PTCL Gene Expression based classification

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

true
true

2018/07/18

Contents

Libraries **2**

Ensembl Library . . . 3

Gene Expression Data . . . 3

Patients Data . . . 4

Database metrics **5**

PCA . . . 5

PCA . . . 6

Heatmap of hierarchical clustering on most variable genes . . . 7

Check relative log expression after batch correction . . . 8

Build Gene Expression Matrix . . . 9

Model fitting procedure **10**

Check Differentially Expressed Genes . . . 11

Calculate significant effects per covariate . . . 12

Extract the list of differentially expressed genes by mutations . . . 12

Example of single gene extraction . . . 12

Plot significant effects per covariate (q<0.01) . . . 13

Print the list of differently expressed genes using the Ensembl annotation . . . 15

Generate a heatmap with AITL, PTCL-NOS with the extracted differentially expressed genes. 16

LOOCV on AILT, PTCLnos based on 16-gene model **20**

Extracting the most significant clusters based on 19-gene signature . . . 21

Plot heatmap AITL, PTCL-NOS, ALCL-neg and the 19-gene model . . . 32

Cibersort to characterize tumour microenviroment composition of each cluster . . . 36

Boxplot comparing the contribution of each cibersort signature between all extracted clusters 38

Built with R version:
3.5.0

Built with R version:
3.5.0

Libraries

Load necessary libraries

```
library(affy)
library(ComplexHeatmap)
library(plot3D)
library(gplots)
library(circlize)
library(AnnotationDbi)
library(limma)
library(lattice)
library(org.Hs.eg.db)
library(MASS)
library(RColorBrewer)
library(AnnotationDbi)
library(rglwidget)
###library(hgu133plus2hsentrezgcdf)
library(VennDiagram)
library(org.Hs.eg.db)
library(GenomicRanges)
library(GenomicFeatures)
library(rtracklayer)
library(biomaRt)
library(glmnet)
library(survival)
library(Hmisc)
library(ConsensusClusterPlus)
library(pheatmap)
library(ggplot2)
library(heatmap.plus)
library(rgl)
#library(caret) ## unselect to generate LOOCV accuracy table
#library(e1071) ## unselect to generate LOOCV accuracy table
library(tmod)

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" )
```

Gene Expression Data

Upload or generate GEP normalized matrix

```
### choice 1: import processed matrix
# data.dir="./Rmd.files/"
data.dir = '/Users/emagene/Dropbox/codes/github/PTCL/'
setwd(data.dir)
load (file.path(data.dir,"/Rmd.files/541_PTCL_batch_adjusted_geo.id.Rdata"))

geneExpr = adj.data
# 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)
batch$cancer = "cancer"

# ### OPTIONAL: CHECK BATCH ON FINAL.MOLEC
#
# #mod = model.matrix(~as.factor(center), data=batch)
# mod = model.matrix(~as.factor(final.molec), data=design)
```

```
# mod0 = model.matrix(~1, data= batch)
# library(sva)
# n.sv = num.sv(adj.data,mod,method="leek")
# svobj = sva(adj.data,mod,mod0,n.sv=n.sv)
#
# pValues = f.pvalue(adj.data,mod,mod0)
# qValues = p.adjust(pValues,method="BH")
# modSv = cbind(mod,svobj$sv)
# mod0Sv = cbind(mod0,svobj$sv)
# pValuesSv = f.pvalue(adj.data,modSv,mod0Sv)
# qValuesSv = p.adjust(pValuesSv,method="BH")

### 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=FALS
# customize colors for categories
levels(as.factor(pts.info.data$final.molec))
```

```
##  [1] "AITL"     "ALCL.neg" "ALCL.pos" "ATLL"     "NKT"      "PTCL.nos"
##  [7] "T.CD30"   "T.CD4"    "T.CD8"    "T.DR"     "T.reg"    "TCR-HL"
```

```
# "AITL"     "ALCL.neg" "ALCL.pos" "ATLL"     "NKT"      "PTCL.nos" "T.CD30"   "T.CD4"    "T.CD8"    "T
colorz = c("black", "yellow","dodgerblue2","brown2","darkorchid1", "orange", "grey42", "grey52","grey62
temp = split (  pts.info.data$sample.nameNEW, pts.info.data$final.molec )
colorx = colnames(geneExpr)
length(colorz)
```

```
## [1] 12
```

```
length(temp)
```

```
## [1] 12
```

```
for (i in 1:length(colorz)) colorx [ which(colorx %in% unlist(temp[i])) ] = colorz[i]
library(gplots)
colorx = col2hex(colorx)

### build design matrix and transform to numerical
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$final.molec=="PTCL.nos"] <- 1
design$final.molec[design$final.molec=="ALCL.neg"] <- 2
design$final.molec[design$final.molec=="ALCL.pos"] <- 3
design$final.molec[design$final.molec=="ATLL"] <- 4
design$final.molec[design$final.molec=="NKT"] <- 5
design$final.molec[477:541] <- 6
design$gender[design$gender=="M"] <- 1
design$gender[design$gender=="F"] <- 0
design$age = NULL
all(pts.info.data$sample.nameNEW == batch$nameNEW)
```

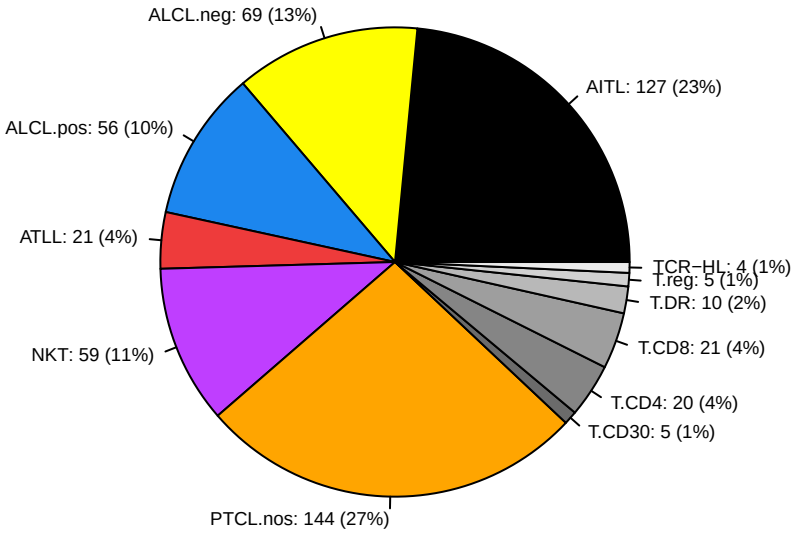
```
## [1] TRUE
```

Database metrics

PCA

```
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
#pdf("Figure_1a_pie_plot.pdf", width = 5, height = 5)
```

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




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

PCA

```
# apply variational filter

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

```
## [1] 1840  541
```

```
# 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 most variable genes

```
mat = as.matrix(data541exprs.vf)
base_mean = rowMeans(mat)
mat_scaled = t(apply(mat, 1, scale))
types = pts.info.data$final.molec
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
column_order(ht)
```

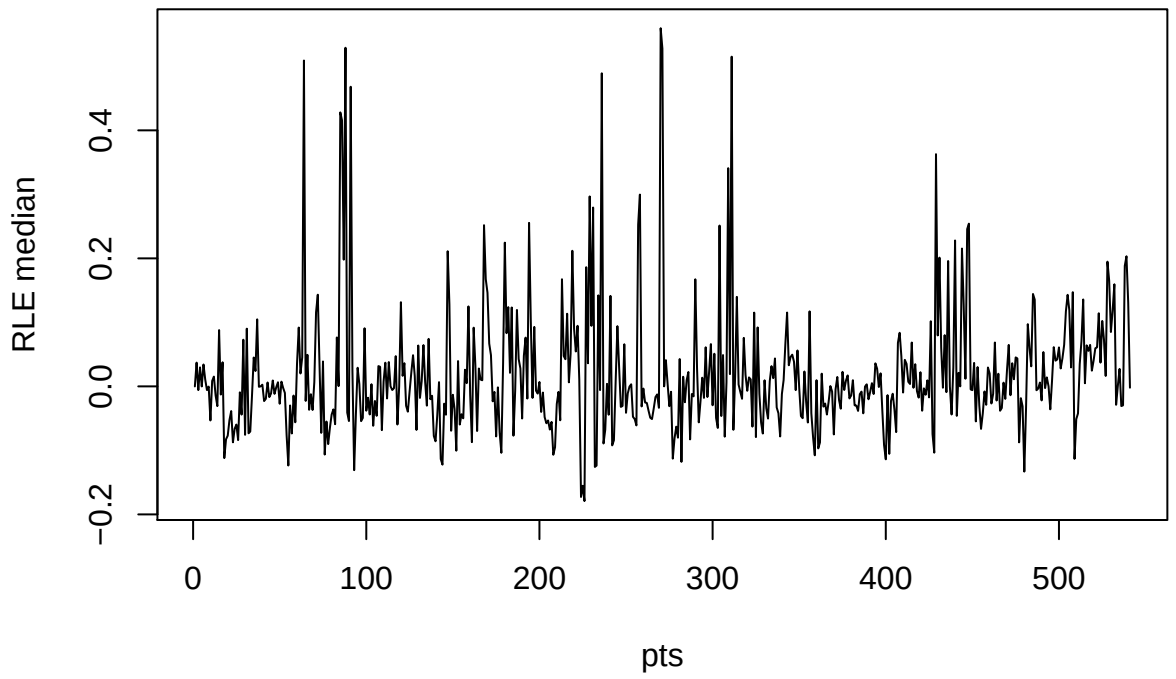
```
##   [1]  91 236  85  86 429  88  64 311 270 271 229 231 180 257  87 227 258
##  [18] 304 309 436 431 234 148 147  31  29 450 281 500 521 499 273 272 269
##  [35] 522 520 501 519 498 518 497 505 506 504 486 485 482 483 484 502 503
##  [52] 524 527 529 528 530 532 531 525 526 523 434  61 182 176  63  83  66
##  [69] 445 230 515 517 513 296 298 516 494 492 496 493 314 297 294 295 293
##  [86] 289 292 512 507 511 510 509 291 290 514 508 448 447 444 440 396 397
```

```
## [103] 395 394 391  50  51 187 192 433 349 363 385 428 435 427 389 366 491
## [120] 401 334 459 392 402 388 457 463 446 380 351 474 372 376 489 487 537
## [137] 488 534 536 495 490 535 533 481 479 478 480 477 264 261 268 259 262
## [154] 253 267 265 266 263  65 254 256 255 318 308 320 319 315 324 326 313
## [171] 280 299 285 316 286 278 344 330 430 331 275 305 307 136 426 184 162
## [188] 167 422 421 469 341 245 160 179 153 443 449 181 149 368 186 164 141
## [205] 194 218 161  98 300 312  84 442 323 301 174 306 248 329 325 178 139
## [222] 145 142 144 185 157 249 172 177 158 183 150 170 171 152 131 134 137
## [239] 216 214 213 241 202 205 203 191 130 133 195 215 244 211 246 238 197
## [256] 224 226 225 169 538 168 539 540 219 223 220 221 222 206 204 232 242
## [273] 200 243 143 250 154 207 198 235 240 247 252 163 155 383 188 239 417
## [290] 276 406  93 303 339 109  53 124  82  26  32  23  22  20 189  18 398
## [307]  41  44  10 193 217 129   2 140 208 251 199 201 209 210 212 233 237
## [324] 288 284 420 439 274 352 287 441 328 283 282 321 317 310 279 322 332
## [341] 302 348 419 399 387 166 159 337 467 151 277 355 359 410 353 470 475
## [358] 466 175 370  42 411 374 393 404 327 456  94 432 541 156 173 146 165
## [375] 135 128 106 458 361 364 354  90 454  27 403 138 196 462 453 338 415
## [392] 407 379 371 464 461 110  79 121  45 405 425  96 102 424 423 360 452
## [409] 451 365 358  89   1  28  19 347 101 260 455  59 418 367  48  69 190
## [426] 333 132  17 378 350 468 346   9 414 409 413 412 408 116  70 122 123
## [443] 340 381 460 382  77  38 108  37 228  49  68  67 416  62  92  71   4
## [460] 103 126  36 120 127  56  74  13  12  72 104 105 100  21  30  25  33
## [477]  24 465 345 117 343 342 471 336 476 356  15  60 390 357 362  99  97
## [494] 437  39  78 438  80 375  40 113 112  52  58 400  57  55  54  76  11
## [511] 114  73  75 472   3   6 377 119  47  35 125   5  95  81  34  46  43
## [528]  16  14 386   8 473 335 111 384   7 118 373 369 107 115
```

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 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)
```

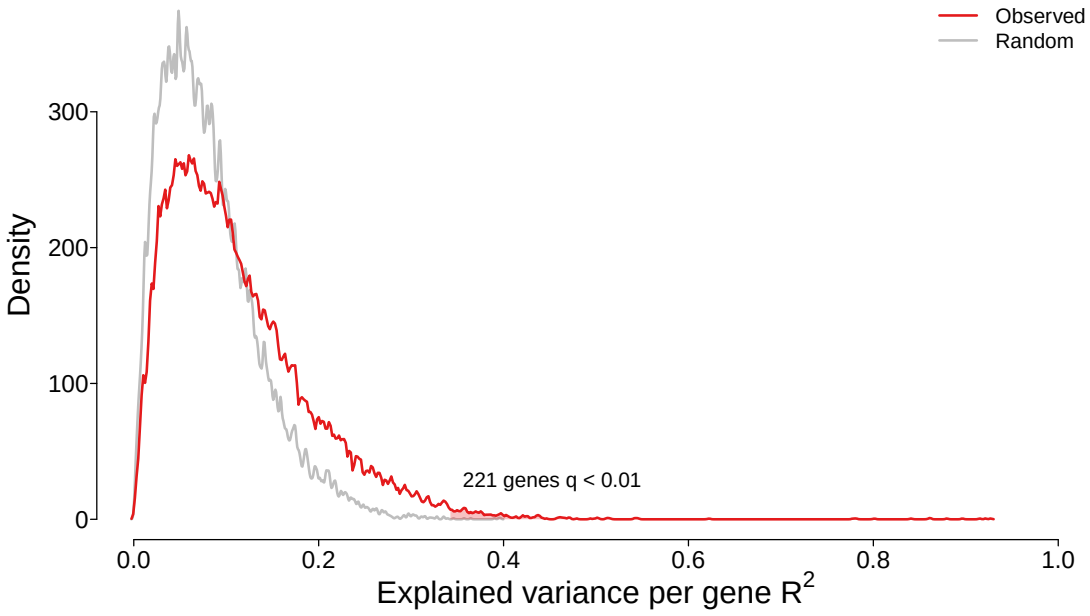
Model fitting procedure

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

#pdf("Figure_2a_MAY.pdf", width = 10, height = 7)
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)
```




```
#dev.off()

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
```

Calculate significant effects per covariate

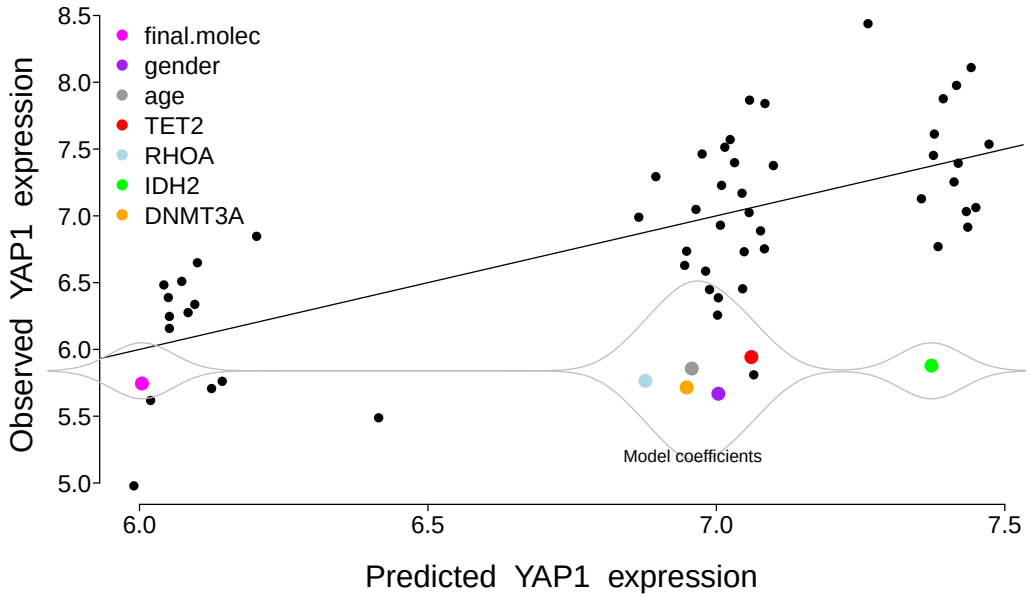
Extract the list of differentially expressed genes by mutations

```
### 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$gene
tab=NULL
for(i in (1:length(kk2$gene)))
{
  gene_single<- gene_code[i]
  y <- glm$coefficients[gene_single,-1]+glm$coefficients[gene_single,1]
  w <- glm$p.value[gene_single,-1] < 0.05
  int<-c(gene_single, as.character(w))
  tab<- rbind(tab, int)
}
rownames(tab)<-seq(1:nrow(tab))
colnames(tab)<- c("gene",colnames(design)[-1])

# Write to disk a file with all significant genes
#write.table(tab, "table_differentially_expressed_gene.txt",sep="\t", quote=F, row.names = F, col.names
```

Example of single gene extraction

```
 # temp_name = unique(getBM( attributes = c("ensembl_transcript_id", "entrezgene", "external_gene_name"
 # mart = ensembl)$external_gene_name)
  #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)
```

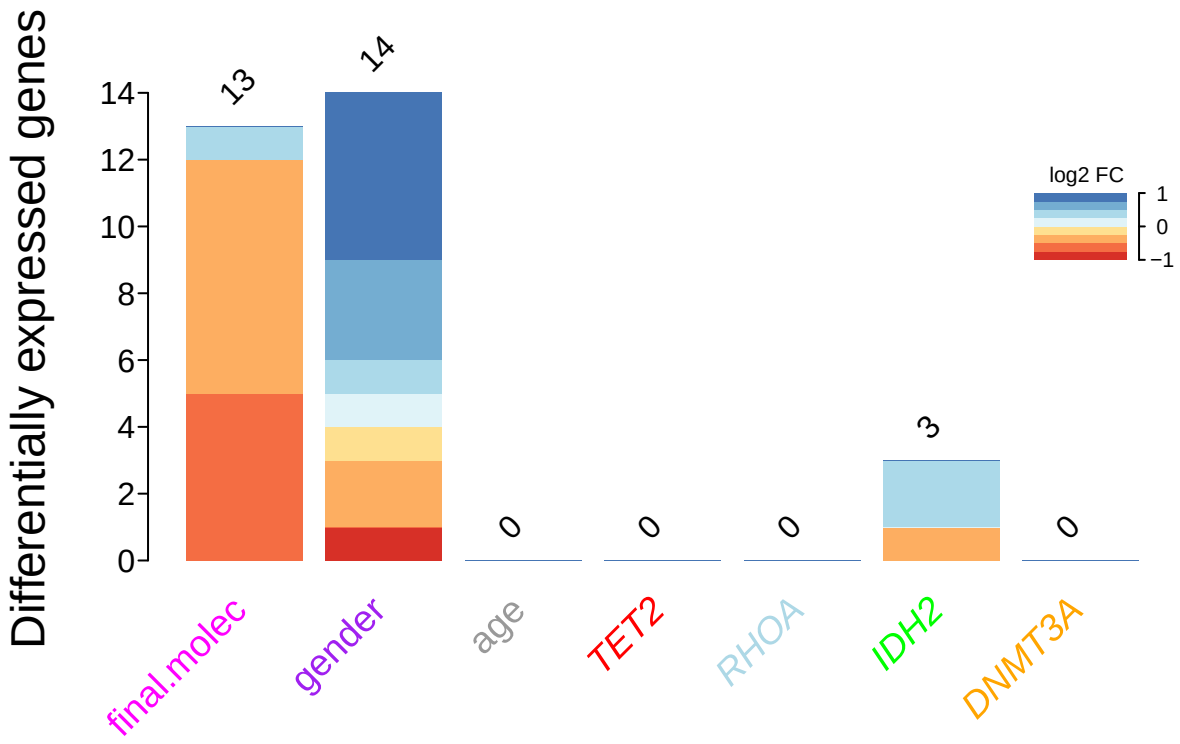



```
#dev.off()
```

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))


#  pdf("Figure_2c.pdf", width = 10, height = 7)
  par(mfrow=c(1,1))
  par(mar=c(8,8,5,5), xpd=F)

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)
```




```
#dev.off()

# par(bty="n", mgp = c(2.5,.33,0), mar=c(3,3.3,3,0)+.1, las=1, tcl=-.25)
# t <- table(rowSums(abs(testResults[,1:6])))
# b <- barplot(t[-1],ylab="Differentially expressed genes", col=rev(brewer.pal(7, "Spectral")[-(4:5)]),
# rotatedLabel(b-0.1, t[-1], t[-1], pos=3, cex=1, srt=45)
# title(xlab="Associated drivers", line=2)
```

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))
```

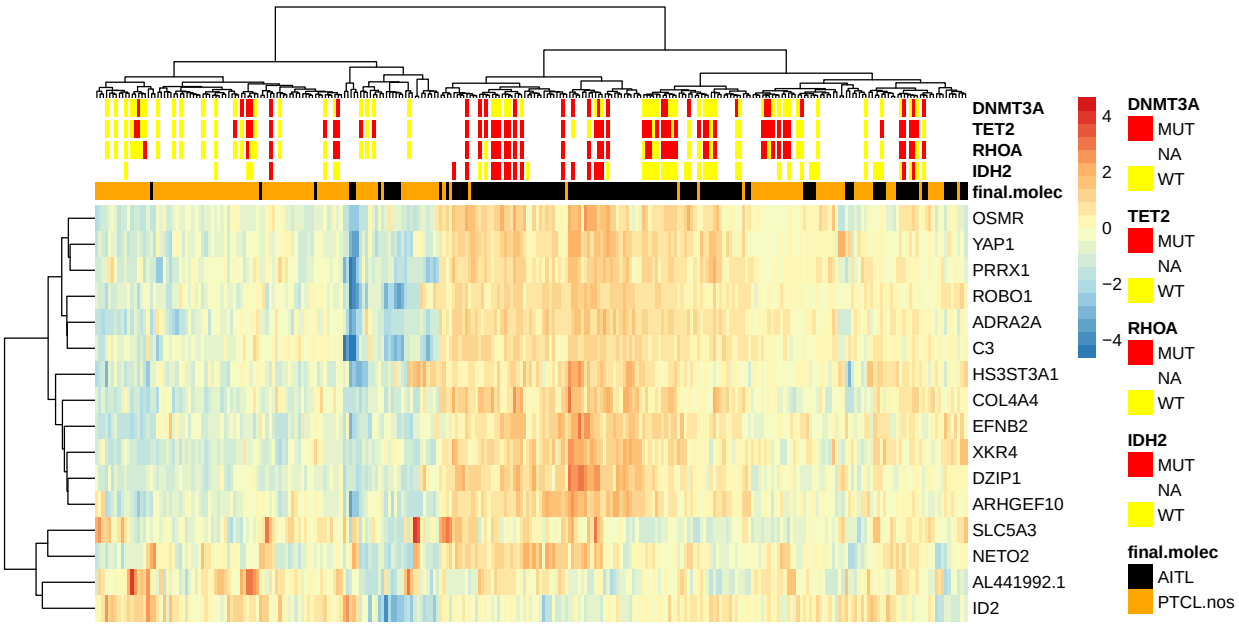
```
##  [1] "ADRA2A"     "AL441992.1" "ARHGEF10"   "C3"         "COL4A4"
##  [6] "DZIP1"      "EFNB2"      "HS3ST3A1"   "ID2"        "NETO2"
## [11] "OSMR"       "PRRX1"      "ROBO1"      "SLC5A3"     "XKR4"
## [16] "YAP1"
```

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

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

setdiff(rownames(mat), paste0(unique(geneannotation1$entrezgene),"_at"))
```

```
## character(0)
```

```
for (ii in 1:nrow(mat)) {
  #if(length (which (paste0(unique(geneannotation1$entrezgene),"_at") == rownames(mat)[ii])) != 0 ) row
  rownames(mat) [ii] = unique(geneannotation1$external_gene_name) [ which (paste0(unique(geneannotation
}

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))
cluster.pts.nr = pheatmap(mat, annotation_col = mylabel.nocol, annotation_colors = list(final.molec = c
                                  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")))(20), scale
```




```
### export pts order
cluster.pts.nr$tree_col$labels [cluster.pts.nr$tree_col$order]
```

```
##   [1] "PTCL.nos..23"  "PTCL.nos..428" "PTCL.nos..448" "PTCL.nos..124"
##   [5] "PTCL.nos..247" "PTCL.nos..463" "PTCL.nos..89"  "PTCL.nos..156"
##   [9] "PTCL.nos..432" "PTCL.nos..216" "PTCL.nos..25"  "PTCL.nos..87"
##  [13] "PTCL.nos..94"  "PTCL.nos..98"  "PTCL.nos..105" "PTCL.nos..93"
##  [17] "PTCL.nos..195" "AITL..413"     "PTCL.nos..531" "PTCL.nos..143"
##  [21] "PTCL.nos..46"  "PTCL.nos..28"  "PTCL.nos..185" "PTCL.nos..416"
##  [25] "PTCL.nos..112" "PTCL.nos..424" "PTCL.nos..134" "PTCL.nos..32"
##  [29] "PTCL.nos..22"  "PTCL.nos..194" "PTCL.nos..30"  "PTCL.nos..211"
##  [33] "PTCL.nos..52"  "PTCL.nos..97"  "PTCL.nos..201" "PTCL.nos..27"
##  [37] "PTCL.nos..68"  "PTCL.nos..139" "PTCL.nos..72"  "PTCL.nos..120"
##  [41] "PTCL.nos..444" "PTCL.nos..24"  "PTCL.nos..15"  "PTCL.nos..109"
##  [45] "PTCL.nos..29"  "PTCL.nos..100" "PTCL.nos..171" "PTCL.nos..104"
##  [49] "PTCL.nos..99"  "PTCL.nos..126" "PTCL.nos..258" "AITL..536"
##  [53] "PTCL.nos..535" "PTCL.nos..20"  "PTCL.nos..102" "PTCL.nos..452"
##  [57] "PTCL.nos..529" "PTCL.nos..90"  "PTCL.nos..230" "PTCL.nos..231"
##  [61] "PTCL.nos..232" "PTCL.nos..236" "PTCL.nos..16"  "PTCL.nos..189"
##  [65] "PTCL.nos..506" "PTCL.nos..519" "PTCL.nos..151" "PTCL.nos..186"
##  [69] "AITL..419"     "PTCL.nos..455" "PTCL.nos..47"  "PTCL.nos..213"
```

```
##  [73] "PTCL.nos..161" "PTCL.nos..61"  "PTCL.nos..209" "PTCL.nos..119"
##  [77] "PTCL.nos..80"  "PTCL.nos..34"  "PTCL.nos..440" "AITL..19"
##  [81] "AITL..18"      "PTCL.nos..504" "PTCL.nos..118" "PTCL.nos..293"
##  [85] "PTCL.nos..92"  "PTCL.nos..251" "PTCL.nos..101" "PTCL.nos..446"
##  [89] "AITL..479"     "PTCL.nos..469" "AITL..411"     "AITL..473"
##  [93] "AITL..472"     "AITL..481"     "AITL..487"     "PTCL.nos..434"
##  [97] "PTCL.nos..180" "PTCL.nos..135" "PTCL.nos..445" "PTCL.nos..408"
## [101] "PTCL.nos..409" "PTCL.nos..460" "PTCL.nos..468" "PTCL.nos..470"
## [105] "PTCL.nos..471" "PTCL.nos..441" "PTCL.nos..451" "AITL..12"
## [109] "PTCL.nos..237" "AITL..165"     "PTCL.nos..178" "AITL..458"
## [113] "AITL..191"     "AITL..163"     "AITL..187"     "AITL..62"
## [117] "PTCL.nos..249" "AITL..250"     "AITL..257"     "AITL..110"
## [121] "AITL..260"     "AITL..60"      "AITL..77"      "AITL..84"
## [125] "AITL..106"     "AITL..74"      "AITL..133"     "AITL..113"
## [129] "AITL..127"     "AITL..44"      "AITL..82"      "AITL..197"
## [133] "AITL..223"     "AITL..17"      "AITL..523"     "AITL..530"
## [137] "AITL..154"     "AITL..45"      "AITL..505"     "AITL..2"
## [141] "AITL..238"     "AITL..11"      "AITL..259"     "AITL..10"
## [145] "AITL..234"     "AITL..435"     "PTCL.nos..239" "AITL..6"
## [149] "AITL..438"     "AITL..518"     "AITL..532"     "AITL..256"
## [153] "AITL..449"     "AITL..129"     "AITL..9"       "AITL..450"
## [157] "AITL..534"     "AITL..66"      "AITL..255"     "AITL..207"
## [161] "AITL..8"       "AITL..1"       "AITL..152"     "AITL..229"
## [165] "AITL..248"     "AITL..235"     "AITL..420"     "AITL..483"
## [169] "AITL..157"     "AITL..179"     "AITL..67"      "AITL..70"
## [173] "AITL..225"     "AITL..71"      "AITL..58"      "AITL..78"
## [177] "AITL..137"     "AITL..206"     "AITL..459"     "AITL..144"
## [181] "AITL..222"     "PTCL.nos..294" "AITL..198"     "AITL..453"
## [185] "AITL..210"     "AITL..26"      "AITL..114"     "PTCL.nos..226"
## [189] "AITL..51"      "AITL..224"     "AITL..69"      "AITL..123"
## [193] "AITL..221"     "AITL..150"     "AITL..43"      "AITL..153"
## [197] "AITL..520"     "AITL..159"     "AITL..79"      "AITL..204"
## [201] "AITL..214"     "PTCL.nos..246" "AITL..167"     "AITL..190"
## [205] "PTCL.nos..289" "PTCL.nos..287" "PTCL.nos..288" "PTCL.nos..128"
## [209] "PTCL.nos..136" "PTCL.nos..91"  "PTCL.nos..86"  "PTCL.nos..196"
## [213] "PTCL.nos..212" "PTCL.nos..13"  "PTCL.nos..96"  "PTCL.nos..95"
## [217] "PTCL.nos..208" "PTCL.nos..290" "PTCL.nos..115" "PTCL.nos..219"
## [221] "AITL..484"     "AITL..7"       "AITL..457"     "AITL..454"
## [225] "PTCL.nos..200" "PTCL.nos..215" "PTCL.nos..252" "PTCL.nos..166"
## [229] "PTCL.nos..218" "PTCL.nos..291" "PTCL.nos..292" "PTCL.nos..467"
## [233] "PTCL.nos..482" "AITL..502"     "AITL..515"     "AITL..406"
## [237] "PTCL.nos..162" "PTCL.nos..465" "PTCL.nos..203" "PTCL.nos..63"
## [241] "PTCL.nos..240" "PTCL.nos..33"  "AITL..510"     "AITL..513"
## [245] "AITL..147"     "AITL..426"     "PTCL.nos..527" "PTCL.nos..414"
## [249] "PTCL.nos..415" "AITL..14"      "AITL..456"     "AITL..205"
## [253] "AITL..55"      "AITL..64"      "AITL..199"     "AITL..65"
## [257] "PTCL.nos..3"   "AITL..121"     "AITL..517"     "PTCL.nos..174"
## [261] "PTCL.nos..524" "PTCL.nos..243" "PTCL.nos..242" "PTCL.nos..244"
## [265] "AITL..392"     "AITL..466"     "AITL..4"       "AITL..5"
## [269] "PTCL.nos..241" "AITL..417"     "AITL..461"
```

```
cluster.pts.nr$tree_col$labels
```

```
##   [1] "AITL..1"       "AITL..10"      "AITL..106"     "AITL..11"
```

```
##   [5] "AITL..110"     "AITL..113"     "AITL..114"     "AITL..12"
##   [9] "AITL..121"     "AITL..123"     "AITL..127"     "AITL..129"
##  [13] "AITL..133"     "AITL..137"     "AITL..14"      "AITL..144"
##  [17] "AITL..147"     "AITL..150"     "AITL..152"     "AITL..153"
##  [21] "AITL..154"     "AITL..157"     "AITL..159"     "AITL..163"
##  [25] "AITL..165"     "AITL..167"     "AITL..17"      "AITL..179"
##  [29] "AITL..18"      "AITL..187"     "AITL..19"      "AITL..190"
##  [33] "AITL..191"     "AITL..197"     "AITL..198"     "AITL..199"
##  [37] "AITL..2"       "AITL..204"     "AITL..205"     "AITL..206"
##  [41] "AITL..207"     "AITL..210"     "AITL..214"     "AITL..221"
##  [45] "AITL..222"     "AITL..223"     "AITL..224"     "AITL..225"
##  [49] "AITL..229"     "AITL..234"     "AITL..235"     "AITL..238"
##  [53] "AITL..248"     "AITL..250"     "AITL..255"     "AITL..256"
##  [57] "AITL..257"     "AITL..259"     "AITL..26"      "AITL..260"
##  [61] "AITL..392"     "AITL..4"       "AITL..406"     "AITL..411"
##  [65] "AITL..413"     "AITL..417"     "AITL..419"     "AITL..420"
##  [69] "AITL..426"     "AITL..43"      "AITL..435"     "AITL..438"
##  [73] "AITL..44"      "AITL..449"     "AITL..45"      "AITL..450"
##  [77] "AITL..453"     "AITL..454"     "AITL..456"     "AITL..457"
##  [81] "AITL..458"     "AITL..459"     "AITL..461"     "AITL..466"
##  [85] "AITL..472"     "AITL..473"     "AITL..479"     "AITL..481"
##  [89] "AITL..483"     "AITL..484"     "AITL..487"     "AITL..5"
##  [93] "AITL..502"     "AITL..505"     "AITL..51"      "AITL..510"
##  [97] "AITL..513"     "AITL..515"     "AITL..517"     "AITL..518"
## [101] "AITL..520"     "AITL..523"     "AITL..530"     "AITL..532"
## [105] "AITL..534"     "AITL..536"     "AITL..55"      "AITL..58"
## [109] "AITL..6"       "AITL..60"      "AITL..62"      "AITL..64"
## [113] "AITL..65"      "AITL..66"      "AITL..67"      "AITL..69"
## [117] "AITL..7"       "AITL..70"      "AITL..71"      "AITL..74"
## [121] "AITL..77"      "AITL..78"      "AITL..79"      "AITL..8"
## [125] "AITL..82"      "AITL..84"      "AITL..9"       "PTCL.nos..100"
## [129] "PTCL.nos..101" "PTCL.nos..102" "PTCL.nos..104" "PTCL.nos..105"
## [133] "PTCL.nos..109" "PTCL.nos..112" "PTCL.nos..115" "PTCL.nos..118"
## [137] "PTCL.nos..119" "PTCL.nos..120" "PTCL.nos..124" "PTCL.nos..126"
## [141] "PTCL.nos..128" "PTCL.nos..13"  "PTCL.nos..134" "PTCL.nos..135"
## [145] "PTCL.nos..136" "PTCL.nos..139" "PTCL.nos..143" "PTCL.nos..15"
## [149] "PTCL.nos..151" "PTCL.nos..156" "PTCL.nos..16"  "PTCL.nos..161"
## [153] "PTCL.nos..162" "PTCL.nos..166" "PTCL.nos..171" "PTCL.nos..174"
## [157] "PTCL.nos..178" "PTCL.nos..180" "PTCL.nos..185" "PTCL.nos..186"
## [161] "PTCL.nos..189" "PTCL.nos..194" "PTCL.nos..195" "PTCL.nos..196"
## [165] "PTCL.nos..20"  "PTCL.nos..200" "PTCL.nos..201" "PTCL.nos..203"
## [169] "PTCL.nos..208" "PTCL.nos..209" "PTCL.nos..211" "PTCL.nos..212"
## [173] "PTCL.nos..213" "PTCL.nos..215" "PTCL.nos..216" "PTCL.nos..218"
## [177] "PTCL.nos..219" "PTCL.nos..22"  "PTCL.nos..226" "PTCL.nos..23"
## [181] "PTCL.nos..230" "PTCL.nos..231" "PTCL.nos..232" "PTCL.nos..236"
## [185] "PTCL.nos..237" "PTCL.nos..239" "PTCL.nos..24"  "PTCL.nos..240"
## [189] "PTCL.nos..241" "PTCL.nos..242" "PTCL.nos..243" "PTCL.nos..244"
## [193] "PTCL.nos..246" "PTCL.nos..247" "PTCL.nos..249" "PTCL.nos..25"
## [197] "PTCL.nos..251" "PTCL.nos..252" "PTCL.nos..258" "PTCL.nos..27"
## [201] "PTCL.nos..28"  "PTCL.nos..287" "PTCL.nos..288" "PTCL.nos..289"
## [205] "PTCL.nos..29"  "PTCL.nos..290" "PTCL.nos..291" "PTCL.nos..292"
## [209] "PTCL.nos..293" "PTCL.nos..294" "PTCL.nos..3"   "PTCL.nos..30"
## [213] "PTCL.nos..32"  "PTCL.nos..33"  "PTCL.nos..34"  "PTCL.nos..408"
## [217] "PTCL.nos..409" "PTCL.nos..414" "PTCL.nos..415" "PTCL.nos..416"
```

```
## [221] "PTCL.nos..424" "PTCL.nos..428" "PTCL.nos..432" "PTCL.nos..434"
## [225] "PTCL.nos..440" "PTCL.nos..441" "PTCL.nos..444" "PTCL.nos..445"
## [229] "PTCL.nos..446" "PTCL.nos..448" "PTCL.nos..451" "PTCL.nos..452"
## [233] "PTCL.nos..455" "PTCL.nos..46"  "PTCL.nos..460" "PTCL.nos..463"
## [237] "PTCL.nos..465" "PTCL.nos..467" "PTCL.nos..468" "PTCL.nos..469"
## [241] "PTCL.nos..47"  "PTCL.nos..470" "PTCL.nos..471" "PTCL.nos..482"
## [245] "PTCL.nos..504" "PTCL.nos..506" "PTCL.nos..519" "PTCL.nos..52"
## [249] "PTCL.nos..524" "PTCL.nos..527" "PTCL.nos..529" "PTCL.nos..531"
## [253] "PTCL.nos..535" "PTCL.nos..61"  "PTCL.nos..63"  "PTCL.nos..68"
## [257] "PTCL.nos..72"  "PTCL.nos..80"  "PTCL.nos..86"  "PTCL.nos..87"
## [261] "PTCL.nos..89"  "PTCL.nos..90"  "PTCL.nos..91"  "PTCL.nos..92"
## [265] "PTCL.nos..93"  "PTCL.nos..94"  "PTCL.nos..95"  "PTCL.nos..96"
## [269] "PTCL.nos..97"  "PTCL.nos..98"  "PTCL.nos..99"
```

```
#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)

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 PTCL
##   AITL  106   21
##   PTCL   16  128
```

```
## unselect to build confusionMatrix
# confusionMatrix(table(output$true.class, output$LOOCV.predicted  ))

# Confusion Matrix and Statistics
#
#
#        AITL PTCL
#   AITL  106   21
#   PTCL   16  128
#
#                Accuracy : 0.8635
#                  95% CI : (0.8168, 0.902)
#     No Information Rate : 0.5498
#     P-Value [Acc > NIR] : <2e-16
#
#                   Kappa : 0.7252
#  Mcnemar's Test P-Value : 0.5108
#
#             Specificity : 0.8591
#          Pos Pred Value : 0.8346
#          Neg Pred Value : 0.8889
#              Prevalence : 0.4502
#          Detection Rate : 0.3911
#    Detection Prevalence : 0.4686
#       Balanced Accuracy : 0.8640
#
#        'Positive' Class : AITL
```

Extracting the most significant clusters based on 19-gene signature

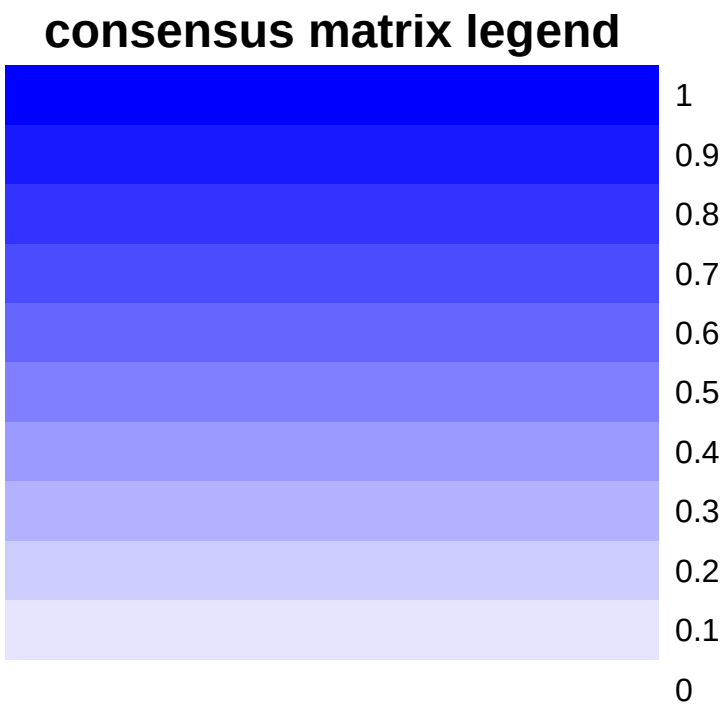
Analyze sample stratification based on the extracted differentially expressed genes betwee AILT and PTCL-nos and the ALCL ALK-negative 3-gene model.

```
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_n

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,
```

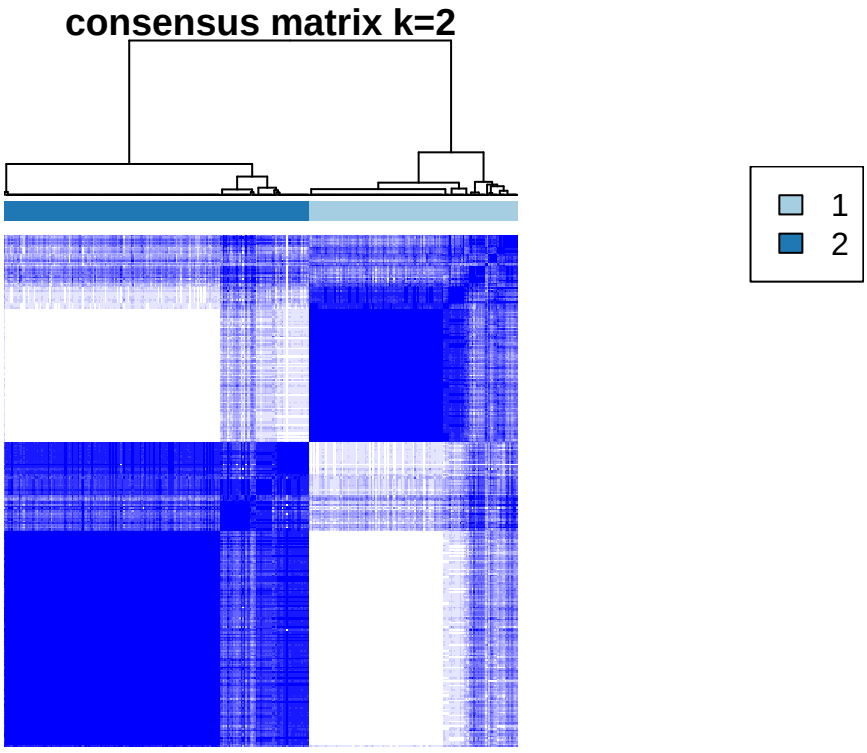
```
                                clusterAlg="hc",
                                innerLinkage="ward.D2",
                                finalLinkage="ward.D2",
                                distance="euclidean",
                                seed=123456789)
```

```
## end fraction
```

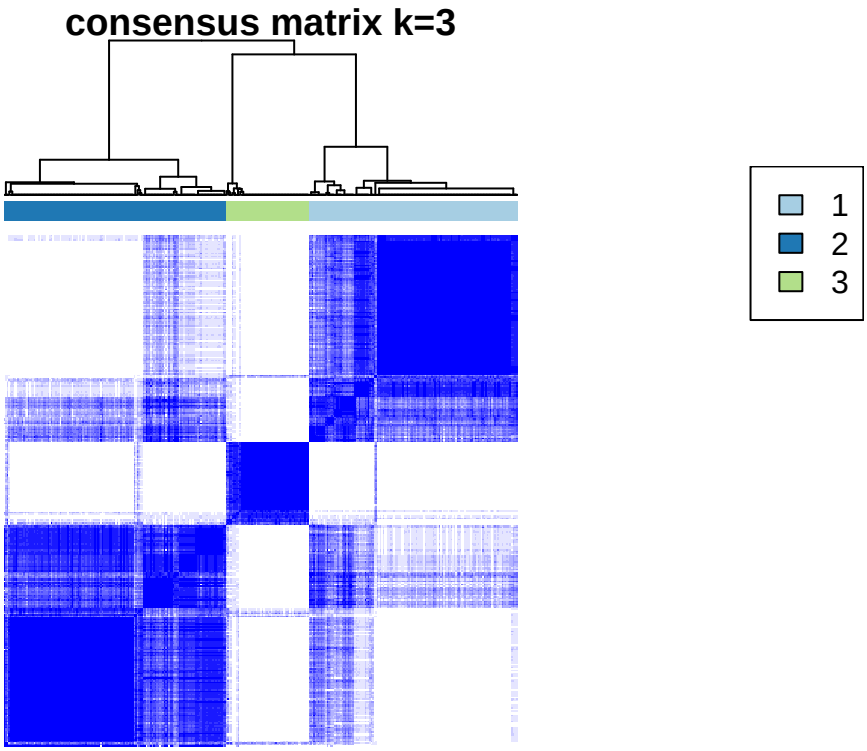
```
## clustered
```



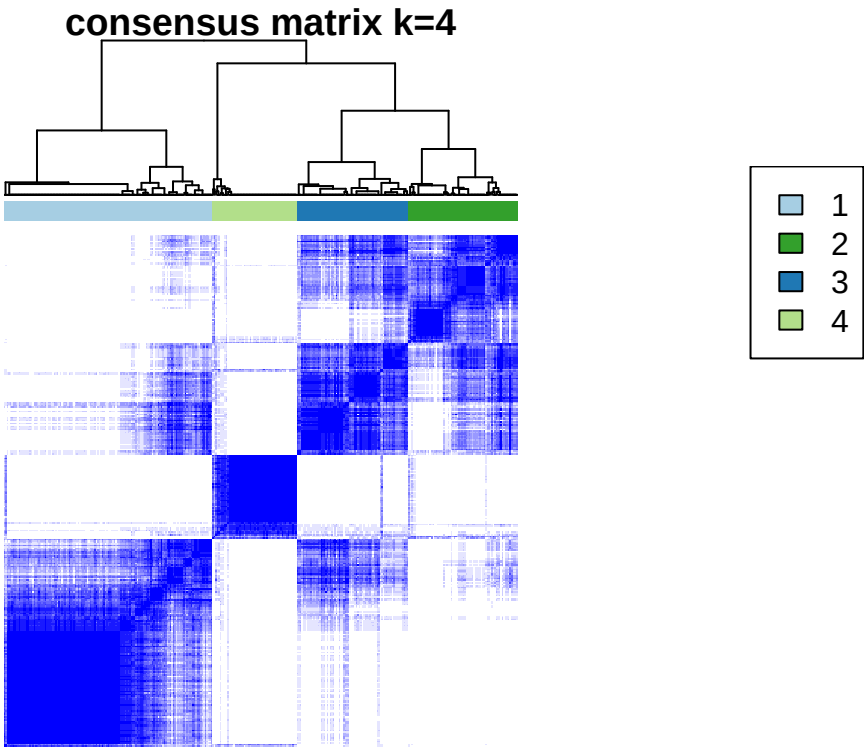

```
## clustered
```



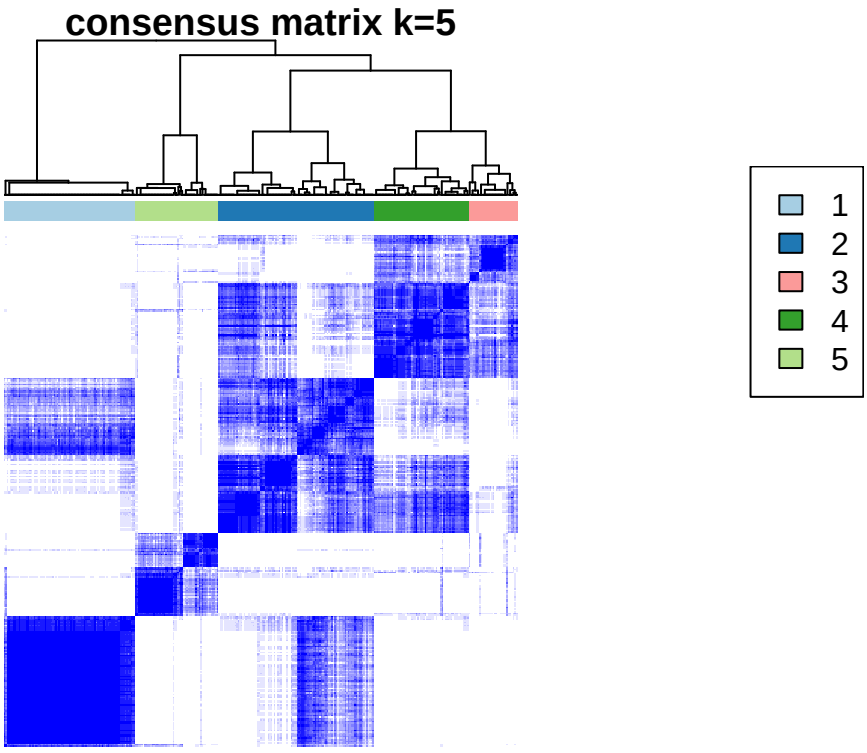

```
## clustered
```



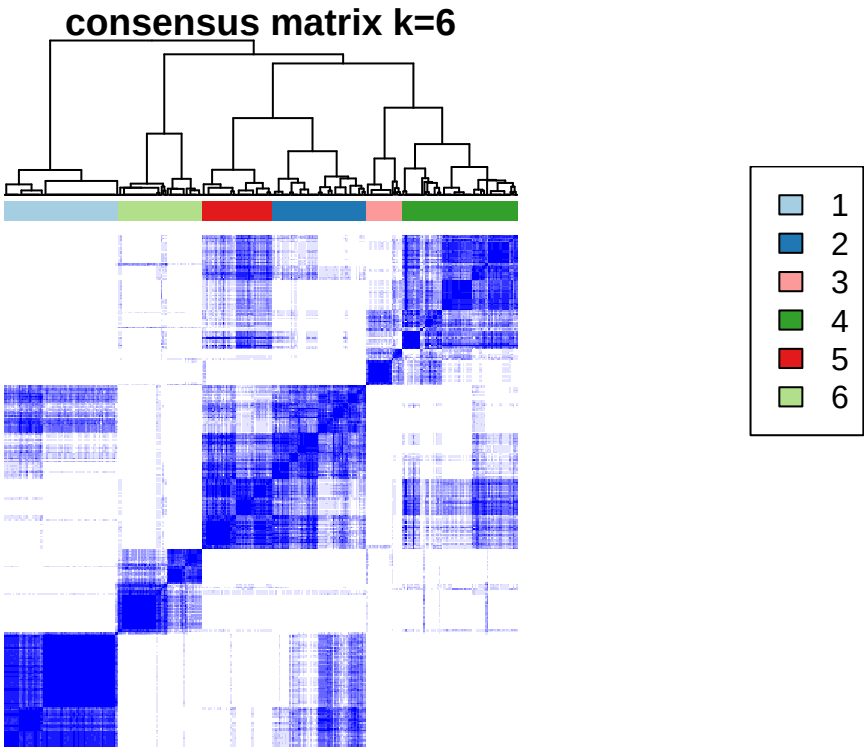

```
## clustered
```



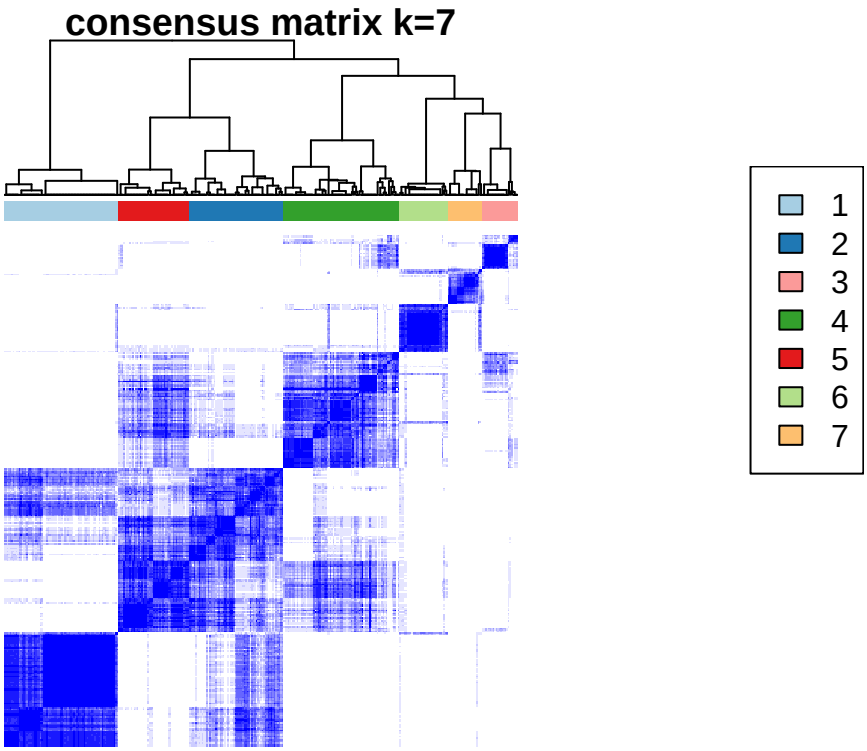

```
## clustered
```

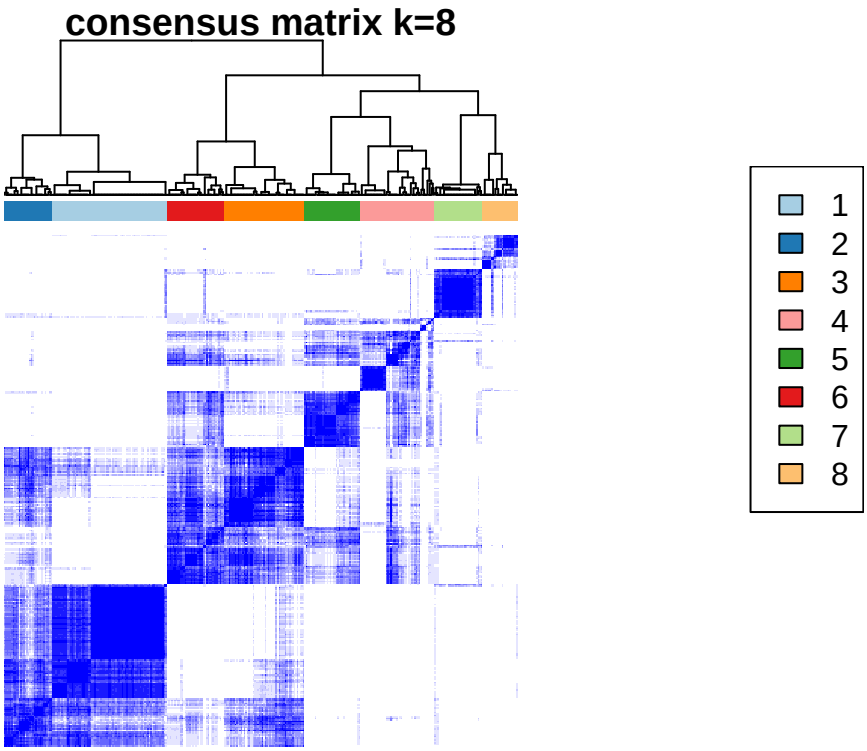


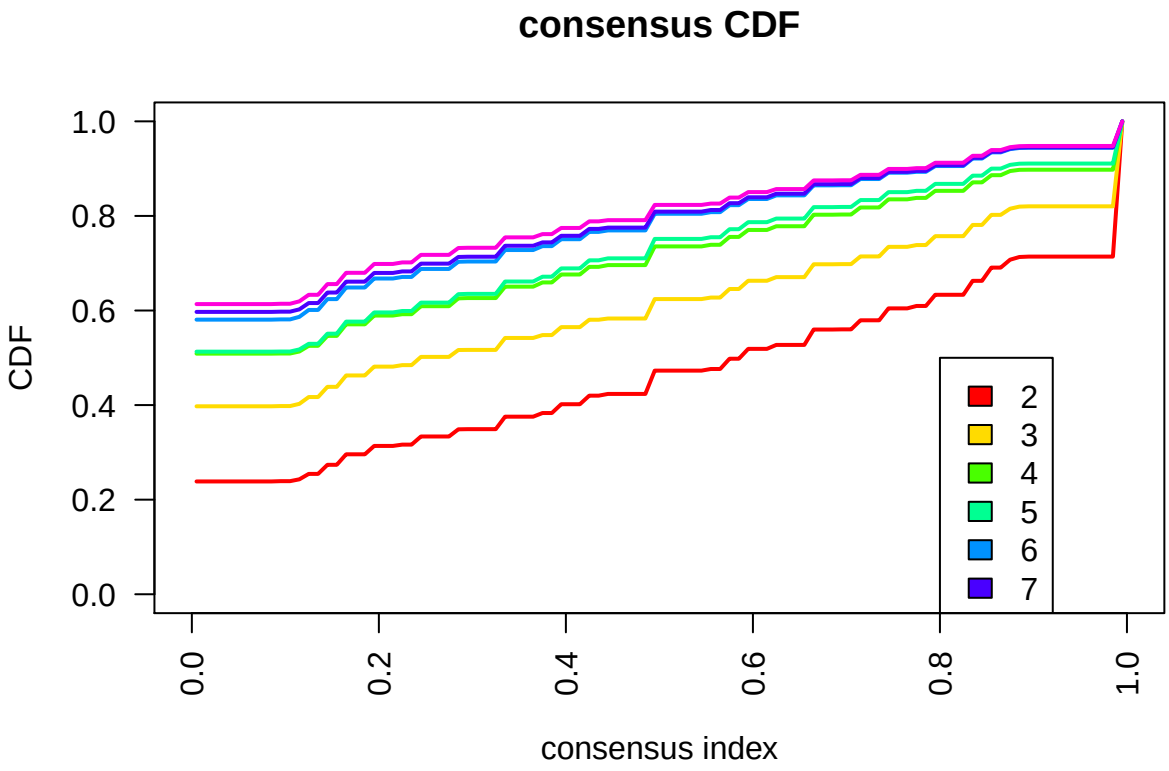

```
## clustered
```

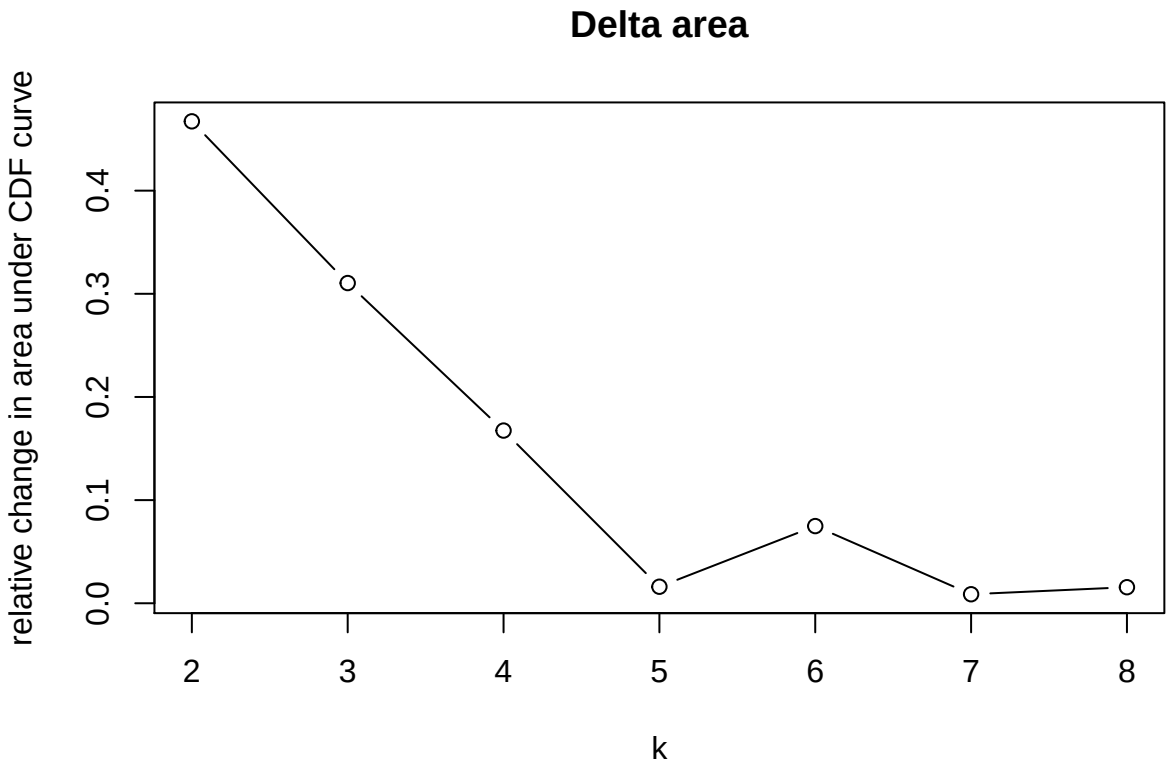


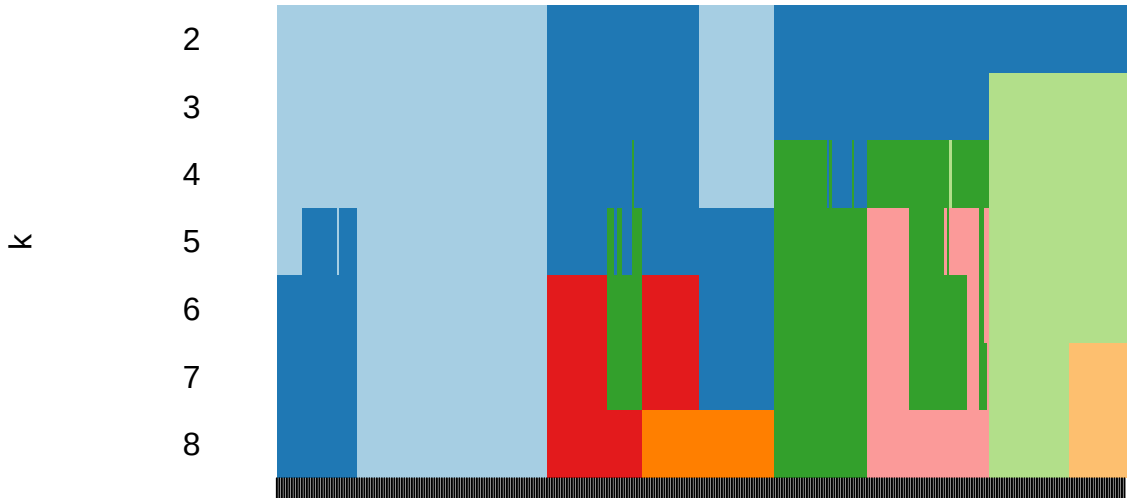

```
## clustered
```


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



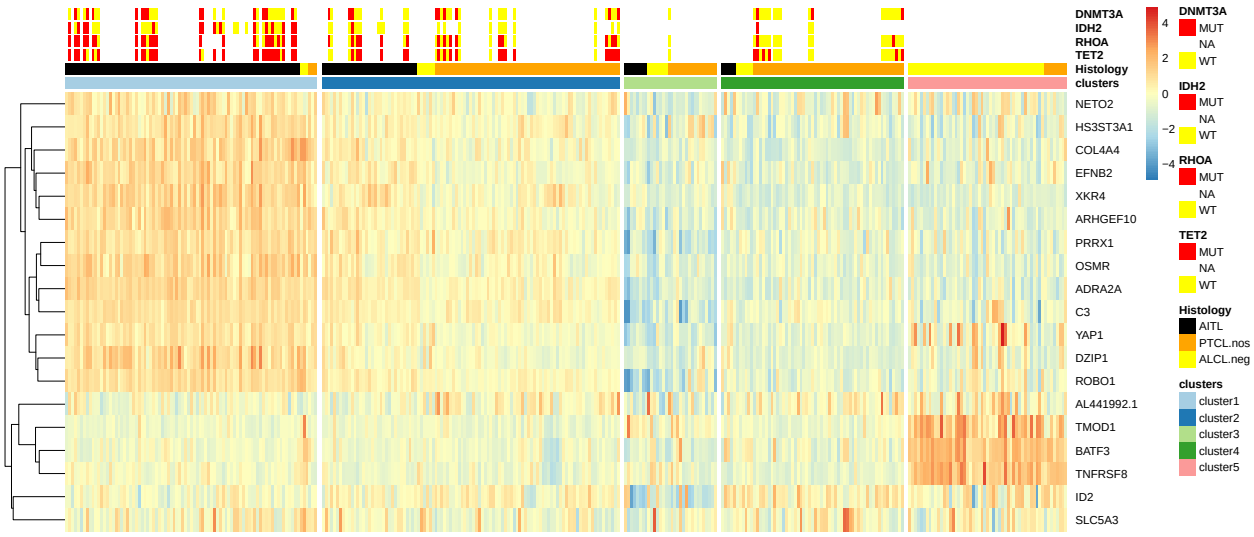

```
kk<- as.data.frame((results[[5]]$consensusClass)) ##### 4 significant cluster
kk$geo.id<- rownames(kk)
colnames(kk)[1]<- "cluster"
table(kk$cluster)
```

```
##
##   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)
```

```r
mylabel.col$Histology[mylabel.col$Histology == "AITL"] = "black"; mylabel.col$Histology[mylabel.col$His
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=""))

par(mfrow=c(1,1))
par(mar=c(5,5,5,5), xpd=F)
mat3<- t(data.matrix(heat2[,2:20]))
colnames(mat3)<-heat2$Row.names
mat3= mat3[order(rownames(mat3)),]
temp_name = getBM( attributes = c("ensembl_transcript_id", "entrezgene", "external_gene_name"), filters
temp_name = temp_name[!duplicated(temp_name[,1]),]
rownames(mat3) = temp_name$external_gene_name
mat3  <- mat3 - rowMeans(mat3)
par(mfrow=c(1,1))
#pheatmap(mat3, annotation_col = mylabel.nocol, annotation_colors = list(clusters = c(cluster1= mycol_p
#dev.off()
# print with gaps
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,
```

```
gep<- geneExpr[,select_hist$sample.nameNEW]
mat<- gep[list_genes_all,]
geneannotation1 <- getBM( attributes = c("ensembl_transcript_id", "entrezgene", "external_gene_name"), i
sort(unique(geneannotation1$external_gene_name))
```

```
##  [1] "ADRA2A"     "AL441992.1" "ARHGEF10"   "BATF3"      "C3"
##  [6] "COL4A4"     "DZIP1"      "EFNB2"      "HS3ST3A1"   "ID2"
## [11] "NETO2"      "OSMR"       "PRRX1"      "ROBO1"      "SLC5A3"
## [16] "TMOD1"      "TNFRSF8"    "XKR4"       "YAP1"
```

```
setdiff(rownames(mat), paste0(unique(geneannotation1$entrezgene),"_at"))
```

```
## character(0)
```

```
for (ii in 1:nrow(mat)) {
  #if(length (which (paste0(unique(geneannotation1$entrezgene),"_at") == rownames(mat)[ii])) != 0 ) row
  rownames(mat) [ii] = unique(geneannotation1$external_gene_name) [ which (paste0(unique(geneannotation
}

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

## unselect below to cluster data

# mat  <- mat - rowMeans(mat)
# par(mfrow=c(1,1))
# pheatmap(mat, annotation_col = mylabel.nocol, annotation_colors = list(final.molec = c(AITL = "black"
#                                   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_colname
#          border_color= NA, color = colorRampPalette(rev(brewer.pal(n = 5 , name = "RdYlBu")))(20), sc

### export pts order
cluster.pts.nr$tree_col$labels [cluster.pts.nr$tree_col$order]
```

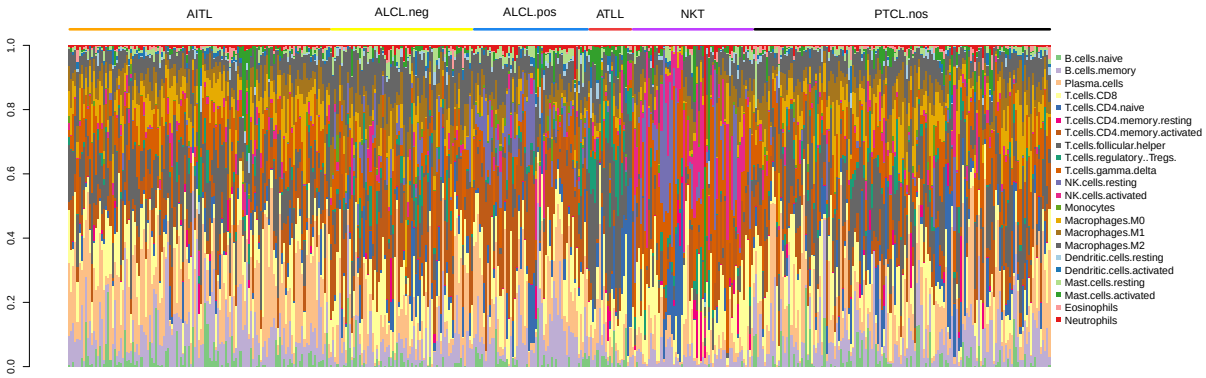
```
##   [1] "PTCL.nos..23"  "PTCL.nos..428" "PTCL.nos..448" "PTCL.nos..124"
##   [5] "PTCL.nos..247" "PTCL.nos..463" "PTCL.nos..89"  "PTCL.nos..156"
##   [9] "PTCL.nos..432" "PTCL.nos..216" "PTCL.nos..25"  "PTCL.nos..87"
##  [13] "PTCL.nos..94"  "PTCL.nos..98"  "PTCL.nos..105" "PTCL.nos..93"
```

```
##  [17] "PTCL.nos..195" "AITL..413"     "PTCL.nos..531" "PTCL.nos..143"
##  [21] "PTCL.nos..46"  "PTCL.nos..28"  "PTCL.nos..185" "PTCL.nos..416"
##  [25] "PTCL.nos..112" "PTCL.nos..424" "PTCL.nos..134" "PTCL.nos..32"
##  [29] "PTCL.nos..22"  "PTCL.nos..194" "PTCL.nos..30"  "PTCL.nos..211"
##  [33] "PTCL.nos..52"  "PTCL.nos..97"  "PTCL.nos..201" "PTCL.nos..27"
##  [37] "PTCL.nos..68"  "PTCL.nos..139" "PTCL.nos..72"  "PTCL.nos..120"
##  [41] "PTCL.nos..444" "PTCL.nos..24"  "PTCL.nos..15"  "PTCL.nos..109"
##  [45] "PTCL.nos..29"  "PTCL.nos..100" "PTCL.nos..171" "PTCL.nos..104"
##  [49] "PTCL.nos..99"  "PTCL.nos..126" "PTCL.nos..258" "AITL..536"
##  [53] "PTCL.nos..535" "PTCL.nos..20"  "PTCL.nos..102" "PTCL.nos..452"
##  [57] "PTCL.nos..529" "PTCL.nos..90"  "PTCL.nos..230" "PTCL.nos..231"
##  [61] "PTCL.nos..232" "PTCL.nos..236" "PTCL.nos..16"  "PTCL.nos..189"
##  [65] "PTCL.nos..506" "PTCL.nos..519" "PTCL.nos..151" "PTCL.nos..186"
##  [69] "AITL..419"     "PTCL.nos..455" "PTCL.nos..47"  "PTCL.nos..213"
##  [73] "PTCL.nos..161" "PTCL.nos..61"  "PTCL.nos..209" "PTCL.nos..119"
##  [77] "PTCL.nos..80"  "PTCL.nos..34"  "PTCL.nos..440" "AITL..19"
##  [81] "AITL..18"      "PTCL.nos..504" "PTCL.nos..118" "PTCL.nos..293"
##  [85] "PTCL.nos..92"  "PTCL.nos..251" "PTCL.nos..101" "PTCL.nos..446"
##  [89] "AITL..479"     "PTCL.nos..469" "AITL..411"     "AITL..473"
##  [93] "AITL..472"     "AITL..481"     "AITL..487"     "PTCL.nos..434"
##  [97] "PTCL.nos..180" "PTCL.nos..135" "PTCL.nos..445" "PTCL.nos..408"
## [101] "PTCL.nos..409" "PTCL.nos..460" "PTCL.nos..468" "PTCL.nos..470"
## [105] "PTCL.nos..471" "PTCL.nos..441" "PTCL.nos..451" "AITL..12"
## [109] "PTCL.nos..237" "AITL..165"     "PTCL.nos..178" "AITL..458"
## [113] "AITL..191"     "AITL..163"     "AITL..187"     "AITL..62"
## [117] "PTCL.nos..249" "AITL..250"     "AITL..257"     "AITL..110"
## [121] "AITL..260"     "AITL..60"      "AITL..77"      "AITL..84"
## [125] "AITL..106"     "AITL..74"      "AITL..133"     "AITL..113"
## [129] "AITL..127"     "AITL..44"      "AITL..82"      "AITL..197"
## [133] "AITL..223"     "AITL..17"      "AITL..523"     "AITL..530"
## [137] "AITL..154"     "AITL..45"      "AITL..505"     "AITL..2"
## [141] "AITL..238"     "AITL..11"      "AITL..259"     "AITL..10"
## [145] "AITL..234"     "AITL..435"     "PTCL.nos..239" "AITL..6"
## [149] "AITL..438"     "AITL..518"     "AITL..532"     "AITL..256"
## [153] "AITL..449"     "AITL..129"     "AITL..9"       "AITL..450"
## [157] "AITL..534"     "AITL..66"      "AITL..255"     "AITL..207"
## [161] "AITL..8"       "AITL..1"       "AITL..152"     "AITL..229"
## [165] "AITL..248"     "AITL..235"     "AITL..420"     "AITL..483"
## [169] "AITL..157"     "AITL..179"     "AITL..67"      "AITL..70"
## [173] "AITL..225"     "AITL..71"      "AITL..58"      "AITL..78"
## [177] "AITL..137"     "AITL..206"     "AITL..459"     "AITL..144"
## [181] "AITL..222"     "PTCL.nos..294" "AITL..198"     "AITL..453"
## [185] "AITL..210"     "AITL..26"      "AITL..114"     "PTCL.nos..226"
## [189] "AITL..51"      "AITL..224"     "AITL..69"      "AITL..123"
## [193] "AITL..221"     "AITL..150"     "AITL..43"      "AITL..153"
## [197] "AITL..520"     "AITL..159"     "AITL..79"      "AITL..204"
## [201] "AITL..214"     "PTCL.nos..246" "AITL..167"     "AITL..190"
## [205] "PTCL.nos..289" "PTCL.nos..287" "PTCL.nos..288" "PTCL.nos..128"
## [209] "PTCL.nos..136" "PTCL.nos..91"  "PTCL.nos..86"  "PTCL.nos..196"
## [213] "PTCL.nos..212" "PTCL.nos..13"  "PTCL.nos..96"  "PTCL.nos..95"
## [217] "PTCL.nos..208" "PTCL.nos..290" "PTCL.nos..115" "PTCL.nos..219"
## [221] "AITL..484"     "AITL..7"       "AITL..457"     "AITL..454"
## [225] "PTCL.nos..200" "PTCL.nos..215" "PTCL.nos..252" "PTCL.nos..166"
## [229] "PTCL.nos..218" "PTCL.nos..291" "PTCL.nos..292" "PTCL.nos..467"
```

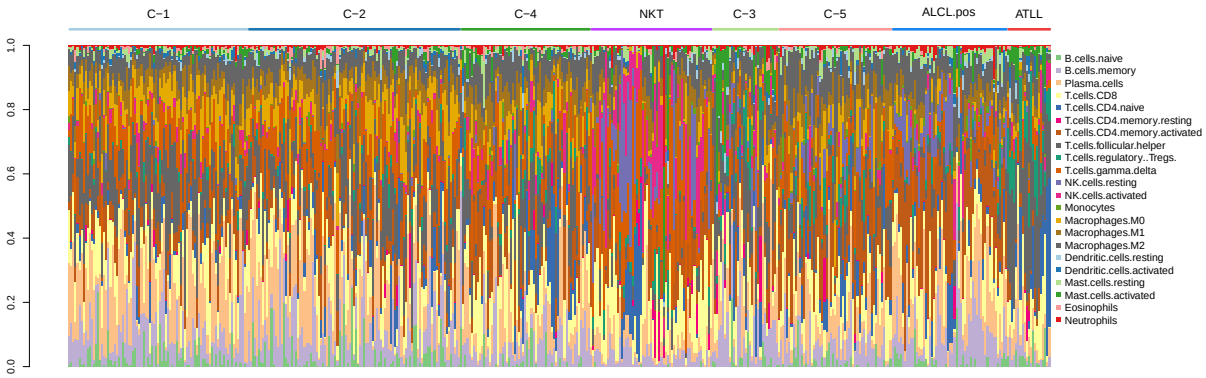
```
## [233] "PTCL.nos..482" "AITL..502"     "AITL..515"     "AITL..406"
## [237] "PTCL.nos..162" "PTCL.nos..465" "PTCL.nos..203" "PTCL.nos..63"
## [241] "PTCL.nos..240" "PTCL.nos..33"  "AITL..510"     "AITL..513"
## [245] "AITL..147"     "AITL..426"     "PTCL.nos..527" "PTCL.nos..414"
## [249] "PTCL.nos..415" "AITL..14"      "AITL..456"     "AITL..205"
## [253] "AITL..55"      "AITL..64"      "AITL..199"     "AITL..65"
## [257] "PTCL.nos..3"   "AITL..121"     "AITL..517"     "PTCL.nos..174"
## [261] "PTCL.nos..524" "PTCL.nos..243" "PTCL.nos..242" "PTCL.nos..244"
## [265] "AITL..392"     "AITL..466"     "AITL..4"       "AITL..5"
## [269] "PTCL.nos..241" "AITL..417"     "AITL..461"
```

Cibersort to characterize tumour microenviroment composition of each cluster

```
##### cibersort and origical molecular histologies
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_v
            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)

}
```



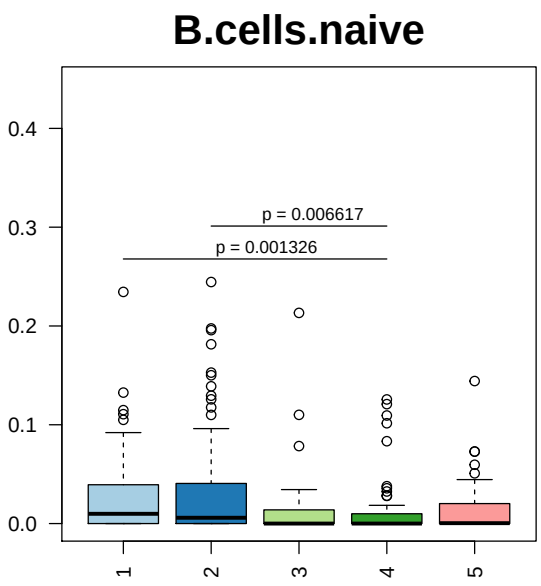

```
##### plot cibersort profile of patients stratified according to histology and clusters

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)

}
```

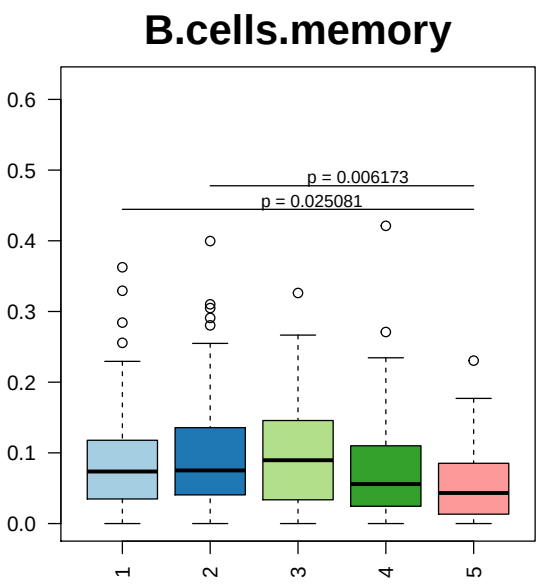


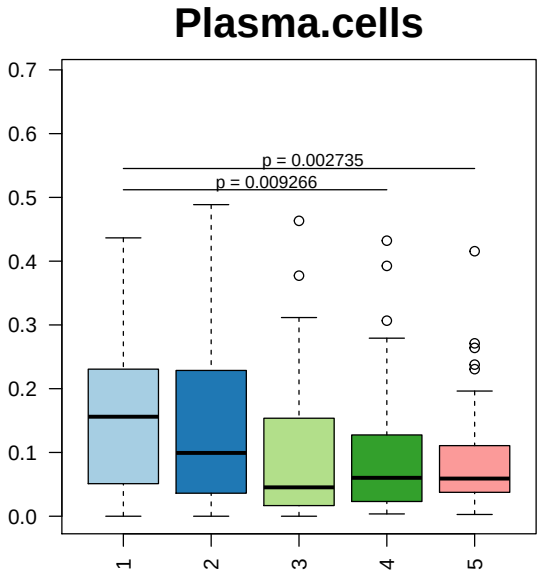

```
# dev.off()
```

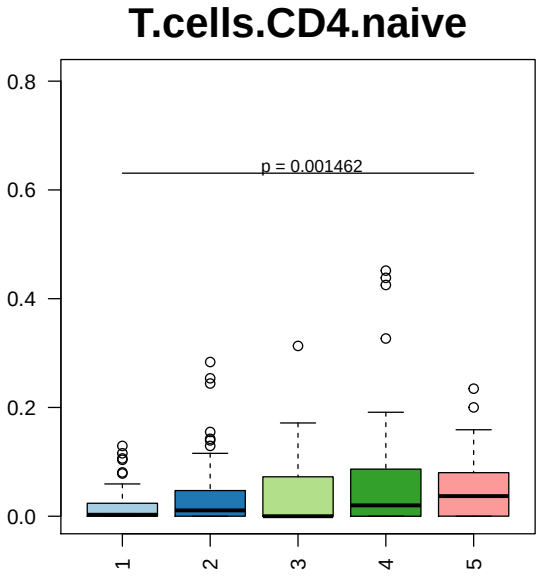
Boxplot comparing the contribution of each cibersort signature between all extracted clusters

```
par(mfrow=c(1,2))
par(mar=c(3,3,3,3), xpd=F)
for(i in (7:27))
{
  #pdf(sprintf("%s_cibersort_ptcl.pdf",i), height=8, width=10)
  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)
  }
    }
  #dev.off()
   }
```

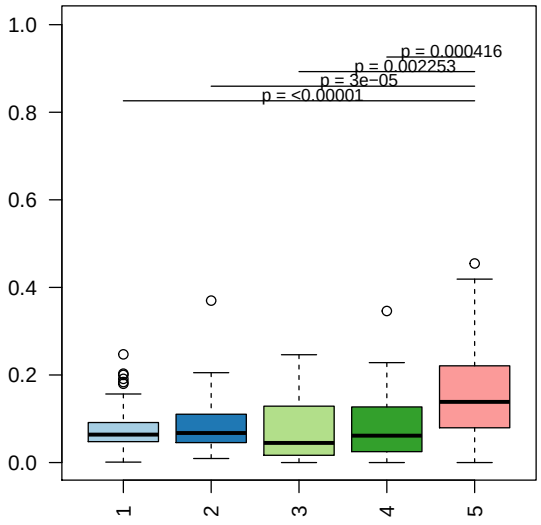

B.cells.naive
p = 0.006617
p = 0.001326
0.0
0.1
0.2
0.3
0.4
1
2
3
4
5


B.cells.memory
p = 0.006173
p = 0.025081
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
2
3
4
5

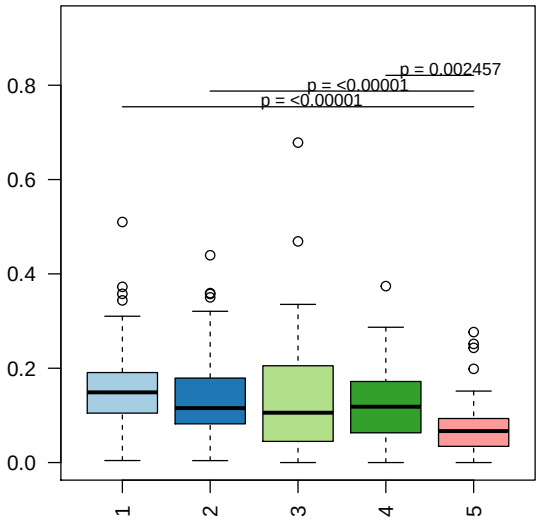

Plasma.cells
p = 0.002735
p = 0.009266
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1
2
3
4
5


T.cells.CD4.naive
p = 0.001462
0.0
0.2
0.4
0.6
0.8
1
2
3
4
5

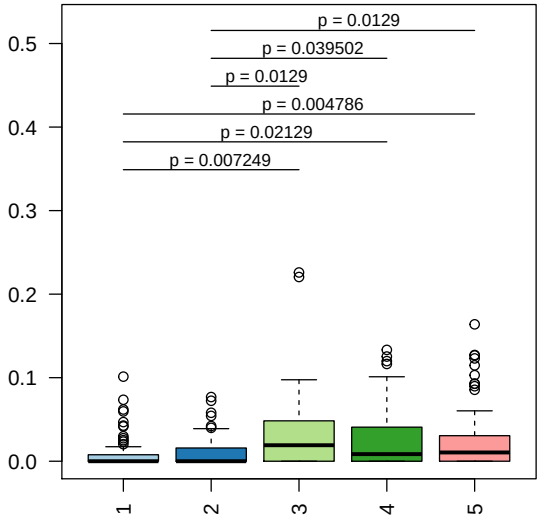
T.cells.CD4.memory.activated



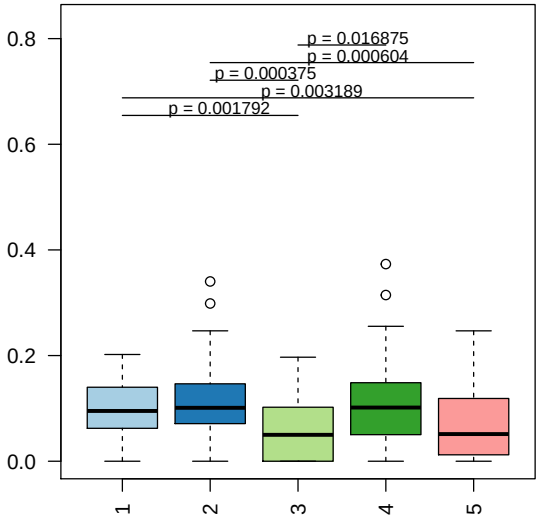
T.cells.follicular.helper

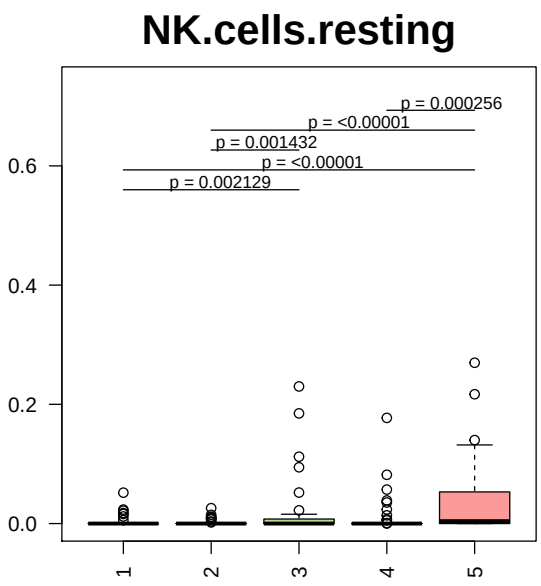


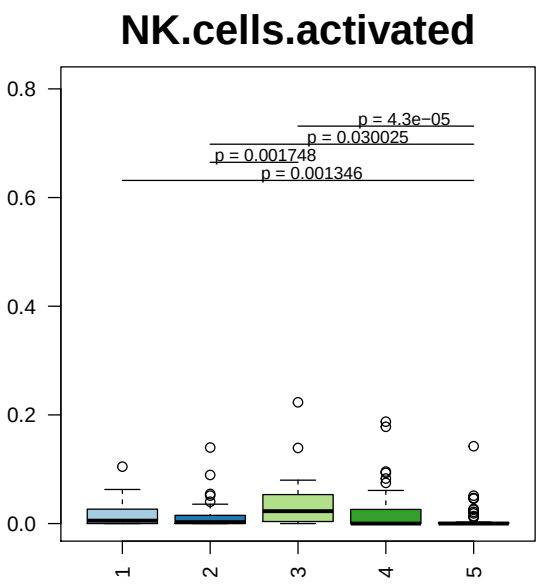
T.cells.regulatory..Tregs.

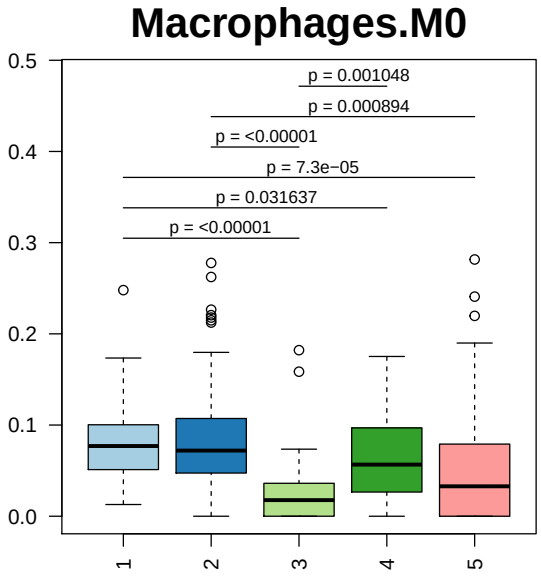


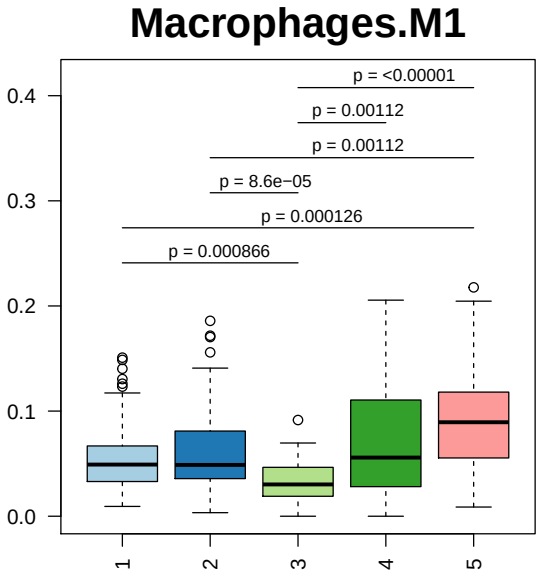
T.cells.gamma.delta

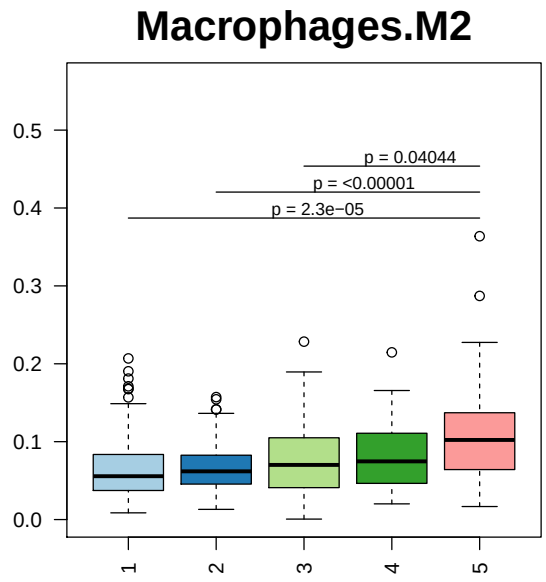


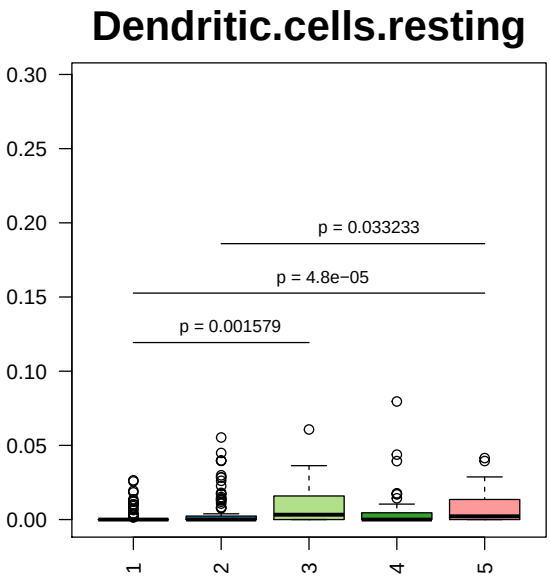

NK.cells.resting
p = 0.000256
p = <0.00001
p = 0.001432
p = <0.00001
p = 0.002129
0.6
0.4
0.2
0.0
1
2
3
4
5

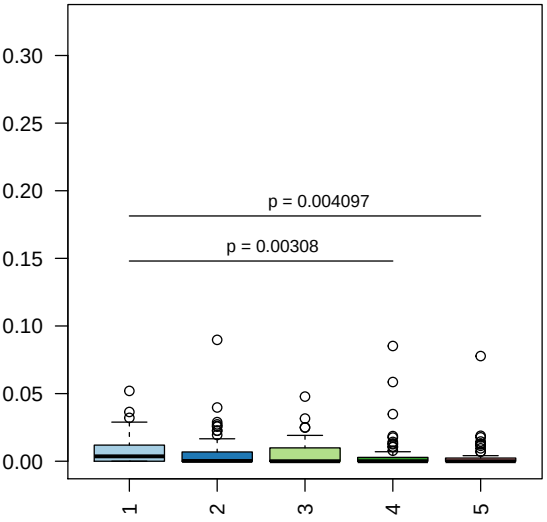

NK.cells.activated
p = 4.3e−05
p = 0.030025
p = 0.001748
p = 0.001346
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5

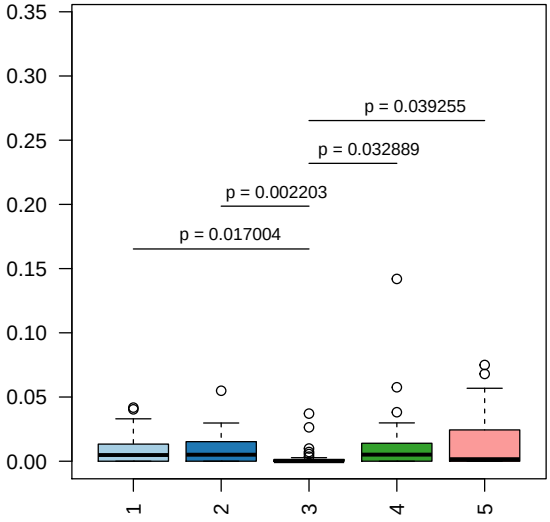

Macrophages.M0
p = 0.001048
p = 0.000894
p = <0.00001
p = 7.3e−05
p = 0.031637
p = <0.00001
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5

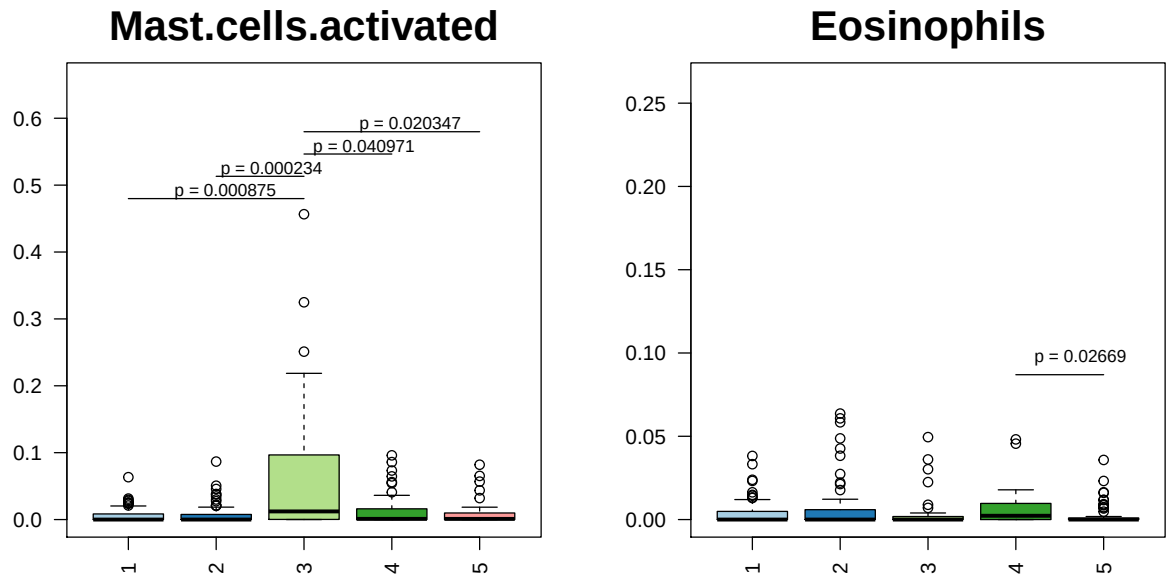

Macrophages.M1
p = <0.00001
p = 0.00112
p = 0.00112
p = 8.6e−05
p = 0.000126
p = 0.000866
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5


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
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1
2
3
4
5


Dendritic.cells.activated
p = 0.004097
p = 0.00308
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1
2
3
4
5


Mast.cells.resting
p = 0.039255
p = 0.032889
p = 0.002203
p = 0.017004
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1
2
3
4
5



R tmod analysis

```
# for convenience: reimport annotated matrix
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_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 <- getBM(attributes=c('entrezgene','hgnc_symbol','hgnc_id'),filters = 'entrezgene', values = gsub(


## Batch submitting query [>-------------------------] 5% eta: 9s Batch
```

```
## submitting query [=>-----------------------] 7% eta: 11s Batch
## submitting query [==>----------------------] 10% eta: 11s Batch
## submitting query [==>----------------------] 12% eta: 11s Batch
## submitting query [===>---------------------] 15% eta: 10s Batch
## submitting query [===>---------------------] 17% eta: 10s Batch
## submitting query [====>--------------------] 20% eta: 10s Batch
## submitting query [=====>-------------------] 22% eta: 10s Batch
## submitting query [=====>-------------------] 24% eta: 10s Batch
## submitting query [======>------------------] 27% eta: 9s Batch
## submitting query [=======>-----------------] 29% eta: 9s Batch
## submitting query [=======>-----------------] 32% eta: 9s Batch
## submitting query [========>----------------] 34% eta: 9s Batch
## submitting query [=========>---------------] 37% eta: 8s Batch
## submitting query [=========>---------------] 39% eta: 8s Batch
## submitting query [==========>--------------] 41% eta: 8s Batch
## submitting query [==========>--------------] 44% eta: 8s Batch
## submitting query [===========>-------------] 46% eta: 7s Batch
## submitting query [============>------------] 49% eta: 7s Batch
## submitting query [============>------------] 51% eta: 7s Batch
## submitting query [=============>-----------] 54% eta: 6s Batch
## submitting query [==============>----------] 56% eta: 6s Batch
## submitting query [==============>----------] 59% eta: 6s Batch
## submitting query [===============>---------] 61% eta: 5s Batch
## submitting query [===============>---------] 63% eta: 5s Batch
## submitting query [================>--------] 66% eta: 5s Batch
## submitting query [=================>-------] 68% eta: 4s Batch
## submitting query [=================>-------] 71% eta: 4s Batch
## submitting query [==================>------] 73% eta: 4s Batch
## submitting query [===================>-----] 76% eta: 3s Batch
## submitting query [===================>-----] 78% eta: 3s Batch
## submitting query [====================>----] 80% eta: 3s Batch
## submitting query [=====================>---] 83% eta: 2s Batch
## submitting query [=====================>---] 85% eta: 2s Batch
## submitting query [======================>--] 88% eta: 2s Batch
## submitting query [======================>--] 90% eta: 1s Batch
## submitting query [=======================>-] 93% eta: 1s Batch
## submitting query [=======================>-] 95% eta: 1s Batch submitting
## query [=======================>-] 98% eta: 0s Batch submitting query
## [========================] 100% eta: 0s
```

```r
geneExpr3<- as.data.frame.matrix(geneExpr2[which(rownames(geneExpr2) %in% paste0(hgnc$entrezgene,"_at")),]

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)

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
                  )

############ 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"
                 "SOX8","ARHGEF10","DMRT1",  "SLC19A21","STK3","PERP","TNFRSF8","TMOD1","BATF3","CDC14B
                 "TMOD1","ATP6V0D1","AXL","CD59","CHI3L1","CLTC","COL6A1","CREG1","CTSB","CTSC","NR1","
                 "PLSCR1","PRDX3","CTSS","CYBB","FABP3","FPR1","FTL","GUCA2A","HCK","IFI30","IL13RA1","
                 "PRKG1PSAP","SLC7A7","SOD2","TCN2","THY1","TYR","UBE2L6","WARS","AXL","FTL","SIRPA","S
                 "SEPT6","GATA3","CD28","STAT1","AXL","CD28","CD40","CD59","CSF2","FTL","IFNG","LILRB1"
                 "MSH6","EGR1","CAT","EGR1","CAT"))
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"
```

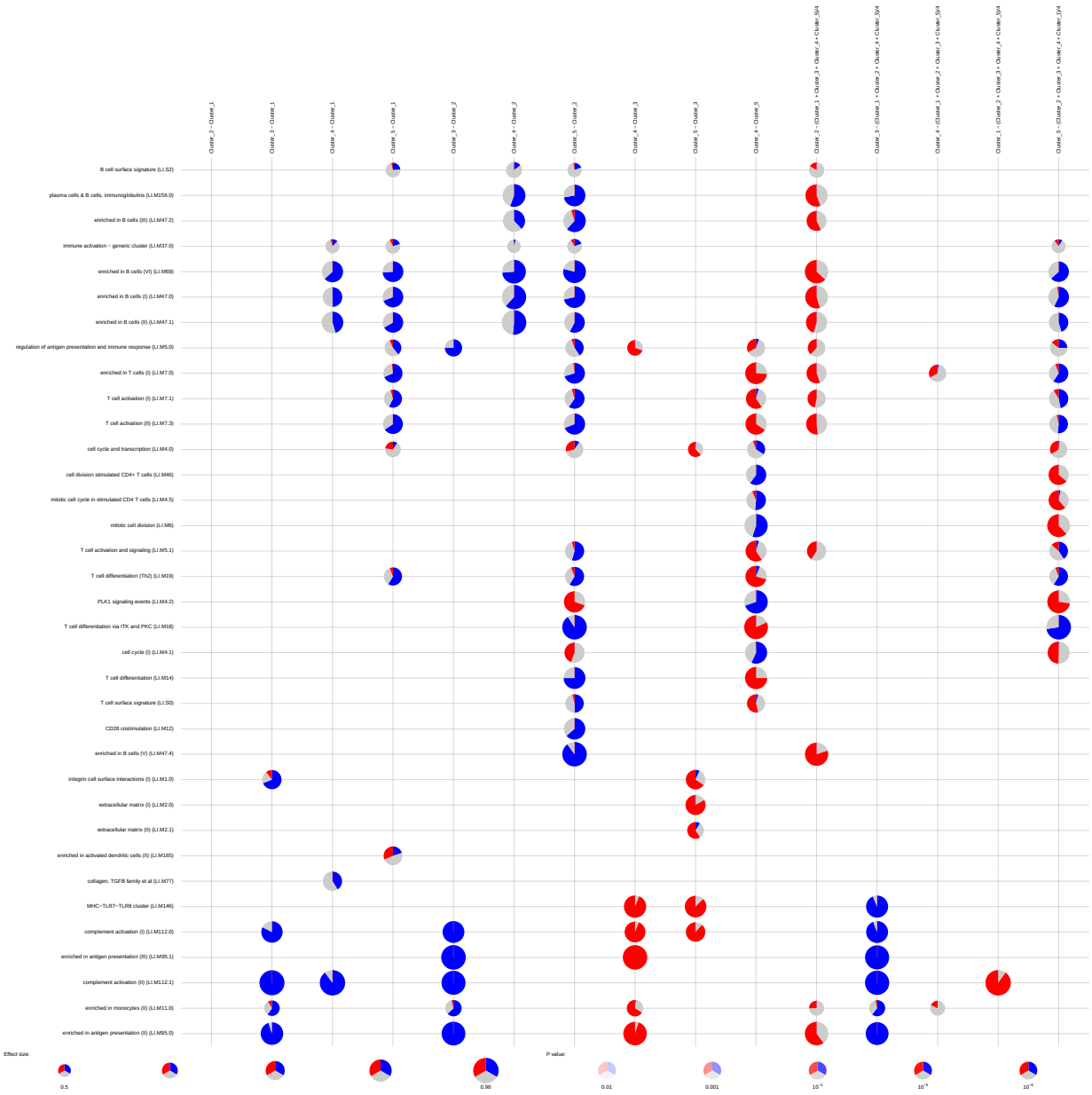
```
fit1 <- lmFit(mat, design)
fit2 <- contrasts.fit(fit1, contrast.matrix)
fit <- eBayes(fit2)
res.l <- tmodLimmaTest(fit, rownames(mat))
length(res.l)
```

```
## [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))
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=0.6) ##### zero = grey, blue down in the first factor and red u
```


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_3 - (Cluster_1 + Cluster_2 + Cluster_4
Cluster_4 - (Cluster_1 + Cluster_2 + Cluster_3
Cluster_1 - (Cluster_2 + Cluster_3 + Cluster_4
Cluster_5 - (Cluster_2 + Cluster_3 + Cluster_4
complement activation (II) (LI.M112.1)
integrin cell surface interactions (I) (LI.M1.0)
cytoskeleton/actin (SRF transcription targets) (LI.M145.0)
enriched in T cells (I) (LI.M7.0)
ytoskeletal remodeling (enriched for SRF targets) (LI.M34)
regulation of signal transduction (LI.M3)
TBA (source: B cells) (LI.M217)
cell adhesion (LI.M51)
myeloid, dendritic cell activation via NFkB (I) (LI.M43.0)
enriched in B cells (VI) (LI.M69)
enriched in B cells (I) (LI.M47.0)
enriched in B cells (III) (LI.M47.1)
complement activation (I) (LI.M112.0)
B cell surface signature (LI.S2)
enriched in myeloid cells and monocytes (LI.M81)
Resting dendritic cell surface signature (LI.S10)
enriched in monocytes (II) (LI.M11.0)
mitotic cell cycle in stimulated CD4 T cells (LI.M4.5)
putative targets of PAX3 (LI.M89.1)
mitotic cell division (LI.M6)
lymphocyte generic cluster (LI.M60)
targets of FOSL1/2 (LI.M0)
MHC−TLR7−TLR8 cluster (LI.M146)
enriched in antigen presentation (III) (LI.M95.1)
collagen, TGFB family et al (LI.M77)
extracellular matrix, collagen (LI.M210)
platelet activation and degranulation (LI.M85)
enriched in antigen presentation (II) (LI.M95.0)
T & B cell development, activation (LI.M62.0)
IL2, IL7, TCR network (LI.M65)
integrin mediated leukocyte migration (LI.M39)
enriched in NK cells (III) (LI.M157)
leukocyte activation and migration (LI.M45)
enriched in NK cells (I) (LI.M7.2)
plasma cells, immunoglobulins (LI.M156.1)
cell adhesion (lymphocyte homing) (LI.M21)
BCR signaling (LI.M54)
cell adhesion (GO) (LI.M117)
T cell activation (III) (LI.M7.4)
double positive thymocytes (LI.M126)
Ran mediated mitosis (LI.M15)
mitotic cell cycle (LI.M4.7)
T cell activation (IV) (LI.M52)
CD4 T cell surface signature Th2−stimulated (LI.S7)
Rho GTPase cycle (LI.M4.14)
E2F transcription factor network (LI.M8)
cell division in stimulated CD4 T cells (LI.M4.6)
mitotic cell cycle − DNA replication (LI.M4.4)
cell cycle (III) (LI.M103)
cell cycle (II) (LI.M4.10)
mitotic cell cycle in stimulated CD4 T cells (LI.M4.9)
small GTPase mediated signal transduction (LI.M215)
immune activation − generic cluster (LI.M37.0)
T cell differentiation (Th2) (LI.M19)
T cell signaling and costimulation (LI.M44)
enriched in naive and memory B cells (LI.M83)
integrins and cell adhesion (LI.M84)
T cell differentiation via ITK and PKC (LI.M18)
cell cycle (I) (LI.M4.1)
es and inflammatory molecules in myeloid cells (LI.M86.0)
putative targets of PAX3 (LI.M89.0)
purine nucleotide biosynthesis (LI.M212)
mismatch repair (I) (LI.M22.0)
C−MYC transcriptional network (LI.M4.12)
enriched in B cells (IV) (LI.M47.3)
chaperonin mediated protein folding (II) (LI.M204.1)
complement and other receptors in DCs (LI.M40)
enriched in antigen presentation (I) (LI.M71)
extracellular matrix (III) (LI.M2.2)
chaperonin mediated protein folding (I) (LI.M204.0)
enriched in monocytes (I) (LI.M4.15)
integrin cell surface interactions (II) (LI.M1.1)
transcription elongation, RNA polymerase II (LI.M234)
enriched in activated dendritic cells (II) (LI.M165)
T cell activation and signaling (LI.M5.1)
activated dendritic cells (LI.M67)
T cell activation (II) (LI.M7.3)
lysosomal/endosomal proteins (LI.M139)
lysosome (LI.M209)
piratory electron transport chain (mitochondrion) (LI.M219)
l sensing & immunity; IRF2 targets network (II) (LI.M111.1)
piratory electron transport chain (mitochondrion) (LI.M238)
chemokine cluster (I) (LI.M27.0)
chemokine cluster (II) (LI.M27.1)
extracellular matrix (II) (LI.M2.1)
cell junction (GO) (LI.M4.13)
extracellular matrix (I) (LI.M2.0)
on of antigen presentation and immune response (LI.M5.0)
adhesion and migration, chemotaxis (LI.M91)
CD28 costimulation (LI.M12)
enriched in B cells (V) (LI.M47.4)
plasma cells & B cells, immunoglobulins (LI.M156.0)
enriched in B cells (III) (LI.M47.2)