GO Enrichment

Run the `render()` function below and everything will be run with report at end.

```
library(rmarkdown)
render("skeleton_GO.Rmd", "pdf_document", output_file = paste(sample1,"_",sample2,"_","GO.pdf",sep=""))
```

Read in YAML guide

```
library(yaml)
yamls <- yaml.load_file("de.yml")
sample1 <- yamls$sample1
sample2 <- yamls$sample2

sample1
```

```
## [1] "wtaother"
```

```
sample2
```

```
## [1] "wtcother"
```

```
library(goseq)
library(GO.db)
```

Setting up the DE table for GO analysis

File Input

Input the output from DE analysis. This is made for a list that includes only the significant genes.

```
sigOnly <- read.table(paste(sample1,"_",sample2,"_DE_sig.txt", sep=""), header = TRUE, fill = TRUE)
head(sigOnly)
```

```
##                   ITAG logFC logCPM    PValue       FDR
## 1 Solyc00g005070.1.1 5.387  7.065 1.184e-16 3.640e-14
## 2 Solyc00g014800.1.1 6.622  7.991 1.547e-22 7.581e-20
## 3 Solyc00g036520.1.1 3.098  3.973 1.275e-05 9.898e-04
## 4 Solyc00g069880.2.1 2.831  3.750 1.865e-04 1.111e-02
## 5 Solyc00g094550.1.1 2.360  4.215 5.883e-04 2.856e-02
## 6 Solyc00g095760.1.1 5.681  5.214 6.736e-14 1.620e-11
##
## 1
## 2
## 3
## 4
## 5
## 6 1-aminocyclopropane-1-carboxylate synthase (AHRD V1 ***- Q96580_SOLLC); contains Interpro domain(s
##         AGI symbol
## 1      <NA>   <NA>
## 2      <NA>   <NA>
## 3      <NA>   <NA>
## 4      <NA>   <NA>
## 5      <NA>   <NA>
## 6 AT3G61510   ACS1
##
## 1
## 2
## 3
## 4
## 5
## 6 Encodes a member of the 1-aminocyclopropane-1-carboxylate (ACC) synthase (S-adenosyl-L-methionine m
##   X..identity alignment.length e.value bit.score percent.query.align
## 1          NA               NA      NA        NA                  NA
## 2          NA               NA      NA        NA                  NA
## 3          NA               NA      NA        NA                  NA
## 4          NA               NA      NA        NA                  NA
## 5          NA               NA      NA        NA                  NA
## 6       58.76              291   2e-95       345               83.53
```

```r
dim(sigOnly)
```

```
## [1] 378  14
```

```r
colnames(sigOnly)
```

```
##  [1] "ITAG"                "logFC"               "logCPM"
##  [4] "PValue"              "FDR"                 "SGN_annotation"
##  [7] "AGI"                 "symbol"              "gene_name"
## [10] "X..identity"         "alignment.length"    "e.value"
## [13] "bit.score"           "percent.query.align"
```

```r
colnames(sigOnly)[1] <- "itag"
```

Subset

First I need to subset the list to up or down regulated, then add a new colum that specififys 1. This column is need to for merging.

```r
upITAG <- subset(sigOnly, logFC > 0, select = c(itag))
upITAG$up <- 1

downITAG <- subset(sigOnly, logFC < 0, select = c(itag))
downITAG$down <- 1

allITAG <- subset(sigOnly, select = c(itag))
allITAG$all <- 1
```

Merge I - with normalized ITAG length gene list

read in guide.

```
geneLength <- read.csv("../normalized_genes_length.csv")
head(geneLength)
```

```
##                 itag length
## 1 Solyc00g005040.2.1    357
## 2 Solyc00g005050.2.1    588
## 3 Solyc00g005060.1.1    273
## 4 Solyc00g005070.1.1     81
## 5 Solyc00g005080.1.1    297
## 6 Solyc00g005150.1.1   1143
```

```
#isolate just the gene list
genes <- subset(geneLength, select = c(itag))
```

First merge each table to geneLength

```
upITAGmerge <- merge(genes, upITAG, by = "itag", all= TRUE)
downITAGmerge <- merge(genes, downITAG, by = "itag", all= TRUE)
allITAGmerge <- merge(genes, allITAG, by = "itag", all= TRUE)
```

Merge II - Merge them all together.

```
matrixGOupdown <- merge(upITAGmerge, downITAGmerge, by = "itag", all = TRUE)
matrixGOupdownall <- merge(matrixGOupdown, allITAG, by = "itag", all = TRUE)
matrixGO <- merge(matrixGOupdownall, geneLength, by = "itag", all = TRUE)
```

Clean Up

```
matrixGO[is.na(matrixGO)] <- 0
head(matrixGO)
```

```
##                 itag up down all length
## 1 Solyc00g005040.2.1  0    0   0    357
## 2 Solyc00g005050.2.1  0    0   0    588
## 3 Solyc00g005060.1.1  0    0   0    273
## 4 Solyc00g005070.1.1  1    0   1     81
## 5 Solyc00g005080.1.1  0    0   0    297
## 6 Solyc00g005150.1.1  0    0   0   1143
```

This is if you want to write out the table of the GO matrix. #write.table(matrixGO, “mydata.txt”, sep=“”̑, quote= FALSE)

GO enrichment

The is the input of the GOslim categories. There are only two columns 1. itag and 2. go

```
pat <- matrixGO
head(pat)
```

```
##                   itag up down all length
## 1 Solyc00g005040.2.1  0    0   0    357
## 2 Solyc00g005050.2.1  0    0   0    588
## 3 Solyc00g005060.1.1  0    0   0    273
## 4 Solyc00g005070.1.1  1    0   1     81
## 5 Solyc00g005080.1.1  0    0   0    297
## 6 Solyc00g005150.1.1  0    0   0   1143
```

```
cate <- read.table("../melted.GOTable.txt",header=TRUE)
head(cate)
```

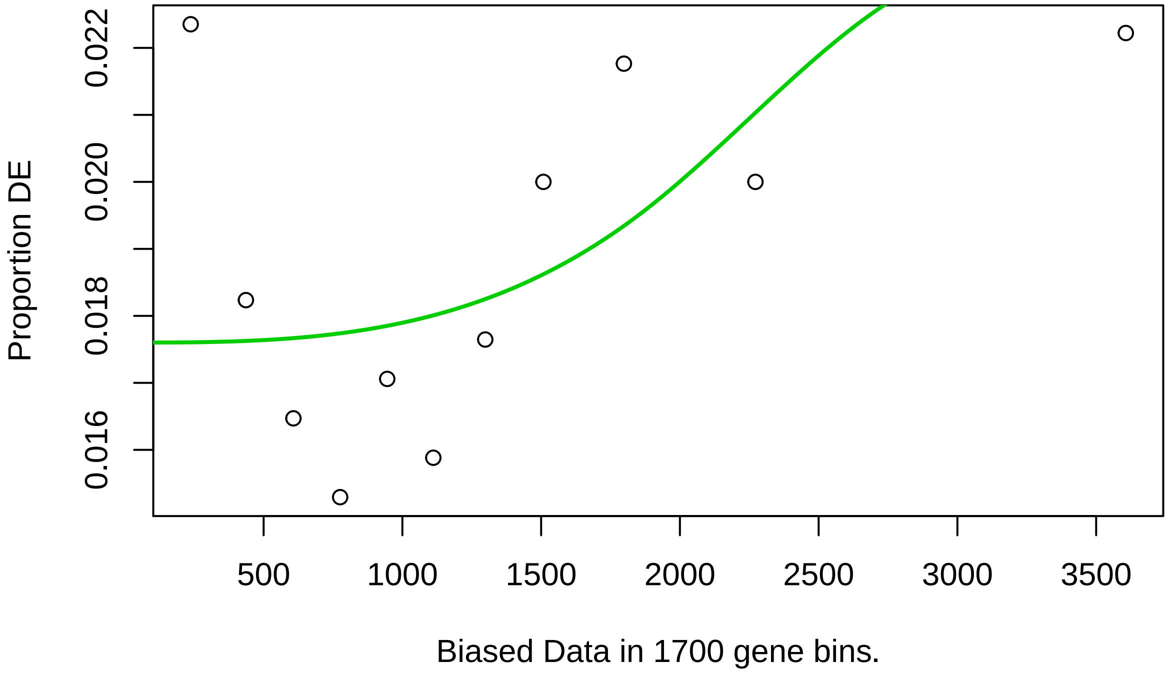
```
##                   itag         go
## 1 Solyc00g005000.2.1 GO:0006508
## 2 Solyc00g005040.2.1 GO:0005774
## 3 Solyc00g005050.2.1 GO:0005829
## 4 Solyc00g005080.1.1 GO:0005524
## 5 Solyc00g005130.1.1 GO:0006508
## 6 Solyc00g005150.1.1 GO:0003676
```

Subseting for GO analysis

Specify the column you are interested in `pat$all` refers to all the DE gene regardless if they are up or down regulated. If you want to specify down regulated, specify `pat$down`. I am going to put this into a loop, where each time the loop goes thought it will perform GO enrichment on all three types of lists of significant genes and them write them to a table.

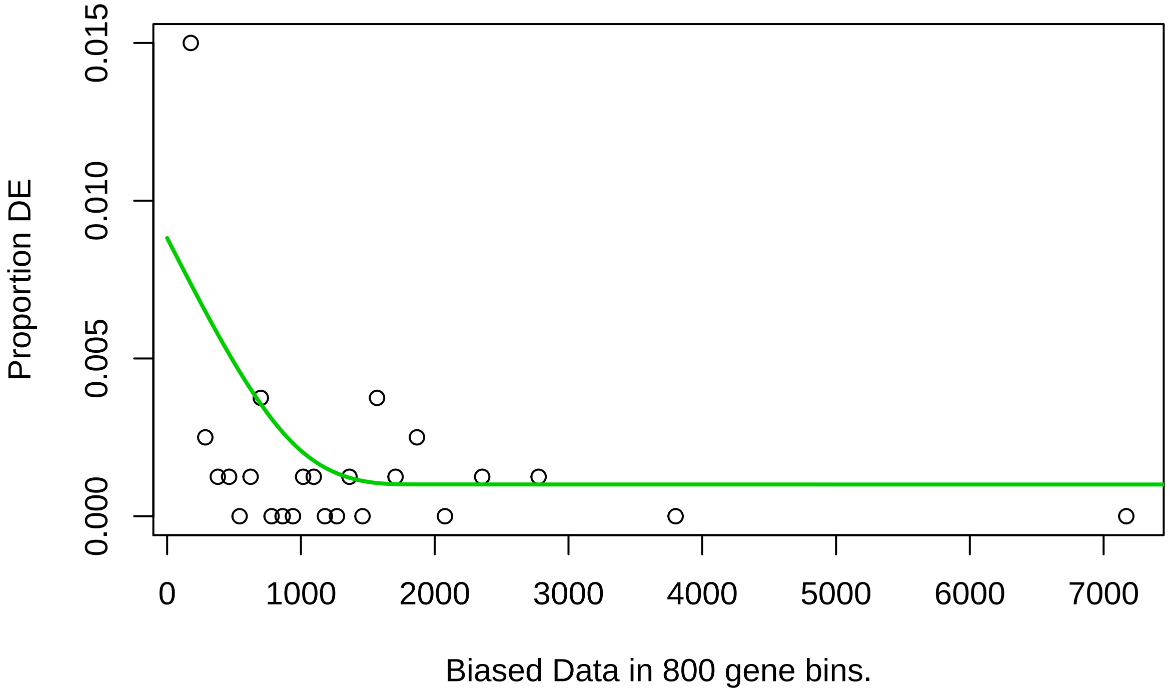
```
sigType <- c("up", "down", "all")

for(type in sigType) {

genes = as.integer(pat[,type])
names(genes) = pat$itag
table(genes)
length(genes)

pwf = nullp(genes,bias.data=pat$length)

GO.wall = goseq(pwf,gene2cat = cate)
head(GO.wall)

#This is going to correct for multiple testing.  You can specify the p-value cut-off of GO categories y

enriched.GO = GO.wall$category[p.adjust(GO.wall$over_represented_pvalue, method = "BH") < 0.05]

enriched.GO
my.GO <- as.character(enriched.GO)
my.GO.table <- Term(my.GO)
my.GO.table
t <- as.matrix(my.GO.table)

print(type) #this is for the knitr document
print(t) #this is for the knitr document

write.table(t, file=paste(sample1,"_",sample2,"DE1_sigonly_",type,"_GO.txt", sep=""))
}
```

```
## Using manually entered categories.
## For 2939 genes, we could not find any categories. These genes will be excluded.
## To force their use, please run with use_genes_without_cat=T (see documentation).
## This was the default behavior for version 1.15.1 and earlier.
## Calculating the p-values...
```



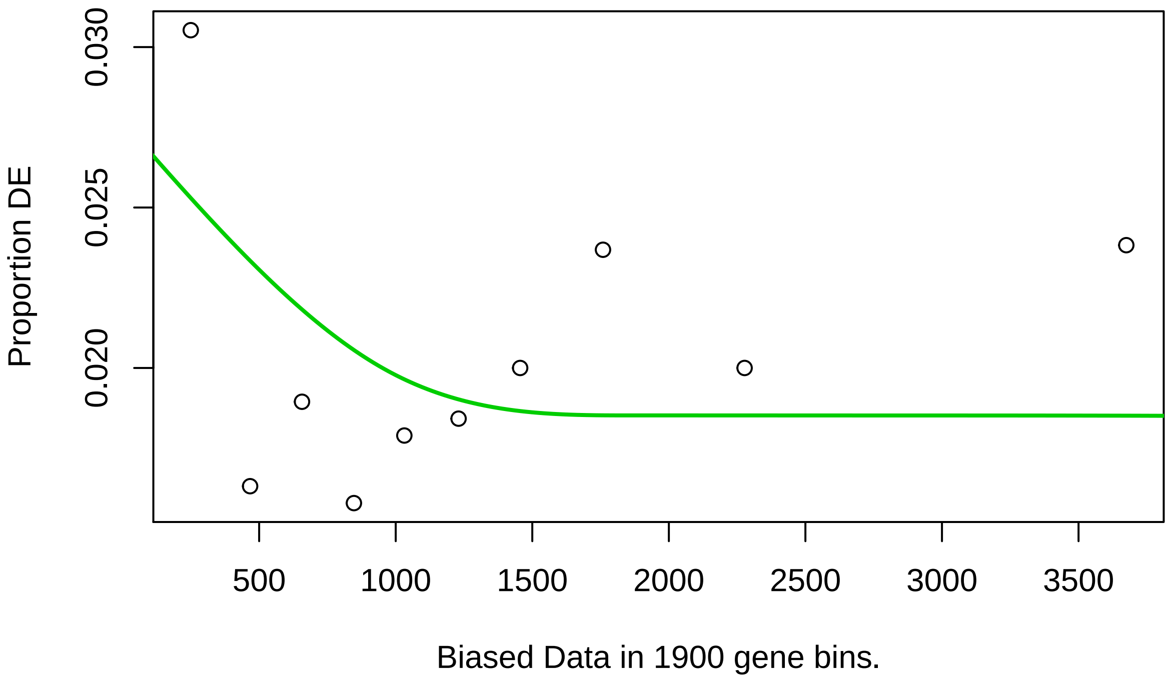
```
## [1] "up"
##            [,1]
## GO:0015074 "DNA integration"
## GO:0003964 "RNA-directed DNA polymerase activity"
## GO:0006278 "RNA-dependent DNA replication"
## GO:0006333 "chromatin assembly or disassembly"
## GO:0000785 "chromatin"
## GO:0003682 "chromatin binding"
## GO:0008270 "zinc ion binding"
## GO:0003677 "DNA binding"
## GO:0043229 "intracellular organelle"
## GO:0003676 "nucleic acid binding"
## GO:0003723 "RNA binding"
## GO:0004190 "aspartic-type endopeptidase activity"
## GO:0006310 "DNA recombination"
## GO:0032549 "ribonucleoside binding"
## GO:0006259 "DNA metabolic process"
## GO:0004518 "nuclease activity"
## GO:0043170 "macromolecule metabolic process"
## GO:0005488 "binding"
## GO:0006508 "proteolysis"
## GO:0016779 "nucleotidyltransferase activity"
```

```
## Using manually entered categories.
## For 2939 genes, we could not find any categories. These genes will be excluded.
## To force their use, please run with use_genes_without_cat=T (see documentation).
## This was the default behavior for version 1.15.1 and earlier.
## Calculating the p-values...
```



```
## [1] "down"
##            [,1]
## GO:0008235 "metalloexopeptidase activity"
## GO:0004177 "aminopeptidase activity"
## GO:0030145 "manganese ion binding"
```

```
## Using manually entered categories.
## For 2939 genes, we could not find any categories. These genes will be excluded.
## To force their use, please run with use_genes_without_cat=T (see documentation).
## This was the default behavior for version 1.15.1 and earlier.
## Calculating the p-values...
```



```
## [1] "all"
##            [,1]
## GO:0015074 "DNA integration"
## GO:0003964 "RNA-directed DNA polymerase activity"
## GO:0006278 "RNA-dependent DNA replication"
## GO:0006333 "chromatin assembly or disassembly"
## GO:0000785 "chromatin"
## GO:0003682 "chromatin binding"
## GO:0008270 "zinc ion binding"
## GO:0003677 "DNA binding"
## GO:0043229 "intracellular organelle"
## GO:0003676 "nucleic acid binding"
## GO:0004190 "aspartic-type endopeptidase activity"
## GO:0003723 "RNA binding"
## GO:0006508 "proteolysis"
## GO:0006310 "DNA recombination"
## GO:0032549 "ribonucleoside binding"
## GO:0006259 "DNA metabolic process"
## GO:0004518 "nuclease activity"
## GO:0043170 "macromolecule metabolic process"
## GO:0005488 "binding"
## GO:0016779 "nucleotidyltransferase activity"
```