Find DE genes from SL4 and SL5 comparing mutant and wildtype genotypes

Ann Loraine

2023-10-22

Prelude - define variables

```
assemblies=c("SL4","SL5")
Q = 0.01
lfcThreshold=1
outfname_prefix="results/MvW"
```

Introduction

We observed in prior work that:

- clustering analysis showed that gene expression in VF36 and F3H lines were similar, and both were very different from *are* (anthocyanin-reduced mutant)
- many more genes were differentially expressed in the *are* mutant in response to temperature than in the other two
- the greatest number of gene expression changes in response to temperature occurred in the 75 minutes time point, the longest duration of heat stress tested in this experiment

We also know from experimental work that pollen tubes in *are* are less able to achieve fertilization.

These observations raise new questions:

- How does gene expression differ between *are* and the other two lines, with and without heat stess?

To answer this question, we will investigate gene expression differences between VF36 and *are* experiencing the same temperature, across and within treatment duration.

We will only consider VF36 here because we know from dimensionality reduction analyses that F3H and VF36 are very similar.

Goal: Find and interpret how gene expression in *are* differs from gene expression in VF36, irrespective of temperature.

Results

Load code:

```
source("Common.R")
```

Read counts data for assemblies SL4, SL5:

```
dfs = list()
for (assembly in assemblies) {
  dfs[[assembly]]=getCounts(assembly=assembly,
                            keep_description = T)
}
```

Fit a model that compares *are* to VF36 at 28 degrees C, no heat stress, including treatment durations as a variable:

```
ddss = list()
for (assembly in assemblies) {
  d = dfs[[assembly]]
  desc_index = which(names(d)=="description")
  d = d[,-desc_index]
  toks = strsplit(names(d),"\\.")
  genotype=sapply(toks,function(x){x[[1]]})
  temperature=sapply(toks,function(x){x[[2]]})
  time=sapply(toks,function(x){x[[3]]})
  v = genotype%in%c("A","V")&temperature==28
  d = d[,v]
  coldata=data.frame(genotype=factor(genotype[v],levels=c("V","A")),
                     time=factor(time[v]))
  row.names(coldata)=names(d)
  cts = round(as.matrix(d))
  dds = DESeqDataSetFromMatrix(countData=cts,
                               colData=coldata,
                               design=~time + genotype)
  featureData = data.frame(gene=rownames(cts))
  mcols(dds) = DataFrame(mcols(dds), featureData)
  dds = DESeq(dds, minReplicatesForReplace=Inf)
  ddss[[assembly]]=dds
}
```

The above code creates a linear model that captures/estimates the contribution of treatment duration and genotype to gene expression levels, per gene.

Collect results:

```
rss=list()
vs = list()
for (assembly in assemblies) {
  dds=ddss[[assembly]]
  rs = results(dds,alpha=Q,contrast=c("genotype","A","V"))
  rs = rs[!is.na(rs$padj)&!is.na(rs$log2FoldChange),]
  rs=cbind(gene=row.names(rs),group1="A.28",group2="V.28",rs)
  rs$description = dfs[[assembly]][rs$gene,"description"]
  o = order(rs$pvalue)
  rs = rs[o,]
  rss[[assembly]] = data.frame(rs)
  vs[[assembly]] = rss[[assembly]]$padj<=Q &
    rss[[assembly]]$log2FoldChange>=lfcThreshold
}
```

The previous code chunk collected results from evaluating the null hypothesis that the difference between *are* and VF36 gene expression, added up over all time points, was non-zero.

The above code identified 978 and 1,007 genes from assemblies SL4 and SL5, respectively, where the adjusted p value was less than or equal to 0.01 and the log2 fold-change, calculated as the logarithm base 2 of the average gene expression of *are* divided by average gene expression of VF36, was greater than or equal to 1.

Next, we test the data again, this time looking for genes that were differentially expressed in *are* versus VF36 samples exposed to the same temperature for the same amount of time.

To start, define a function that compares samples and collects results:

```
compareTwoGroups = function(counts,group1_name,group2_name) {
  group1_indexes=grep(group1_name,names(counts))
  group2_indexes=grep(group2_name,names(counts))
  indexes=c(group1_indexes,group2_indexes)
  condition = factor(c(rep(group1_name,
                           length(group1_indexes)),
                       rep(group2_name,length(group2_indexes))))
  m = round(as.matrix(counts[,indexes]))
  dds = DESeqDataSetFromMatrix(m,
                               DataFrame(condition),
                               ~ condition)
  featureData = data.frame(gene=rownames(m))
  mcols(dds) = DataFrame(mcols(dds),featureData)
  dds = DESeq(dds, minReplicatesForReplace=Inf)
  return(dds)
}
```

Create a data frame that defines the comparisons we will do, which include comparing *are* to VF36, same temperature and same treatment duration:

```
A = c("A.28.15","A.28.30","A.28.45","A.28.75")
A = c(A,str_replace_all(A,"28","34"))
V = str_replace_all(A,"A","V")
comps = data.frame(group1=V,group2=A)
```

The previous code chunk defined 8 comparisons, one per treatment duration, per treatment.

Perform the comparisons, using the previously defined function, and collect the results:

```
for (assembly in assemblies) {
  counts = dfs[[assembly]]
  for (iter in 1:nrow(comps)) {
    group_1 = comps[iter,1]
    group_2 = comps[iter,2]
    dds = compareTwoGroups(counts,group_1,group_2)
    rs = results(dds,alpha=Q)
    rs = rs[!is.na(rs$padj),]
    rs=cbind(gene=row.names(rs),group1=group_1,
             group2=group_2,rs)
    rs$description = counts[rs$gene,"description"]
    o = order(rs$pvalue)
    rs = data.frame(rs[o,])
    rss[[assembly]] = rbind(rss[[assembly]],rs)
  }
}
```

Summarize the results from each comparison:

```
for (assembly in assemblies) {
  comps[,assembly]=0
  for (iter in 1:nrow(comps)) {
    group_1 = comps[iter,1]
    group_2 = comps[iter,2]
    rs = rss[[assembly]]
    rs = rs[rs$group1 == group_1 & rs$group2 == group_2,]
    v = rs$padj<=Q & rs$log2FoldChange>=lfcThreshold
    num_genes=sum(v)
    comps[iter,assembly]=num_genes
  }
}
```

Summarize results:

```
comps
```

```
##    group1  group2 SL4 SL5
## 1 V.28.15 A.28.15 126 109
## 2 V.28.30 A.28.30 128 136
## 3 V.28.45 A.28.45 104 118
## 4 V.28.75 A.28.75 111 117
## 5 V.34.15 A.34.15 130 133
## 6 V.34.30 A.34.30 146 157
## 7 V.34.45 A.34.45 108 121
## 8 V.34.75 A.34.75 135 137
```

The preceding table reports the number of genes per comparison with adjusted p value less than or equal to 0.01 and log2(fold-change) greater than or equal to 1.

Write the results to files, one file per assembly:

```
for (assembly in assemblies) {
  fname = paste0(outfname_prefix,"-",assembly,".txt")
  rs = rss[[assembly]]
  rs = rs[,c("gene","group1","group2","baseMean",
             "log2FoldChange","lfcSE","padj",
             "pvalue","stat","description")]
  for (i in 4:9) {
    rs[,i]=signif(rs[,i],3)
  }
  if (assembly == "SL5") {
    SL4_gene_names = getSL4GeneNames(rs$description)
    rs$SL4 = SL4_gene_names
  }
  write.table(rs,fname,row.names = F,col.names = T,
              quote=F,sep="\t")
}
```

Files were created created named results/MvW-[assembly].txt with all results, with no cutoffs for reporting results.

All numeric values were rounded to three significant digits.

Explanation of columns:

- gene - gene measured (from SL4 or SL5)

- group 1 - fold-change denominator; control group
- group 2 - fold-change numerator; treatment group
- baseMean - mean across samples
- log2FoldChange - log2(group 2 average/group 1 average)
- lfcsE - log2FoldChange standard error
- padj - false discovery rate; adjusted p-value computed using method of Benjamini and Hochberg
- stat - test statistic used to assess significance
- description - gene description
- SL4 - SL5 assembly results only; putative SL4 (June 2019 assembly release) gene counterpart

The first rows of the results files contain the results from comparing *are* to VF36, including duration times points, together.

Discussion

In nearly all comparisons, more genes were identified in the SL5 annotations, perhaps reflecting improvements in the annotation for this newer assembly.

As expected from the previously observed clustering results, many genes were differentially expressed between the two genotypes.

Conclusions

- Genes differently expressed between *are* and VF36 were identified.
- Files were written to the "results" directory, one per assembly.

Session Info

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Big Sur 11.7.9
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
```

```
##  [1] EnhancedVolcano_1.18.0      ggrepel_0.9.3
##  [3] ggplot2_3.4.3               DESeq2_1.40.2
##  [5] SummarizedExperiment_1.30.2 Biobase_2.60.0
##  [7] MatrixGenerics_1.12.3       matrixStats_1.0.0
##  [9] GenomicRanges_1.52.0        GenomeInfoDb_1.36.3
## [11] IRanges_2.34.1              S4Vectors_0.38.1
## [13] BiocGenerics_0.46.0         edgeR_3.42.4
## [15] limma_3.56.2                readxl_1.4.3
## [17] readr_2.1.4                 stringr_1.5.0
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.4           xfun_0.40              lattice_0.21-8
##  [4] tzdb_0.4.0             vctrs_0.6.3            tools_4.3.1
##  [7] bitops_1.0-7           generics_0.1.3         parallel_4.3.1
## [10] tibble_3.2.1           fansi_1.0.4            pkgconfig_2.0.3
## [13] Matrix_1.6-1           lifecycle_1.0.3        GenomeInfoDbData_1.2.10
## [16] compiler_4.3.1         munsell_0.5.0          codetools_0.2-19
## [19] htmltools_0.5.6        RCurl_1.98-1.12        yaml_2.3.7
## [22] pillar_1.9.0           crayon_1.5.2           BiocParallel_1.34.2
## [25] DelayedArray_0.26.7    abind_1.4-5            tidyselect_1.2.0
## [28] locfit_1.5-9.8         digest_0.6.33          stringi_1.7.12
## [31] dplyr_1.1.3            fastmap_1.1.1          grid_4.3.1
## [34] colorspace_2.1-0       cli_3.6.1              magrittr_2.0.3
## [37] S4Arrays_1.0.6         utf8_1.2.3             withr_2.5.0
## [40] scales_1.2.1           rmarkdown_2.24         XVector_0.40.0
## [43] cellranger_1.1.0       hms_1.1.3              evaluate_0.21
## [46] knitr_1.44             rlang_1.1.1            Rcpp_1.0.11
## [49] glue_1.6.2             rstudioapi_0.15.0      R6_2.5.1
## [52] zlibbioc_1.46.0
```