University of Tampare

BioMedTech

High-throughput data analysis

Gene Set Enrichment analysis using topGO package

By: Nigatu Adossa

Na95542

Submitted to: Ville Kytölä

Data: June, 2014

|  |
| --- |
|  |

**Content**

1. Introduction to gene set enrichment analysis (GSEA) 3
2. GSEA using topGO package 4
   1. Data preparation 4
   2. Running enrichment test 6
   3. Interpretation of the result 7

1. **Introduction to gene set enrichment analysis (GSEA)**

Gene set enrichment analysis (GSEA) is one of the analytical method used to find a biological insights from a genome wide gene expression data. Extraction of biological meaning from gene expression data depends on the gene set, S, which are group of genes sharing common biological functions, chromosomal locations and regulations based on prior biological knowledge.

The genes are ordered based on the phenotypic correlation between the class labels. Then the goal of GSEA is to determine whether the member of gene set is among the top or bottom of the ranked gene list. If the gene set is at the top or bottom of the ranked list, then gene set S is correlated with the phenotypic class distinction.

n external file that holds a picture, illustration, etc.
Object name is zpq0370595180001.jpg

Figure 1. GSEA

In figure 1A, the genes are assorted based on the correlation between the phenotypic class labels A & B. In figure 1B, we got the Gene set S. Combining the ranked gene list with correlation of phenotypes together with gene set information, we can plot a run sum score from which the enrichment score (ES) is calculated. We increase the run sum score by coloration weight if we find a gene in the gene set, S. At the same time we decrease the run sum score by a correlation weight if we do not find a gene in the gene set, S. Then the Enrichment score is calculated to be a maximum deviation of the run sum score from the zero.

Then the significance p-value of the enrichment score can be calculated by first permuting the phenotypic correlated gene lists keeping the gene-gene correlation. Then the run sum plot will give us a null distribution of enrichment score. After this we can calculate the significance levels of p-values of enrichment score.

The leading age subset from the figure 1B shows that the sub set of the genes in the gene set, S that gives rise to the enrichment score or the subset of gene set that contributed for the enrichment of the gene set S.

2. **GSEA using topGO package**

TopGo is a package designed for the semi automated enrichment analysis using gene ontology (GO) terms. TopGo can be downloaded and installed from the bioconductor website as follows:

source("http://bioconductor.org/biocLite.R")

biocLite()

biocLite("topGO")

Gene set enrichment analysis using the TopGo package can be categorized in to three parts: data preparation, running the GSEA and interpreting the result. We will look at each of the steps as we try to perform an enrichment analysis for our SPINK1 positive and negative samples.

2.1 **Data preparation**

In the data preparation step, we give list of genes identifiers, gene scores, list of differentially expressed genes or a criteria for selecting genes based on their scores, as well as gene-to-GO annotations as a one topGo object.

From my pervious project we have three sample classes, but in this case we want to have only two sample class labels. Therefore, We have to resample the class labels.

GSEAsamples<-c(rep(1,82))

SPINK1Sample<-match(highlyExpressedSPINK1Sample,colnames(normalizedLogSummerizedSampleData))

GSEAsamples[SPINK1Sample]<-2

Then we calculate the differential expression between the two samples using lima package:

biocLite("limma")

library("limma")

design <- model.matrix(~ 0+factor(GSEAsamples))

colnames(design)<-c("unknownSample","SPINK1Sample")

fit <- lmFit(normalizedLogSummerizedSampleData, design)

fit <- lmFit(normalizedLogSummerizedSampleData, design)

contrast.matrix <- makeContrasts(SPINK1Sample-unknownSample, levels=design)

fit2 <- contrasts.fit(fit, contrast.matrix)

fit2 <- eBayes(fit2)

sigPvalueMatrix<-fit2$p.value

Since the topGo object requires a vector of geneList with corresponding p-value, we can make a vector out of sigPvalueMatrix with p-value and gene names associated with it as names of the vector.

geneScore<-sigPvalueMatrix[,1]

geneListProbid<-notDuplicatedByGeneNameProbId[,1]

names(geneScore)<-geneListProbid

> head(geneScore)

1007\_s\_at 1053\_at 117\_at 121\_at 1255\_g\_at 1294\_at

0.927857876 0.017538634 0.134668301 0.168855133 0.424073405 0.006322472

After we get all p-values associated with each genes, then we have to create a function that selects list of interesting genes based on the given p-value threshold.

topSigGenes<-function(score){

return(score<0.01)

}

Based on the p-value threshold criteria of 0.01, we can see the number of interesting genes to be:

topDiffGenes<-topSigGenes(geneScore)

numOfTopGenes<-sum(topDiffGenes,na.rm=TRUE)

> numOfTopGenes

[1] 877

I am a bit suspicious about the number of significant genes, may be there might be some mistake on differential calculation but I could not figure it out.

After this we are good to create a topGO object as follows:

GOdata <- new("topGOdata",

description = "GSEA for additional project work",

ontology = "BP",

allGenes = geneScore,

geneSel = topSigGenes,

annot = annFUN.db,

nodeSize = 10,

affyLib ="hgu133plus2.db")

Building most specific GOs ..... ( 8999 GO terms found. )

Build GO DAG topology .......... ( 12201 GO terms and 27725 relations. )

Annotating nodes ............... ( 13977 genes annotated to the GO terms. )

> GOdata

------------------------- topGOdata object -------------------------

Description:

- GSEA for additional project work

Ontology:

- BP

20126 available genes (all genes from the array):

- symbol: 1007\_s\_at 1053\_at 117\_at 121\_at 1255\_g\_at ...

- score : 0.9113125 0.015484 0.1362 0.1655 0.4578704 ...

- NA significant genes.

13977 feasible genes (genes that can be used in the analysis):

- symbol: 1007\_s\_at 1053\_at 117\_at 121\_at 1255\_g\_at ...

- score : 0.9113125 0.015484 0.1362 0.1655 0.4578704 ...

- NA significant genes.

GO graph (nodes with at least 10 genes):

- a graph with directed edges

- number of nodes = 4904

- number of edges = 10769

------------------------- topGOdata object -------------------------

The topGO object seems a bit suspicious that it has got NA values for the significant genes. Keeping this in mind lets perform an enrichment test.

2.2 **Running enrichment test**

We can run the enrichment test, first by creating the test statistics object specifying which type of statistics and algorithm we want to use for the enrichment analysis, for example, we can use fisher testing, Kolmogorov-Smirnov test and so on.

Now let’s use fisher test and create to test statistics object and run the enrichment test:

test.stat <- new("classicCount", testStatistic = GOFisherTest, name = "Fisher test")

> resultFisher <- getSigGroups(GOdata, test.stat)

-- Classic Algorithm --

the algorithm is scoring 3694 nontrivial nodes

parameters:

test statistic: Fisher test

> resultFisher

Description: GSEA for additional project work

Ontology: BP

'classic' algorithm with the 'Fisher test' test

4904 GO terms scored: 143 terms with p < 0.01

Annotation data:

Annotated genes: 13977

Significant genes: NA

Min. no. of genes annotated to a GO: 10

Nontrivial nodes: 3694

> geneData(resultFisher)

Annotated Significant NodeSize SigTerms

13977 NA 10 3694

2.3 **Interpretation of the result**

Finally, let see the top ten enriched biological processes using the the genTable function from the topGO package:

enrichmentResult <- GenTable(GOdata, classicFisher = resultFisher,topNodes = 10)

> enrichmentResult

GO.ID Term Annotated Significant

1 GO:0003012 muscle system process 320 28

2 GO:0006928 cellular component movement 1415 78

3 GO:0007155 cell adhesion 938 56

4 GO:0022610 biological adhesion 940 56

5 GO:0040011 locomotion 1257 69

6 GO:0006936 muscle contraction 282 23

7 GO:0030595 leukocyte chemotaxis 129 14

8 GO:0006935 chemotaxis 583 37

9 GO:0042330 taxis 583 37

10 GO:0046475 glycerophospholipid catabolic process 10 4

Expected classicFisher

1 NA 3.4e-06

2 NA 5.6e-06

3 NA 1.5e-05

4 NA 1.6e-05

5 NA 2.4e-05

6 NA 7.6e-05

7 NA 0.00010

8 NA 0.00014

9 NA 0.00014

10 NA 0.00022

Although the enrichment result has NA value for the expected number of annotation, which I could not figure it out why, the result can be analyzed as muscle system process is functioning differently between the two sample classes. Because from the top 10 enriched biological processes, most of them are related to the muscle contraction, cell adhesion and cellular movement. Therefor, the muscular system process might have something related to the difference in gene expiration between the two tested sample classes.