GSEA

James Adler

9/8/2021

1. Packages and files

Files:

- /projects/bgmp/shared/Bi623/Assignment_GSEA/CompCoag_pouch_multivar.tsv
- /projects/bgmp/shared/Bi623/Assignment_GSEA/pouch_RNAseq.tsv
- /projects/bgmp/shared/Bi623/Assignment_GSEA/pouch_TMM_values.csv
- /projects/bgmp/shared/Bi623/Assignment_GSEA/stickleback_CPM.tsv

```
# load in kegg_pouch
kegg_pouch = read.delim("pouch_RNAseq.tsv", sep="\t", stringsAsFactors=FALSE)

head(kegg_pouch)
```

2. Perform GSEA and KEGG pathways using gage

```
# load data
data(kegg.sets.ko)
data(sigmet.idx.ko)
kegg.sets.ko = kegg.sets.ko[sigmet.idx.ko]
head(kegg.sets.ko, 3)

# check class
class(kegg.sets.ko)
```

What class of object is kegg.sets.ko, and what kind of information does it contain?

Class: List

Contents: Many different pathways and their associated IDs.

```
# important variables
pouch_foldchanges = kegg_pouch$logFC

# assign kegg ids if present
names(pouch_foldchanges) = kegg_pouch$ko_ID

head(pouch_foldchanges)
```

```
# test for enrichment genes with extreme values
pouch_test = gage(pouch_foldchanges, gsets=kegg.sets.ko,
                  same.dir=FALSE)

# look at top entries
lapply(pouch_test, head)
head(pouch_test$greater, 30)

# subset for FDR controlled at 0.1
str(pouch_test$greater)
pouch_test.greater.01 = pouch_test$greater[which(pouch_test$greater[,'q.val'] < 0.1),] # subset functio

head(pouch_test.greater.01)
```

Which KEGG pathways are enriched for genes with exceptional pregnancy-specific gene expression?

- Cytokine-cytokine receptor interaction
- Neuroactive ligand-receptor interaction
- Calcium signaling pathway
- Complement and coagulation cascades

Which two of these are related to the immune system?

- Cytokine-cytokine receptor interaction
- Complement and coagulation cascades

3. Visualize pregnancy fold change magnitudes for all genes in the "coagaulation and complement" KEGG pathway

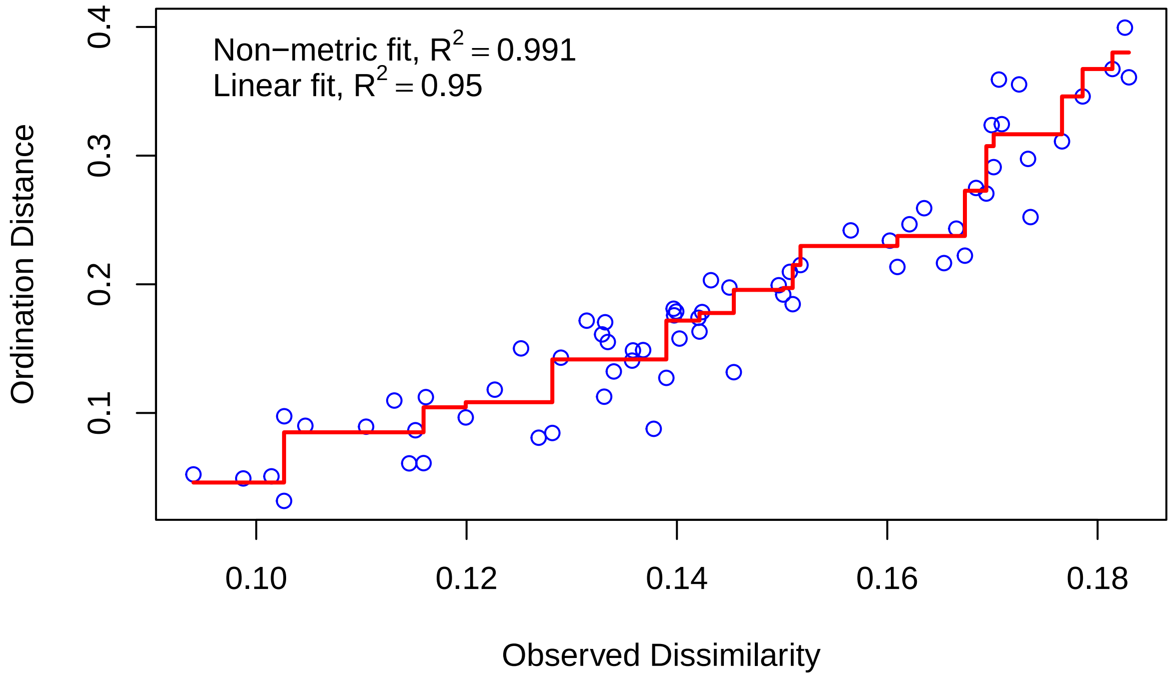
```
# isolate KEGG id of 'coagulation and complement' pathway
pouch_pathways = rownames(pouch_test$greater)[4]
pouch_ids = substr(pouch_pathways, start=1, stop=7)
pouch_ids

# draw pathway with color scale that reflects log2 fold change for each gene
pathview(gene.data=pouch_foldchanges,
         pathway.id=pouch_ids, species="ko", new.signature=FALSE,
         trans.fun = list(gene = NULL, cpd = NULL),
         low = list(gene = "green", cpd = "blue"),
         mid = list(gene = "yellow", cpd = "gray"),
         high = list(gene = "red", cpd = "yellow"), na.col = "transparent")
```

What does the plot tell you about components of this pathway and male pregnancy? The plot provides us with a visualization of the components of this pathway that are differentially expressed during male pregnancy, colored according to the magnitude of the log2 fold change observed in our study.

4. Visualize multi-genic expression patterns in pregnant and non-pregnant pouch tissues using non-metric multidimensional scaling.

What happened to our data frame after using the t() function? The t() function transposes the dataframe, switching the columns and rows so that each row is now represented by a sample and each column is now represented by a gene id.

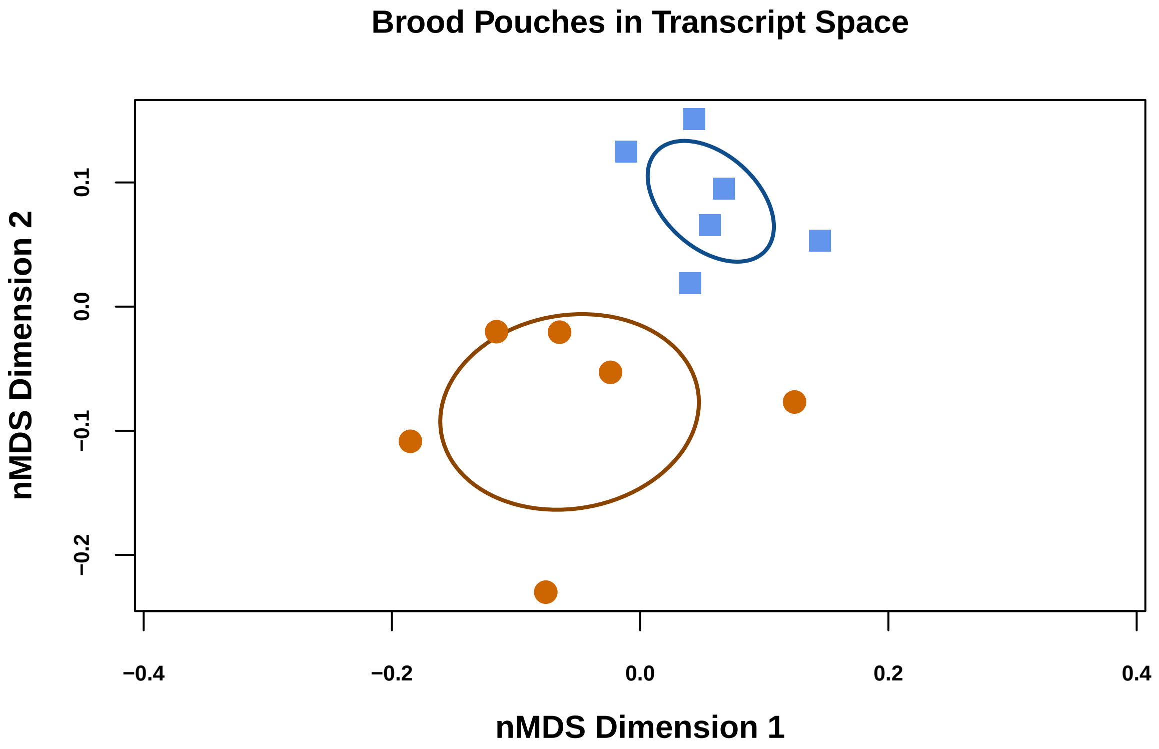
```
# compute dissimilarity matrix
pipe.dis = vegdist(pipe_TMMvals)

# perform multidimensional scaling with 2 dimensions
pipe.mds0 = isoMDS(pipe.dis,k=2)
stressplot(pipe.mds0,pipe.dis)
```



What do you think is meant by "convergence"? Convergence is reaching agreement on the separation of points based on the distances related to many all other points within the matrix.

```
# construct dataframe linking pouch sample ids with pregnancy status
targets = as.data.frame(rownames(pipe_TMMvals))
targets$PregStat = factor(c(rep("preg",6),rep("nonpreg",6)))
colnames(targets) = c("ID", "PregStat")

# define ordination plotting parameters
par(mgp=c(2.5, 1, 0))
preg = as.character(targets$PregStat)
```

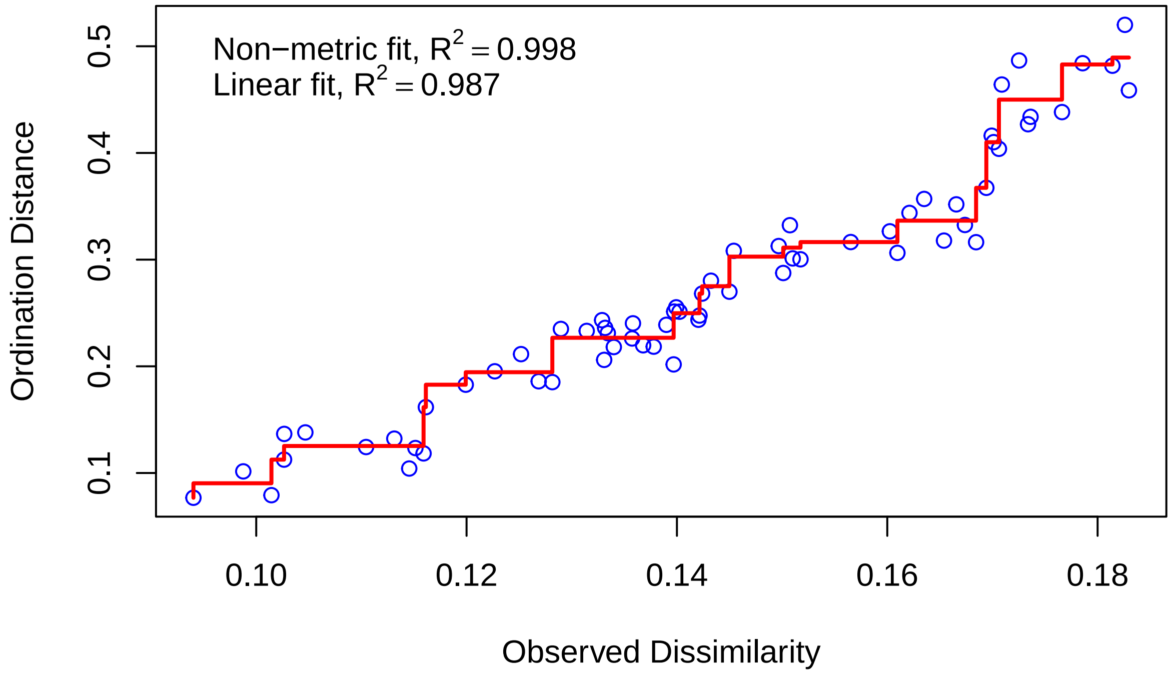
```
fig = ordiplot(
  pipe.mds0,
  main="Brood Pouches in Transcript Space",
  ylab="nMDS Dimension 2",
  xlab="nMDS Dimension 1",
  font.lab=2,
  font.axis=2,
  cex.axis=.7,
  type="none",
  cex.main=1,
  xlim=(c(-.2,0.2))
)

# then add 'confidence ellipses'
ordiellipse(
  pipe.mds0,
  groups=preg,
  label=FALSE,
  lwd=2,
  show.groups=preg[1:6],
  col="darkorange4",
  draw="lines"
)

ordiellipse(
  pipe.mds0,
  groups=preg,
  label=FALSE,
  lwd=2,
  show.groups=preg[7:12],
  col="dodgerblue4",
  draw="lines"
)

#  add individual samples as points
points(
  fig,
  "sites",
  pch=c(rep(19,6),rep(15,6)),
  col=c(rep("darkorange3",6),rep("cornflowerblue",6)),
  cex=1.5
)
```



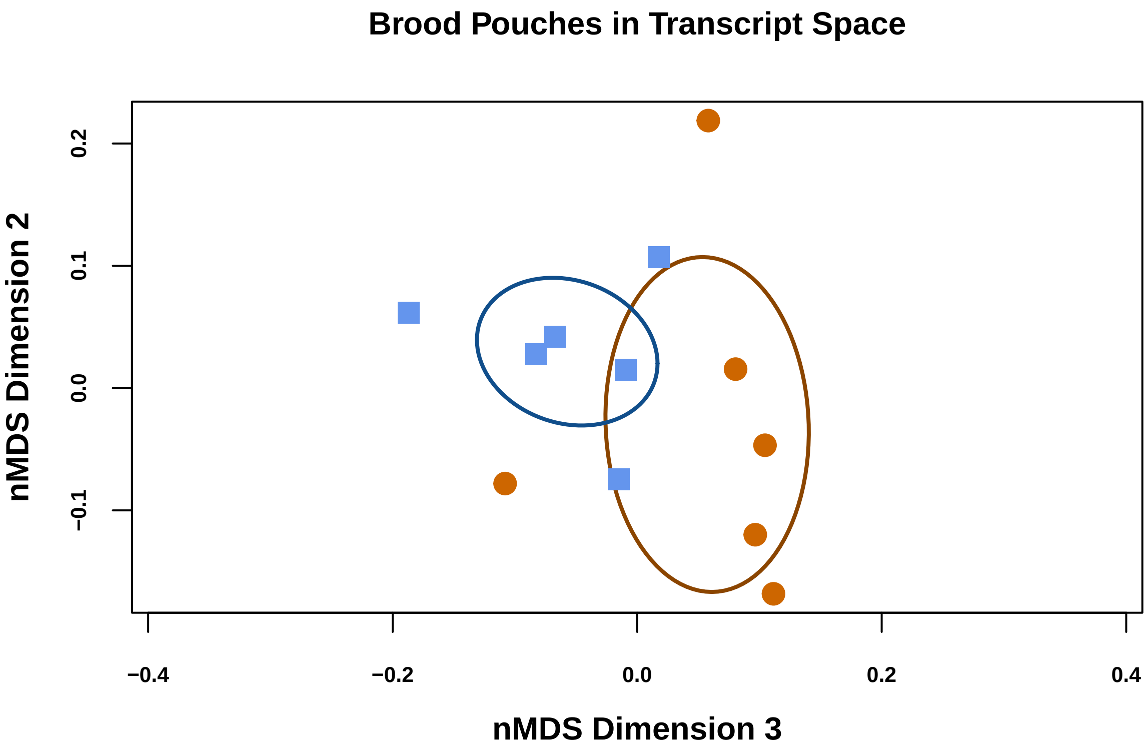
What does this plot tell you about the brood pouch transcriptomes profiled in this study? This plot tells us that the brood pouch transcriptomes profiled in this study group nicely dependent upon the status of male pregnancy.

Repeat the ordination with 3 nMDS dimensions

```
# perform multidimensional scaling with 3 dimensions
pipe.mds0 = isoMDS(pipe.dis,k=3)
stressplot(pipe.mds0,pipe.dis)
```



```
# construct dataframe linking pouch sample ids with pregnancy status
targets = as.data.frame(rownames(pipe_TMMvals))
targets$PregStat = factor(c(rep("preg",6),rep("nonpreg",6)))
colnames(targets) = c("ID", "PregStat")

# define ordination plotting parameters
par(mgp=c(2.5, 1, 0))
preg = as.character(targets$PregStat)

fig = ordiplot(
  pipe.mds0$points[,2:3],
  main="Brood Pouches in Transcript Space",
  ylab="nMDS Dimension 2",
  xlab="nMDS Dimension 3",
  font.lab=2,
  font.axis=2,
  cex.axis=.7,
  type="none",
  cex.main=1,
  xlim=(c(-.2,0.2))
)

# then add 'confidence ellipses'
ordiellipse(
  pipe.mds0$points[,2:3],
  groups=preg,
  label=FALSE,
  lwd=2,
  show.groups=preg[1:6],
  col="darkorange4",
  draw="lines"
)

ordiellipse(
  pipe.mds0$points[,2:3],
  groups=preg,
  label=FALSE,
  lwd=2,
  show.groups=preg[7:12],
  col="dodgerblue4",
  draw="lines"
)

#  add individual samples as points
points(
  fig,
  "sites",
  pch=c(rep(19,6),rep(15,6)),
  col=c(rep("darkorange3",6),rep("cornflowerblue",6)),
  cex=1.5
)
```



5. Permutational Multivariate Analysis of Variance (perMANOVA)

```
# run perMANOVA
otu.env = targets
adonis(pipe.dis ~ PregStat, otu.env, perm=999)
```

Based on output of adonis(), do we see a significant effect of pregnancy status? Yes, the reported p-value is 0.003 which would be considered significant if we use the typical p-value of 0.05.

6. Constructing a heatmap with clustering dendrograms for Coagulation and Complement Cascade KEGG pathway genes

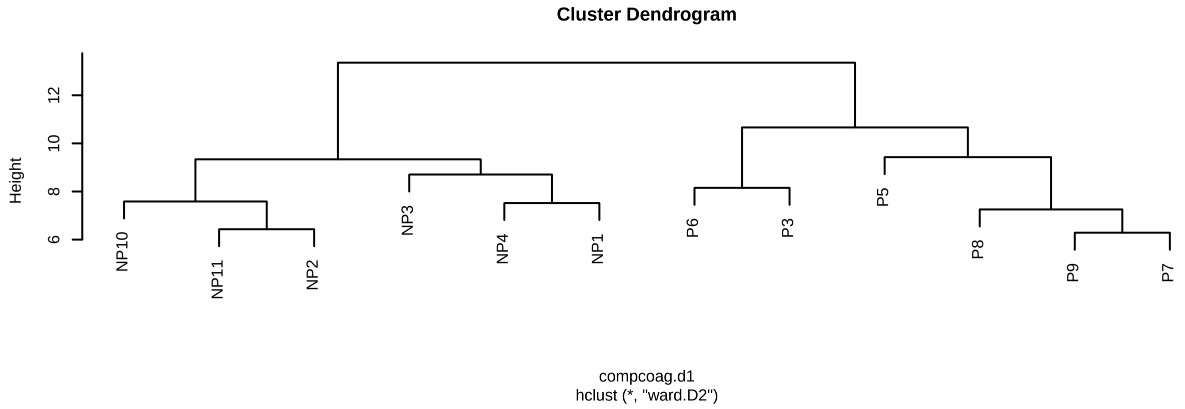
```
# read in expression data for pipefish genes that mapped to KEGG 'coagulation and complement cascade' p
pouch_compcoag = read.delim(
  "CompCoag_pouch_multivar.tsv",
  sep="\t",
  row.names=1,
  header=F
)

# define colnames
colnames(pouch_compcoag) = c(
  "KO",
  "name",
  "P9",
  "P8",
  "P7",
  "P6",
  "P5",
  "P3",
  "NP11",
  "NP10",
  "NP4",
  "NP3",
  "NP2",
  "NP1"
)

# define vector of gene names
names_compcoag = pouch_compcoag$name

# reduce pouch_compcoag to CPM values
pouch_compcoag = pouch_compcoag[,!names(pouch_compcoag) %in% c("KO","name")]

# log2 transform and add 0.01 to each value
pouch_compcoag = log2(pouch_compcoag + 0.01)
```

Why do we add 0.01 to all values? We add 0.01 to all values to remove any 0 values since log2(0) is equal to -Inf. The log2(0.01) is instead -6.643856, which is a more manageable number for downstream visualization.

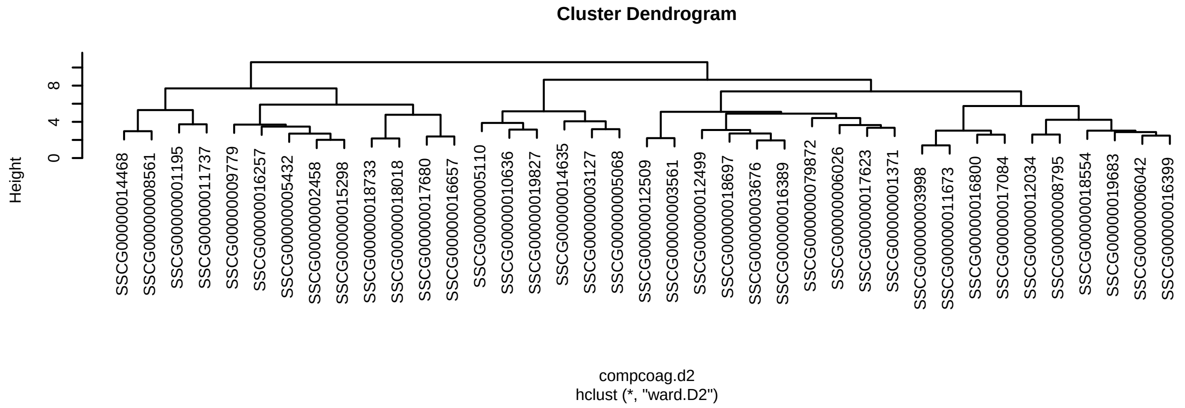
```r
# mean-center and range data (mean=0, sd=1)
pouch_compcoag.n = scale(t(pouch_compcoag))
pouch_compcoag.tn = t(pouch_compcoag.n)

class(pouch_compcoag.tn)
```

What class of object are we dealing with now? Now we are dealing with a "matrix" or "array".

```r
# calculate multivariate dissimilarity for all sample pairs, using Euclidean Distance
compcoag.d1 = dist(
  pouch_compcoag.n,
  method="euclidean",
  diag=FALSE,
  upper=FALSE
)


max(round(compcoag.d1,3))
round(compcoag.d1,3)

# caluclate multivariate dissimilarity for all gene pairs
compcoag.d2 = dist(
  pouch_compcoag.tn,
  method="euclidean",
  diag=FALSE,
  upper=TRUE
)

# cluster samples, then genes, using Ward linkage clustering
compcoag.c1 = hclust(
  compcoag.d1,
  method="ward.D2",
  members=NULL
)

compcoag.c2 = hclust(
  compcoag.d2,
  method="ward.D2",
  members=NULL
)

# take a look at dendrograms based on the clustering
par(mfrow=c(2,1),cex=0.5)
plot(compcoag.c1)
plot(compcoag.c2)
```



Cluster Dendrogram

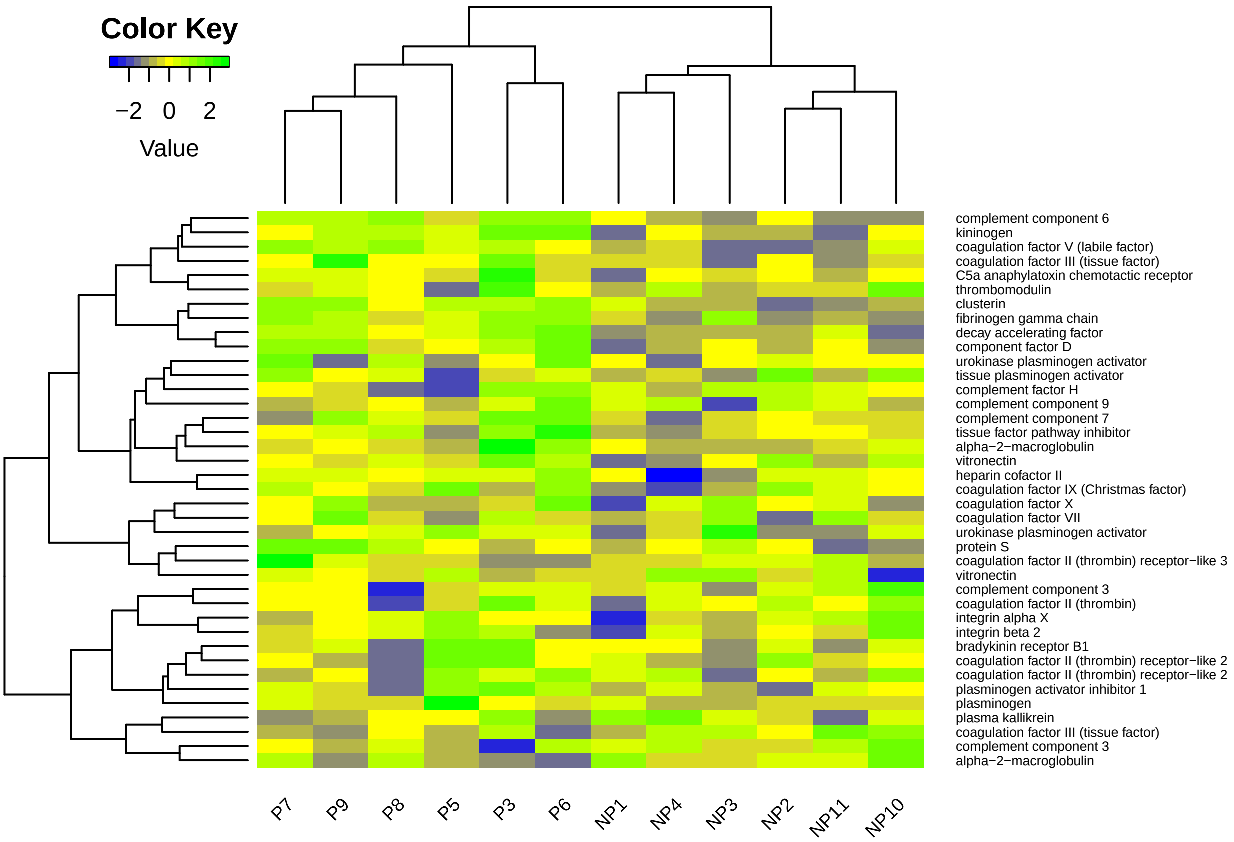


Cluster Dendrogram

Which two samples are the most dissimilar based on Euclidean Distance? NP1 and P3

What does the sample dendrogram tell us about pregnant (P) and non-pregnant (NP) pouch transcriptomes? The sample dendrogram displays pregnant (P) and non-pregnant (NP) samples form two separate clusters.

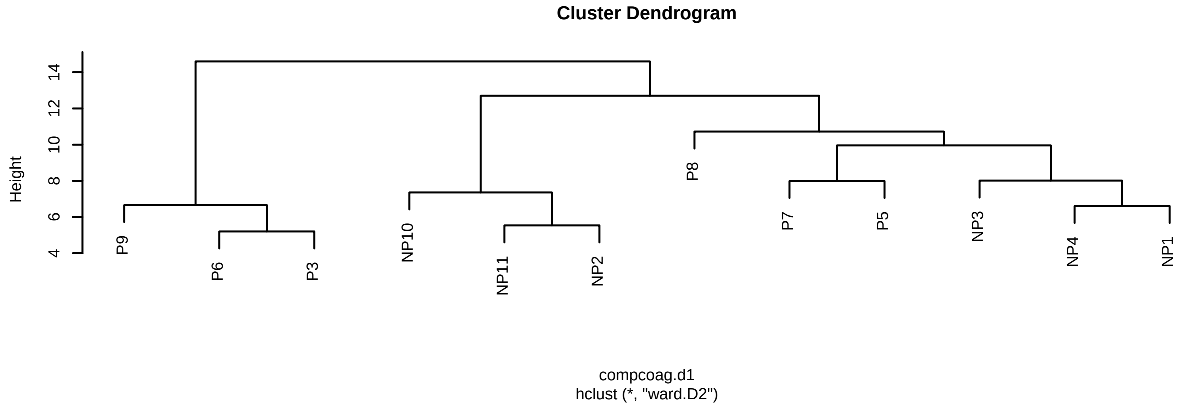
```
# print order of samples in the tree
compcoag.c1$order

# set color scale for the heatmap
compcoag_pal = colorRampPalette(
  c("blue","yellow","green"),
  # n=299
)

# plot the heatmap
heatmap.2(
  pouch_compcoag.tn,
  Colv=rotate(as.dendrogram(compcoag.c1),order=c(12,11,10,9,8,7,6,5,4,3,2,1)),
  Rowv=as.dendrogram(compcoag.c2),
  labRow=names_compcoag,
  density.info="none",
  trace="none",
  scale="none",
  col=compcoag_pal,
  cexRow=0.5,
  cexCol=0.75,
  margins=c(3,10),
  lwid=c(0.8,3),
  # lhei=c(0.8,3),
  srtCol=45,
  adjCol=c(1,1),
  keysize=1.3
)
```

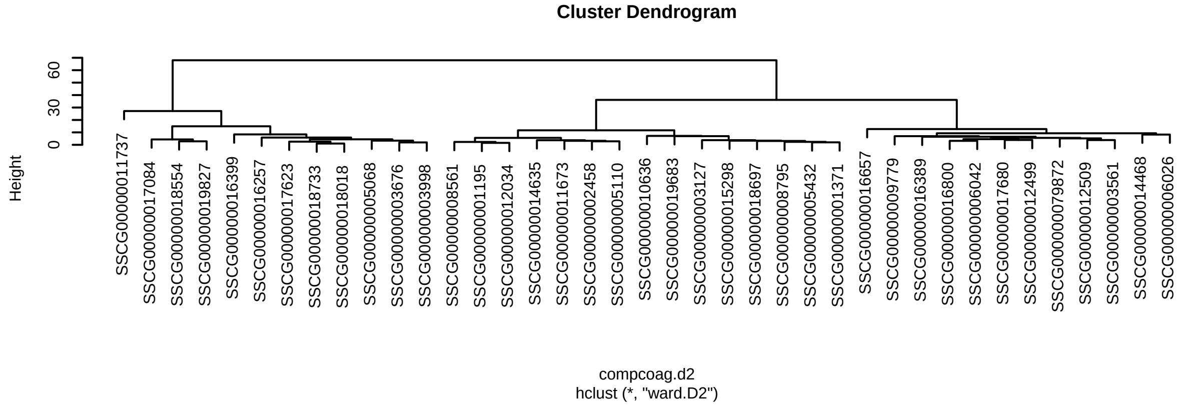


Are there any groups of genes that differentiate between pregnant and non-pregnant males particularly well? If so, name those genes? The top quarter(ish) of the plot appears to cluster fairly consistently between the two groups. These genes include the top branch at the point that there are 5 branches on the left side of the heatmap. These genes include complement component 6, kininogen, coagulation factor V (labile factor), coagulation factor III (tissue factor), C5a anaphylatoxin chemotactic receptor, thrombomodulin, clusterin, fibinogen gamma chain, decay accelerating factor, and component factor D.

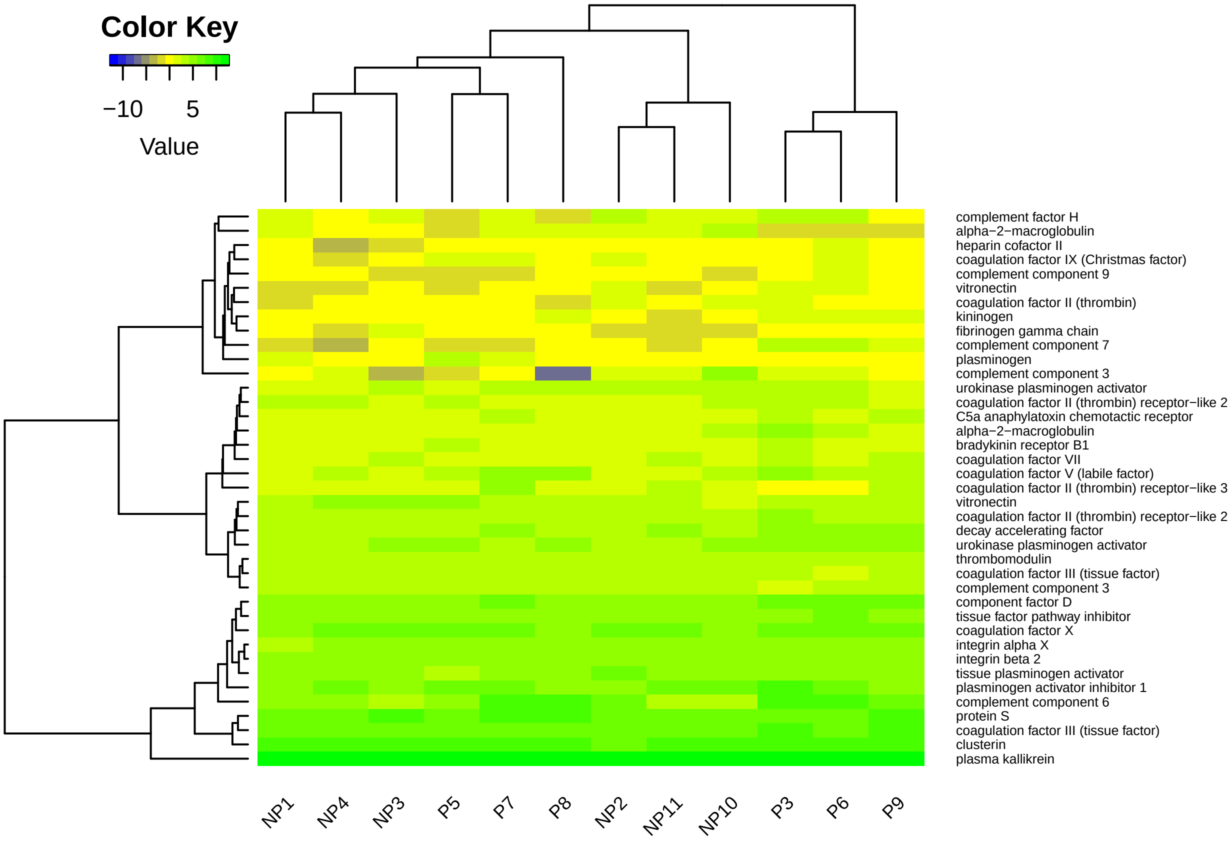
```
pouch_compcoag.n = t(pouch_compcoag)
pouch_compcoag.tn = t(pouch_compcoag.n)

# calculate multivariate dissimilarity for all sample pairs, using Euclidean Distance
compcoag.d1 = dist(
  pouch_compcoag.n,
  method="euclidean",
  diag=FALSE,
  upper=FALSE
)
```

```
max(round(compcoag.d1,3))
round(compcoag.d1,3)

# caluclate multivariate dissimilarity for all gene pairs
compcoag.d2 = dist(
  pouch_compcoag.tn,
  method="euclidean",
  diag=FALSE,
  upper=TRUE
)

# cluster samples, then genes, using Ward linkage clustering
compcoag.c1 = hclust(
  compcoag.d1,
  method="ward.D2",
  members=NULL
)

compcoag.c2 = hclust(
  compcoag.d2,
  method="ward.D2",
  members=NULL
)

# take a look at dendrograms based on the clustering
par(mfrow=c(2,1),cex=0.5)
plot(compcoag.c1)
plot(compcoag.c2)
```





```
# print order of samples in the tree
compcoag.c1$order

# set color scale for the heatmap
compcoag_pal = colorRampPalette(
  c("blue","yellow","green"),
  # n=299
)

# plot the heatmap
heatmap.2(
  pouch_compcoag.tn,
  Colv=rotate(as.dendrogram(compcoag.c1),order=c(12,11,10,9,8,7,6,5,4,3,2,1)),
  Rowv=as.dendrogram(compcoag.c2),
  labRow=names_compcoag,
  density.info="none",
  trace="none",
  scale="none",
  col=compcoag_pal,
  cexRow=0.5,
  cexCol=0.75,
  margins=c(3,10),
  lwid=c(0.8,3),
  # lhei=c(0.8,3),
  srtCol=45,
  adjCol=c(1,1),
  keysize=1.3
)
```



Does your heatmap look any different? If so, explain why this might be. Yes, the heatmap and dendrogram are very different. The NP and P samples are no longer clustered together. The range of values in the heatmap is much narrower. This is likely a result of major skew in the original data. The scale() function gets rid of tails that exist in the distribution.

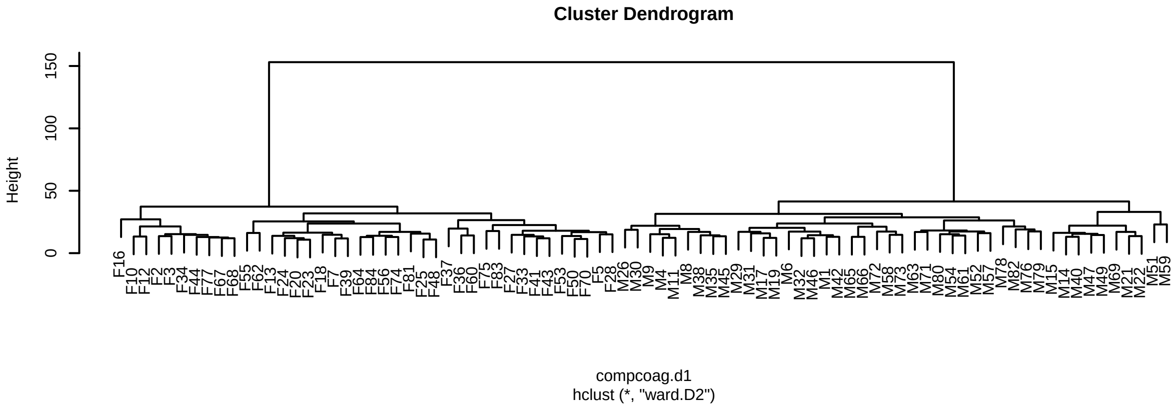
7. Construct heatmaps with clustering dendrograms for stickleback gene espression data

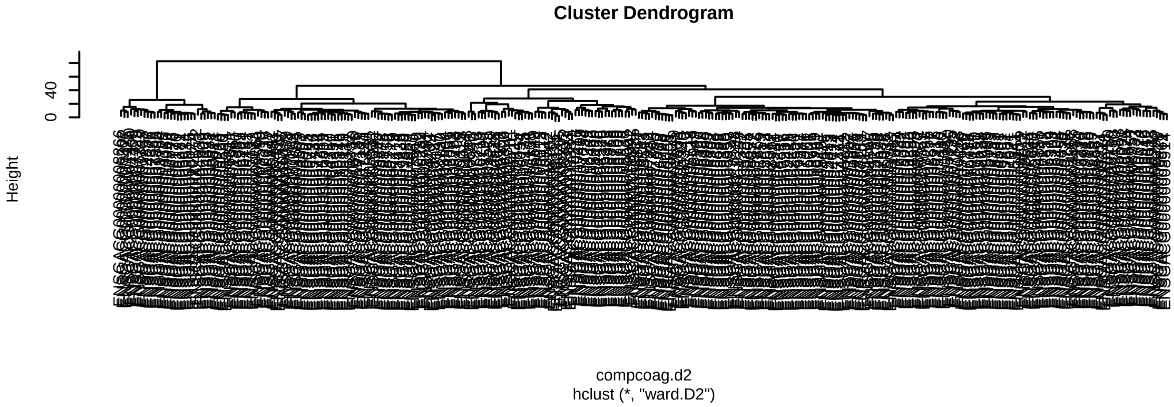
```
# load stickleback data
stick_TMMvals = read.csv("stickleback_CPM.tsv",sep="\t",head=TRUE,row.names=1,stringsAsFactors=FALSE)
dim(stick_TMMvals)
head(stick_TMMvals)

stick_TMMvals = na.omit(stick_TMMvals)
dim(stick_TMMvals)
head(stick_TMMvals)

stick_TMMvals[,3:ncol(stick_TMMvals)] = lapply(stick_TMMvals[,3:ncol(stick_TMMvals)],as.numeric)
head(stick_TMMvals)
```

```
# # transpose so genes are columns
# stick_TMMvals = t(stick_TMMvals)
# dim(stick_TMMvals)
```

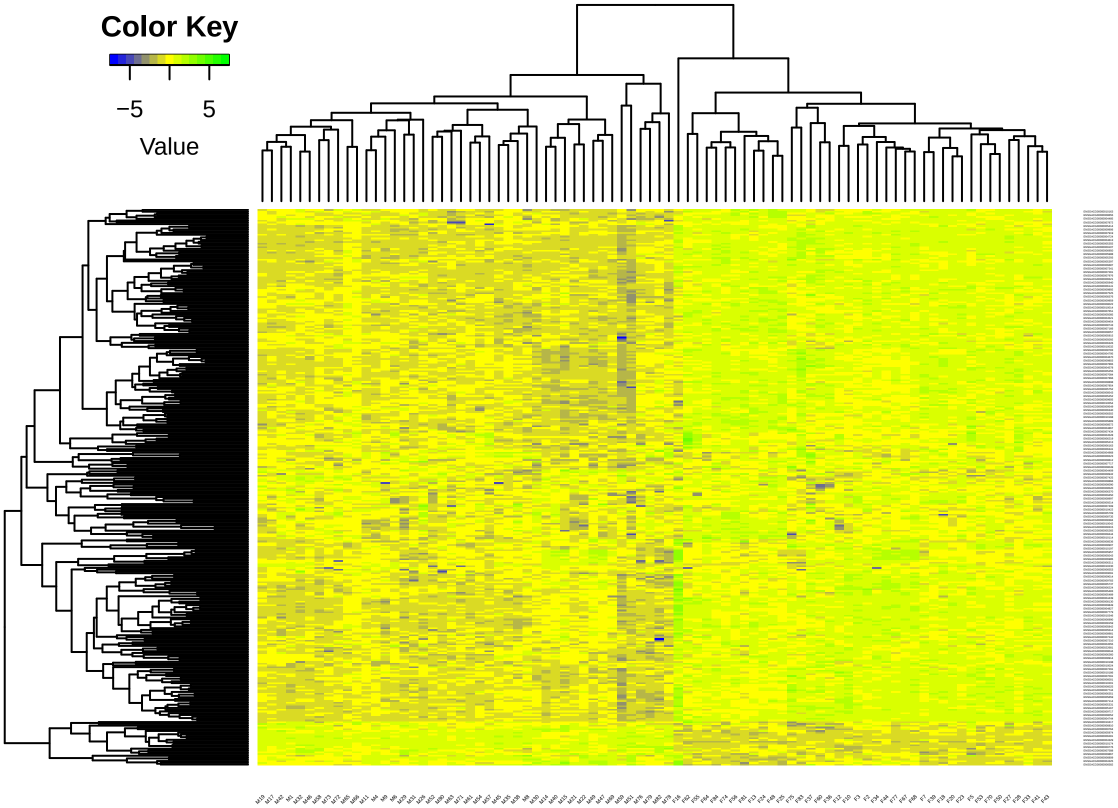
Generate stickleback heat map with GroupXIX and start position between 6000000-120000000

```
# subset to groupXIX and between 6000000-12000000 gene start position
stick_TMMvals.groupxix = stick_TMMvals[which(stick_TMMvals[,"Genome_Loc"] == "groupXIX"),]
stick_TMMvals.groupxix.6.12 = stick_TMMvals.groupxix[which(stick_TMMvals.groupxix[,"Gene_Start_Pos"] > 6

# reduce pouch_compcoag to CPM values
stick_TMMvals.groupxix.6.12 = stick_TMMvals.groupxix.6.12[,!names(stick_TMMvals.groupxix.6.12) %in% c("

# log2 transform and add 0.01 to each value
stick_TMMvals.groupxix.6.12 = log2(stick_TMMvals.groupxix.6.12 + 0.01)
head(stick_TMMvals.groupxix.6.12)

# mean-center and range data (mean=0, sd=1)
stick_TMMvals.groupxix.6.12.n = scale(t(stick_TMMvals.groupxix.6.12))
stick_TMMvals.groupxix.6.12.tn = t(stick_TMMvals.groupxix.6.12.n)

head(stick_TMMvals.groupxix.6.12.tn)

# calculate multivariate dissimilarity for all sample pairs, using Euclidean Distance
compcoag.d1 = dist(
  stick_TMMvals.groupxix.6.12.n,
  method="euclidean",
  diag=FALSE,
  upper=FALSE
)

max(round(compcoag.d1,3))
round(compcoag.d1,3)

# caluclate multivariate dissimilarity for all gene pairs
compcoag.d2 = dist(
  stick_TMMvals.groupxix.6.12.tn,
  method="euclidean",
  diag=FALSE,
  upper=TRUE
)

# cluster samples, then genes, using Ward linkage clustering
compcoag.c1 = hclust(
  compcoag.d1,
  method="ward.D2",
  members=NULL
)

compcoag.c2 = hclust(
  compcoag.d2,
  method="ward.D2",
  members=NULL
)

# take a look at dendrograms based on the clustering
par(mfrow=c(2,1),cex=0.5)
plot(compcoag.c1)
plot(compcoag.c2)
```





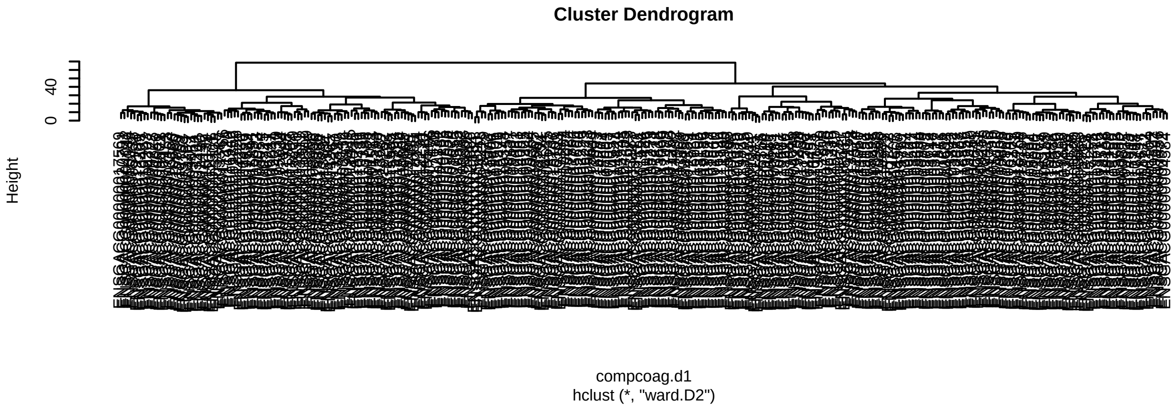
```
# print order of samples in the tree
compcoag.c1$order

# set color scale for the heatmap
compcoag_pal = colorRampPalette(
  c("blue","yellow","green"),
  # n=299
)

# plot the heatmap
heatmap.2(
  stick_TMMvals.groupxix.6.12.tn,
  # Colv=rotate(as.dendrogram(compcoag.c1),order=c(12,11,10,9,8,7,6,5,4,3,2,1)),
  # Rowv=as.dendrogram(compcoag.c2),
  # labRow=names_compcoag,
  density.info="none",
  trace="none",
  scale="none",
  col=compcoag_pal,
  cexRow=0.15,
  cexCol=0.25,
  margins=c(3,6),
  lwid=c(0.8,3),
  # lhei=c(0.8,3),
  srtCol=45,
  adjCol=c(1,1),
  keysize=1.3
)
```

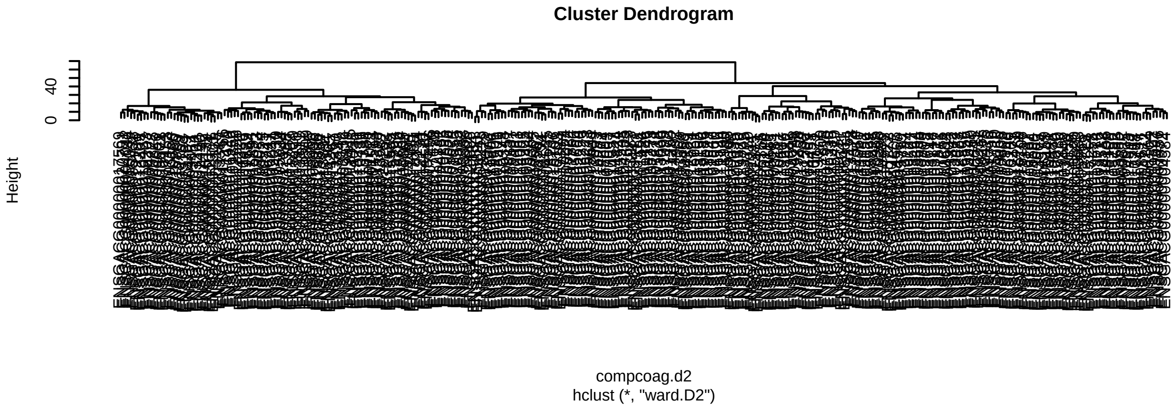


Run stickleback with random subset of genes

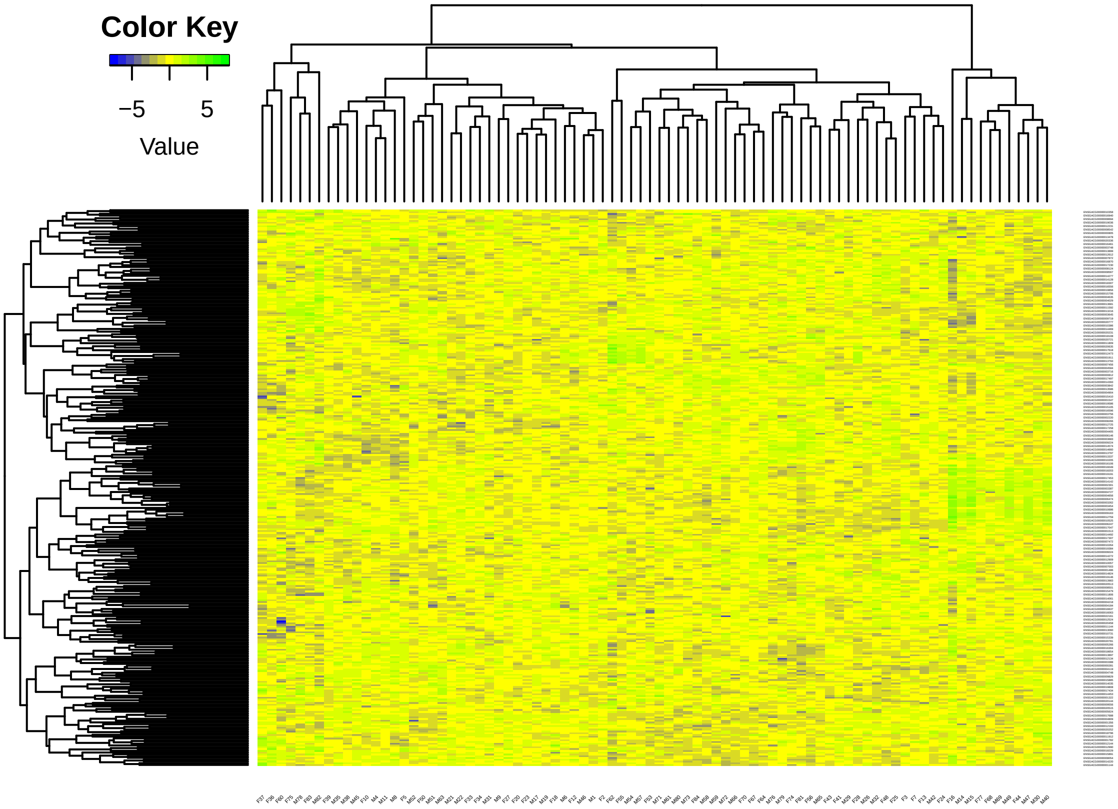
```
# subsample random set of genes (equal to the number of genes in first map)
stick_TMMvals.random = stick_TMMvals[sample(rownames(stick_TMMvals),nrow(stick_TMMvals.groupxix.6.12)),

# reduce pouch_compcoag to CPM values
stick_TMMvals.random = stick_TMMvals.random[,3:ncol(stick_TMMvals.random)]

# log2 transform and add 0.01 to each value
```

```r
stick_TMMvals.random = log2(stick_TMMvals.random + 0.01)

# mean-center and range data (mean=0, sd=1)
stick_TMMvals.random = scale(t(stick_TMMvals.random))
stick_TMMvals.random = t(stick_TMMvals.random)

head(stick_TMMvals.random)

# calculate multivariate dissimilarity for all sample pairs, using Euclidean Distance
compcoag.d1 = dist(
  stick_TMMvals.random,
  method="euclidean",
  diag=FALSE,
  upper=FALSE
)

max(round(compcoag.d1,3))
round(compcoag.d1,3)

# caluclate multivariate dissimilarity for all gene pairs
compcoag.d2 = dist(
  stick_TMMvals.random,
  method="euclidean",
  diag=FALSE,
  upper=TRUE
)

# cluster samples, then genes, using Ward linkage clustering
compcoag.c1 = hclust(
  compcoag.d1,
  method="ward.D2",
  members=NULL
)

compcoag.c2 = hclust(
  compcoag.d2,
  method="ward.D2",
  members=NULL
)

# take a look at dendrograms based on the clustering
par(mfrow=c(2,1),cex=0.5)
plot(compcoag.c1)
plot(compcoag.c2)
```





```
# print order of samples in the tree
compcoag.c1$order

# set color scale for the heatmap
compcoag_pal = colorRampPalette(
  c("blue","yellow","green"),
  # n=299
)

# plot the heatmap
heatmap.2(
  stick_TMMvals.random,
  # Colv=rotate(as.dendrogram(compcoag.c1),order=c(12,11,10,9,8,7,6,5,4,3,2,1)),
  # Rowv=as.dendrogram(compcoag.c2),
  # labRow=names_compcoag,
  density.info="none",
  trace="none",
  scale="none",
  col=compcoag_pal,
  cexRow=0.15,
  cexCol=0.25,
  margins=c(3,6),
  lwid=c(0.8,3),
  # lhei=c(0.8,3),
  srtCol=45,
  adjCol=c(1,1),
  keysize=1.3
)
```



Do you notice anything different about the clustering patterns for males and females between the two subsets? Do a quick literature search to find out why we might expect a difference given the biology of stickleback chromosome 19, and interpret what may be going on, especially regarding the clusters of genes that yield different sex-specific expression patterns in the groupXIX heatmap. In the non-random subset, the male vs. female samples clustered together in two separate groups by sex. The genes expressed in these two separate groups appear very similar to each other, also clustering according to sample sex. However, in the random subset, the male vs. female samples are interspersed and there is no clear separation of samples based on sex. The heat map does not have clear clusters gene expression levels. Chromosome 19 is the sex-determination chromsome in stickleback, so it makes sense that we observe clustering of gene expression levels when subsetting for groupXIX. Alternatively, the random subset returns an array of random expression levels among the random genes that are not necessarily associated with sex, and therefore we do not see the same level of clustering as in groupXIX subset.