:2]),sum(num_clust[1:3]),sum(num_clust[1:4]),sum(num_clust[1:5]) )
par(mfrow=c(1,1))
pheatmap(mat3, annotation_col = mylabel.nocol, annotation_colors = list(clusters = c(cluster1= mycol_plu
         # gaps_col=c(0,rep(0,num[1]-1), 40,rep(0,num[2]-1), 1000,rep(0,num[3]-1), 40,rep(0,num[4]-1),
         #            40,rep(0,num[5]-1)))
         gaps_col = num)
```



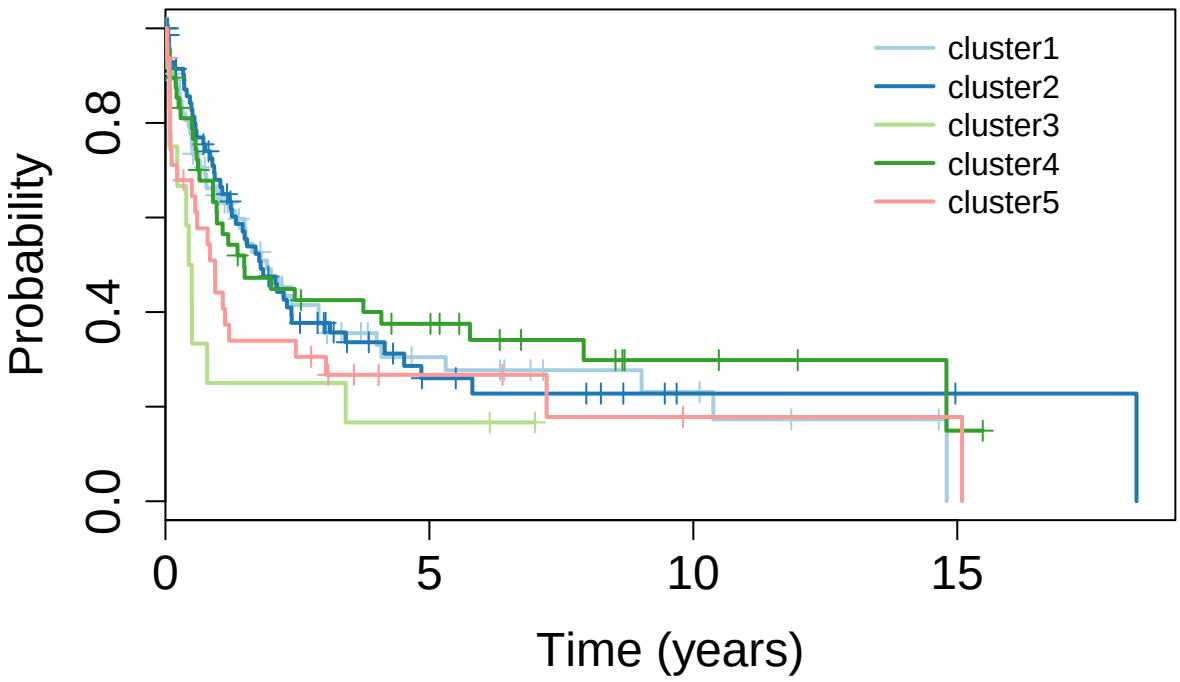
```
# dev.off()
```

Clinical impact of each cluster

```
tab_2<- mylabel.nocol
tab_2$sample.nameNEW<- rownames(tab_2)
pts.info.data <- read.table("./Rmd.files/541_paz_info_MUT.txt", sep="\t", header=TRUE, check.names=FALSE)

prog<- merge(pts.info.data[,c(1,5,7,8,9,10,11,12,13)], tab_2, by="sample.nameNEW")
prog<- prog[complete.cases(prog$time),]


summary(coxph(Surv(time, status) ~ clusters, data=prog))
```

```
## Call:
## coxph(formula = Surv(time, status) ~ clusters, data = prog)
##
##   n= 239, number of events= 160
##
##                       coef exp(coef) se(coef)      z Pr(>|z|)
## clusterscluster2 -0.05210   0.94924  0.20654 -0.252   0.8009
## clusterscluster3  0.65350   1.92226  0.34938  1.870   0.0614 .
## clusterscluster4 -0.09999   0.90485  0.23218 -0.431   0.6667
## clusterscluster5  0.32114   1.37869  0.25190  1.275   0.2024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## clusterscluster2    0.9492     1.0535    0.6332     1.423
## clusterscluster3    1.9223     0.5202    0.9692     3.812
## clusterscluster4    0.9048     1.1052    0.5740     1.426
## clusterscluster5    1.3787     0.7253    0.8415     2.259
##
## Concordance= 0.552  (se = 0.024 )
## Rsquare= 0.024   (max possible= 0.998 )
## Likelihood ratio test= 5.92  on 4 df,   p=0.2
## Wald test            = 6.66  on 4 df,   p=0.2
## Score (logrank) test = 6.85  on 4 df,   p=0.1
```

```
plot(survfit(Surv(time, status) ~ clusters, data=prog), lwd =2,  mark.time = TRUE,  ylab = "Probability
     xlab = "Time (years)", cex.axis = 1.5, cex.lab = 1.5, col=mycol_plus, lty=1)
legend("topright", legend=sort(unique(prog$clusters)),
       col=mycol_plus,bty = "n", lty=1, lwd=2, cex=1, pt.cex=0.5,
       inset=c(+0.1,0.0), x.intersp = 0.5)
```



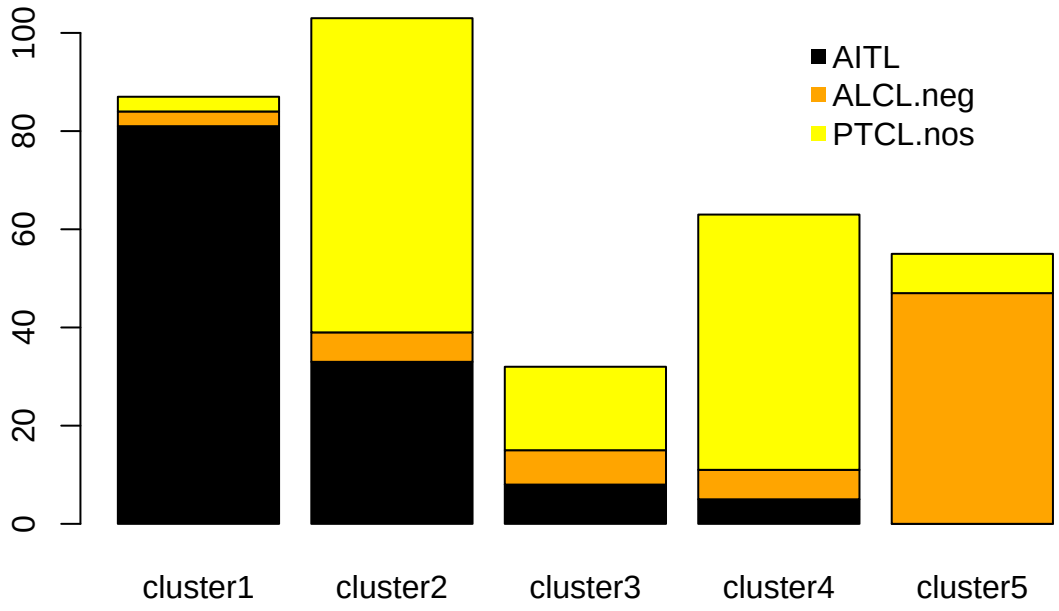

```
prog$age<- as.numeric(as.character(prog$age))
summary(coxph(Surv(time, status) ~ clusters + age + stage + ipi, data=prog))
```

```
## Call:
## coxph(formula = Surv(time, status) ~ clusters + age + stage +
##     ipi, data = prog)
##
##   n= 49, number of events= 30
##    (190 observations deleted due to missingness)
##
##                       coef exp(coef) se(coef)      z Pr(>|z|)
## clusterscluster2 -0.35570   0.70068  0.49993 -0.711   0.4768
## clusterscluster3 -0.39088   0.67646  1.07600 -0.363   0.7164
## clusterscluster4  0.68481   1.98340  0.53819  1.272   0.2032
## clusterscluster5 -0.45744   0.63290  0.67354 -0.679   0.4970
## age               0.02038   1.02059  0.01472  1.385   0.1661
## stage             0.53861   1.71363  0.32394  1.663   0.0964 .
## ipi              -0.12117   0.88588  0.23752 -0.510   0.6100
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## clusterscluster2    0.7007     1.4272    0.2630     1.867
## clusterscluster3    0.6765     1.4783    0.0821     5.574
## clusterscluster4    1.9834     0.5042    0.6907     5.695
```

```
## clusterscluster5    0.6329     1.5800    0.1691     2.369
## age                 1.0206     0.9798    0.9916     1.050
## stage               1.7136     0.5836    0.9082     3.233
## ipi                 0.8859     1.1288    0.5562     1.411
##
## Concordance= 0.619  (se = 0.057 )
## Rsquare= 0.133   (max possible= 0.985 )
## Likelihood ratio test= 6.99  on 7 df,   p=0.4
## Wald test            = 6.79  on 7 df,   p=0.5
## Score (logrank) test = 7.04  on 7 df,   p=0.4
```

Histological Composition of each cluster

```
prog<- merge(pts.info.data[,c(1,5,7,8,9,10,11,12,13)], tab_2, by="sample.nameNEW")
barplot(table(prog$Histology, prog$clusters), col=c("black","orange","yellow"))
legend("topright", legend=sort(unique(prog$Histology)),
       col=c("black","orange","yellow"),bty = "n", pch=15,  cex=1, pt.cex=1,
       inset=c(+0.1,0.0), x.intersp = 0.5)
```



LOOCV on AILT, ALCL neg, PTCLnos based on 19-gene model

```r
y = t(mat3)
cl.orig = c()
for (u in 1:nrow(y)) cl.orig [u] = unlist(strsplit(rownames(y)[u],"\\."))[1]

perm.mother = rownames(y)
perm.son = combn (perm.mother, length(perm.mother)-1)

output <- cbind(perm.mother, NA)

for (i in 1:length(perm.mother)) {
  train <- y [ perm.son[,i], ]
  test <- y [ ! ( rownames(y) %in% perm.son[,i]) , ]
  cl <- cl.orig [which(rownames(y)%in%perm.son[,i])]
  z <- lda(train, cl)
  p <- predict(z,test)$class
  output  [ setdiff(1:271, which( rownames(y) %in% perm.son[,i]) ) , 2  ] = as.character(p)
#  output  [ output[,1] == rownames(test) , 3  ] = z$scaling [1,1]
#  output  [ output[,1] == rownames(test) , 4  ] = z$scaling [2,1]
#  output  [ output[,1] == rownames(test) , 5  ] = z$scaling [3,1]
}

colnames(output) = c("true","LOOCV.predicted")
output = as.data.frame(output)
output$true.class = cl.orig

table(output$true.class, output$LOOCV.predicted  )
```

```
##
##        AITL ALCL PTCL
##   AITL  109    1   17
##   ALCL    4    3   15
##   PTCL   13    4  105
```

```r
library(caret)
library(e1071)
confusionMatrix(table(output$true.class, output$LOOCV.predicted))
```

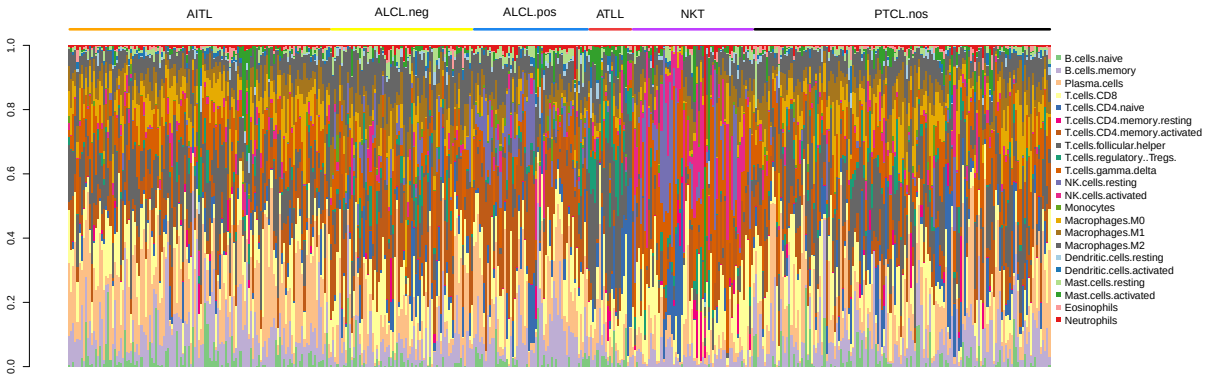
```
## Confusion Matrix and Statistics
##
##
##        AITL ALCL PTCL
##   AITL  109    1   17
##   ALCL    4    3   15
##   PTCL   13    4  105
##
## Overall Statistics
##
##                Accuracy : 0.8007
##                  95% CI : (0.7481, 0.8466)
##     No Information Rate : 0.5055
##     P-Value [Acc > NIR] : < 2e-16
##
```

```
##                   Kappa : 0.6391
##  Mcnemar's Test P-Value : 0.03353
##
## Statistics by Class:
##
##                      Class: AITL Class: ALCL Class: PTCL
## Sensitivity               0.8651     0.37500      0.7664
## Specificity               0.8759     0.92776      0.8731
## Pos Pred Value            0.8583     0.13636      0.8607
## Neg Pred Value            0.8819     0.97992      0.7852
## Prevalence                0.4649     0.02952      0.5055
## Detection Rate            0.4022     0.01107      0.3875
## Detection Prevalence      0.4686     0.08118      0.4502
## Balanced Accuracy         0.8705     0.65138      0.8198
```

Cibersort to characterize tumour microenviroment composition of each cluster

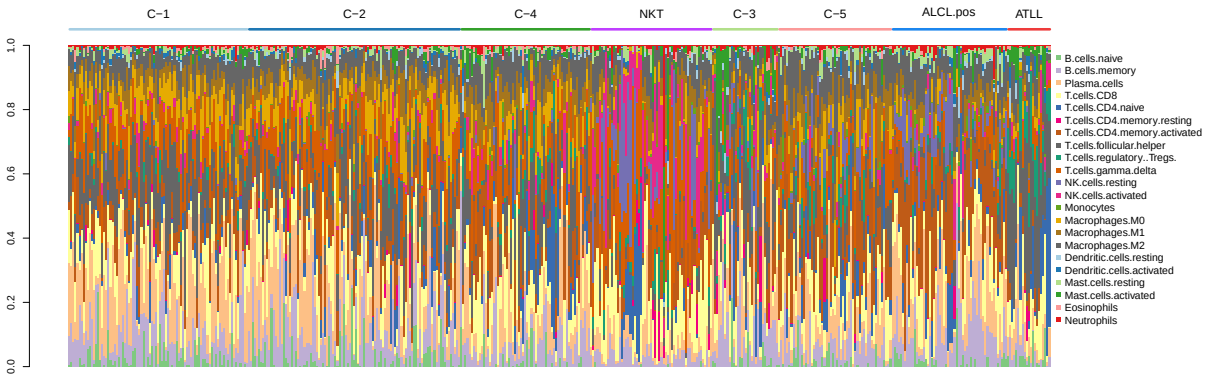
Focus the analysis on AITL, PTCL-NOS and ALCL-neg

```
#cibersort.percentages<- read.delim("./Rmd.files/cibersort.percentages.txt", sep="\t", stringsAsFactors
load("./Rmd.files/cibersort.all.Rdata")
ciber_all<-as.data.frame.matrix(t(cibersort.percentages))
ciber_all$sample.nameNEW <- rownames(ciber_all)
colnames(kk)[2]<-"sample.nameNEW"
require(plyr)
final <-join(ciber_all, kk, by = "sample.nameNEW",  type="left")
final2<-merge(pts.info.data[,c(1,6,14:17)], final, by="sample.nameNEW")
final3<- subset(final2, final.molec %in% c("AITL","ALCL.neg","ALCL.pos","ATLL","NKT","PTCL.nos"))
final3<- final3[order(final3$final.molec),]
library(RColorBrewer)
n <- 22
qual_col_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxcolors, rownames(qual_col_pals)))

par(mar=c(2,5,7,10), xpd=TRUE)
x<- barplot(t(final3[7:28]), names.arg = rep("", length(final3$final.molec)), cex.names = 0.7, col=col_
            space=rep(0, nrow(final3)))
legend("topright",legend=colnames(final3)[7:28], col=col_vector, pch=c(15), inset=c(-0.11,0), pt.cex= 1
cex = 1, bty = "n",  x.intersp = 0.7)


names_hist<- unique(final3$final.molec)
col_hist<- c("orange","yellow","dodgerblue2","brown2","darkorchid1","black")
num<- as.numeric(table(final3$final.molec))
for(i in (1:length(num)))
{
  segments(x[sum(num[1:i])+1-num[i]], 1.05,x[sum(num[1:i])],1.05,lwd=4, col=col_hist[i])
  text(x[(sum(num[1:i])-num[i] +1+ sum(num[1:i]))/2], 1.1, names_hist[i], cex=1.2, srt=0)

}
```



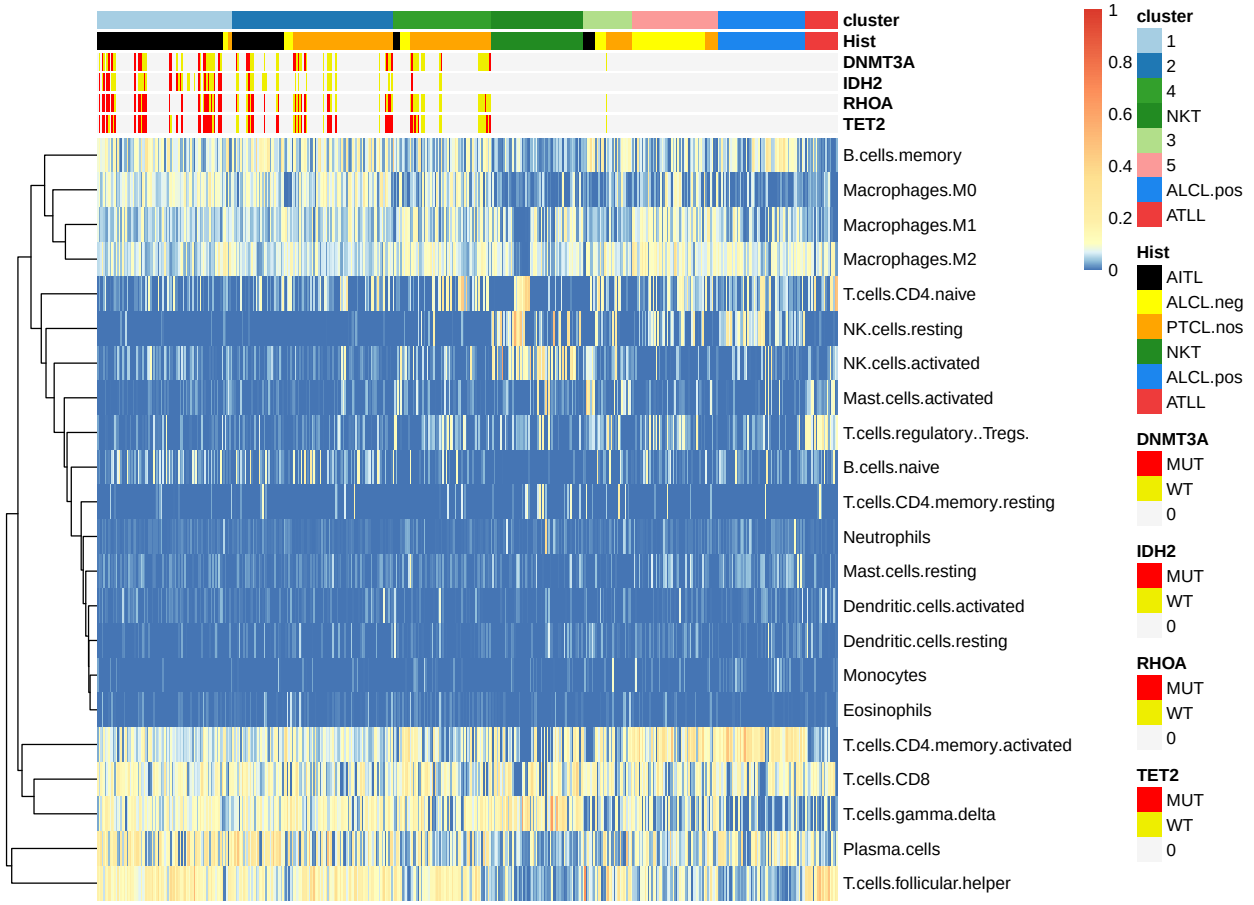
Boxplot comparing the contribution of each cibersort signature between all extracted clusters

```
for(i in (1:nrow(final3)))
{
final3$cluster[i][is.na(final3$cluster[i])]<- final3$final.molec[i]
}

final3$cluster <- factor(final3$cluster, levels = c( "1","2","4","NKT","3","5","ALCL.pos", "ATLL"))

final3<- final3[order(final3$cluster),]

#pdf("barplot_cibersort.pdf", width = 20, height = 7)
par(mar=c(2,5,7,10), xpd=TRUE)
x<- barplot(t(final3[7:28]), names.arg = rep("", length(final3$final.molec)), cex.names = 0.7, col=col_
            space=rep(0, nrow(final3)))
legend("topright",legend=colnames(final3)[7:28], col=col_vector, pch=c(15), inset=c(-0.11,0), pt.cex= 1
cex = 1, bty = "n",  x.intersp = 0.7)

mycol_plus<- c(brewer.pal(11,"Paired"),brewer.pal(6,"Dark2"))
names_hist<- c("C-1","C-2", "C-4","NKT","C-3","C-5","ALCL.pos","ATLL")
col_hist<- c(mycol_plus[1],mycol_plus[2],mycol_plus[4],"darkorchid1",mycol_plus[3],mycol_plus[5],"dodge
num<- as.numeric(table(final3$cluster))
  par(new=TRUE)
for(i in (1:(length(num))))
{
  segments(x[sum(num[1:i])+1-num[i]], 1.05,x[sum(num[1:i])],1.05,lwd=4, col=col_hist[i])
  text(x[(sum(num[1:i])-num[i] +1+ sum(num[1:i]))/2], 1.1, names_hist[i], cex=1.2, srt=0)

}
```



Cibersort Heatmap

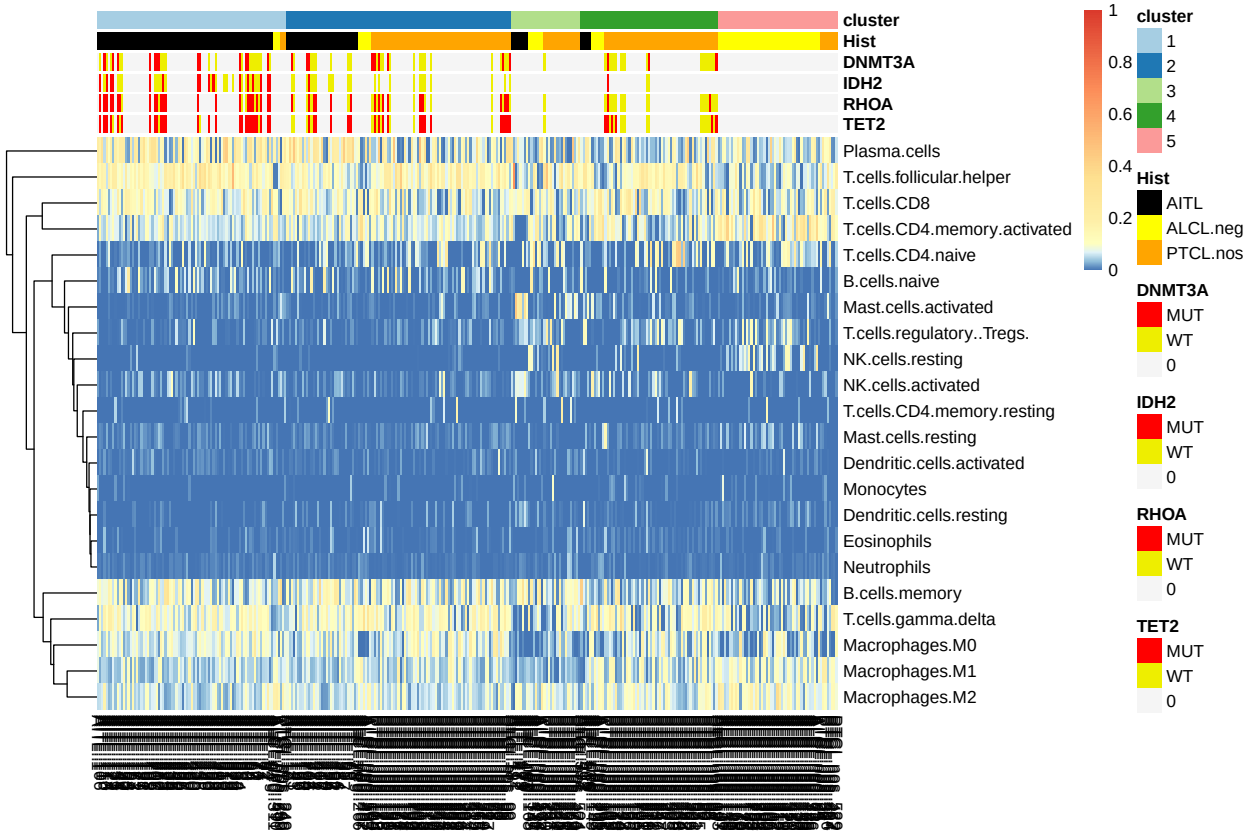
```
annotation_col<- final3[,c("TET2","RHOA","IDH2","DNMT3A","final.molec","cluster")]
annotation_col[is.na(annotation_col)]<-0
rownames(annotation_col)<- final3$sample.nameNEW
colnames(annotation_col)[5]<-c("Hist")
A <- function(x) (as.factor(as.character(x))) ##### lapply function for all columns to generate the rela
annotation_col[,1:ncol(annotation_col)] = apply(annotation_col[,1:ncol(annotation_col)], 2, function(x)
annotation_col<- as.data.frame(annotation_col)
mycol_plus<- c(brewer.pal(11,"Paired"),brewer.pal(6,"Dark2"))
color.annot<- as.character(color.annot)
ann_colors = list(Hist=c( "AITL"=color.annot[1],"ALCL.neg"=color.annot[2],"PTCL.nos"=color.annot[6], "NI
                          "ALCL.pos" =color.annot[3],"ATLL"=color.annot[4]),
                  cluster=c("1" = mycol_plus[1],"2" = mycol_plus[2],"4" = mycol_plus[4],"NKT" = color.ar
                            "3" = mycol_plus[3],"5" = mycol_plus[5], "ALCL.pos" = color.annot[3],"ATLL"
                  DNMT3A = c("MUT"="red","WT"= "yellow2","0"="grey96"),
                  IDH2 = c("MUT"="red","WT"= "yellow2","0"="grey96"),
                  RHOA = c("MUT"="red","WT"= "yellow2","0"="grey96"),
                  TET2 = c("MUT"="red","WT"= "yellow2","0"="grey96")
                  )
edata<- as.matrix(((final3[,c(7:28)])))
rownames(edata)<-final3$sample.nameNEW
library(pheatmap)
pheatmap(as.matrix( t(edata)), annotation_col=annotation_col, annotation_colors = ann_colors,
         breaks = c(seq(0, 0.1, by= 0.001), seq(0.101, 0.2, by= 0.005),seq(0.21, 1, by= 0.01 )) , color
         cluster_cols =  F, border_color="NA", show_colnames = F)
```



focus the analysis on AITL, PTCL-NOS and ALCL-neg

```
load(file.path(data.dir,"./Rmd.files//cibersort.all.Rdata"))
ciber<-as.data.frame.matrix(t(cibersort.percentages))
ciber_select<-ciber[rownames(kk),]
ciber_select$sample.nameNEW<- rownames(ciber_select)
final<- merge(ciber_select, kk, by="sample.nameNEW")
final2<- merge(final, pts.info.data, by="sample.nameNEW")
final2<- final2[order(final2$cluster),]

annotation_col<- final2[,c("TET2","RHOA","IDH2","DNMT3A","final.molec","cluster")]
annotation_col[is.na(annotation_col)]<-0
```

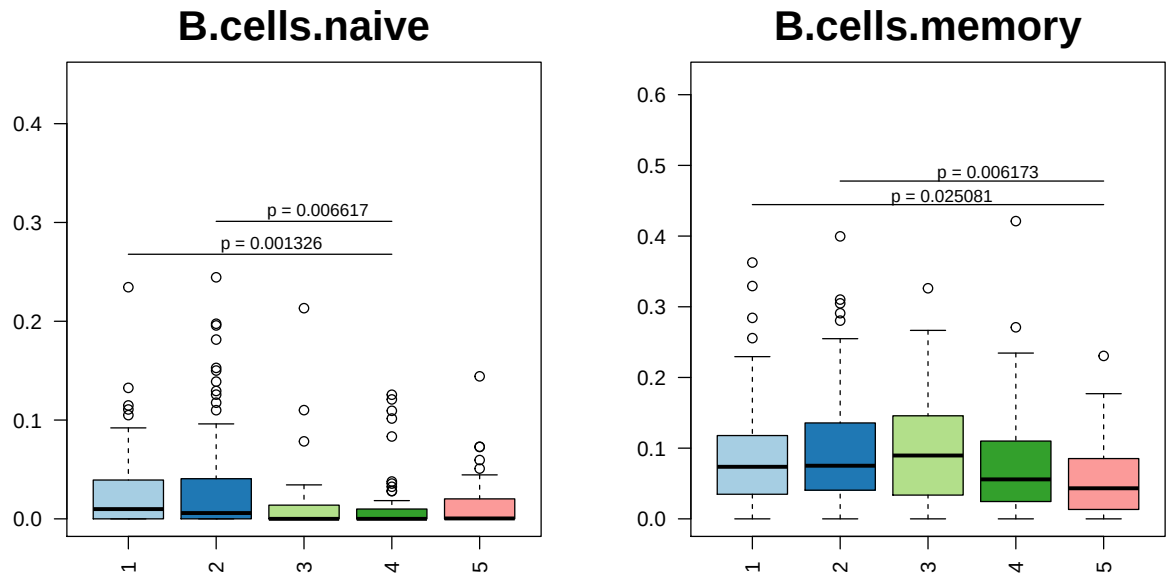
```
rownames(annotation_col)<- final2$sample.nameNEW
colnames(annotation_col)[5]<-c("Hist")

A <- function(x) (as.factor(as.character(x))) ##### lapply function for all columns to generate the rela
annotation_col[,1:ncol(annotation_col)] = apply(annotation_col[,1:ncol(annotation_col)], 2, function(x)
annotation_col<- as.data.frame(annotation_col)
mycol_plus<- c(brewer.pal(11,"Paired"),brewer.pal(6,"Dark2"))

mycol_plus<- c(brewer.pal(11,"Paired"),brewer.pal(6,"Set2"))
color.annot<- as.character(color.annot)
ann_colors = list(Hist=c( "AITL"=color.annot[1],"ALCL.neg"=color.annot[2],"PTCL.nos"=color.annot[6]),
                  cluster=c("1" = mycol_plus[1],"2" = mycol_plus[2],"3" = mycol_plus[3],"4" = mycol_plus
                  DNMT3A = c("MUT"="red","WT"= "yellow2","0"="grey96"),
                  IDH2 = c("MUT"="red","WT"= "yellow2","0"="grey96"),
                  RHOA = c("MUT"="red","WT"= "yellow2","0"="grey96"),
                  TET2 = c("MUT"="red","WT"= "yellow2","0"="grey96")
                  )
edata<- as.matrix(((final2[,c(2:23)])))
rownames(edata)<-final2$sample.nameNEW
library(pheatmap)
pheatmap(as.matrix( t(edata)), annotation_col=annotation_col, annotation_colors = ann_colors,
         breaks = c(seq(0, 0.1, by= 0.001), seq(0.101, 0.2, by= 0.005),seq(0.21, 1, by= 0.01 )) , color
         cluster_cols =  F, border_color="NA")
```

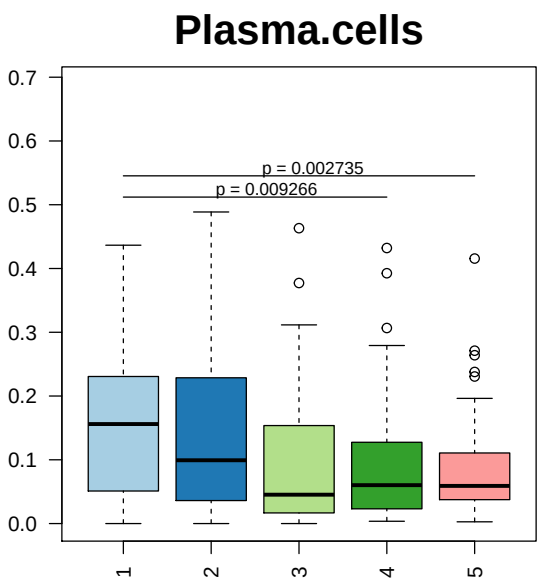


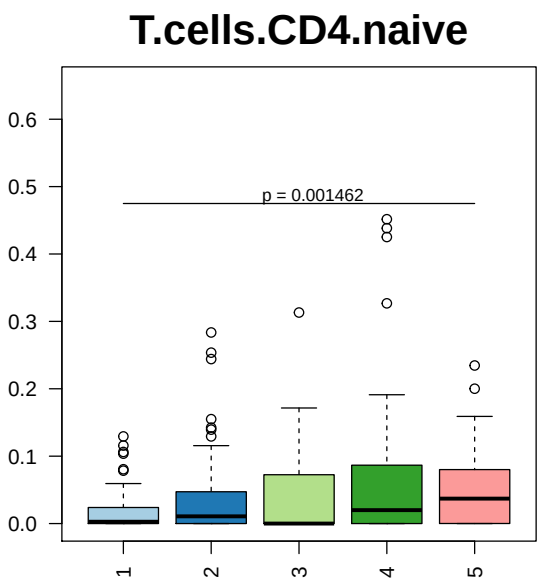
Boxplot comparing the contribution of each cibersort signature between all extracted clusters

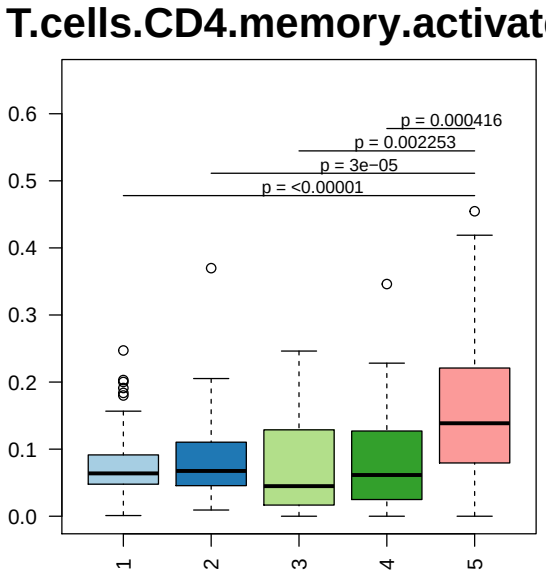
```
par(mfrow=c(1,2))
par(mar=c(3,3,3,3), xpd=F)
for(i in (2:23))
{
  k<- as.numeric(final2[,i])
  table_wilk<- pairwise.wilcox.test(k,final2$cluster,p.adjust.methods = "bonferroni" )$p.value
  df_wilk <- data.frame(expand.grid(dimnames(table_wilk)),array(table_wilk))
  df_wilk2<-na.omit(df_wilk)
  df_wilk2_sig<- df_wilk2[df_wilk2$array.table_wilk.<0.05,]
```

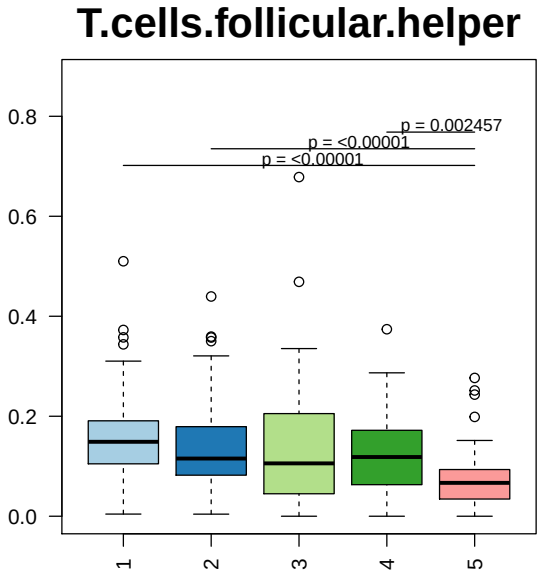
```
df_wilk2_sig$Var1<-as.numeric(as.character(df_wilk2_sig$Var1))
df_wilk2_sig$Var2<-as.numeric(as.character(df_wilk2_sig$Var2))
if(nrow(df_wilk2_sig)>0)
{
boxplot(k~final2$cluster, ylim=c(0,(max(k)+0.2)), main=colnames(final2)[i], cex.main=2, col=mycol_plus
for(j in (1:nrow(df_wilk2_sig)))
{
  segments(df_wilk2_sig$Var1[j], max(k)-0.01+j/30, df_wilk2_sig$Var2[j],max(k)-0.01+j/30)
  p<-df_wilk2_sig$array.table_wilk.[j]
  if(p<0.00001){p2 = "<0.00001"}else{
  p2<-as.numeric(formatC(p,digits=6,format="f"))}
  pval <- paste("p =",p2,sep=" ")
  text((df_wilk2_sig$Var1[j]+ df_wilk2_sig$Var2[j])/1.9,  max(k) +j/30, pval, cex=0.8)
}
  }
 }
```

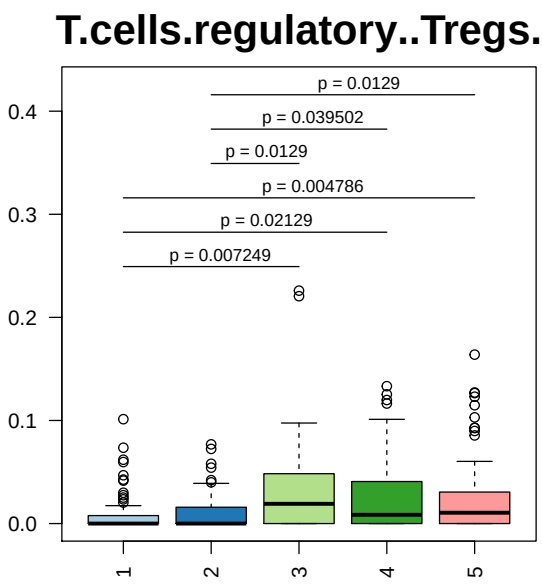


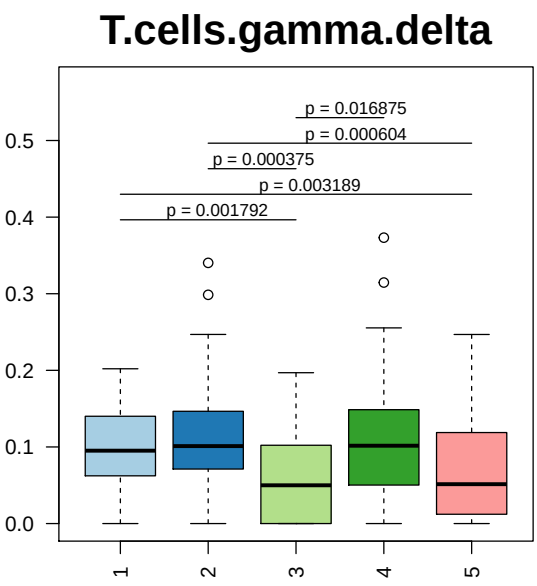

Plasma.cells
p = 0.002735
p = 0.009266
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

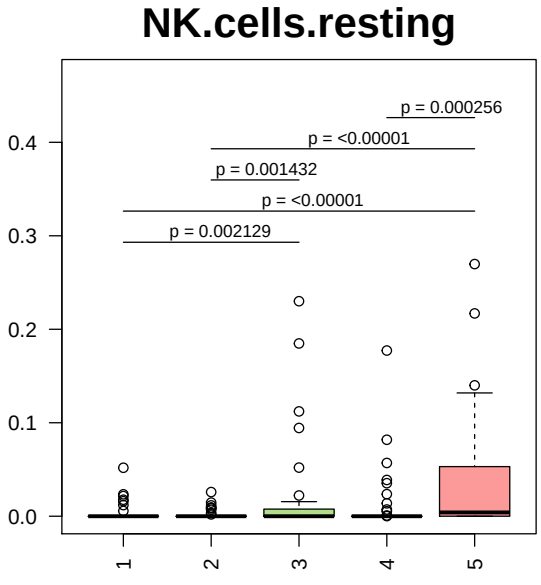

T.cells.CD4.naive
p = 0.001462
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

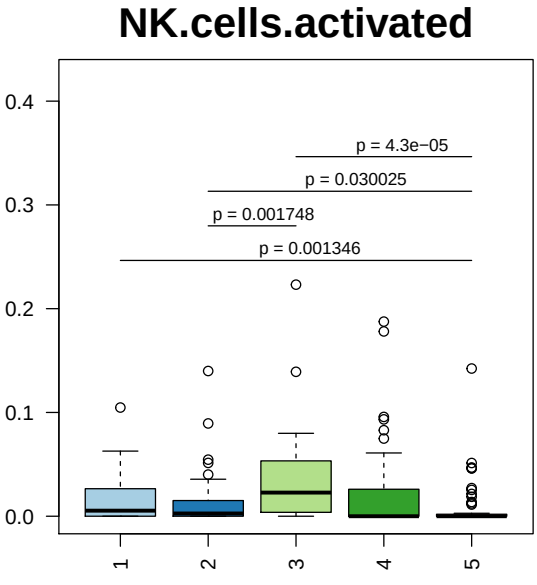

T.cells.CD4.memory.activated
p = 0.000416
p = 0.002253
p = 3e−05
p = <0.00001
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

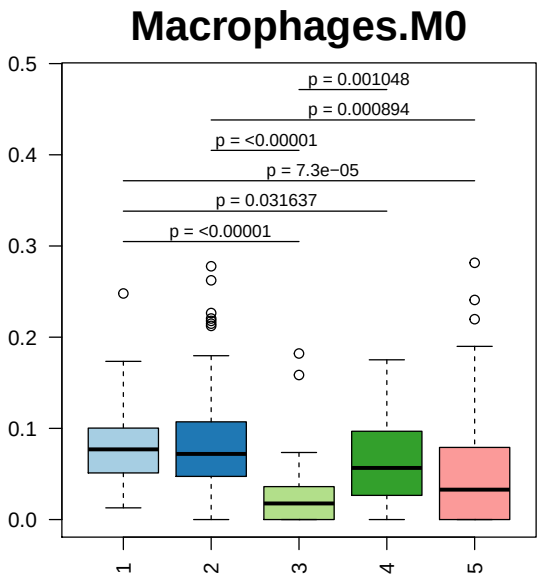

T.cells.follicular.helper
p = 0.002457
p = <0.00001
p = <0.00001
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5

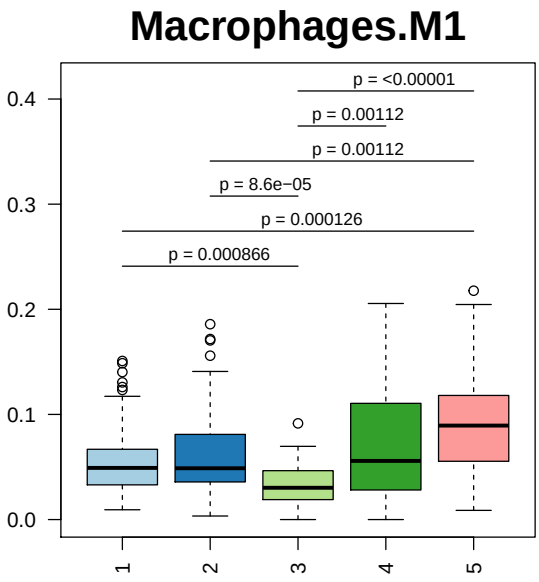

T.cells.regulatory..Tregs.
p = 0.0129
p = 0.039502
p = 0.0129
p = 0.004786
p = 0.02129
p = 0.007249
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

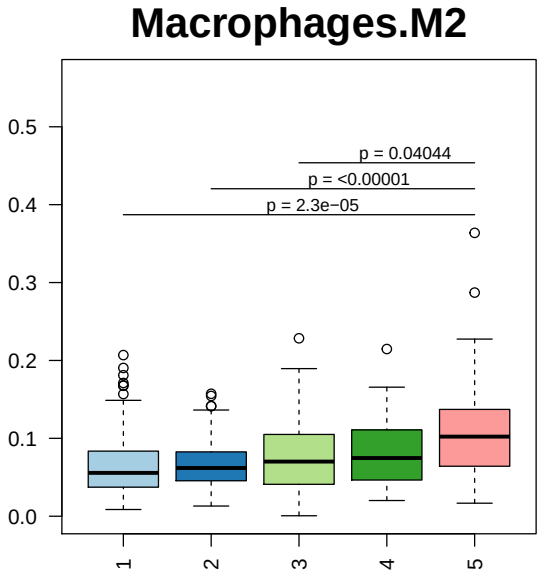

T.cells.gamma.delta
p = 0.016875
p = 0.000604
p = 0.000375
p = 0.003189
p = 0.001792
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

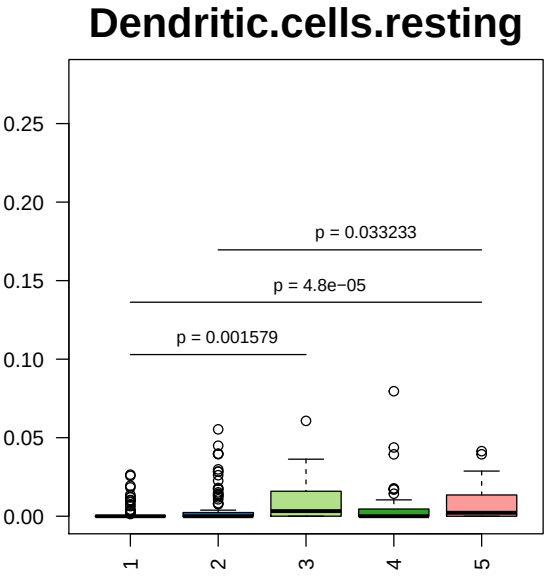

NK.cells.resting
p = 0.000256
p = <0.00001
p = 0.001432
p = <0.00001
p = 0.002129
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

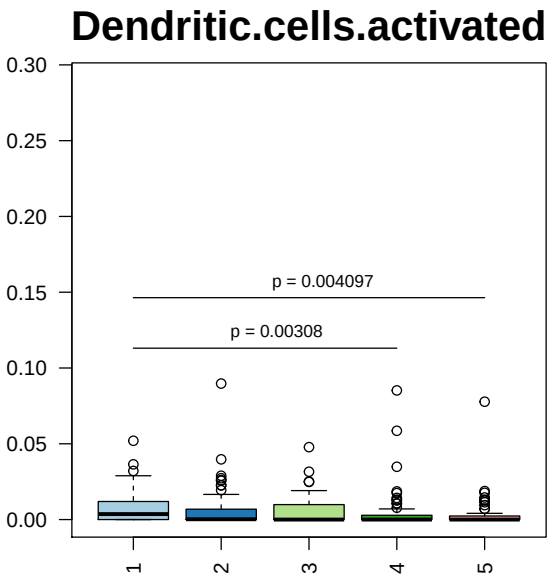

NK.cells.activated
p = 4.3e−05
p = 0.030025
p = 0.001748
p = 0.001346
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

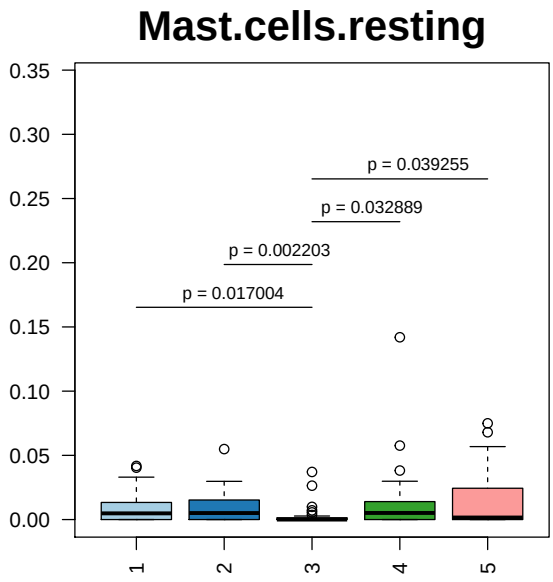

Macrophages.M0
p = 0.001048
p = 0.000894
p = <0.00001
p = 7.3e−05
p = 0.031637
p = <0.00001
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

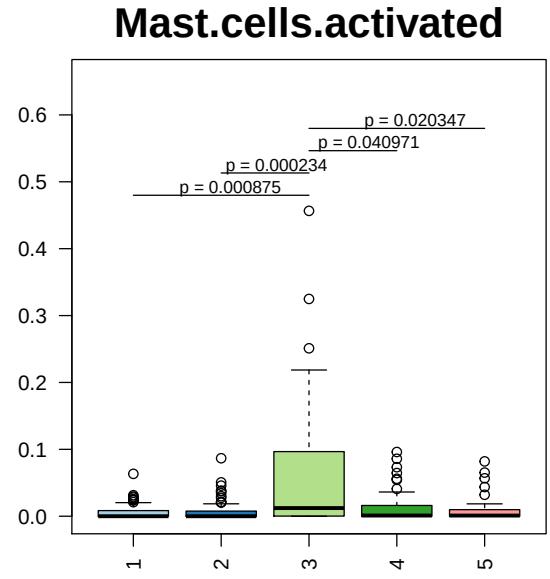

Macrophages.M1
p = <0.00001
p = 0.00112
p = 0.00112
p = 8.6e−05
p = 0.000126
p = 0.000866
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

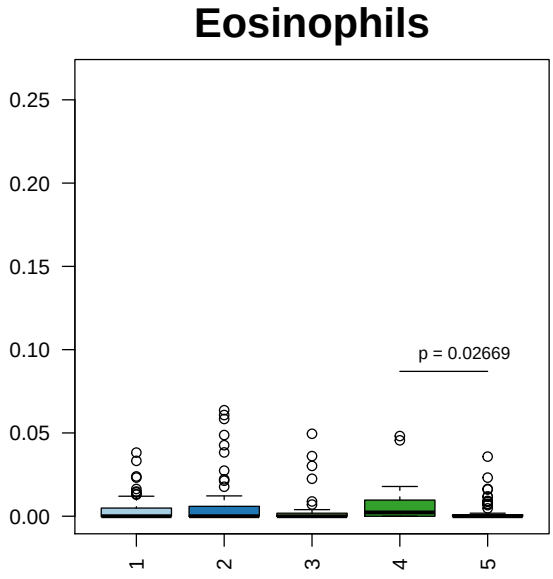

Macrophages.M2
p = 0.04044
p = <0.00001
p = 2.3e−05
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5


Dendritic.cells.resting
p = 0.033233
p = 4.8e−05
p = 0.001579
0.25
0.20
0.15
0.10
0.05
0.00
1
2
3
4
5









R tmod analysis

```
final<- read.delim("./Rmd.files/aitl_nos_alcl_clsutering.txt",sep="\t",header = T,stringsAsFactors = F)
final2<- final[,c("Row.names","hist","cluster")]
mat<- read.delim("./Rmd.files/ensembl_annotated_matrix.txt", sep="\t", stringsAsFactors = F)

design <- model.matrix(~ 0+factor(final2$cluster)) ##### create matrix
colnames(design)<-paste0("Cluster_",c(1:5))

contrast.matrix <- makeContrasts(Cluster_2-Cluster_1,Cluster_3-Cluster_1,Cluster_4-Cluster_1,
                                 Cluster_5-Cluster_1,Cluster_3-Cluster_2,
```

```
                                Cluster_4-Cluster_2,Cluster_5-Cluster_2,Cluster_4-Cluster_3,
                                Cluster_5-Cluster_3,Cluster_4-Cluster_5,
                                Cluster_2-(Cluster_1 + Cluster_3 + Cluster_4 + Cluster_5)/4,
                                Cluster_3-(Cluster_1 + Cluster_2 + Cluster_4 + Cluster_5)/4,
                                Cluster_4-(Cluster_1 + Cluster_2 + Cluster_3 + Cluster_5)/4,
                                Cluster_1-(Cluster_2 + Cluster_3 + Cluster_4 + Cluster_5)/4,
                                Cluster_5-(Cluster_2 + Cluster_3 + Cluster_4 + Cluster_1)/4,
                                levels=design)

fit1 <- lmFit(mat, design)
fit2 <- contrasts.fit(fit1, contrast.matrix)
fit <- eBayes(fit2)

geneExpr = adj.data
geneExpr2<- geneExpr[,colnames(geneExpr) %in% final2$Row.names ]
geneExpr2<- geneExpr2[,final2$Row.names]
ensembl = useMart( "ensembl", dataset = "hsapiens_gene_ensembl" )
hgnc_swissprot <- getBM(attributes=c('entrezgene','hgnc_symbol','hgnc_id'),filters = 'entrezgene',
values = gsub("_at","",rownames(geneExpr2)),mart = ensembl)
geneExpr3<- as.data.frame.matrix(geneExpr2[which(rownames(geneExpr2) %in% paste0(hgnc_swissprot$entrezg


### pairwise for pathway using tmod (https://cran.r-project.org/web/packages/tmod/vignettes/tmod.pdf)

fit1 <- lmFit(mat, design)
fit2 <- contrasts.fit(fit1, contrast.matrix)
fit <- eBayes(fit2)
res.l <- tmodLimmaTest(fit, rownames(mat))
length(res.l)
```

```
## [1] 15
```

```
names(res.l)
```

```
##  [1] "Cluster_2 - Cluster_1"
##  [2] "Cluster_3 - Cluster_1"
##  [3] "Cluster_4 - Cluster_1"
##  [4] "Cluster_5 - Cluster_1"
##  [5] "Cluster_3 - Cluster_2"
##  [6] "Cluster_4 - Cluster_2"
##  [7] "Cluster_5 - Cluster_2"
##  [8] "Cluster_4 - Cluster_3"
##  [9] "Cluster_5 - Cluster_3"
## [10] "Cluster_4 - Cluster_5"
## [11] "Cluster_2 - (Cluster_1 + Cluster_3 + Cluster_4 + Cluster_5)/4"
## [12] "Cluster_3 - (Cluster_1 + Cluster_2 + Cluster_4 + Cluster_5)/4"
## [13] "Cluster_4 - (Cluster_1 + Cluster_2 + Cluster_3 + Cluster_5)/4"
## [14] "Cluster_1 - (Cluster_2 + Cluster_3 + Cluster_4 + Cluster_5)/4"
## [15] "Cluster_5 - (Cluster_2 + Cluster_3 + Cluster_4 + Cluster_1)/4"
```

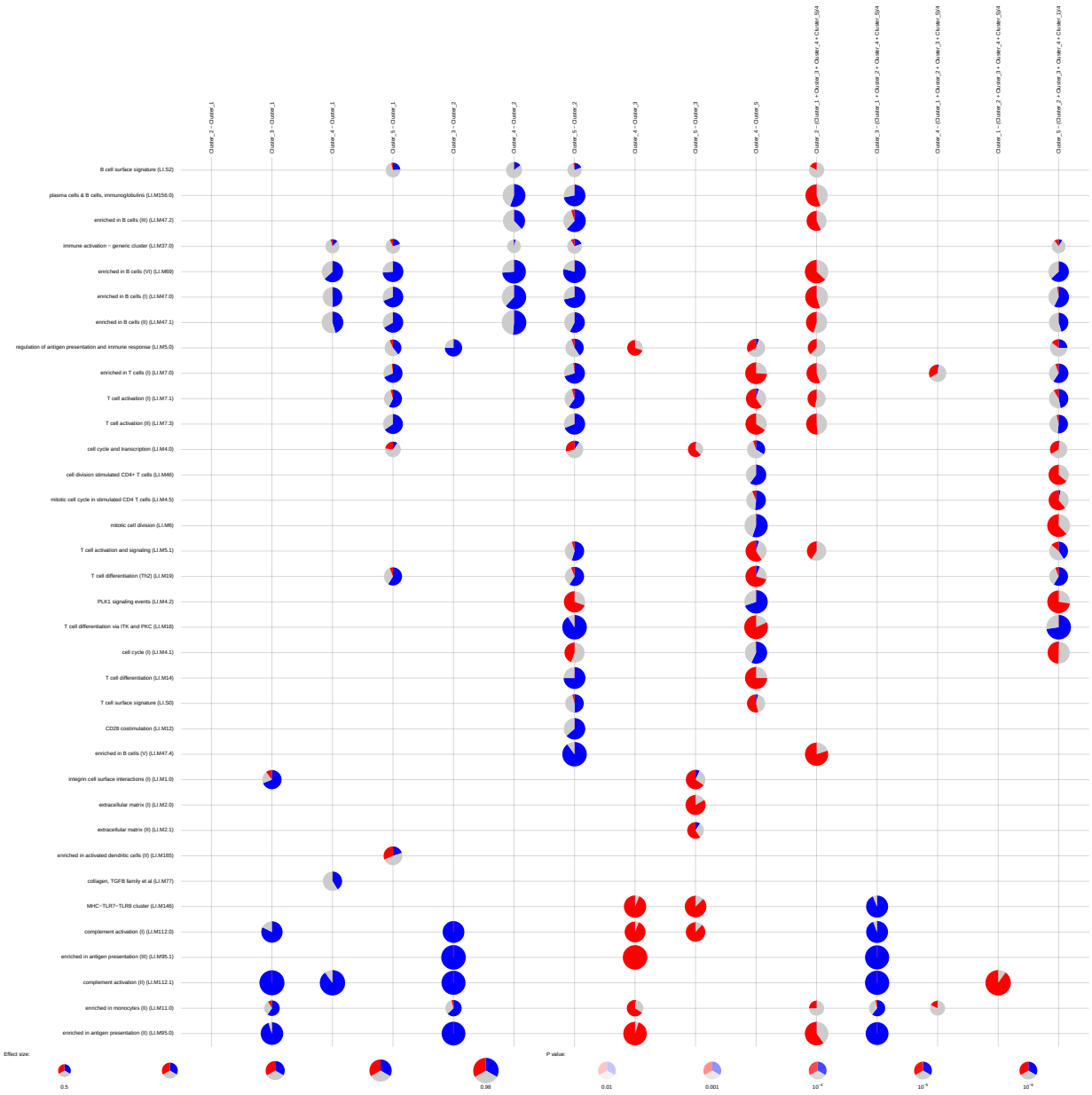
```
pie <- tmodLimmaDecideTests(fit, genes=rownames(mat))
head(tmodSummary(res.l), 5)
```

```
##                ID                                    Title
## LI.M0       LI.M0                       targets of FOSL1/2
## LI.M1.0   LI.M1.0  integrin cell surface interactions (I)
## LI.M1.1   LI.M1.1 integrin cell surface interactions (II)
## LI.M10.0 LI.M10.0                        E2F1 targets (Q3)
## LI.M10.1 LI.M10.1                        E2F1 targets (Q4)
##          AUC.Cluster_2 - Cluster_1 q.Cluster_2 - Cluster_1
## LI.M0                           NA                       NA
## LI.M1.0                  0.7086866             4.263926e-06
## LI.M1.1                  0.7519449             1.923230e-03
## LI.M10.0                        NA                       NA
## LI.M10.1                        NA                       NA
##          AUC.Cluster_3 - Cluster_1 q.Cluster_3 - Cluster_1
## LI.M0                    0.7904401             7.937733e-05
## LI.M1.0                  0.7463549             6.014734e-08
## LI.M1.1                  0.8062779             1.129604e-04
## LI.M10.0                        NA                       NA
## LI.M10.1                        NA                       NA
##          AUC.Cluster_4 - Cluster_1 q.Cluster_4 - Cluster_1
## LI.M0                    0.5243252             1.817202e-02
## LI.M1.0                  0.7560506             6.024021e-06
## LI.M1.1                  0.8509507             2.501998e-04
## LI.M10.0                        NA                       NA
## LI.M10.1                        NA                       NA
##          AUC.Cluster_5 - Cluster_1 q.Cluster_5 - Cluster_1
## LI.M0                           NA                       NA
## LI.M1.0                         NA                       NA
## LI.M1.1                         NA                       NA
## LI.M10.0                        NA                       NA
## LI.M10.1                        NA                       NA
##          AUC.Cluster_3 - Cluster_2 q.Cluster_3 - Cluster_2
## LI.M0                    0.7349997             0.0003016984
## LI.M1.0                  0.6804175             0.0001357877
## LI.M1.1                  0.7368450             0.0064224854
## LI.M10.0                        NA                       NA
## LI.M10.1                        NA                       NA
##          AUC.Cluster_4 - Cluster_2 q.Cluster_4 - Cluster_2
## LI.M0                    0.4458492              0.04574528
## LI.M1.0                         NA                       NA
## LI.M1.1                         NA                       NA
## LI.M10.0                        NA                       NA
## LI.M10.1                        NA                       NA
##          AUC.Cluster_5 - Cluster_2 q.Cluster_5 - Cluster_2
## LI.M0                           NA                       NA
## LI.M1.0                         NA                       NA
## LI.M1.1                         NA                       NA
## LI.M10.0                 0.6268481              0.005215744
## LI.M10.1                        NA                       NA
##          AUC.Cluster_4 - Cluster_3 q.Cluster_4 - Cluster_3
## LI.M0                    0.7195404             0.002995659
## LI.M1.0                  0.6079516             0.003575630
## LI.M1.1                         NA                       NA
## LI.M10.0                        NA                       NA
## LI.M10.1                        NA                       NA
```

```
##          AUC.Cluster_5 - Cluster_3 q.Cluster_5 - Cluster_3
## LI.M0                    0.8007786            4.719991e-04
## LI.M1.0                  0.7580298            6.629245e-10
## LI.M1.1                  0.7753771            2.452147e-05
## LI.M10.0                        NA                     NA
## LI.M10.1                        NA                     NA
##          AUC.Cluster_4 - Cluster_5 q.Cluster_4 - Cluster_5
## LI.M0                           NA                     NA
## LI.M1.0                  0.5793860            0.0414285291
## LI.M1.1                  0.7011807            0.0483845346
## LI.M10.0                 0.6935247            0.0001611948
## LI.M10.1                 0.6809144            0.0186586321
##          AUC.Cluster_2 - (Cluster_1 + Cluster_3 + Cluster_4 + Cluster_5)/4
## LI.M0                                                              0.5388099
## LI.M1.0                                                                   NA
## LI.M1.1                                                                   NA
## LI.M10.0                                                                  NA
## LI.M10.1                                                                  NA
##          q.Cluster_2 - (Cluster_1 + Cluster_3 + Cluster_4 + Cluster_5)/4
## LI.M0                                                             0.0313183
## LI.M1.0                                                                  NA
## LI.M1.1                                                                  NA
## LI.M10.0                                                                 NA
## LI.M10.1                                                                 NA
##          AUC.Cluster_3 - (Cluster_1 + Cluster_2 + Cluster_4 + Cluster_5)/4
## LI.M0                                                              0.7610722
## LI.M1.0                                                            0.7181696
## LI.M1.1                                                            0.7802837
## LI.M10.0                                                                  NA
## LI.M10.1                                                                  NA
##          q.Cluster_3 - (Cluster_1 + Cluster_2 + Cluster_4 + Cluster_5)/4
## LI.M0                                                         1.684751e-04
## LI.M1.0                                                       7.332335e-07
## LI.M1.1                                                       5.275817e-04
## LI.M10.0                                                                NA
## LI.M10.1                                                                NA
##          AUC.Cluster_4 - (Cluster_1 + Cluster_2 + Cluster_3 + Cluster_5)/4
## LI.M0                                                                     NA
## LI.M1.0                                                                   NA
## LI.M1.1                                                                   NA
## LI.M10.0                                                                  NA
## LI.M10.1                                                                  NA
##          q.Cluster_4 - (Cluster_1 + Cluster_2 + Cluster_3 + Cluster_5)/4
## LI.M0                                                                   NA
## LI.M1.0                                                                 NA
## LI.M1.1                                                                 NA
## LI.M10.0                                                                NA
## LI.M10.1                                                                NA
##          AUC.Cluster_1 - (Cluster_2 + Cluster_3 + Cluster_4 + Cluster_5)/4
## LI.M0                                                              0.7041510
## LI.M1.0                                                            0.7594978
## LI.M1.1                                                            0.7807329
## LI.M10.0                                                                  NA
## LI.M10.1                                                                  NA
```

```
##          q.Cluster_1 - (Cluster_2 + Cluster_3 + Cluster_4 + Cluster_5)/4
## LI.M0                                                        6.515199e-04
## LI.M1.0                                                      9.608212e-07
## LI.M1.1                                                      2.996175e-04
## LI.M10.0                                                               NA
## LI.M10.1                                                               NA
##          AUC.Cluster_5 - (Cluster_2 + Cluster_3 + Cluster_4 + Cluster_1)/4
## LI.M0                                                           0.6430191
## LI.M1.0                                                         0.6242903
## LI.M1.1                                                         0.7192819
## LI.M10.0                                                        0.7013382
## LI.M10.1                                                        0.6653520
##          q.Cluster_5 - (Cluster_2 + Cluster_3 + Cluster_4 + Cluster_1)/4
## LI.M0                                                       0.0217931595
## LI.M1.0                                                     0.0002196092
## LI.M1.1                                                     0.0048978471
## LI.M10.0                                                    0.0001601024
## LI.M10.1                                                    0.0369700700
```

```
par(mfrow=c(1,1))
res.l2<- lapply(res.l, function(x) {x[x$adj.P.Val<10e-8,]})
tmodPanelPlot(res.l2, pie=pie, text.cex=0.6) ##### zero = grey, blue down in the first factor and red u
```




```
res.l2<- lapply(res.l, function(x) {x[x$adj.P.Val>10e-8 & x$adj.P.Val<10e-5,]})
tmodPanelPlot(res.l2, pie=pie, text.cex=0.5) ##### zero = grey, blue down in the first factor and red u
```

Supervised Analysis between clusters

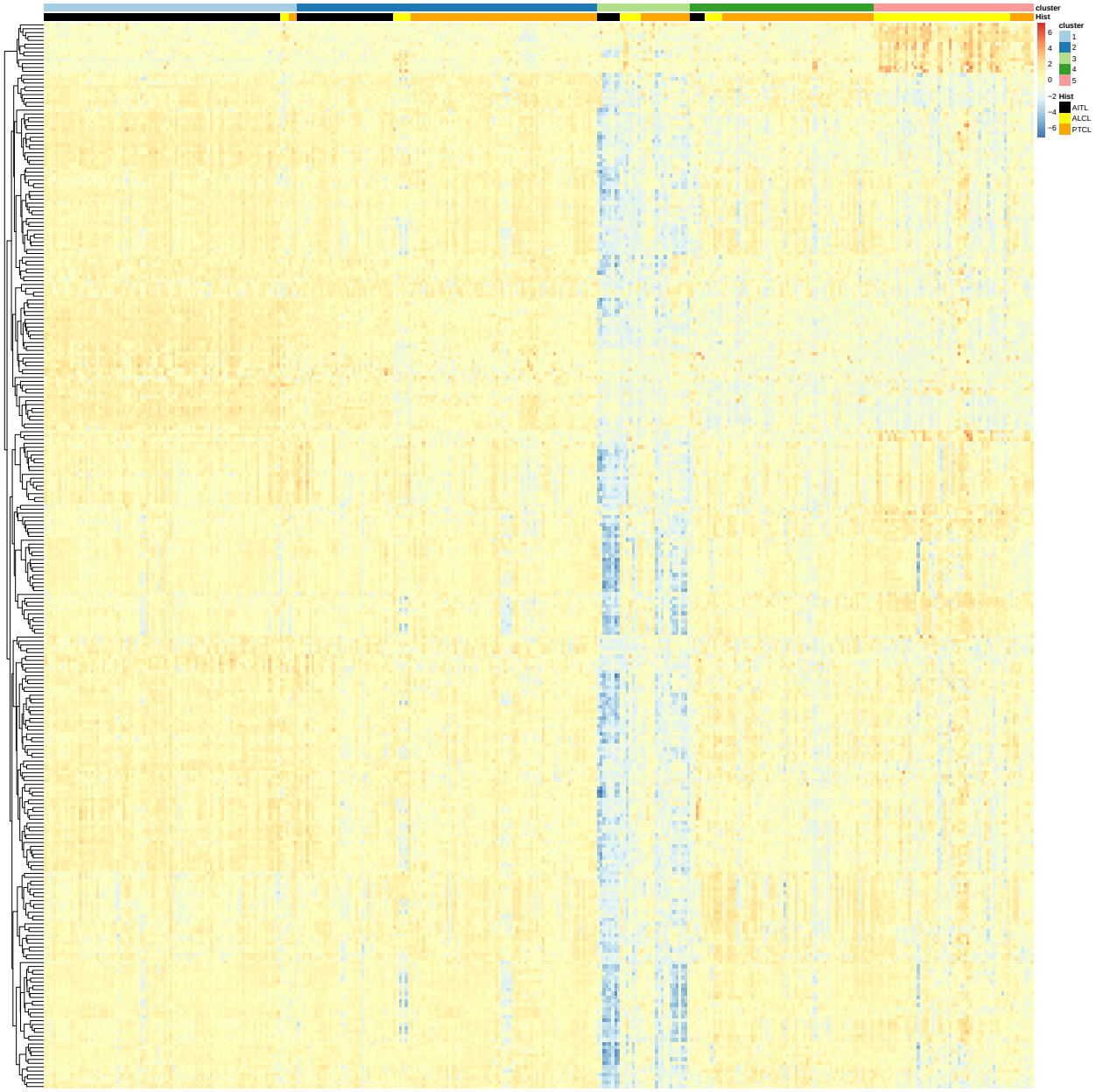
```
levels_design<- c("Cluster_2-Cluster_1","Cluster_3-Cluster_1","Cluster_4-Cluster_1","Cluster_5-Cluster_1
                  "Cluster_3-Cluster_2","Cluster_4-Cluster_2","Cluster_5-Cluster_2","Cluster_4-Cluster_3
                  "Cluster_5-Cluster_3","Cluster_4-Cluster_5",
                  "Cluster_2-(Cluster_1 + Cluster_3 + Cluster_4 + Cluster_5)/4",
                  "Cluster_3-(Cluster_1 + Cluster_2 + Cluster_4 + Cluster_5)/4",
                  "Cluster_4-(Cluster_1 + Cluster_2 + Cluster_3 + Cluster_5)/4",
                  "Cluster_1-(Cluster_2 + Cluster_3 + Cluster_4 + Cluster_5)/4",
                  "Cluster_5-(Cluster_2 + Cluster_3 + Cluster_4 + Cluster_1)/4")
df_diff_all=NULL
```

```
for(i in (1:length(levels_design)))
{
tt <- topTable(fit, coef=i, number=Inf, genelist=rownames(geneExpr3))
tt$ID<- rownames(tt)
colnames(tt)[1]<-"GENE_SYMBOL"
head(tt, 10)
fg <- tt$GENE_SYMBOL[tt$adj.P.Val < 0.001 & abs( tt$logFC ) > 2]
length(fg)
df_diff<- cbind(fg, rep(levels_design[i], length(fg)))
df_diff_all<-rbind(df_diff_all, df_diff)
#plot(tt$logFC, -log10(tt$adj.P.Val))
}
df_diff_all<- as.data.frame.matrix(df_diff_all)
table(df_diff_all$V2)
```

```
##
## Cluster_1-(Cluster_2 + Cluster_3 + Cluster_4 + Cluster_5)/4
##                                                         18
## Cluster_3-(Cluster_1 + Cluster_2 + Cluster_4 + Cluster_5)/4
##                                                         84
##                                          Cluster_3-Cluster_1
##                                                        203
##                                          Cluster_3-Cluster_2
##                                                        121
##                                          Cluster_4-Cluster_1
##                                                         15
##                                          Cluster_4-Cluster_2
##                                                          2
##                                          Cluster_4-Cluster_3
##                                                         78
##                                          Cluster_4-Cluster_5
##                                                          8
## Cluster_5-(Cluster_2 + Cluster_3 + Cluster_4 + Cluster_1)/4
##                                                          5
##                                          Cluster_5-Cluster_1
##                                                         41
##                                          Cluster_5-Cluster_2
##                                                         19
##                                          Cluster_5-Cluster_3
##                                                         74
```

```
annotation_col<- final2
colnames(annotation_col)<-c("sampleID","Hist","cluster")
A <- function(x) (as.factor(as.character(x))) ##### lapply function for all columns to generate the rela
annotation_col[,1:ncol(annotation_col)] = apply(annotation_col[,1:ncol(annotation_col)], 2, function(x)
annotation_col<- as.data.frame(annotation_col[,-1])
mycol_plus<- c(brewer.pal(11,"Paired"),brewer.pal(6,"Dark2"))
ann_colors = list(Hist=c( "AITL"="black","ALCL"="yellow","PTCL"="orange"),
                  cluster=c("1" = mycol_plus[1],"2" = mycol_plus[2],"3" = mycol_plus[3],"4" = mycol_plus
                  )

edata3<- mat[rownames(mat) %in% unique(df_diff_all$fg),]
pheatmap(as.matrix( as.matrix(edata3)), annotation_col=annotation_col, annotation_colors = ann_colors, l
```

```
        scale = "row", cluster_cols = FALSE, show_colnames= F, show_rownames = FALSE)
```



overlap between differentially expressed genes and the list published by Iqbal et al Blood 2014

```
############ table of genes

df_diff_all_tab=NULL
for(i in (1:length(levels_design)))
{
  tt <- topTable(fit, coef=i, number=Inf, genelist=rownames(geneExpr3))
```

```
  tt$ID<- rownames(tt)
  colnames(tt)[1]<-"GENE_SYMBOL"
  head(tt,10)
  fg <- tt[tt$adj.P.Val < 0.001 & abs( tt$logFC ) > 2,]
  if(nrow(fg)>0){
    fg$design<- levels_design[i]

  df_diff_all_tab<-rbind.data.frame(df_diff_all_tab, fg)
  #plot(tt$logFC"," -log10(tt$adj.P.Val))
  }
  }

nrow(df_diff_all_tab) #### number of genes differentially expressed between C-1, C-2, C-3, C-4, C-5
```

```
## [1] 668
```

```
##### list gene from Iqbal et al. blood 2014
iqbal<- unique(c("EFNB2","ROBO1","S1PR3","ANK2","LPAR1","SNAP91","SOX8","LPAR1","RAMP3","S1PR3","ROBO1"

intersect(iqbal, unique(df_diff_all_tab$GENE_SYMBOL))
```

```
##  [1] "ROBO1"    "LPAR1"    "SOX8"     "TUBB2B"   "TNFRSF8"  "TMOD1"
##  [7] "BATF3"    "ATP6V0D1" "CHI3L1"   "CREG1"    "CTSB"     "CTSC"
## [13] "FTL"      "HCK"
```