Exploratory analysis of RNA-seq dataset

gg

2015-12-01

To run this file: Rscript -e "rmarkdown::render('GO_enrichment.Rmd')"

Examining gene set enrichment in the Barton dataset

Sources:

correspondence list between the identifiers in our dataset and uniprot identifiers: http://www.uniprot.org/docs/yeast

replaced all ";" with blank

replaced all spaced column separators with a tab

removed all (3) instances. AAAARGH!!!

removed all GAG,POL instances. AAAARGH!!!

saved into uniprot_sgd_correspondence.txt

```
cor.table <- read.table("uniprot_sgd_correspondence.txt", header=F, row.names=2)

aldex.all <- read.table("aldex_all.txt", header=T, row.names=1)
aldex.all.g <- read.table("aldex_all.g.txt", header=T, row.names=1)
aldex.all.b <- read.table("aldex_all.b.txt", header=T, row.names=1)

edgeR.all <- read.table("edgeR_all.txt", header=T, row.names=1)
edgeR.all.g <- read.table("edgeR_all.g.txt", header=T, row.names=1)
edgeR.all.b <- read.table("edgeR_all.b.txt", header=T, row.names=1)

aldex.all.up <- rownames(aldex.all)[which(aldex.all$effect >= 2)]
aldex.all.up.uniprot.na <- as.vector(cor.table[aldex.all.up,2])
aldex.all.up.uniprot <- aldex.all.up.uniprot.na[!is.na(aldex.all.up.uniprot.na)]

aldex.all.down <- rownames(aldex.all)[which(aldex.all$effect <= -2)]
aldex.all.down.uniprot.na <- as.vector(cor.table[aldex.all.down,2])
aldex.all.down.uniprot <- aldex.all.down.uniprot.na[!is.na(aldex.all.down.uniprot.na)]

aldex.all.uniprot <- c(aldex.all.down.uniprot, aldex.all.up.uniprot)

bg <- c(rep("pink", length(aldex.all.down.uniprot)), rep("cyan", length(aldex.all.up.uniprot)) )

aldex.2 <- cbind(aldex.all.uniprot, bg)

write.table(aldex.2, file="aldex.effect2.txt", row.names=F, quote=F)
```

Generate the dataset

```
# load the required packages
library(zCompositions)
```

```
library(compositions)
library(ALDEx2)
library(edgeR)
library(gplots)

# read the dataset
meta <- read.table("metadata.txt", header=T, row.names=1, check.names=F)

# Group:sample:lane
nms <- paste(meta[,2], meta[,3], meta[,1], sep=":")

d <- read.table("countfinal2.file", header=T, row.names=1, check.names=F)

# double check that the column names of d and rownames of meta
# are congruent - they are

# change the column names to something more informative
colnames(d) <- nms
# remove rows with 0 counts
d.gt0 <- d[apply(d,1,sum) > 0,]

###########
# aggregate all replicates
nms.agg <- paste(meta[,2], meta[,3], sep=":")

# make an aggregated dataset by sample
# sum gene counts across samples
d.agg <- aggregate(t(d), by=list(nms.agg), FUN=sum)
rownames(d.agg) <- d.agg$Group.1
d.agg$Group.1 <- NULL

# remove rows with 0 counts
d.agg.gt0 <- t(d.agg[,apply(d.agg, 2, sum) > 0])

# estimate 0 values (zCompositions)
d.agg.n0 <- cmultRepl(t(d.agg.gt0), method="CZM", label=0)

# clr transform
d.agg.n0.clr <- t(apply(d.agg.n0, 1, function(x){log2(x) - mean(log2(x))}))

# check each independently
mvar.s <- mvar(d.agg.n0.clr[1:48,])
pcx.s <- prcomp(d.agg.n0.clr[1:48,])

mvar.w <- mvar(d.agg.n0.clr[49:96,])
pcx.w <- prcomp(d.agg.n0.clr[49:96,])

# find samples that contribute 2X or more of the IQR to the variance
cut.s <- median(apply(pcx.s$x,1,function(x){sum(x^2/mvar.s)})) +
    2 * IQR(apply(pcx.s$x,1,function(x){sum(x^2/mvar.s)}))

cut.w <- median(apply(pcx.w$x,1,function(x){sum(x^2/mvar.w)})) +
    2 * IQR(apply(pcx.w$x,1,function(x){sum(x^2/mvar.w)}))
```

```
# get a vector of names of the outliers
bad.s <- names(which(apply(pcx.s$x,1,function(x){sum(x^2/mvar.s)})>cut.s))
bad.w <- names(which(apply(pcx.w$x,1,function(x){sum(x^2/mvar.w)})>cut.w))

bad <- c(bad.s, bad.w)

good <- rownames(d.agg)[! rownames(d.agg) %in% bad]

d.good <- d.agg[good,]
# remove rows with 0 counts
d.good.gt0 <- t(d.good[,apply(d.good, 2, sum) > 0])

d.bad <- d.agg[bad,]
# remove rows with 0 counts
d.bad.gt0 <- t(d.bad[,apply(d.bad, 2, sum) > 0])

# ALDEx of all
d.aldex <- data.frame(d.agg.gt0)
conds <- c(rep("S", 48), rep("W",48))
x <- aldex.clr(d.aldex, mc.samples=16)
x.e <- aldex.effect(x, conds, verbose=FALSE)
x.t <- aldex.ttest(x, conds)
aldex.de <- rownames(x.t)[which(x.t$wi.eBH < 0.05)]

# ALDEx of good
conds.g <- c(rep("S", length(grep("SNF", good))), rep("W", length(grep("WT", good))))
d.aldex.g <- data.frame(d.good.gt0)
x.g <- aldex.clr(d.aldex.g, mc.samples=16)
x.e.g <- aldex.effect(x.g, conds.g, verbose=FALSE)
x.t.g <- aldex.ttest(x.g, conds.g)

# ALDEx of bad
conds.b <- c(rep("S", length(grep("SNF", bad))), rep("W", length(grep("WT", bad))))
d.aldex.b <- data.frame(d.bad.gt0)
x.b <- aldex.clr(d.aldex.b, mc.samples=16)
x.e.b <- aldex.effect(x.b, conds.b, verbose=FALSE)
x.t.b <- aldex.ttest(x.b, conds.b)

x.all <- data.frame(x.e, x.t)
write.table(x.all, file="aldex_all.txt", sep="\t", quote=F, col.names=NA)

x.all.g <- data.frame(x.e.g, x.t.g)
write.table(x.all.g, file="aldex_all.g.txt", sep="\t", quote=F, col.names=NA)

x.all.b <- data.frame(x.e.b, x.t.b)
write.table(x.all.b, file="aldex_all.b.txt", sep="\t", quote=F, col.names=NA)
```