

# coXpress significance testing

Katie Wang

2024-11-28

Based on code.R, code for “CoXpress: differential co-expression in gene expression data” submitted to BMC Bioinformatics, Michael Watson, 20/7/06.

```
# import libraries  
library(coXpress)  
library(plotrix) # for plotting
```

Integrate this into the existing pipeline.

## Loading data

When integrating this in a pipeline with Go, do the pre-processing in Go and then load that file rather than loading the dataframe from golub.

```
data(golub) # loads our data golub.df; this is included in the coXpress package
```

## Running the coXpress analysis

```
# cluster on ALL subset  
hc.gene <- cluster.gene(golub.df[,1:27], "pearson", "average")  
  
# cut the tree at a correlation of 0.6  
g <- cutree(hc.gene, h=0.4) # the list of modules!
```

## Significance testing

This is the part that we can reimplement in Go. We generate some matrices in Go, and also the t-distribution.

In the future, add the functionality to allow a user to select a module to plot, rather than hardcoding g==3. Also make it so that it works with the rat data from the DiffCoEx paper, as this is currently hard-coded to separate into conditions based on the dimensions of golub.df. (Maybe use a conditional: if Golub, separate it so c1 uses 1:27 and c2 uses 28:38, and if rat data, c1 uses 1:36 while c2 uses 37:72)

```
# figure 2  
# note the position and values of the labels for the t statistics will  
# need to be changed each time coXpress is run  
  
# observed data for group 3; will be plotted in blue  
g3 <- golub.df[g==3,] # g was generated in line 40; this subsets golub.df to just genes in group 3 from  
c1 <- cor(t(g3[,1:27])) # condition 1  
c2 <- cor(t(g3[,28:38]))  
v1 <- c1[lower.tri(c1)]
```

```

v2 <- c2[lower.tri(c2)]

# true random data from the random uniform distribution
tr <- matrix(runif(1292), nrow=34)
c1r <- cor(t(tr[,1:27]))
c2r <- cor(t(tr[,28:38]))
v1r <- c1r[lower.tri(c1r)]
v2r <- c2r[lower.tri(c2r)]

# random data generated by resampling
rs <- golub.df[trunc(runif(34,1,nrow(golub.df)+1)),]
c1s <- cor(t(rs[,1:27]))
c2s <- cor(t(rs[,28:38]))
v1s <- c1s[lower.tri(c1s)]
v2s <- c2s[lower.tri(c2s)]

# create distributions of t statistics by randomly generating 10000 groups
# of size 34, calculating the correlation matrix, and then calculating t
num <- 1000 # number of iterations for t-distribution
# create.dists is a coXpress function
ds1 <- create.dists(golub.df, 1:27, times=num, sizes=34)
ds2 <- create.dists(golub.df, 28:38, times=num, sizes=34)

```

## Import data from Go back into R

TODO: read things in from the file

## A bit more processing in R before generating figures

```

t=155.7 # TODO: once you get the t-value from the file, use that value instead of hard-coding it

# run the density function to plot the distribution
# density is an R stats function for computing kernel density estimates
dens <- density(ds1[[1]], adjust=1.5) # the element at ds1[[1]] is a list
x <- c(dens$x, t) # need to extend the x axis out so that the t-value can be plotted
y <- c(dens$y,0)

```

## Generating the figures

This step should be done in R.

```

# plot the graphs
split.screen(c(2,2))

## [1] 1 2 3 4

screen(1)
par(ps=8,mar=c(3,2.7,2,1),mgp=c(2,1,0))
plot(density(v1,adjust=1.5), xlim=c(-1,1), ylim=c(0,5), main="A", xlab="correlation", col="blue")
lines(density(v1r,adjust=1.5), col="red")
lines(density(v1s,adjust=1.5), col="green")
text(mean(v1), 4, "t=155.7", col="blue", ps=8)

```

```

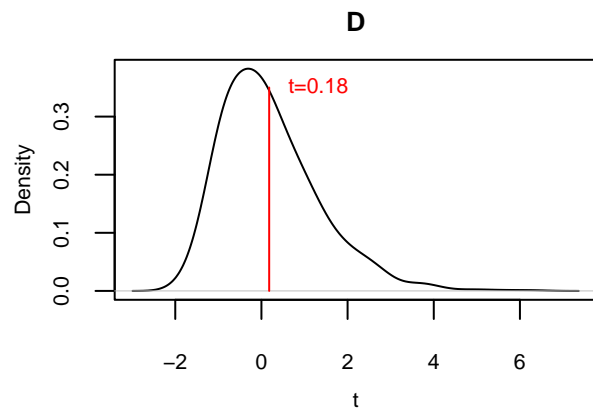
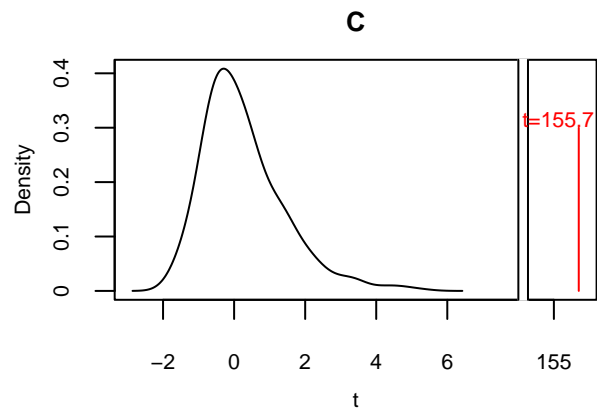
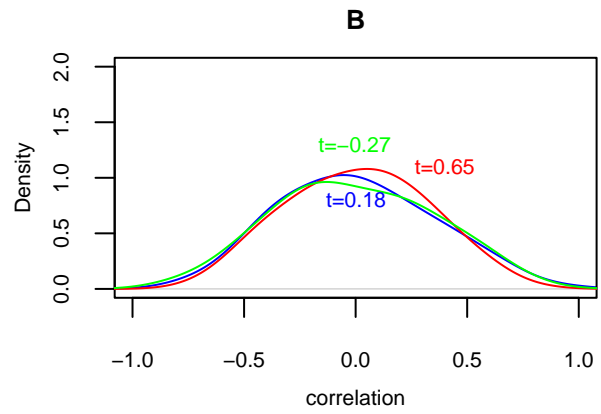
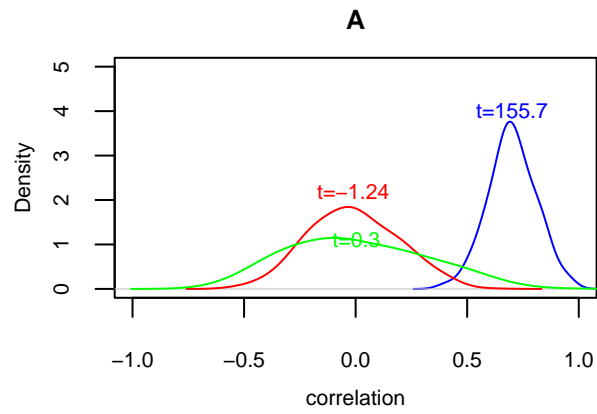
text(mean(v1r), 2.2, paste("t=", round(t.test(v1r)$statistic,2), sep=""), col="red")
text(mean(v1s), 1.1, paste("t=", round(t.test(v1s)$statistic,2), sep=""), col="green")

screen(2)
par(ps=8,mar=c(3,2.7,2,1),mgp=c(2,1,0))
plot(density(v2,adjust=1.5), xlim=c(-1,1), ylim=c(0,2), main="B", xlab="correlation", col="blue")
lines(density(v2r,adjust=1.5), col="red")
lines(density(v2s,adjust=1.5), col="green")
text(mean(v2), .8, "t=0.18", col="blue")
text(0.4, 1.1, paste("t=", round(t.test(v2r)$statistic,2), sep=""), col="red")
text(mean(v2s), 1.3, paste("t=", round(t.test(v2s)$statistic,2), sep=""), col="green")

screen(3)
par(ps=8,mar=c(3,2.7,2,1),mgp=c(2,1,0))
par(pch=".")
gap.plot(x, y, gap=c(8,154), gap.axis="x", xlab="t", ylab="Density", main="C", type="l", xtics=c(-2,0,2))
lines(xy.coords(c(155.7-146,155.7-146),c(0,.3)), col="red")
par(ps=8)
text(155.7-146, 0.31, "t=155.7", col="red", pos=2, offset=-.4)

screen(4)
par(ps=8,mar=c(3,2.7,2,1),mgp=c(2,1,0))
plot(density(ds2[["n34"]], adjust=1.5), xlab="t", main="D")
lines(xy.coords(c(0.18,0.18),c(0,.35)), col="red")
par(ps=8)
text(0.18,0.35,"t=0.18", col="red", pos=4)

```



```
close.screen(all=TRUE)
```