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] "wtambr"
```

```
sample2
```

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

```
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 SGN_annotation  AGI
## 1 Solyc00g005070.1.1 -2.439  4.357 2.113e-04 5.309e-03           <NA> <NA>
## 2 Solyc00g006470.1.1 -2.954 11.691 4.068e-08 3.682e-06           <NA> <NA>
## 3 Solyc00g006670.2.1 -4.253  6.268 2.419e-12 5.573e-10           <NA> <NA>
## 4 Solyc00g006680.1.1 -3.682 12.345 2.069e-11 3.885e-09           <NA> <NA>
## 5 Solyc00g006690.2.1 -3.609  8.065 9.840e-11 1.700e-08           <NA> <NA>
## 6 Solyc00g006810.2.1  1.900  4.670 4.059e-03 4.937e-02           <NA> <NA>
##   symbol gene_name X..identity alignment.length e.value bit.score
## 1   <NA>      <NA>          NA               NA      NA        NA
## 2   <NA>      <NA>          NA               NA      NA        NA
## 3   <NA>      <NA>          NA               NA      NA        NA
## 4   <NA>      <NA>          NA               NA      NA        NA
## 5   <NA>      <NA>          NA                NA       NA         NA
## 6   <NA>      <NA>          NA                NA       NA         NA
##   percent.query.align
## 1                  NA
## 2                  NA
## 3                  NA
## 4                  NA
## 5                  NA
## 6                  NA
```

```
dim(sigOnly)
```

```
## [1] 1251   14
```

```
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"
```

```
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.

```
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
```

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

First merge each table to geneLength

```r
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.

```r
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

```r
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  0    1   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

```r
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  0    1   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)
```

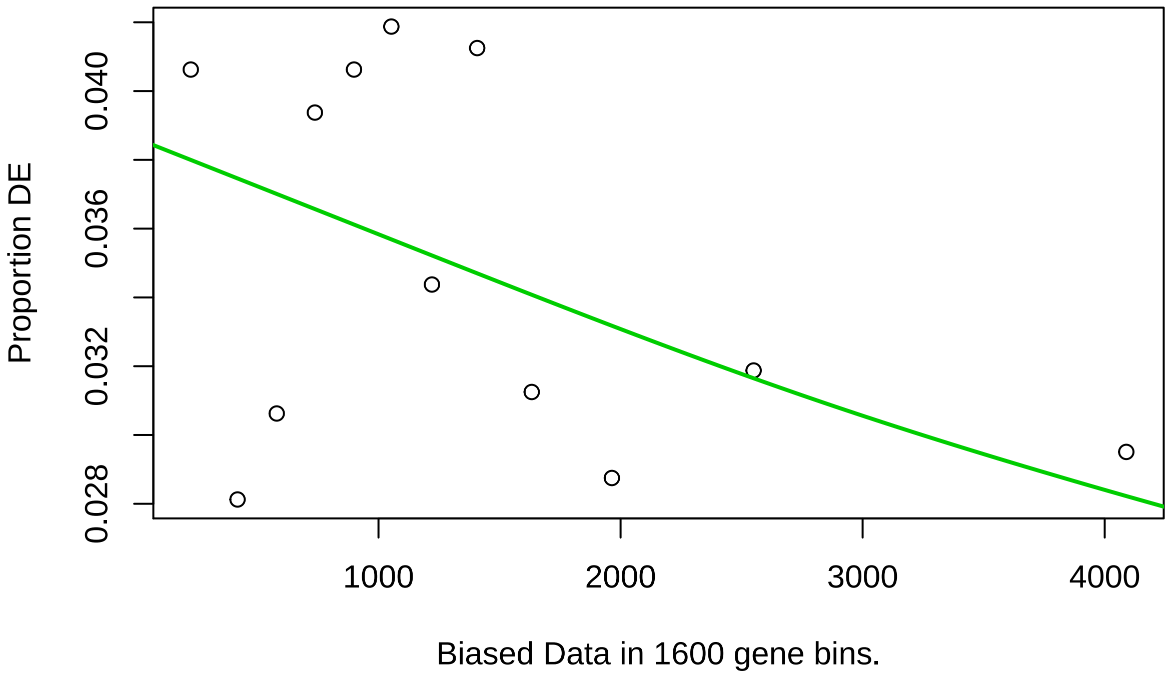
```
##                  itag         go
## 1 Solyc00g005000.2.1 GO:0006508
## 2 Solyc00g005040.2.1 GO:0005774
## 3 Solyc00g005050.2.1 GO:0005829
## 4 Solyc00g005080.1.1 GO:0005524
## 5 Solyc00g005130.1.1 GO:0006508
## 6 Solyc00g005150.1.1 GO:0003676
```

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.

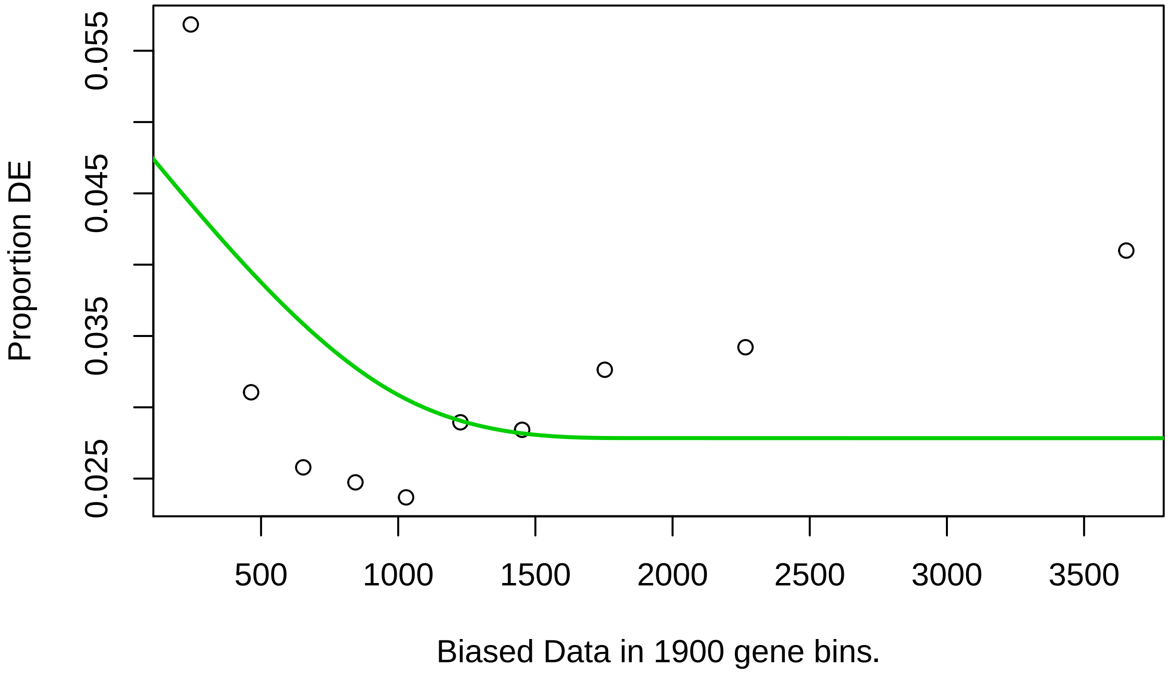
```
sigType <- c("up", "down", "all")

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 2942 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...
```



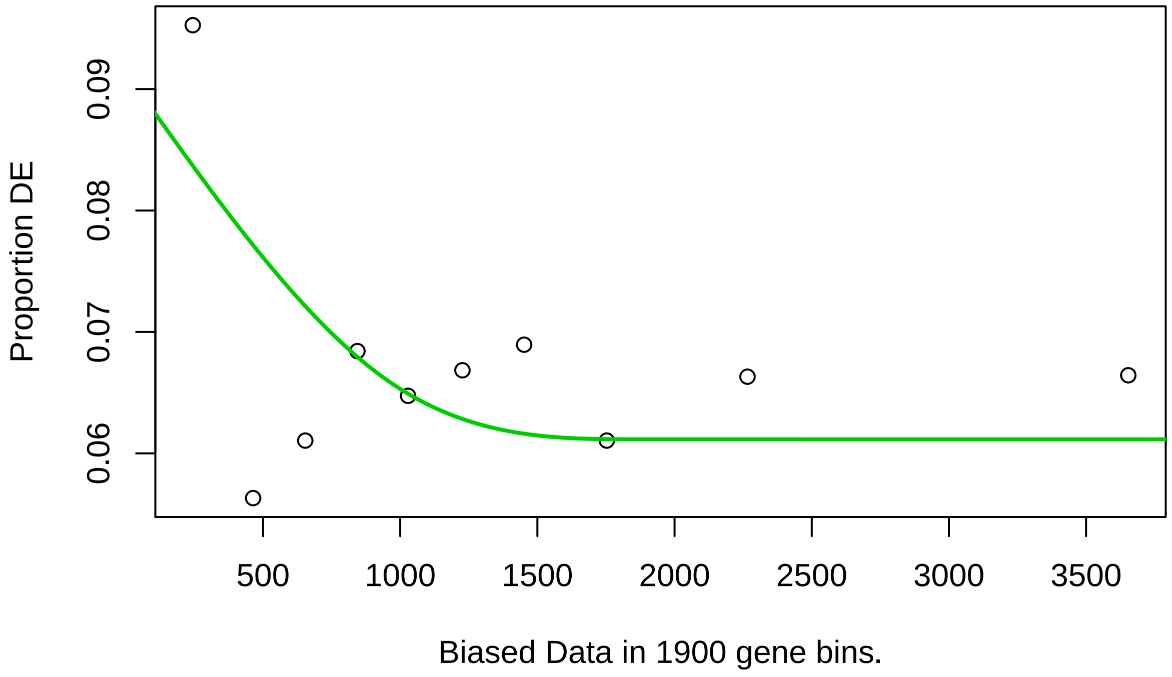
```
## [1] "up"
##            [,1]
## GO:0005985 "sucrose metabolic process"
## <NA>       NA
## GO:0005982 "starch metabolic process"
## GO:0003700 "sequence-specific DNA binding transcription factor activity"
## GO:0005667 "transcription factor complex"
## GO:0043086 "negative regulation of catalytic activity"
## GO:0030001 "metal ion transport"
## GO:0006833 "water transport"
## GO:0006869 "lipid transport"
## GO:0008289 "lipid binding"
## GO:0006857 "oligopeptide transport"
## GO:0008356 "asymmetric cell division"
## GO:0015250 "water channel activity"
## GO:0046910 "pectinesterase inhibitor activity"
## GO:0055085 "transmembrane transport"
## GO:0015706 "nitrate transport"
## GO:0008810 "cellulase activity"
## GO:0005975 "carbohydrate metabolic process"
## GO:0006949 "syncytium formation"
## GO:0030599 "pectinesterase activity"
## GO:0000024 "maltose biosynthetic process"
## GO:0009765 "photosynthesis, light harvesting"
## GO:0043169 "cation binding"
## GO:0006355 "regulation of transcription, DNA-templated"
## GO:0012505 "endomembrane system"
```

```
## Using manually entered categories.
## For 2942 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:0015074 "DNA integration"
## GO:0003964 "RNA-directed DNA polymerase activity"
## GO:0006278 "RNA-dependent DNA replication"
## GO:0003676 "nucleic acid binding"
## GO:0006333 "chromatin assembly or disassembly"
## GO:0006259 "DNA metabolic process"
## GO:0009575 "chromoplast stroma"
## GO:0043229 "intracellular organelle"
## GO:0000785 "chromatin"
## GO:0010466 "negative regulation of peptidase activity"
## GO:0003682 "chromatin binding"
## GO:0004867 "serine-type endopeptidase inhibitor activity"
## GO:0070330 "aromatase activity"
## GO:0048825 "cotyledon development"
## GO:0008270 "zinc ion binding"
## GO:0003677 "DNA binding"
## GO:0004866 "endopeptidase inhibitor activity"
## GO:0006310 "DNA recombination"
```

```
## Using manually entered categories.
## For 2942 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"
## <NA>       NA
## GO:0003700 "sequence-specific DNA binding transcription factor activity"
## GO:0005667 "transcription factor complex"
## GO:0005985 "sucrose metabolic process"
## GO:0003964 "RNA-directed DNA polymerase activity"
## GO:0006278 "RNA-dependent DNA replication"
## GO:0005982 "starch metabolic process"
## GO:0043565 "sequence-specific DNA binding"
## <NA>       NA
## GO:0006949 "syncytium formation"
## GO:0006355 "regulation of transcription, DNA-templated"
## GO:0008810 "cellulase activity"
## GO:0043086 "negative regulation of catalytic activity"
```