PloGO: An R package for plotting Gene Ontology annotation and abundance

D. Pascovici and T. Keighley

October 27, 2011

Abstract

This R package contains tools for plotting gene ontology information in a similar fashion to the WEGO web gene ontology plotting tool Ye et al. (2006). However it was designed to incorporate information about abundance in addition to the gene ontology annotation, as well as handle multiple files and allow for a small selection of gene ontology categories of interest. It arose out of the need to easily produce text and graphical summaries of functional annotation with incorporated abundance for multiple files simultaneously. It builds on the functionality provided by the GOstats Falcon and Gentleman (2007) package for handling the GO graph structure, and on the functionality provided by the biomaRt Durinck et al. package for accessing available Biomart repositories. The package includes sample data from Mirzaei et al., which motivated several aspects of this work. NB: While the Mirzaei et al. article is under review the actual experimental data has been replaced with randomly generated values.

1 Introduction

The Gene Ontology hierarchy as developed by the GO consortium is widely used to provide some first-glance insight into the function, biological process or cellular location of collections of genes or proteins. The GO data is organized as a directed acyclic graph starting from one parent node, which means that particular ontology categories can have multiple parents as well as multiple children. Given the complexity of the graph, multiple tools have been designed to navigate it and summarize GO information.

One simple and useful such tool is Wego Ye et al. (2006), a web-based gene ontology plotter, which is readily available over the web and plots gene ontology summaries as well as comparing percentages of GO annotation in different samples. While working in the context of proteomics label free experiments, we found it useful to reproduce and extend some of the WEGO functionality, and generate the PloGO gene ontology annotation and abundance plotter.

Why generate a new package instead of a readily available web tool?

- We needed an easy way to extract particular GO categories of interest and generate summaries in text or excel format for publication for a large number of files
- We found that often researchers wanted to concentrate on particular categories of interest for an experiment (e.g. stress response) across many annotation files, instead of working with a particular level of the GO hierarchy.

- We needed to add protein abundance information such as NSAF values where available to the annotation provided by WEGO
- We needed to be able to analyze multiple files at once (often many more than three which is the Wego limit)
- We wanted to script against the GO plotter and integrate it with our other R analyses as opposed to just access via the web interface
- We preferred to take advantage of and build on existing R packages for GO handling (Falcon and Gentleman (2007)) and annotation (Durinck et al.)
- We planned to include other information available from Biomart such as pathway details and summarize and plot in a similar fashion.

2 Package contents

The R project R Development Core Team (2008) website contains all the needed information for installing R or other packages as needed. The PloGO package can then be installed from the zip file provided.

The package itself contains several subfolders as below, in the standard format of any R package.

- R: The subfolder containing the R code for all the implemented functions
- man: The subfolder containing the help pages for each function
- doc: The subfolder containing the documentation, including the file that you are reading right now, and the dynamic document used to generate it.
- files: The subfolders containing some sample gene ontology files from Mirzaei et al., and a protein abundance file.

3 Package use

First load up PloGO; most of the required packages such as biomaRt and lattice will be loaded up automatically; we are also loading up the xtable packages so we can display some tables nicely in this document. And we'll keep track of the time so we can say how long it all takes.

```
> library(PloGO)
> library(xtable)
> ptm <- proc.time()
```

4 Generate some gene ontology files from scratch

The starting point of a gene ontology analysis is always a collection of identifiers with all their available gene ontology annotation. The GO annotation can be obtained from biomaRt. At this point, only Ensembl human or Uniprot identifiers are accepted, though this can be easily extended. The Uniprot access via biomaRt has been very poor lately, so we switched the examples over to Ensembl human.

The code fragment below generates a very small such file in the local folder where R is running from. Keep in mind that this will require online accesing of the Ensembl biomart repository and as such can take a while (or can fail if the repository is not available for whatever reason).

```
> v <- c("ENSP00000003100", "ENSP00000075120", "ENSP00000160262",
+     "ENSP00000172229", "ENSP00000184956")
> genWegoFile(v, fname = "F1.txt")
```

Now a quick look at the files generated: the format is very simple, with the identifier in the first column followed by space separated GO identifiers.

```
> print(xtable(read.annot.file("F1.txt"), align = "rp{4cm}p{10cm}"))
```

	IDS	V1			
1	ENSP00000003100	GO:0006695	GO:0006805	GO:0005783	GO:0005789
		GO:0005792	GO:0016020	GO:0016021	GO:0004497
		GO:0008398	GO:0009055	GO:0016705	GO:0020037
		GO:0046872			
2	ENSP00000075120	GO:0005975	GO:0006766	GO:0006767	GO:0008643
		GO:0008645	GO:0015758	GO:0055085	GO:0005886
		GO:0016020	GO:0016021	GO:0005355	GO:0022891
3	ENSP00000160262	GO:0016337	GO:0050776	GO:0005886	GO:0005887
		GO:0005102	GO:0005178	GO:0005515	
4	ENSP00000172229	GO:0006915	GO:0006916	GO:0006917	GO:0007165
		GO:0007275	GO:0007411	GO:0007417	GO:0008624
		GO:0009611	GO:0010468	GO:0016048	GO:0021675
		GO:0030154	GO:0031069	GO:0031293	GO:0040037
		GO:0042488	GO:0043588	GO:0045786	GO:0048011
		GO:0048146	GO:0048635	GO:0050770	GO:0050771
		GO:0050772	GO:0051799	GO:0005576	GO:0005634
		GO:0005654	GO:0005737	GO:0005768	GO:0005829
		GO:0005886	GO:0005887	GO:0009986	GO:0004871
		GO:0004872	GO:0004888	GO:0005035	GO:0005488
		GO:0005515	GO:0048406		
5	ENSP00000184956	GO:0005488			

Alternatively, one can obtain some gene ontology files in "long" format, namely identifier followed by GO annotation separated by white space (spaces or tabs). One online program that generates such files is GORetriever (part of the McCarthy et al. (2006) set of tools); other options for obtaining similar output are a direct download of ID and GO information only from places such as Biomart or Uniprot. Only the first two columns of the files will be processed, the rest (if any) will be discarded. Two such sample files, one from GoRetriever and another a Biomart download are included in the package, and we show the first few lines below.

```
> path <- system.file("files", package = "PloGO")
> goRet <- file.path(path, "goRetOutput.txt")
> print(xtable(read.annot.file(goRet)[1:10, ]))
```

	IDS	V1
1	P00359	GO:0001950 C
2	P00359	GO:0003824 F
3	P00359	GO:0004365 F
4	P00359	GO:0005488 F
5	P00359	GO:0005515 F
6	P00359	GO:0005737 C
7	P00359	GO:0005739 C
8	P00359	GO:0005811 C
9	P00359	GO:0006006 P
10	P00359	GO:0006094 P

```
> bioMt <- file.path(path, "mart_export.txt")
> print(xtable(read.annot.file(bioMt)[1:10, ]))
```

	IDS	V1
1	Ensembl	Gene ID GO Term Accession (bp) Ensembl Protein ID
2	ENSG00000072786	GO:0006468 ENSP00000176763
3	ENSG00000072786	GO:0000166 ENSP00000176763
4	ENSG00000072786	GO:0004674 ENSP00000176763
5	ENSG00000072786	GO:0005524 ENSP00000176763
6	ENSG00000072786	GO:0016740 ENSP00000176763
7	ENSG00000072786	GO:0004672 ENSP00000176763
8	ENSG00000075415	ENSP00000188376
9	ENSG00000072110	GO:0002576 ENSP00000193403
10	ENSG00000072110	GO:0007596 ENSP00000193403

5 Process existing gene ontology files

From now on we assume we have several gene ontology files, whether generated with PloGO or obtained elsewhere. For a realistic example, there are 5 sample gene ontology files included in the package; they are a subset of all those analyzed for the paper by Mirzaei et al. and correspond to various presence-absence categories in which the total number of proteins were partitioned. The same package also contains a protein abundance file which has protein abundance values for each identifier, as well as protein names.

```
> file.names <- file.path(path, c("00100.txt", "01111.txt", "10000.txt",
+     "11111.txt", "Control.txt"))
> datafile <- file.path(path, "NSAFDesc.csv")
```

As a next step we could either look at a few categories of interest, or extract all categories at a particular level of the gene ontology graph.

The following code fragment would extract all nodes at the Level 2 (3 or 4) of the GO hierarchy:

```
> GOIDlist <- GOTermList("BP", level = 2)
```

While perhaps restrictive, the list at level 2 could be quite informative; at levels 3 or 4 it is very long. By some manual input or processing you can choose to enter for instance a GO slim of interest; for instance below we selected the cellular component part of the generic GO slim developed by the GO consortium.

```
> GOSlimCC <- c("GO:0000228", "GO:0000229", "GO:0005575",
+     "GO:0005576", "GO:0005578", "GO:0005615",
+     "GO:0005618", "GO:0005622", "GO:0005623",
+     "GO:0005634", "GO:0005635", "GO:0005654",
+     "GO:0005694", "GO:0005730", "GO:0005737",
+     "GO:0005739", "GO:0005764", "GO:0005768",
+     "GO:0005773", "GO:0005777", "GO:0005783",
+     "GO:0005794", "GO:0005811", "GO:0005815",
+     "GO:0005829", "GO:0005840", "GO:0005856",
+     "GO:0005886", "GO:0005929", "GO:0009536",
+     "GO:0009579", "GO:0016023", "GO:0030312",
+     "GO:0043226", "GO:0043234")
```

For the purpose of this analysis, a more targeted fixed list of categories of interest was preferred by Mirzaei et al., as below.

```
> termList <- c("response to stimulus", "transport",
+     "protein folding", "protein metabolic process",
+     "carbohydrate metabolic process", "oxidation reduction",
+     "photosynthesis", "lipid metabolic process",
+     "cell redox homeostasis", "cellular amino acid and derivative metabolic process",
+     "nucleobase, nucleoside and nucleotide metabolic process",
```

```
+     "vitamin metabolic process", "generation of precursor metabolites and energy",
+     "metabolic process", "signaling")
> GOIDmap <- getGoID(termList)
> GOIDlist <- names(GOIDmap)
```

Once you have the files and the GO categories, you need to process the files one by one to extract summaries of the categories of interest. The file by file processing is done by the processWegoFile function.

```
> processGoFile("F1.txt", GOIDlist)
```

The bulk of processing a set of annotation files is done by the processAnnotation file, which can print annotation listings for each file, and merge with abundance information if any is available.

```
> res.list <- processAnnotation(file.names, GOIDlist,
+     printFiles = TRUE)
```

After processing the files, the annotationPlot function produces some visual summaries and generates the counts and percentages.

```
> annot.res <- annotationPlot(res.list)
```

Below are the summaries generated for the protein annotation files considered.

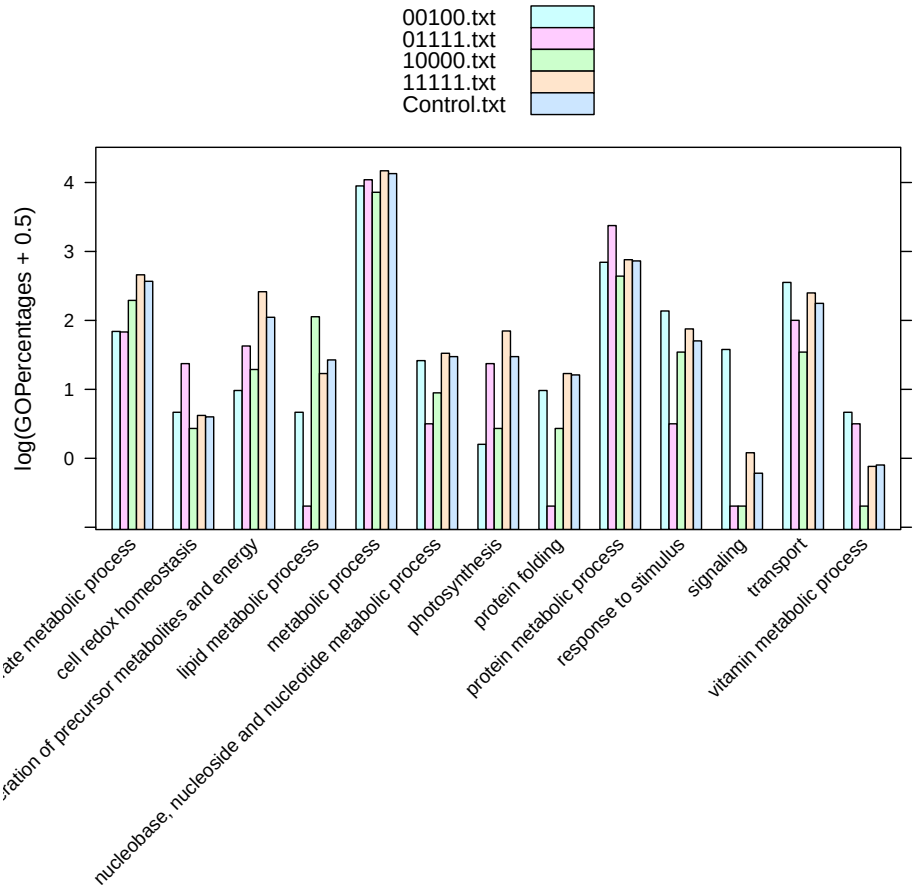
```
> print(xtable(annot.res$counts, align = "rp{1.2cm}p{1.2cm}p{1.2cm}p{1.2cm}p{1.2cm}"))
```

	00100.txt	01111.txt	10000.txt	11111.txt	Control.txt
response to stimulus	11	1	4	31	49
transport	17	6	4	54	88
protein folding	3	0	1	15	28
protein metabolic process	23	25	13	89	167
carbohydrate metabolic process	8	5	9	71	123
photosynthesis	1	3	1	30	38
lipid metabolic process	2	0	7	15	36
cell redox homeostasis	2	3	1	7	13
nucleobase, nucleoside and nucleotide metabolic process	5	1	2	21	38
vitamin metabolic process	2	1	0	2	4
generation of precursor metabolites and energy	3	4	3	55	71
metabolic process	71	49	45	330	605
signaling	6	0	0	3	3

```
> print(xtable(annot.res$percentages, align = "rp{1.2cm}p{1.2cm}p{1.2cm}p{1.2cm}p{1.2cm}"))
```

	00100.txt	01111.txt	10000.txt	11111.txt	Control.txt
response to stimulus	7.97	1.15	4.17	6.03	4.99
transport	12.32	6.90	4.17	10.51	8.96
protein folding	2.17	0.00	1.04	2.92	2.85
protein metabolic process	16.67	28.74	13.54	17.32	17.01
carbohydrate metabolic process	5.80	5.75	9.38	13.81	12.53
photosynthesis	0.72	3.45	1.04	5.84	3.87
lipid metabolic process	1.45	0.00	7.29	2.92	3.67
cell redox homeostasis	1.45	3.45	1.04	1.36	1.32
nucleobase, nucleoside and nucleotide metabolic process	3.62	1.15	2.08	4.09	3.87
vitamin metabolic process	1.45	1.15	0.00	0.39	0.41
generation of precursor metabolites and energy	2.17	4.60	3.12	10.70	7.23
metabolic process	51.45	56.32	46.88	64.20	61.61
signaling	4.35	0.00	0.00	0.58	0.31

Figure 1: Annotation plot: a barplot of the percentage of identification in each selected category and for each selected file, on a log scale



6 Compare annotation to a given reference

One can choose to compare the percentages of annotation in various subsets to a selected reference, to check whether there is an association between the presence in a particular gene ontology category and presence in the particular subset (e.g. are there more carbohydrate metabolism proteins in the subset X than expected by chance considering the whole population?) . This is done by means of applying Fisher's exact test for each gene ontology category and for each subset as compared to the selected reference. The test returns a p-value, which is only recorded if the counts for the respective category are not too small (n $>$ 5).

```
> res <- compareAnnot(res.list, "Control")
> print(xtable(res))
```

	00100.txt	01111.txt	10000.txt	11111.txt	Cont
response to stimulus	0.16			0.40	
transport	0.21	0.69		0.35	
protein folding				1.00	
protein metabolic process	1.00	0.01	0.47	0.89	
carbohydrate metabolic process	0.02	0.08	0.42	0.52	
photosynthesis				0.09	
lipid metabolic process			0.10	0.55	
cell redox homeostasis				1.00	
nucleobase, nucleoside and nucleotide metabolic process	1.00			0.89	
vitamin metabolic process					
generation of precursor metabolites and energy				0.02	
metabolic process	0.03	0.36	0.01	0.34	
signaling					

Given the large number of tests, and the fact that multiple testing corrections are not applied, such a table should be regarded as a suggestion for selecting further categories and protein subsets for further consideration. In the example at hand for instance, there is a clear indication that there are more signalling proteins in the "00100" subset (Protein present at extreme stress conditions only) than expected by chance.

7 Add abundance data

For a last analysis, we now consider the protein abundance data. We process the annotation again, this time indicating that we have an abundance datafile. Note that the abundance file has two descriptive columns (a protein ID and a protein description), so we indicate that by setting the datafile.ignore.cols.

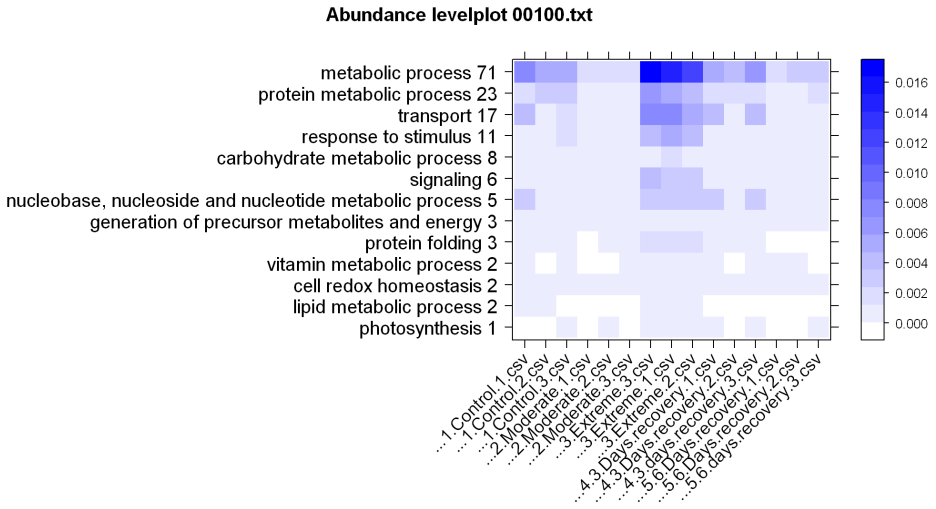
```
> res.list <- processAnnotation(file.names, GOIDlist,
+     datafile = datafile, printFiles = TRUE, datafile.ignore.cols = 2)
```

If the printFiles parameter is set to TRUE, a tab separated annotation file will be printed in the current directory for each of the GO files processed. The format can be changed to a CSV matrix containing identifiers as rows and GO categories as columns. You can inspect for instance the "Annot 11111.csv" and "Annot 11111.txt" to see the different formats. The matrix one might be preferred if one wishes to see combinations of GO identifiers ("Which of my ID's were involved in both transport and signaling?").

```
> writeAnnotation(res.list, datafile = datafile,
+     format = "matrix")
```

After processing the abundance files, some graphs can be generated by abundancePlot and are included below. The graphs are of two kinds, one levelplot for each file, and one barchart for each GO category, provided there was a small number of GO categories being considered.

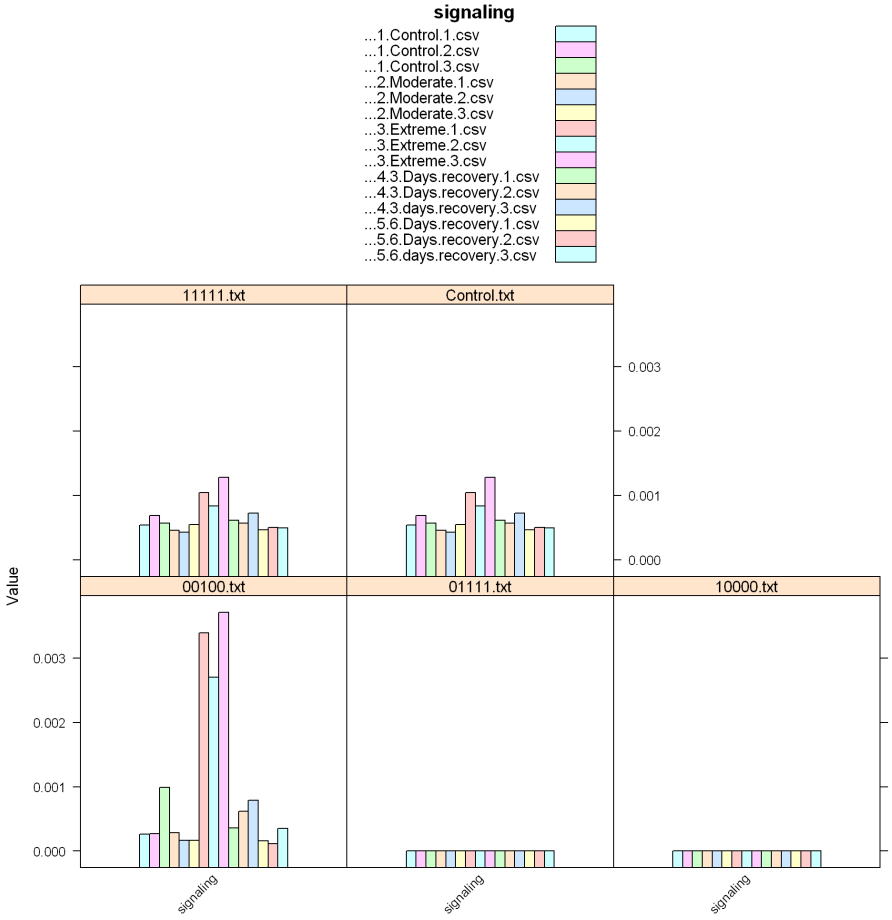
Figure 2: Abundance levelplot for File 1, showing the total abundance for each GO category in each file. One such image is generated for each file under consideration.




```
> tp <- abundancePlot(res.list)
> proc.time() - ptm

   user  system elapsed
  91.58    2.42  112.20
```

Figure 3: Abundance totals for one GO category, in this particular case "signaling", showing the overall abundance in each sample and replicate separately for each file. One such image is generated for each GO category under consideration. In this case, note the abundance peak at Extreme temperatures, file "00100". This abundance view complements the annotation view which showed more signaling proteins at Extreme temperatures only.



8 Conclusions

The PloGO package is a simple gene annotation summarizing and plotting tool, building on existing R packages such as biomaRt and GOstats. It provides for integration of abundance information where such information is present, and allows easy selection of multiple categories of interest as well as allowing for many files to be analyzed at the same time. In future work we plan to extend it to summarize pathway information in a similar manner.

References

Steffen Durinck, Wolfgang Huber, and Sean Davis. *biomaRt: Interface to BioMart databases (e.g. Ensembl, Wormbase and Gramene)*. R package version 2.2.0.

S Falcon and R Gentleman. Using GOstats to test gene lists for GO term association. *Bioinformatics*, 23(2):257–8, 2007.

F. McCarthy, N. Wang, G. B. Magee, B. Nanduri, M. Lawrence, E. Camon, D. Barrell, D. Hill, M. Dolan, W. P. Williams, D. Luthe, S. Bridges, and S. Burgess. Agbase: a functional genomics resource for agriculture. *BMC Genomics*, 7(1):229, 2006. doi: 10.1186/1471-2164-7-229. URL `http://www.biomedcentral.com/1471-2164/7/229`.

M Mirzaei, D Pascovici, B Atwell, and P Haynes. Differential regulation of aqualporins and small gtpases proteins in rice leaves subjected to drought stress and recovery. *Under review*.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Jia Ye, Lin Fang, Hongkun Zheng, Yong Zhang, Jie Chen, Zengjin Zhang, Jing Wang, Shengting Li, Ruiqiang Li, Lars Bolund, and Jun Wang. Wego: a web tool for plotting go annotations. *Nucleic Acids Research*, 34:W293–W297, 2006. Web Server Issue.