

# A quick gene selection, annotation and GO analysis

Alex Sánchez<sup>1,2</sup>

<sup>1</sup>Statistics and Bioinformatics Unit (VHIR)

<sup>2</sup> Statistics Department. University of Barcelona

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Obtaining gene lists . . . . .	1
<b>2</b>	<b>Annotating the genes</b>	<b>2</b>
2.1	Using a microarray annotation package . . . . .	2
2.2	Using BiomaRt . . . . .	3
2.3	The gene list for pathway Analysis . . . . .	5
<b>3</b>	<b>Pathway Analysis</b>	<b>5</b>
<b>4</b>	<b>Analysis of Functional Profiles</b>	<b>7</b>

## 1 Introduction

Most gene expression studies undergo one phase where, after gene selection has been performed, one wishes to:

1. Annotate the genes or transcripts, that is associate, to each probeset or transcript, some identifiers in the appropriate databases that can be used to understand better the results or that are needed to proceed with further analyses (for instance GO Analysis needs “Entrez” identifiers).
2. Do some type of Gene Set Enrichment Analyses such as Overrepresentation Analysis (ORA) or classical Gene Set Enrichment Analysis (GSEA).

This document is an illustration which does not intend to be exhaustive, on how to do this with some of these packages.

### 1.1 Obtaining gene lists

The first step in annotation analysis is to obtain the gene lists, usually as the output of some differential expression analysis.

```
topTab <- read.table("https://raw.githubusercontent.com/alexsanchezpla/Ejemplo_de_MDA_con_Bioconductor/main/data/topTab.txt",
colnames=topTab)

## [1] "SymbolsA"      "EntrezsA"      "logFC"         "AveExpr"       "t"
## [6] "P.Value"       "adj.P.Val"     "B"             "GSM26878.CEL"  "GSM26883.CEL"
## [11] "GSM26887.CEL"  "GSM26903.CEL"  "GSM26910.CEL"  "GSM26888.CEL"  "GSM26889.CEL"
## [16] "GSM26892.CEL"  "GSM26898.CEL"  "GSM26906.CEL"

head(topTab)
```

##	SymbolsA	EntrezsA	logFC	AveExpr	t	P.Value
## 204667_at	FOXA1	3169	-3.038	8.651	-14.362	0.00000000005742
## 215729_s_at	VGLL1	51442	3.452	6.138	12.815	0.00000000034385
## 220192_x_at	SPDEF	25803	-3.016	9.522	-10.859	0.00000000433617
## 214451_at	TFAP2B	7021	-5.665	7.433	-10.830	0.00000000451614
## 217528_at	CLCA2	9635	-5.622	6.763	-9.666	0.00000002429965
## 217284_x_at	SERHL2	253190	-4.313	9.133	-9.528	0.00000002994504
##	adj.P.Val	B	GSM26878.CEL	GSM26883.CEL	GSM26887.CEL	
## 204667_at	0.000000357	14.650	9.822	9.514	9.919	
## 215729_s_at	0.000001069	13.150	4.737	4.761	6.255	
## 220192_x_at	0.000007020	10.929	10.484	10.915	10.511	
## 214451_at	0.000007020	10.893	10.177	10.060	11.201	
## 217528_at	0.000030219	9.364	10.534	10.036	11.326	
## 217284_x_at	0.000031033	9.172	11.727	9.741	11.436	
##	GSM26903.CEL	GSM26910.CEL	GSM26888.CEL	GSM26889.CEL	GSM26892.CEL	
## 204667_at	9.601	9.592	6.484	6.551	7.001	
## 215729_s_at	4.820	4.848	8.266	8.963	8.304	
## 220192_x_at	11.510	10.265	7.824	7.810	7.522	
## 214451_at	10.889	10.404	4.818	4.784	4.976	
## 217528_at	8.053	10.619	4.581	4.538	4.519	
## 217284_x_at	12.819	12.687	7.274	7.298	7.491	
##	GSM26898.CEL	GSM26906.CEL				
## 204667_at	6.685	6.535				
## 215729_s_at	8.769	8.381				
## 220192_x_at	8.427	7.020				
## 214451_at	4.912	4.916				
## 217528_at	4.357	4.463				
## 217284_x_at	7.562	7.217				

## 2 Annotating the genes

This table has already been “annotated” in the script that has performed the original analysis, but, *what would we have had to do if it hadn't been?*

We might have used either a specific annotation package for the array or the BioMaRt package.

### 2.1 Using a microarray annotation package

If we hadn't had 'Entrez' Identifiers, but only the probeset identifiers which depend on the array type we might have done as follows:

```
probeIDsAll <- rownames(topTab)
probeIDsUp <- probeIDsAll [topTab$adj.P.Val<0.05 & topTab$logFC > 0]
probeIDsDown <- probeIDsAll [topTab$adj.P.Val<0.05 & topTab$logFC < 0]

require(hgu133a.db)
keytypes(hgu133a.db)

## [1] "ACCNUM"      "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
## [6] "ENTREZID"    "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GO"          "GOALL"        "IPI"          "MAP"          "OMIM"
## [16] "ONTOLOGY"    "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"
## [21] "PROBEID"     "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [26] "UNIGENE"     "UNIPROT"
```

```

geneEntrezsUp <- select(hgu133a.db, keys = probeIDsUp, columns=c("ENTREZID", "SYMBOL"))
## 'select()' returned 1:1 mapping between keys and columns

geneEntrezsDown <- select(hgu133a.db, keys = probeIDsUp, columns=c("ENTREZID", "SYMBOL"))
## 'select()' returned 1:1 mapping between keys and columns

geneEntrezsUniverse <- select(hgu133a.db, keys = probeIDsAll, columns=c("ENTREZID", "SYMBOL"))
## 'select()' returned 1:1 mapping between keys and columns

head(geneEntrezsUp)

##      PROBEID ENTREZID SYMBOL
## 1 215729_s_at   51442  VGLL1
## 2 205044_at    2568  GABRP
## 3 209337_at   11168  PSIP1
## 4 209786_at   10473  HMGH4
## 5 204061_at    5613  PRKX
## 6 207039_at    1029  CDKN2A

```

## 2.2 Using BiomaRt

BiomaRt is a powerful engine for linking identifiers. It is a bit cryptic at the first approach because in order to use it we must define *filters* (what we input for searching), *attributes* (what we output) and *values* (which values we input).

```

biodatset <- useDataset("hsapiens_gene_ensembl", useMart("ensembl"))
listDatasets(biodatset)$dataset

##      [1] "acalliptera_gene_ensembl"      "acarolinensis_gene_ensembl"
##      [3] "acitrinellus_gene_ensembl"     "amelanoleuca_gene_ensembl"
##      [5] "amexicanus_gene_ensembl"       "anancymae_gene_ensembl"
##      [7] "aocellaris_gene_ensembl"       "apercula_gene_ensembl"
##      [9] "aplatyrhynchos_gene_ensembl"   "apolyacanthus_gene_ensembl"
##     [11] "atestudineus_gene_ensembl"     "btaurus_gene_ensembl"
##     [13] "caperea_gene_ensembl"          "catys_gene_ensembl"
##     [15] "ccapucinus_gene_ensembl"       "cchok1gshd_gene_ensembl"
##     [17] "ccrigri_gene_ensembl"          "celegans_gene_ensembl"
##     [19] "cfamiliaris_gene_ensembl"      "chircus_gene_ensembl"
##     [21] "choffmanni_gene_ensembl"       "cintestinalis_gene_ensembl"
##     [23] "cjacchus_gene_ensembl"         "clanigera_gene_ensembl"
##     [25] "cpalliatus_gene_ensembl"       "cporcellus_gene_ensembl"
##     [27] "csabaeus_gene_ensembl"         "csavignyi_gene_ensembl"
##     [29] "csemilaavis_gene_ensembl"      "csyrichta_gene_ensembl"
##     [31] "cvariegatus_gene_ensembl"      "dmelanogaster_gene_ensembl"
##     [33] "dnovemcinctus_gene_ensembl"    "dordii_gene_ensembl"
##     [35] "drerio_gene_ensembl"           "eburgeri_gene_ensembl"
##     [37] "ecaballus_gene_ensembl"        "eeuropaeus_gene_ensembl"
##     [39] "elucius_gene_ensembl"          "etelfairi_gene_ensembl"
##     [41] "falbicollis_gene_ensembl"      "fcatus_gene_ensembl"
##     [43] "fdamarensis_gene_ensembl"      "fheteroclitus_gene_ensembl"
##     [45] "gaculeatus_gene_ensembl"       "gaffinis_gene_ensembl"
##     [47] "ggallus_gene_ensembl"          "ggorilla_gene_ensembl"

```

```
## [49] "gmorhua_gene_ensembl" "hburtoni_gene_ensembl"
## [51] "hcomes_gene_ensembl" "hfemale_gene_ensembl"
## [53] "hmale_gene_ensembl" "hsapiens_gene_ensembl"
## [55] "ipunctatus_gene_ensembl" "itridecemlineatus_gene_ensembl"
## [57] "jjaculus_gene_ensembl" "kmarmoratus_gene_ensembl"
## [59] "lafricana_gene_ensembl" "lbergylta_gene_ensembl"
## [61] "lchalumnae_gene_ensembl" "loculatus_gene_ensembl"
## [63] "malbus_gene_ensembl" "marmatus_gene_ensembl"
## [65] "mauratus_gene_ensembl" "mcaroli_gene_ensembl"
## [67] "mdomestica_gene_ensembl" "mfascicularis_gene_ensembl"
## [69] "mfuro_gene_ensembl" "mgallopavo_gene_ensembl"
## [71] "mleucophaeus_gene_ensembl" "mlucifugus_gene_ensembl"
## [73] "mmola_gene_ensembl" "mmulatta_gene_ensembl"
## [75] "mmurinus_gene_ensembl" "mmusculus_gene_ensembl"
## [77] "mnemestrina_gene_ensembl" "mochrogaster_gene_ensembl"
## [79] "mpahari_gene_ensembl" "mspretus_gene_ensembl"
## [81] "mzebra_gene_ensembl" "nbrichardi_gene_ensembl"
## [83] "neugenii_gene_ensembl" "ngalili_gene_ensembl"
## [85] "nleucogenys_gene_ensembl" "oanatinus_gene_ensembl"
## [87] "oaries_gene_ensembl" "ocuniculus_gene_ensembl"
## [89] "odegus_gene_ensembl" "ogarnettii_gene_ensembl"
## [91] "ohni_gene_ensembl" "ohsok_gene_ensembl"
## [93] "olatipes_gene_ensembl" "omelastigma_gene_ensembl"
## [95] "oniloticus_gene_ensembl" "oprinceps_gene_ensembl"
## [97] "pabelii_gene_ensembl" "paltaica_gene_ensembl"
## [99] "panubis_gene_ensembl" "pbairdii_gene_ensembl"
## [101] "pcapensis_gene_ensembl" "pcoquereli_gene_ensembl"
## [103] "pformosa_gene_ensembl" "pkingsleyae_gene_ensembl"
## [105] "platipinna_gene_ensembl" "pmagnuspinnatus_gene_ensembl"
## [107] "pmarinus_gene_ensembl" "pmexicana_gene_ensembl"
## [109] "pnattereri_gene_ensembl" "pnyererei_gene_ensembl"
## [111] "ppaniscus_gene_ensembl" "ppardus_gene_ensembl"
## [113] "pretilulata_gene_ensembl" "psinensis_gene_ensembl"
## [115] "ptroglodytes_gene_ensembl" "pvampyrus_gene_ensembl"
## [117] "rbieti_gene_ensembl" "rnorvegicus_gene_ensembl"
## [119] "rroxellana_gene_ensembl" "saraneus_gene_ensembl"
## [121] "sboliviensis_gene_ensembl" "scerevisiae_gene_ensembl"
## [123] "sdorsalis_gene_ensembl" "sdumerili_gene_ensembl"
## [125] "sformosus_gene_ensembl" "sharrisii_gene_ensembl"
## [127] "smaximus_gene_ensembl" "spartitus_gene_ensembl"
## [129] "sscrofa_gene_ensembl" "tbelangeri_gene_ensembl"
## [131] "tguttata_gene_ensembl" "tnigroviridis_gene_ensembl"
## [133] "trubripes_gene_ensembl" "ttruncatus_gene_ensembl"
## [135] "vpacos_gene_ensembl" "xcouchianus_gene_ensembl"
## [137] "xmaculatus_gene_ensembl" "xtropicalis_gene_ensembl"
```

```
filters<-listFilters(biodataset)
# We need to find the filter to link with Affymetrix arrays hgu133a
u133aFilters<- grep("u133a", filters[,1] )
u133aFilters <- filters[u133aFilters,]
myu133aFilter <- u133aFilters[3,1]
myu133aFilter

## [1] "affy_hg_u133a"
```

```

atributs<- listAttributes(biodataset)
entrezAtributs<- grep("entrez", atributs[,1])
entrezAtribut <- atributs[entrezAtributs,]
myentrezAtribut <- entrezAtribut[2,1]
myentrezAtribut

## [1] "entrezgene"

# Now we can do the search
entrezfromProbesUp <- getBM(filters= myu133aFilter,
                           attributes= c(myentrezAtribut, myu133aFilter),
                           values= probeIDsUp,
                           mart= biodataset,uniqueRows=TRUE)
head(entrezfromProbesUp)

##   entrezgene affy_hg_u133a
## 1         NA      202396_at
## 2      29078      219006_at
## 3       6515      202499_s_at
## 4 100529262      206560_s_at
## 5      51389      219598_s_at
## 6       6877      210053_at

```

## 2.3 The gene list for pathway Analysis

In this example we had already had the Entrez and Symbol identifiers so we can extract these directly from the topTable.

Although we skip it here it may be interesting to compare the entrez identifiers obtained from the three distinct approaches. They should be identical, but there may be small discrepancies...

```

geneListUp <- topTab$EntrezsA [topTab$adj.P.Val<0.05 & topTab$logFC > 0]
head(geneListUp)

## [1] 51442 2568 11168 10473 5613 1029

geneListDown <- topTab$EntrezsA [topTab$adj.P.Val<0.05 & topTab$logFC < 0]
length(geneListDown)

## [1] 268

geneUniverse <- topTab$EntrezsA
length(geneUniverse)

## [1] 6218

writeGeneLists<- FALSE
if(writeGeneLists){
  write.csv(geneListUp, file="selectedAvsB.up.csv")
  write.csv(geneListDown, file="selectedAvsB.down.csv")
  write.csv(geneUniverse, file="geneUniverse.csv")
}

```

### 3 Pathway Analysis

We start by removing NA's (if any) and ensuring that we have unique identifiers.

```
# Remove potential NAs values
geneEntrezUp <- unique(geneListUp[!is.na(geneListUp)])
geneEntrezDown <- unique(geneListDown[!is.na(geneListDown)])
geneEntrezUniverse <- unique(geneUniverse[!is.na(geneUniverse)])
```

We will use the `G0stats` package which proceeds in two steps:

1. First we create the appropriate objects
2. Next we use them to do the enrichment analysis
3. In a final step we generate an html report with the test results

First we create the appropriate objects

```
require(G0stats)
## Creamos los "hiperparametros" en que se basa el analisis
G0params = new("GOHyperGParams",
  geneIds=geneEntrezUp, universeGeneIds=geneEntrezUniverse,
  annotation="org.Hs.eg.db", # might have use hgu133a.db instead
  ontology="BP",
  pvalueCutoff=0.001, conditional=FALSE,
  testDirection="over")
KEGGparams = new("KEGGHyperGParams",
  geneIds=geneEntrezUp, universeGeneIds=geneEntrezUniverse,
  annotation="org.Hs.eg.db", # might have use hgu133a.db instead
  pvalueCutoff=0.01, testDirection="over")
```

Next we use them to do the enrichment analysis

```
G0hyper = hyperGTest(G0params)
KEGGhyper = hyperGTest(KEGGparams)
cat("GO\n")

## GO

print(head(summary(G0hyper)))

##          GOBPID          Pvalue OddsRatio ExpCount Count Size
## 1 GO:0000278 0.000000005457      2.585    27.989    60  479
## 2 GO:1903047 0.000000463897      2.406    24.308    50  416
## 3 GO:0007049 0.000000491386      2.006    47.856    81  819
## 4 GO:0048584 0.000002983857      1.842    58.257    91  997
## 5 GO:1905819 0.000013104020      8.247      1.578     9   27
## 6 GO:2000816 0.000013104020      8.247      1.578     9   27
##
##                                     Term
## 1                                     mitotic cell cycle
## 2                               mitotic cell cycle process
## 3                                     cell cycle
## 4          positive regulation of response to stimulus
## 5          negative regulation of chromosome separation
## 6 negative regulation of mitotic sister chromatid separation
```

```
cat("KEGG\n")

## KEGG

print(head(summary(KEGGhyper)))

##
## KEGG.db contains mappings based on older data because the original
## resource was removed from the the public domain before the most
## recent update was produced. This package should now be considered
## deprecated and future versions of Bioconductor may not have it
## available. Users who want more current data are encouraged to look
## at the KEGGREST or reactome.db packages

##      KEGGID      Pvalue OddsRatio ExpCount Count Size
## 1  04110 0.001294      2.878    5.724    14    89
## 2  04114 0.002082      3.167    4.116    11    64
## 3  04914 0.002461      3.590    3.023     9    47
## 4  04010 0.004909      2.352    7.267    15   113
## 5  04062 0.006140      2.452    6.045    13    94
## 6  04971 0.007421      4.082    1.801     6    28
##
##                                     Term
## 1                                     Cell cycle
## 2                                     Oocyte meiosis
## 3 Progesterone-mediated oocyte maturation
## 4                                     MAPK signaling pathway
## 5                                     Chemokine signaling pathway
## 6                                     Gastric acid secretion
```

In a final step we generate an html report with the test results

```
# Creamos un informe html con los resultados
GOfilename =file.path(paste("GOResults.AvsB.up", ".html", sep=""))
KEGGfilename =file.path(paste("KEGGResults.AvsB.up", ".html", sep=""))
htmlReport(GOhyper, file = GOfilename, summary.args=list("htmlLinks"=TRUE))
htmlReport(KEGGhyper, file = KEGGfilename, summary.args=list("htmlLinks"=TRUE))
```

## 4 Analysis of Functional Profiles

The `goProfiles` package provides a different approach to Pathway Analysis. Its most distinctive characteristic is the possibility of projecting gene lists on “levels” of the Gene Ontology and compare these projections between lists (compare lists based on their projections).

```
require(goProfiles)

## Loading required package: goProfiles
## Loading required package: CompQuadForm
## Loading required package: stringr
##
## Attaching package: 'stringr'
## The following object is masked from 'package:graph':
##
##      boundary
```

```
BPprofile1<- basicProfile(geneListUp, onto="BP", orgPackage="org.Hs.eg.db", empty.cats=FALSE,
head(BPprofile1)
```

```
##           Description          GOID Frequency
## 8           behavior GO:0007610           10
## 15  biological adhesion GO:0022610           36
## 21           biological phase GO:0044848           1
## 29 biological regulation GO:0065007          271
## 3           cell killing GO:0001906           5
## 9           cell proliferation GO:0008283          82
```

Now we might want to annotate the GO categories with their genes. First we look the reverse, which GO terms are associated with each gene in the list

```
require(org.Hs.eg.db)
keytypes(org.Hs.eg.db)
```

```
## [1] "ACCNUM"      "ALIAS"       "ENSEMBL"     "ENSEMBLPROT" "ENSEMBLTRANS"
## [6] "ENTREZID"    "ENZYME"      "EVIDENCE"    "EVIDENCEALL"  "GENENAME"
## [11] "GO"          "GOALL"       "IPI"         "MAP"          "OMIM"
## [16] "ONTOLOGY"    "ONTOLOGYALL" "PATH"        "PFAM"         "PMID"
## [21] "PROSITE"     "REFSEQ"      "SYMBOL"      "UCSCCKG"      "UNIGENE"
## [26] "UNIPROT"
```

```
entrezUp2GO <- select(org.Hs.eg.db, keys = as.character(geneListUp), columns=c("SYMBOL", "GOALL"))
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
head(entrezUp2GO)
```

```
##  ENTREZID SYMBOL      GOALL EVIDENCEALL ONTOLOGYALL
## 1    51442  VGLL1 GO:0000981          IBA          MF
## 2    51442  VGLL1 GO:0003674          IBA          MF
## 3    51442  VGLL1 GO:0003674          TAS          MF
## 4    51442  VGLL1 GO:0003700          IBA          MF
## 5    51442  VGLL1 GO:0003705          IBA          MF
## 6    51442  VGLL1 GO:0003712          TAS          MF
```

```
entrezUp2GOBP<- entrezUp2GO[entrezUp2GO$ONTOLOGY=="BP",]
BPprofileWithGenes<- cbind(BPprofile1, genes=rep("", nrow(BPprofile1)))
BPprofileWithGenes$genes<- as.character(BPprofileWithGenes$genes)
for (i in 1:nrow(BPprofile1)){
  GOIDi<- BPprofile1[i,"GOID"]
  genesi <-unique(entrezUp2GOBP[entrezUp2GOBP$GO==GOIDi,"ENTREZID"])
  genesi <- paste(genesi[!is.na(genesi)], collapse = " ")
  BPprofileWithGenes[i,"genes"]=genesi
}
head(BPprofileWithGenes)
```

```
##           Description          GOID Frequency
## 8           behavior GO:0007610           10
## 15  biological adhesion GO:0022610           36
## 21           biological phase GO:0044848           1
## 29 biological regulation GO:0065007          271
## 3           cell killing GO:0001906           5
```



```
## 9      cell proliferation GO:0008283      82
##
## 8
## 15
## 21
## 29 51442 2568 5613 1029 3251 10644 2171 2744 81611 467 23708 6566 2824 113 51444 7447 4281 3843 7298
## 3
## 9
```