FoldGO: a tool for fold-change-specific functional enrichment analysis

Daniil Wiebe

2018-07-10

Typical scenario of transcriptome data analysis is identification of differentially expressed genes (DEGs), those with significant changes in the number of transcripts (fold-change), and functional enrichment analysis of DEGs lists using Gene Ontology. Classical gene set enrichment analysis ignores the difference in the fold-changes that lead to loss of valuable information. FoldGO method created to identify the GO terms significantly overrepresented for the genes responded to the factor within a narrow fold-change-interval. FoldGO processes the DEGs list in three steps: At the first step, FoldGO sorts the genes by their fold-change values, subdivides them into gene sets of equal size (quantiles) and generates the gene sets for all combinations of the neighboring quantiles, including the whole lists of up-regulated or down-regulated genes. On the second step, FoldGO generates the list of GO terms annotated to at least one differentially expressed gene (DEG). Optionally, GO terms significantly enriched in the DEGs annotation can be selected at this step. On the third step, FoldGO performs enrichment analysis for all selected GO terms and for all gene sublists. FoldGO measures the bias in the portion of the genes responded fold-change-specifically versus whole DEGs for the genes associated with a GO term or the background. If the result of the test is significant, such GO terms are considered as fold-change-specific. See an example of the FoldGO performance in the analysis of the transcriptome data on expression changes of Arabidopsis thaliana genes in response to plant hormone auxin treatment (Omelyanchuk et al., 2017)[1].

Workflow

FoldGO pipeline consists of three steps:

- gene sublists generation;
- GO terms preselection (optional);
- fold-change-specific enrichment analysis.

As input data the algorithm uses the tables for up- and down-regulated genes that contain Gene IDs and their fold-change values:

GeneID	fold-change
AT3G65420	3.6
AT1G78450	1.5
AT2G66890	2.1
…	

First, one have to separate initial set of genes in to quantiles and generate unions of all neighbouring quantiles. For example we will use built-in data derived from experiment on auxin treatment of Arabidopsis thaliana roots. GeneGroups function will take only first two columns, so be sure that your data have gene identifiers in the first column and fold-change values (FC) in the second one. Note that data for up- and down-regulation must be processed separately. In the following example demonstrating GeneGroups function usage only data for up-regulation is used.

[1] Omelyanchuk, N. A. et al. Auxin regulates functional gene groups in a fold-change-specific manner in Arabidopsis thaliana roots // Nat Sci Rep – 2017. – N 7(1) – p 2489

```r
head(degenes)
```

GeneID	FC	pval	qval
AT1G01120	2.069	2.70e-10	2.60e-08
AT1G02520	1.641	7.79e-05	2.06e-03
AT1G02730	1.290	3.37e-05	9.95e-04
AT1G02850	1.978	2.50e-06	1.01e-04
AT1G02900	1.138	1.22e-03	1.95e-02
AT1G03110	2.686	8.44e-05	2.22e-03

```r
up_groups <- GeneGroups(degenes, quannumber=6)
```

On the next step we will conduct functional enrichment analysis of generated groups. For functional enrichment analysis FoldGO uses TopGO package.

Functional annotation

Custom annotaion

For custom annotation one have to provide GO id -> Gene id list. One of the common formats for gene annotation is GAF. FoldGO provides simple and convinient parser for annotation presented in GAF format.

```r
gaf_path <- system.file("extdata", "gene_association.tair", package = "FoldGO")
gaf <- GAFReader(file = gaf_path)
```

One can retrieve direct annotations with some meta information and version of GAF file using following methods:

```r
getVersion(gaf)
#> [1] "2.0"
```

```r
getAnnotation(gaf)
```

and retrieve direct annotations as list object:

```r
gaf_list <- convertToList(gaf)
```

To annotate our gene groups we will use FuncAnnotGroupsTopGO function which uses topGO package for singular enrichment analysis. The minimal set of arguments needed for this function to work is:

- groups - object of GeneGroups class
- namespace - character string specifing GO namespace. This argument accepts following values: “BP”, “MF” or “CC”, where
  - BP - biological process
  - MF - molecular function
  - CC - cellular component
- GO2genes - from topGO manual: named list of character vectors. The list names are GO identifiers. For each GO the character vector contains the genes identifiers which are mapped to it.
- annot - functions used to compile a list of GO terms such that each element in the list is a character vector containing all the gene identifiers that are mapped to the respective GO term. Here it must be assigned with topGO::annFUN.GO2genes
- bggenes - vector contains background set of genes

```r
library(topGO)
annotobj <- FuncAnnotGroupsTopGO(genegroups = up_groups, namespace = "MF", GO2genes = gaf_list, annot =
```

Using annotaions from Bioconductor packages

Another possibility is to use bioconductor packages containing annotations for specific organism. For example "org.Hs.eg.db" (Human) and "org.Mm.eg.db" (Mouse), package name must be assigned to `mapping` argument. In this case one have to assign `topGO::annFUN.org` to annot argument. Specify `ID` - From topGO package manual: character string specifing the gene identifier to use. Currently only the following identifiers can be used: `c("entrez", "genbank", "alias", "ensembl", "symbol", "genename", "unigene")`.

```
up_groups <- GeneGroups(degenes_hum, quannumber=6)
annotobj <- FuncAnnotGroupsTopGO(up_groups,"MF", mapping = "org.Hs.eg.db", annot = topGO::annFUN.org, I
```

Testing for fold-specificity

The fold-specificity recognition procedure consists of GO terms preselection from DEGs annotation and fold-change-specific enrichment analysis. On each step the FDR threshold must be established. By default FDR threshold for GO terms preselection (fdrstep1) is set to 1 (no preselection) and FDR threshold for fold-change-specific enrichment analysis (fdrstep2) is set to 0.05. As a default method for mutltiple testing correction FoldGO uses Benjamini-Hochberg correction procedure.

```
up_fsobj <- FoldSpecTest(up_annotobj, fdrstep1 = 0.05, fdrstep2 = 0.01)
down_fsobj <- FoldSpecTest(down_annotobj, fdrstep1 = 0.05, fdrstep2 = 0.01)
```

It is possible choose another correction procedure from R base which can be listed via `p.adjust.methods`. Here Benjamini-Yekutieli correction procedure is selected.

```
FoldSpecTest(up_annotobj, padjmethod = "BY")
```

One can inspect the results of enrichment analysis as dataframes. Access dataframe with fold-specific terms

```
fs_table <- getFStable(up_fsobj)
```

```
head(fs_table)
```

ids	namespace	name	wholeint_pval	wholeint_padj	min_pval	padj
GO:0000166	MF	nucleotide binding	2.00e-09	2.19e-07	1.16e-03	3.43e-03
GO:0003676	MF	nucleic acid binding	7.70e-29	3.65e-26	5.08e-11	6.30e-10
GO:0003723	MF	RNA binding	1.00e-30	5.69e-28	5.02e-17	7.78e-16
GO:0003729	MF	mRNA binding	1.50e-28	6.09e-26	1.02e-20	2.11e-19
GO:0003735	MF	structural constituent of ribosome	1.00e-30	5.69e-28	1.62e-36	5.02e-35
GO:0003743	MF	translation initiation factor activity	1.50e-08	1.33e-06	5.46e-06	3.39e-05

where:

- ids - GO term identifier
- namespace - GO term namespace
- name - GO term full name
- wholeint_pval - p-value for specific GO term derived from annotation for all differentially expressed genes set
- wholeint_padj - q-value for specific GO term derived from annotation for all differentially expressed genes set
- min_pval - minimal p-value for specific GO term across all intervals
- padj - adjusted minimal p-value for specific GO term across all intervals
- interval - interval that corresponds to minimal p-value for specific GO term

And with not fold-specific:

```
nfs_table <- getNFStable(up_fsobj)
```

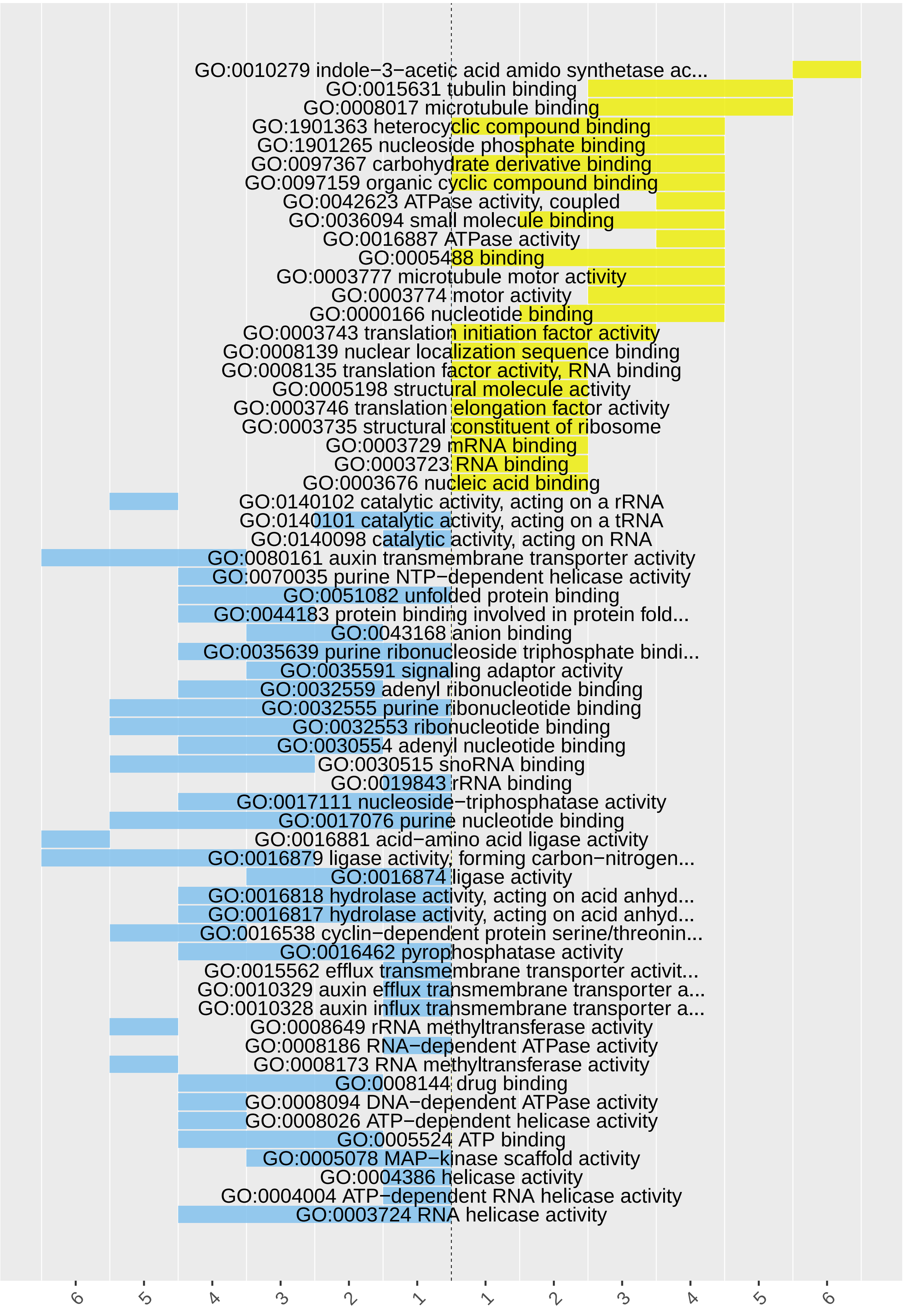
```
head(nfs_table)
```

ids	namespace	name	wholeint__pval	wholeint__padj	min__pval	padj
GO:0003724	MF	RNA helicase activity	6.10e-11	8.26e-09	1.55e-02	2.23e-
GO:0004004	MF	ATP-dependent RNA helicase activity	2.30e-10	2.84e-08	9.47e-03	1.63e-
GO:0004386	MF	helicase activity	9.30e-13	1.56e-10	1.75e-01	1.75e-
GO:0005078	MF	MAP-kinase scaffold activity	6.50e-04	3.08e-02	1.28e-01	1.45e-
GO:0005524	MF	ATP binding	2.30e-09	2.42e-07	1.34e-02	2.03e-
GO:0008026	MF	ATP-dependent helicase activity	7.50e-15	1.52e-12	8.70e-02	1.02e-

Plot results

Via `plot` function one can plot “Fold-change specific GO Profile” on which the GO terms significantly associated with a certain fold-change intervals are presented in yellow and blue boxes for up- and down-regulated genes, correspondingly. Here the result for six equal in size fold-change intervals is presented. The diagram presents only fold-change-specific terms. If the gene was associated fold-specifically with down-regulation but not fold-specifically with up-regulation (or vise versa) than not fold-change-specific interval (1-6 here) will be also shown.

```
plot(up_fsobj, down_fsobj)
```


GO:0010279 indole−3−acetic acid amido synthetase ac...
GO:0015631 tubulin binding
GO:0008017 microtubule binding
GO:1901363 heterocyclic compound binding
GO:1901265 nucleoside phosphate binding
GO:0097367 carbohydrate derivative binding
GO:0097159 organic cyclic compound binding
GO:0042623 ATPase activity, coupled
GO:0036094 small molecule binding
GO:0016887 ATPase activity
GO:0005488 binding
GO:0003777 microtubule motor activity
GO:0003774 motor activity
GO:0000166 nucleotide binding
GO:0003743 translation initiation factor activity
GO:0008139 nuclear localization sequence binding
GO:0008135 translation factor activity, RNA binding
GO:0005198 structural molecule activity
GO:0003746 translation elongation factor activity
GO:0003735 structural constituent of ribosome
GO:0003729 mRNA binding
GO:0003723 RNA binding
GO:0003676 nucleic acid binding
GO:0140102 catalytic activity, acting on a rRNA
GO:0140101 catalytic activity, acting on a tRNA
GO:0140098 catalytic activity, acting on RNA
GO:0080161 auxin transmembrane transporter activity
GO:0070035 purine NTP−dependent helicase activity
GO:0051082 unfolded protein binding
GO:0044183 protein binding involved in protein fold...
GO:0043168 anion binding
GO:0035639 purine ribonucleoside triphosphate bindi...
GO:0035591 signaling adaptor activity
GO:0032559 adenyl ribonucleotide binding
GO:0032555 purine ribonucleotide binding
GO:0032553 ribonucleotide binding
GO:0030554 adenyl nucleotide binding
GO:0030515 snoRNA binding
GO:0019843 rRNA binding
GO:0017111 nucleoside−triphosphatase activity
GO:0017076 purine nucleotide binding
GO:0016881 acid−amino acid ligase activity
GO:0016879 ligase activity, forming carbon−nitrogen...
GO:0016874 ligase activity
GO:0016818 hydrolase activity, acting on acid anhyd...
GO:0016817 hydrolase activity, acting on acid anhyd...
GO:0016538 cyclin−dependent protein serine/threonin...
GO:0016462 pyrophosphatase activity
GO:0015562 efflux transmembrane transporter activit...
GO:0010329 auxin efflux transmembrane transporter a...
GO:0010328 auxin influx transmembrane transporter a...
GO:0008649 rRNA methyltransferase activity
GO:0008186 RNA−dependent ATPase activity
GO:0008173 RNA methyltransferase activity
GO:0008144 drug binding
GO:0008094 DNA−dependent ATPase activity
GO:0008026 ATP−dependent helicase activity
GO:0005524 ATP binding
GO:0005078 MAP−kinase scaffold activity
GO:0004386 helicase activity
GO:0004004 ATP−dependent RNA helicase activity
GO:0003724 RNA helicase activity
6
5
4
3
2
1
1
2
3
4
5
6