PloGO2: An R package for plotting GO or Pathway annotation and abundance

J.Wu and D.Pascovici

September 15, 2022

Abstract

This R package is an upgrade of the PloGO package described in Pascovici et al. (2012), and contains tools for plotting gene ontology or KEGG pathway information for multiple data subsets at the same time. It is designed to incorporate additive information about data abundance (if available) in addition to the gene ontology and pathway annotation, handle multiple input files corresponding to multiple data subsets of interest, and allow for a small selection of gene ontology categories of interest. It allows to potentially compare all subsets to a baseline condition, typically the set of all proteins identified in an experiment, to determine enrichment. The package includes samples of the publicly available data of Mirzaei et al. and Wu et al. (2016).

1 Introduction

Both GO (Gene Ontology) and KEGG pathway data summaries are routinely used by proteomic publications to provide some first-glance insight into the function, process, location, interaction and relation 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. The KEGG pathway data, on the other hand, is simply orgnaised into seven big categories including metabolism, generic information processing and so on.

This package was designed for the following tasks

- Generate GO or KEGG summaries in text or excel format for a large number of data subsets
- Allow for selection of 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.
- Summarise aditive protein abundance information (such as percentages of total abundance, or log fold changes) for all data sets and all categories of interest, if available
- Determine enrichment by comparing all sets to one of the selected sets, typically all proteins identified in the experiment

2 Package use

First load up PloGO2; we are also loading up the xtable packages so we can display some tables nicely in this document.

```
> library(PloGO2)
> library(xtable)
```

Fig 1 shows the high level PloGO2 workflow usage. The workflow contains two parts: Gene Ontology analysis and pathway analysis.

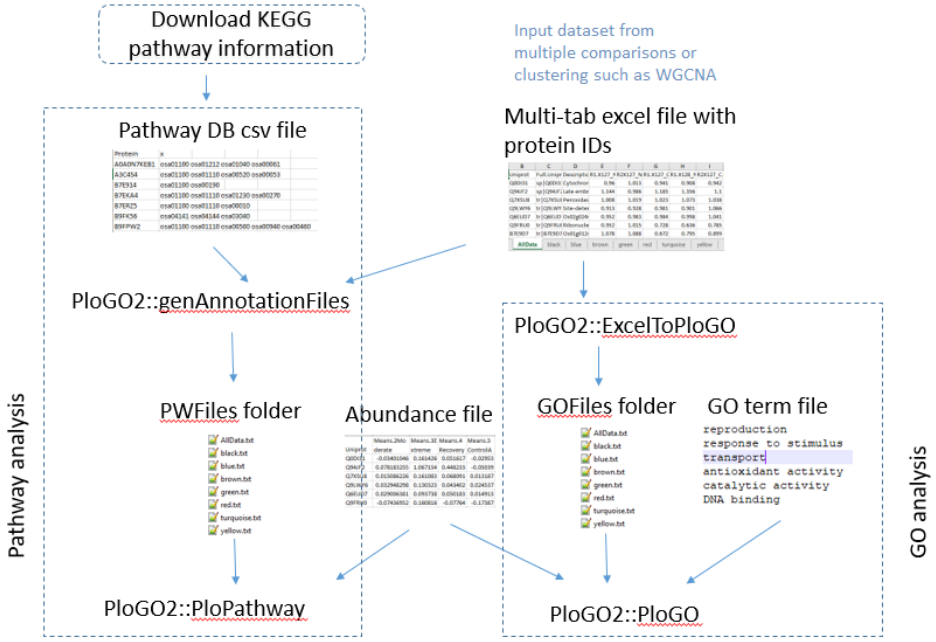


Figure 1: PloGO2 workflow usage

3 Gene Ontology analysis

3.1 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. At this point, only Ensembl human or Uniprot identifiers are accepted, though this can be easily extended. The GO annotation is obtained from biomaRt or directly from the Uniprot website.

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("Q9HWC9","Q9HWD0","Q9I4N8","Q9HW18","Q9HWC9","Q9HWD0")
> wego.file <- 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(wego.file), align = "rp{4cm}p{10cm}"))
```

	IDS	V1
1	Q9HWC9	GO:0000287 GO:0000428 GO:0003677 GO:0003899 GO:0006351 GO:0008270
2	Q9HWD0	GO:0000049 GO:0003735 GO:0005840 GO:0006412 GO:0015935 GO:0019843
3	Q9I4N8	GO:0016020 GO:0047355
4	Q9HW18	

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 = "PloGO2")
> 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, format="long")[1:10, ]))
```

	IDS	V1
1	ENSG00000010256	GO:0006119 GO:0006122 GO:0006508 GO:0006810 GO:0009060 GO:0014823 GO:00229
2	ENSG00000072110	GO:0002576 GO:0007596 GO:0030168 GO:0034329 GO:0042981 GO:0048041 GO:00512
3	ENSG00000072786	GO:0006468 GO:0000166 GO:0004674 GO:0005524 GO:0016740 GO:0004672
4	ENSG00000075415	
5	ENSG00000080824	GO:0000086 GO:0000278 GO:0006839 GO:0006986 GO:0007165 GO:0007411 GO:00346
6	ENSG00000083720	GO:0007420 GO:0007507 GO:0007584 GO:0008152 GO:0009725 GO:0014823 GO:00320
7	ENSG00000083845	GO:0006412 GO:0006413 GO:0006414 GO:0006415 GO:0006450 GO:0010467 GO:00160
8	ENSG00000085719	GO:0006629 GO:0016192 GO:0005737 GO:0005829 GO:0004674 GO:0005215 GO:00055
9	ENSG00000085872	GO:0006874 GO:0007399 GO:0008285 GO:0006396 GO:0005737 GO:0005783 GO:00484
10	ENSG00000086589	GO:0000060 GO:0006397 GO:0008380 GO:0005634 GO:0005681 GO:0005737 GO:00001

3.2 Process existing gene ontology files

From now on we assume we have several gene ontology files, whether generated with PloGO2 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.

```
> # Extract a few categories of interest
> 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 e
+         "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 processGoFile function.

```
> go.file <- processGoFile(wego.file, 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 = FALSE)
```

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{1.2c
```

	00100.txt	01111.txt	10000.txt	11111.txt	Control.txt
response to stimulus	11	1	4	24	42
transport	15	5	4	39	66
protein folding	3	0	1	15	28
protein metabolic process	20	25	12	74	139
carbohydrate metabolic process	8	5	8	64	112
photosynthesis	1	3	1	30	38
lipid metabolic process	1	0	7	15	35
cell redox homeostasis	2	3	1	7	13
vitamin metabolic process	2	1	0	2	4
generation of precursor metabolites and energy	3	4	4	62	81
metabolic process	62	44	41	297	537
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
```

	00100.txt	01111.txt	10000.txt	11111.txt	Control.txt
response to stimulus	7.97	1.15	4.17	4.67	4.28
transport	10.87	5.75	4.17	7.59	6.72
protein folding	2.17	0.00	1.04	2.92	2.85
protein metabolic process	14.49	28.74	12.50	14.40	14.15
carbohydrate metabolic process	5.80	5.75	8.33	12.45	11.41
photosynthesis	0.72	3.45	1.04	5.84	3.87
lipid metabolic process	0.72	0.00	7.29	2.92	3.56
cell redox homeostasis	1.45	3.45	1.04	1.36	1.32
vitamin metabolic process	1.45	1.15	0.00	0.39	0.41
generation of precursor metabolites and energy	2.17	4.60	4.17	12.06	8.25
metabolic process	44.93	50.57	42.71	57.78	54.68
signaling	4.35	0.00	0.00	0.58	0.31

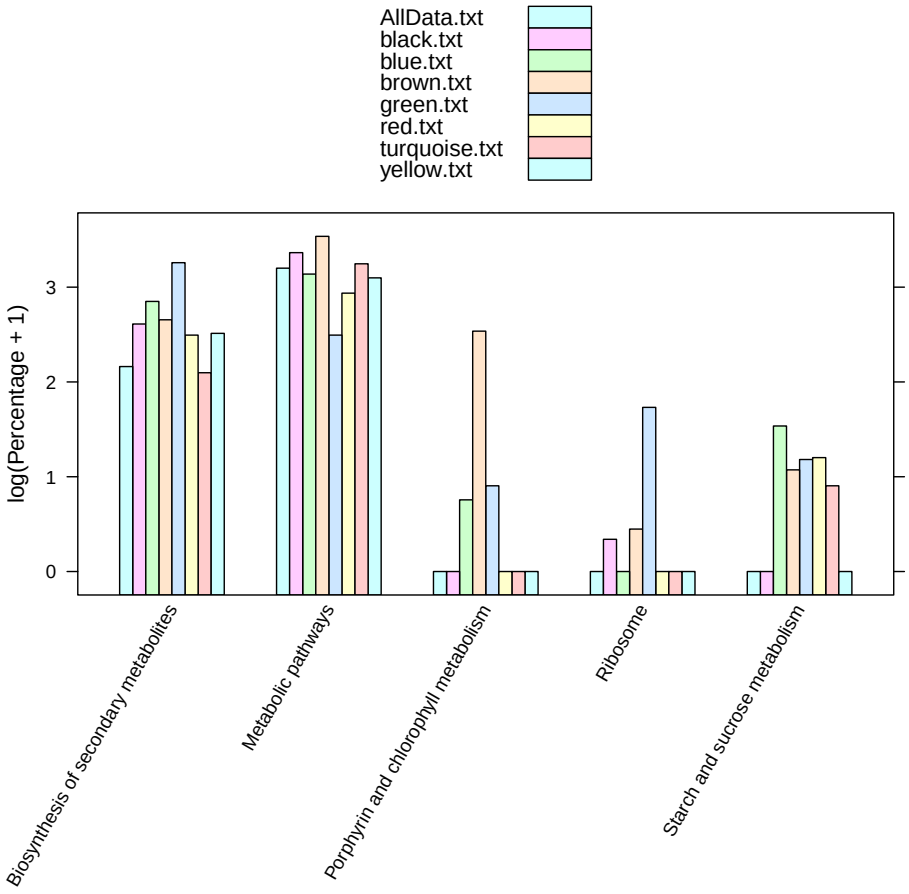


Figure 2: Annotation plot: a barplot of the percentage of identification in each selected category and for each selected file, on a log scale

3.3 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	Control.txt
response to stimulus	0.11		1.00	1.00	
transport	0.11	1.00	0.72	0.87	
protein folding	1.00			1.00	
protein metabolic process	1.00	0.01	0.89	1.00	
carbohydrate metabolic process	0.11	0.35	0.72	0.87	
photosynthesis		1.00		0.49	
lipid metabolic process			0.32	0.87	
cell redox homeostasis		0.35		1.00	
vitamin metabolic process					
generation of precursor metabolites and energy	0.03	0.53	0.54	0.22	
metabolic process	0.10	0.70	0.22	0.87	
signaling	0.00			0.87	

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.

3.4 Add abundance data

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, data.file.name = datafile,
+         printFiles = FALSE, 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

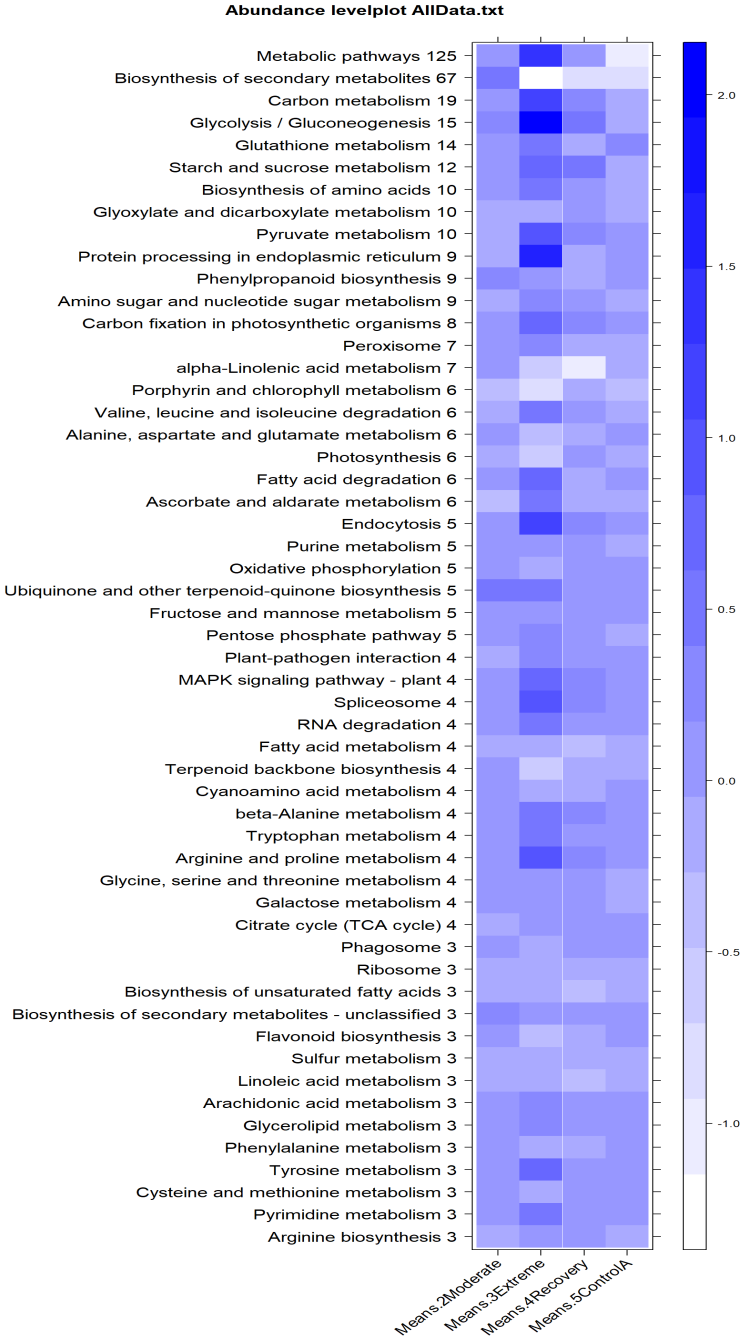
one wishes to see combinations of GO identifiers ("Which of my ID's were involved in both transport and signaling?").

```
> annot.file <- writeAnnotation(res.list, datafile = datafile, format = "matrix")
```

After processing the abundance files, some graphs can be generated by abundancePlot and are included below. One levelplot for each file will be generated.

```
> tp <- abundancePlot(res.list)
```

Figure 3: 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.



4 KEGG pathway analysis

In this section, we will describe the steps involved in KEGG pathway analysis using PloGO2, which is a major extension of our PloGO R packagePascovici et al. (2012). We will use a public dataset in Wu et al. (2016) as an example to help go through the steps of the PloGO2 pathway analysis.

4.1 Prepare the pathway DB file

Unlike the publicly available GO annotation databases, KEGG pathway database requires paid subscription for using their APIs. As such, PloGO2 does not provide functions of accessing KEGG pathway annotations on the fly, hence the pathway annotations for all proteins of interest have to be pre-downloaded. Besides using subscribed APIs, there are a number of ways for downloading KEGG pathway annotations such as using the KEGG website or a third-party tool such as DAVID.

The downloaded pathway annotations need to be saved in a simple two-column CSV file format which is the same as the GO annotation described in 3.1. Each row has one protein ID and one or more pathway IDs separated by space. An example of the pathway DB file is as below.

```
> path <- system.file("files", package = "PloGO2")
> filedb <- file.path(path, "pathwayDB.csv")
> print(xtable(read.csv(filedb)[1:5,]))
```

	Protein	x
1	A0A0N7KEB1	osa01100 osa01212 osa01040 osa00061
2	A3C4S4	osa01100 osa01110 osa00520 osa00053
3	B7E914	osa01100 osa00190
4	B7EKA4	osa01100 osa01110 osa01230 osa00270
5	B7ERZ5	osa01100 osa01110 osa00010

4.2 Generate pathway annotation files

Similar to the GO analysis, the starting point of a pathway analysis is also a collection of identifiers with all their available KEGG pathway IDs in a WegoYe et al. (2006) file format. Function genAnnotationFiles will take a multi-tabbed Excel file with some protein IDs as input and generate an annotation .txt file for each tab.

The 531 differentially expressed proteins from Wu et al. (2016) were clustered into 7 clusters using WGCNA (please refer to PloGO2WithWGCNA vignette for details). Results were saved in an Excel file with each tab containing proteins from one cluster. The KEGG pathway annotations for all proteins were downloaded and formatted in a pathway DB file. Use the following commands to generate the annotation files for all clusters. Parameter colName is used to specify the name of the column that contains the protein IDs, by default, it is “Uniprot". DB.name specifies the pathway DB file name and parameter folder specifies the folder that the generated annotation files will be stored.

```
> path <- system.file("files", package = "PloGO2")
> res.annot <- genAnnotationFiles(fExcelName = file.path(path,
```

```
+                                          "ResultsWGCNA_Input4PloGO2.xlsx"),
+                                    colName="Uniprot",
+                                    DB.name = file.path(path, "pathwayDB.csv"),
+                                    folder="PWFiles")
```

The generated pathway annotation file is in the same file format as the GO annotation file. Below is an example.

```
> print(xtable(read.annot.file(file.path(path, "PWFiles", "red.txt"))[1:5,]))
```

	IDS	V1
1	Q53RM0	osa01100 osa01110 osa00860
2	Q2QP54	
3	Q6ZJ16	
4	Q7XQQ7	
5	Q6ZG77	osa01100 osa01110 osa01230 osa00300

4.3 Semi-automated pathway analysis

When the pathway annotation files have been generated, we can proceed to perform the pathway analysis. PloGO2 provides two ways of executing a pathway analysis: semi-automated and automated.

Using the semi-automated approach, the users can control the analysis and inspect the intermediate output from each step. The first step in this mode is to process the list of annotated files generated above.

Similar to the processGoFile, an individual pathway annotation file can be processed using function processPathFile and the identifiers belonging to each pathway category will be extracted. The following code fragment shows how to process the annotation for the “AllData" file given a list of pathway categories of interest.

```
> fname <- file.path(path,"PWFiles/AllData.txt")
> AnnotIDlist <- c("osa01100","osa01110","osa01230","osa00300","osa00860")
> res.pathfile <- processPathFile(fname, AnnotIDlist)
```

If the abundance data file is available, it can be added as the value of the datafile parameter and the abundance for each pathway category will be aggregated.

```
> datafile <- paste(path, "Abundance_data.csv", sep="/")
> res.pathfile.abun <- processPathFile(fname, AnnotIDlist, datafile=datafile)
```

Again, function processAnnotation can be used to batch processing a list of annotation files. An example is as below.

```
> file.list <- file.path(path,"PWFiles", c("AllData.txt", "red.txt", "yellow.txt",
> AnnotIDlist <- c("osa01100","osa01110","osa01230","osa00300","osa00860")
> res.list <- processAnnotation(file.list, AnnotIDlist, data.file.name=datafile)
```

If the abundance data file exists, function abundancePlot can be used to generate some plots for each annotation file.

```
> Group <- names(read.csv(datafile))[-1]
> abundance <- abundancePlot(res.list, Group=Group)
```

If the reference (usually the whole experimental data) is given, function compareAnnot can be used to perform the pathway enrichment analysis with Fisher exact test.

```
> reference="AllData"
> comp <- compareAnnot(res.list, reference)
> print(xtable(comp, digits=4))
```

	AllData.txt	red.txt	yellow.txt
Metabolic pathways		0.5766	0.1587
Biosynthesis of secondary metabolites		0.5766	0.0518
Biosynthesis of amino acids			
Lysine biosynthesis			
Porphyrin and chlorophyll metabolism		0.0019	

4.4 Automated pathway analysis

In the automated mode, the process described above can be automatically executed using one wrapper function PloPathway. The inputs can be a zip file or file folder, with an optional abundance data file. An example is as below.

```
> path <- system.file("files", package = "PloGO2")
> res <- PloPathway( zipFile=paste(path, "PWFiles.zip", sep="/"),
+          reference="AllData",
+          data.file.name = paste(path, "Abundance_data.csv", sep="/"),
+          datafile.ignore.cols = 1)
```

4.5 Output summary file and plot

A PloGO2 summary file can be printed out as an xlsx file using printSummary.

```
> printSummary(res)
```

A barplot of the aggregated abundance can also be plotted (4.

```
> abudanceBar <- plotAbundanceBar(res$aggregatedAbundance, res$Counts)
```

5 Conclusions

The PloGO2 package is a simple functional annotation summarizing and plotting tool. 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.

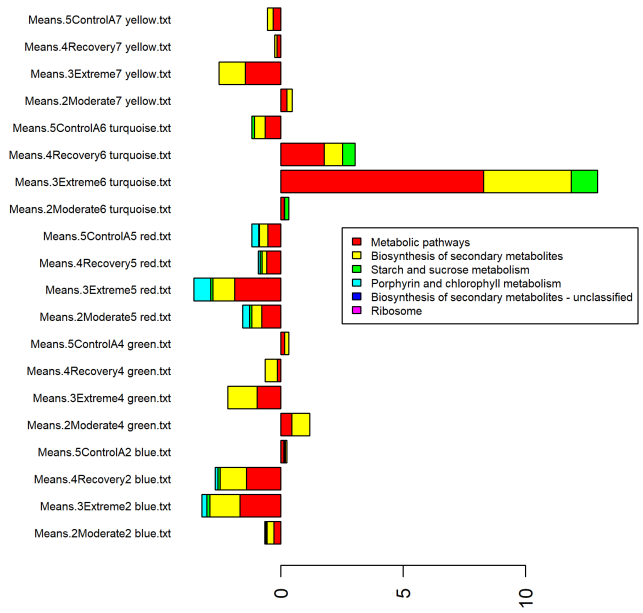


Figure 4: Aggregated abundance barplot for pathways with count >= 5

References

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*.

Dana Pascovici, Tim Keighley, Mehdi Mirzaei, Paul A Haynes, and Brett Cooke. Plogo: Plotting gene ontology annotation and abundance in multi-condition proteomics experiments. *Proteomics*, 12(3): 406–410, 2012.

Yunqi Wu, Mehdi Mirzaei, Dana Pascovici, Joel M Chick, Brian J Atwell, and Paul A Haynes. Quantitative proteomic analysis of two different rice varieties reveals that drought tolerance is correlated with reduced abundance of photosynthetic machinery and increased abundance of clpd1 protease. *Journal of proteomics*, 143:73–82, 2016.

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.