Institut für Informatik Wintersemester 2021/2022

Praktische Informatik und Bioinformatik
Prof. Dr. Ralf Zimmer
Armin Hadziahmetovic

# GoBi: Excercise 5

## Gene Ontology and Gene Set Enrichment Analysis

**Deadline**: Tuesday, 14.02.2022, 12:00

### Aim:

Many high-throughput analysis methods, e.g. differential expression (DE) methods, yield a large number of ''statistically significant" results often indicated by small or very small p-values. In order to biologically interpret long result gene lists, the genes are functionally annotated and mapped to gene sets, e.g. from Gene Ontology (GO) or pathways, e.g. pathway database such as KEGG or Reactome. Sets or pathways which contain more than the expected number of result genes are called enriched with associated enrichment score and p-value. Enriched sets with functional annotation could hint to functional interpretation of the result list and the experiment under investigation.

In this task you will learn about the structure of the Gene Ontology and how to use the GO. You will implement a basic analysis tool for Gene Set Enrichment Analysis (GSEA) and interpret the results of a GSEA.

In the block phase you will apply DE and DAS methods and try to represent, visualize and interpret the results with GSEA. Thus, think of meaningful plots helping in the interpretation of results.

### Guideline:

Save your solution to `/mnt/biocluster/praktikum/genprakt/${user}/Solution5`. Provide also an executable jar file (including sources!) in this directory that allows reproducing your results. The jar should print a usage info if invoked without parameters.
Submit your jar file also to the `<abgabeserver>` (template `goenrich`) at

`https://services.bio.ifi.lmu.de:1047/abgabeserver/`

### Gene Ontology Set Enrichment Analysis (200P):

You find all input and example files in the directory:
`/mnt/biosoft/praktikum/genprakt/GOEnrich`

Implement a program to analyse the Gene Ontology(GO) DAG overlap properties and perform set enrichment analysis on differential expression data with the following parameters:

- `-obo <obo_file>`: the OBO-File.
  For details on the OBO-File format see `http://www.geneontology.org/faq/what-obo-file-format`, for the used OBO-File see the file `go.obo`.
  **Hint**: Skip all obsolete entries upon read-in and use the `is_a` relation to build the DAG structure.

- `-root <GO namespace>`: the option `root` is one of GO-s namespaces: `biological_process`, `cellular_component`, and `molecular_function`. It specifies which DAG is to be used in the analysis.

- `-mapping <gene2go_mapping_file>`: defines which genes are associated to which GO-classes. The format of how this mapping is provided depends on the parameter `mappingtype`.

- `-mappingtype [ensembl|go]`: specifies the format of the mapping-File.
  For `mappingtype go` the file format `gaf` is specified in:
  `http://www.geneontology.org/page/go-annotation-file-format-20`. For the assignment we will need only columns 3-5 of the non-comment lines (comment lines start with !): the third (gene id), fourth (association qualifier modifier), and fifth (GO category id). Use only the associations without any association qualifier modifier. The `go` mapping file used in the evaluation is: `goa_human.gaf.gz`.
  **Hint**: do not unzip the file, use the class *java.util.zip.GZIPInputStream* to directly iterate over the compressed file.
  For `mappingtype ensembl` the mapping is provided in a more sparse tsv format defined by three columns: `ensembl`, `hgnc` and `gos`. The first two specify the gene id and gene name (we will use the `hgnc` symbols in the assignment). `gos` specifies the associated GO classes (from all namespaces) as a list, separated by the symbol '|'. The used `ensembl` mapping in the evaluation is g: `goa_human_ensembl.tsv`.
  **Important:** Do not forget to propagate associations from child entries to their parents!

- `-enrich <diffexp_file>`: specifies the input file for the enrichment analysis. The file is a tsv file with three columns:

  - `id`: the gene id
  - `fc`: the $\log_2$ fold change estimation of the differential expression of the gene. This value is to be used to calculate the Kolmogorov-Smirnov (KS) distribution based enrichment values.
  - `signif`: a boolean field defining if the gene was categorized as significantly differentially expressed. The gene ids with `signif`=`true` are to be used to calculate the over-representation based enrichment values.

  You are given an example input file named `simul_exp_go_bp_ensembl.tsv` and corresponding outputs for the parameters:
  `-root biological_process -mappingtype ensembl -mapping goa_human_ensembl.tsv`
  It also provides simulated, truly enriched, GO-classes (the standard-of-truth): the lines starting with the symbol # specify the simulated enriched GO-ids.

- `-o <output_tsv>`: specifies the path to the output tsv file containing the resulting enrichment information. It must contain the following columns:
    - `term`: id of the GO entry
    - `name`: name of the GO entry
    - `size`: number of **measured** associated genes to the GO categories (i.e. the number of gene ids both occurring in the file given by the `enrich` option and associated to the GO entry by the provided mapping (see option `mapping`).
    - `is_true`: boolean (the standard-of-truth), defined by the lines starting with # in the input file given by the `enrich` option.
    - `noverlap`: number of associated `signif` (see parameter `enrich`) genes in the GO entry.
    - `hg_pval`: enrichment p-value given by the hypergeometric distribution.
    - `hg_fdr`: Benjamini-Hochberg corrected `hg_pval`.
    - `fej_pval`: enrichment p-value given by Fischer's exact test using jack-knifing.
    - `fej_fdr`: Benjamini-Hochberg corrected `fej_pval`.
    - `ks_stat`: the statistics value given by the KS-test comparing the $\log_2$ fold change distribution of the measured genes associated to the GO entry versus the background fold change distribution.
    - `ks_pval`: enrichment p-value given by the KS-test comparing the $\log_2$ fold change distribution of the measured genes associated to the GO entry versus the background fold change distribution.
    - `ks_fdr`: Benjamini-Hochberg corrected `ks_pval`
    - `shortest_path_to_a_true`: empty if `is_true` is true or if no true entries are provided; otherwise the shortest path in the used GO DAG to a GO entry. The path should be given as a '|' separated list of the names of the DAG entries, starting from the analysed GO entry and ending with the nearest true GO entry and marking the least common ancestor (LCA), between the nearest true and the `term`, with the suffix ' * '. We define LCA as (one of) the common ancestor(s) of the two DAG entries leading to minimal path length between the two DAG entries.

      Example (on one line!):

      ```
      regulation of lymphocyte activation|
      regulation of leukocyte activation|
      regulation of immune system process*|
      negative regulation of immune system process
      ```

      In this example the analysed GO entry is 'regulation of lymphocyte activation', the nearest true GO entry is 'negative regulation of immune system process'. These two entries have a LCA - 'regulation of immune system process', which is a direct parent of 'negative regulation of immune system process' and the parent of 'regulation of leukocyte activation', which is in turn a parent of the used GO entry.

**Hint**: There might exist both measured genes without any GO associations and non-measured genes with GO associations, so use for all enrichment tests only the measured genes associated to any GO entry of the given namespace. Perform this also for the background distributions.

Here is an example output for an enrichment:

`simul_exp_go_bp_ensembl_min50_max500.enrich.out`

for the parameters:

```
-obo go.obo -root biological_process -mappingtype ensembl
-mapping goa_human_ensembl.tsv -minsize 50 -maxsize 500
-enrich simul_exp_go_bp_ensembl.tsv
```

- `-minsize <int>` and `-maxsize <int>`: define which GO entries are considered in the analysis. The output tsv must contain all GO entries with `minsize` ≤ `size` ≤ `maxsize` where `size` is defined by the number of associated genes to the GO entry (whether these are measured or not).
- `[-overlapout overlap_out_tsv]`: **optional** parameter that specifies an output file. if set, information about DAG entries, with shared mapped genes is written into this file. The `[overlapout` tsv file must have the following columns:
  - `term1`: GO-id (example: GO:1902554) of the first of the two overlapping DAG entries
  - `term2`: GO-id of the second of the two overlapping DAG entries
  - `is_relative`: `true` if the associated DAG entry to `term1` is ascendant or descendent of the one associated to `term2`, `false` otherwise.
  - `path_length`: the length of shortest path between the two DAG entries. The length is defined as the minimal number of edges between `term1` and `term2`.
    **Hint**: there may exist a shorter path between relatives than the direct one.
  - `num_overlapping`: the number of gene ids associated to both DAG entries
  - `max_ov_percent`: the maximum percentage (a float value between 0.0 and 100.0) of the shared gene ids to all associated gene ids to `term1` or `term2`

**Hint**: output only the GO entry pairs both passing the `minsize`, `maxsize` criteria.

You find an example output for `overlapout` in `go_bp_mapping_go_50_500.overlapout`

for the parameters:

```
-root biological_process -mappingtype go -mapping goa_human.gaf.gz
-minsize 50 -maxsize 500
```

The size of the given gene ontology can be large and the GO structure quite complicated. Therefore interpretation of GSEA is not easy.

Compute and provide a number of plots and tables giving an overview of the given GO, e.g. for all GO namespaces:

- No of genes, no of gene sets, no of leafs, shortest and longest path to root, number of ''short-cuts" in the DAG, ...
- Distribution of number of genes in the gene sets.
- Same, for gene sets between minsize and maxsize
- - of path lengths from all leafs to the root
- - of size of set differences between child and parent sets
- ...

Compute and provide a number of plots and tables as overview of the results:

- Distribution of significant gens (SGs) in the all gene sets
- Same, for gene sets between minsize and maxsize
- - of the enrichment scores and p-values
- Scatter of Score against size
- Scatter of p-value against size
- ...
- Sorted tables of genes sets according to no of SGs, score, p-value, ...
- Same, for ranks (mean, median, top x, ...) of SGs in gene sets in sorted list of genes
- ...
- Distribution of SGs in how many sets, are there unique SGs (in only one set)
- - of SGs in intersections of gene sets
- ...

Compare your program with established tools such as DAVID.

Combine your plots in the form of a report into a pdf- and html-file (`go_enrichment.pdf` and `go_enrichment.html`). As a supplement to your report you should briefly describe your approach for implementing this task. This includes analysis of the correctness, time complexity, and actual time usage of your method.

Please also send Report + Supplement **before the deadline** to gobi@bio.ifi.lmu.de with the subject Topic5 Report - Group X, where X is your group id.

To exercise each member can/should make individual submissions to the <abgabeserver>, but via e-mail submit only one report, supplement and program per group.

Weighting of the various deliverables:

- up to 120 pts - <abgabeserver> submission
- up to 45 pts - Report
- up to 35 pts - Report Supplement

**Hints:** To perform the statistical tests we suggest to use the apache common math3 package:

- For the hypergeometric distribution the class:
  org.apache.commons.math3.distribution.HypergeometricDistribution
- For the KS-test the class:
  org.apache.commons.math3.stat.inference.KolmogorovSmirnovTest.