GSEAL (Gene Set Enrichment Analysis with LASSO) Package Vignette (Version 0.99.0)

David G. Robinson

March 25, 2014

1 Introduction

This is a vignette for the `GSEAL` package, which performs gene set enrichment analysis using Lasso (L1-constrained) regression. *etc*

2 Example

2.1 Data

First we load in data from the RNA-Seq/Microarray factorial experiment comparing glucose and ethanol:

```
library(GSEAL)
data(factorial)

dim(factorial)

## [1] 6115    7

head(factorial)

##          ORF microarray.pvalue microarray.logFC RNA.Seq.pvalue
## 1:   YAL007C            0.2066          0.14276         0.5351
## 2:   YAL020C            0.6958          0.04087         0.3262
## 3:   YAL021C            0.7921         -0.03111         0.2646
## 4:   YAL025C            0.6919          0.02531         0.7734
## 5:   YAL033W            0.3531         -0.02469         0.6516
## 6: YAL034C-B            0.7972         -0.03067         0.7779
##    RNA.Seq.logFC Intensity Depth
## 1:       0.04898   1909.71 20900
## 2:      -0.13189   2742.05  3959
## 3:      -0.09346    668.01 19964
```

```
## 4:       0.02524   2361.25  8415
## 5:      -0.04532   1478.97  4776
## 6:      -0.27623      3.17    12
```

The only inputs we need are the systematic name of each gene (`factorial$ORF`) and some metric in which we are measuring enrichment. Here we have four choices: the p-values or the log fold-changes from either the RNA-Seq or the microarray parallel experiment. We'll use the RNA-Seq log fold changes for this experiment.

2.2 Membership Matrix

The central data structure of this package is the GeneMatrix class. You can create one quite easily by providing the GO map for that species as found in the AnnotationData Packages.

Organism	Package	GO map
Yeast	`org.Sc.sgd.db`	`org.Sc.sgdGO`
Human	`org.Hs.eg.db`	`org.Hs.egGO`
Mouse	`org.Mm.eg.db`	`org.Mm.egGO`
E. coli K12	`org.EcK12.eg.db`	`org.EcK12.egGO`

```
library(org.Sc.sgd.db)
mm = GOMembershipMatrix(org.Sc.sgdGO, ontology = "BP", min.size = 5, max.size = 250)
```

The membership matrix includes useful information about each gene set:

```
mm@colData[398, ]

##            ID               Term
## 1: GO:0006829 zinc ion transport
##
## 1: The directed movement of zinc (Zn) ions into, out of or within a cell, or between cell
##    Count
## 1:     9
```

As well as a table of information about each gene:

```
mm@geneData

##             ID Count
##    1: 15S_rRNA     3
##    2: 21S_rRNA     3
##    3:     AWA1     1
##    4:     ENA6     4
##    5:     ENS2     1
```

```
##   ---
## 6376:  YPR200C     1
## 6377:  YPR201W     6
## 6378:  YPR202W     0
## 6379:  YPR203W     0
## 6380:  YPR204W     0
```

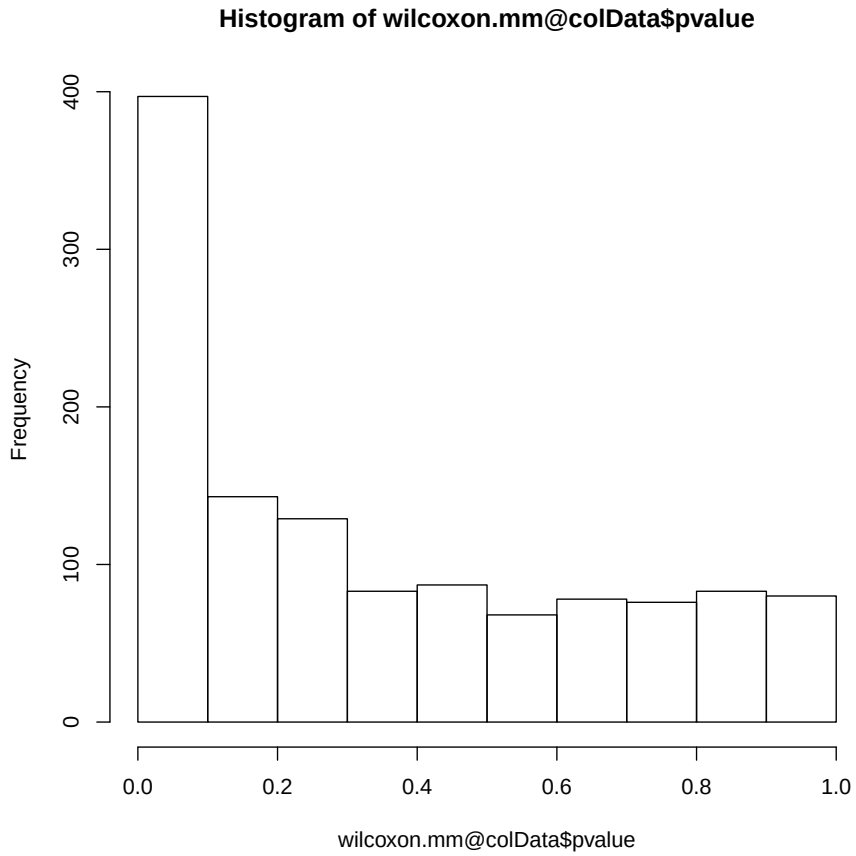
2.3 Wilcoxon Test

One simple test you can perform is a Wilcoxon rank sum test, comparing the y within a set to the y outside that set. This can be done with the `TestAssociation` function:

```
wilcoxon.mm = TestAssociation(mm, factorial$ORF, factorial$RNA.Seq.logFC, method = "wilcoxon
```

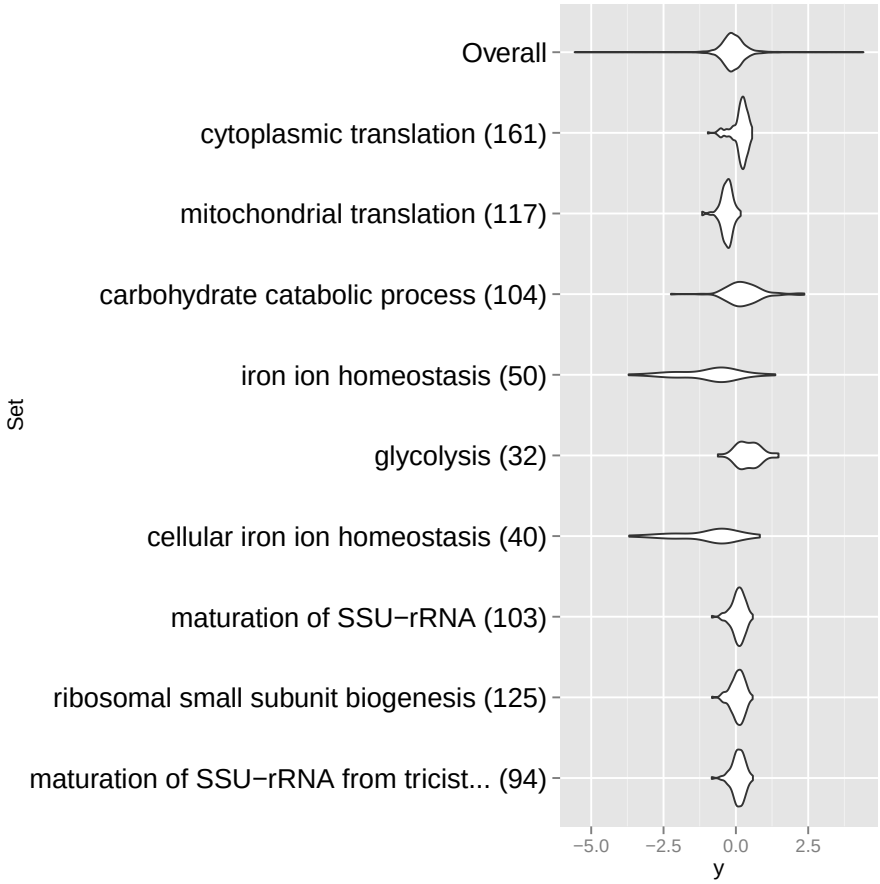
It returns a MembershipMatrix object as well. The p-values from the Wilcoxon test have been added to the `colData` table (one p-value for each set).

```
hist(wilcoxon.mm@colData$pvalue)
```



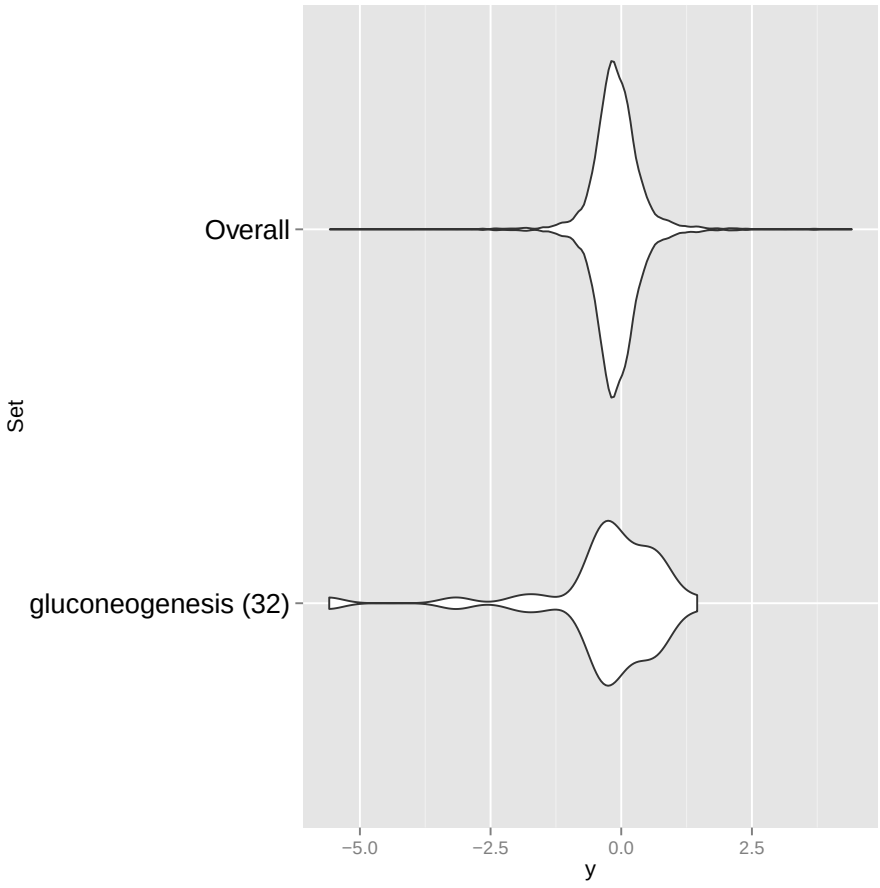
The `CompareTopColumns` function is useful for looking at the actual distribution of y within each of the top significant sets:

```
CompareTopColumns(wilcoxon.mm)
```



If you are interested in other specific gene sets, you can use the CompareY function to compare them one at a time, by ID:

```
CompareY(wilcoxon.mm, "GO:0006094")
```

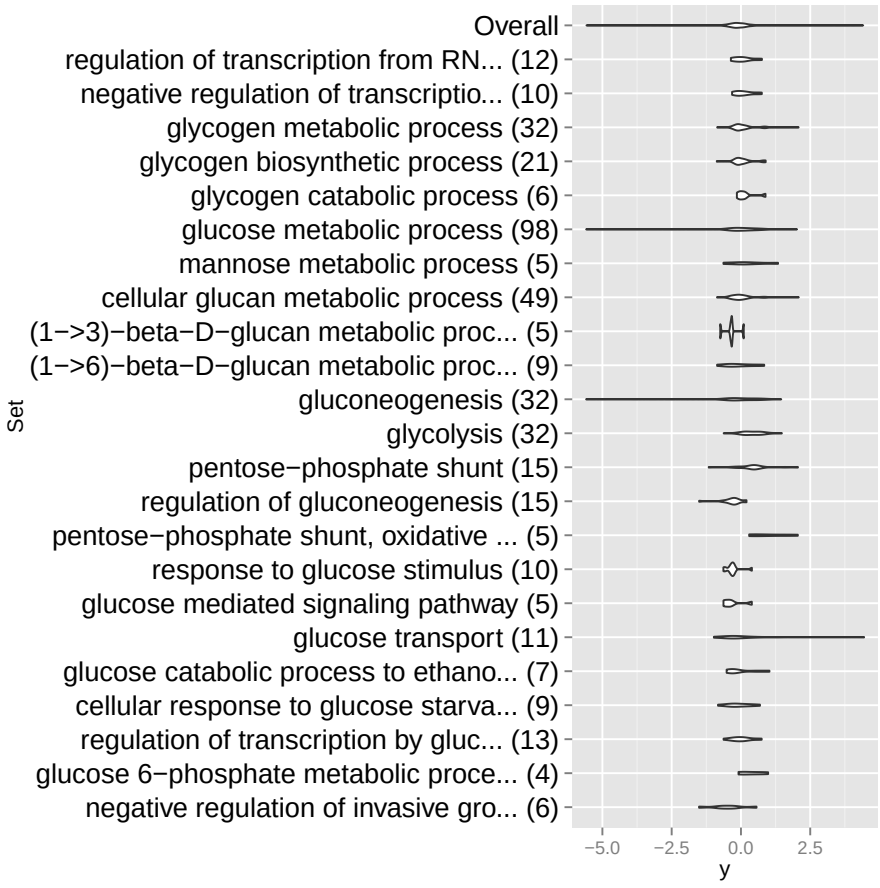


Or we can look at multiple sets that we're interested in:

```
glucose.genes = wilcoxon.mm@colData[grep("glucose", Definition), ]$ID
glucose.genes
```

```
##  [1] "GO:0000430" "GO:0000433" "GO:0005977" "GO:0005978" "GO:0005980"
##  [6] "GO:0006006" "GO:0006013" "GO:0006073" "GO:0006074" "GO:0006077"
## [11] "GO:0006094" "GO:0006096" "GO:0006098" "GO:0006111" "GO:0009051"
## [16] "GO:0009749" "GO:0010255" "GO:0015758" "GO:0019655" "GO:0042149"
## [21] "GO:0046015" "GO:0051156" "GO:2000218"
```

```
CompareY(wilcoxon.mm, glucose.genes)
```

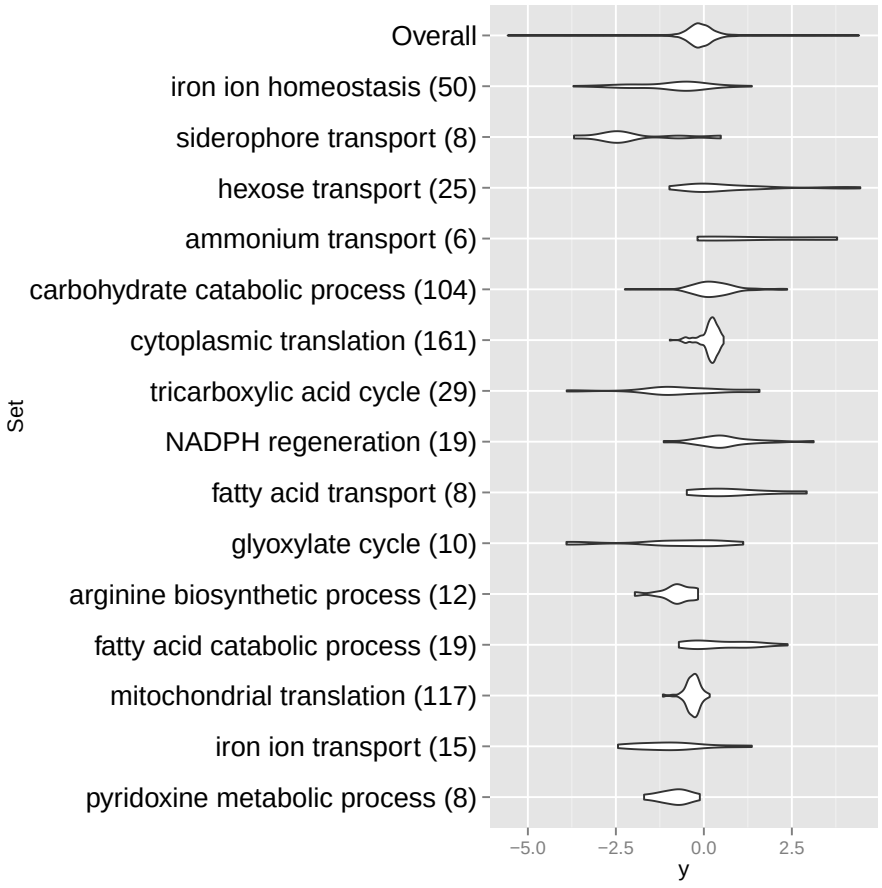


2.4 LASSO

One flaw with the Wilcoxon test is that it treats every hypothesis as being separate, when in fact they are likely highly correlated. For example, gene sets are highly redundant: all gene sets are contained within "parent" gene sets, and some heavily overlap.

```
lasso.mm = TestAssociation(mm, factorial$ORF, factorial$RNA.Seq.logFC, method = "lasso")
```

```
CompareTopColumns(lasso.mm, n = 15)
```


Overall
iron ion homeostasis (50)
siderophore transport (8)
hexose transport (25)
ammonium transport (6)
carbohydrate catabolic process (104)
cytoplasmic translation (161)
tricarboxylic acid cycle (29)
NADPH regeneration (19)
fatty acid transport (8)
glyoxylate cycle (10)
arginine biosynthetic process (12)
fatty acid catabolic process (19)
mitochondrial translation (117)
iron ion transport (15)
pyridoxine metabolic process (8)
Set
−5.0
−2.5
0.0
2.5
y