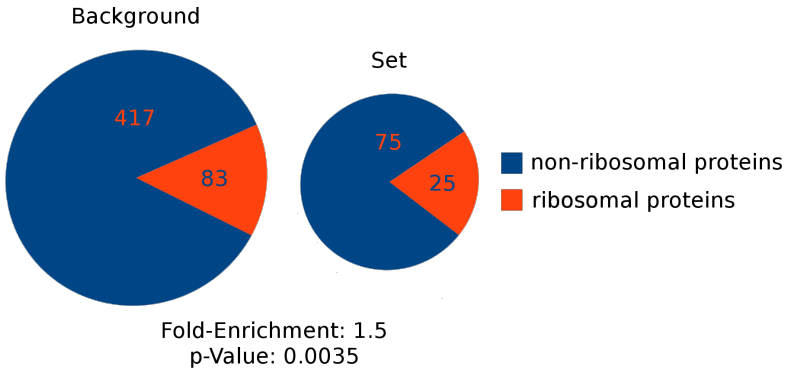
Fuento: functional enrichment for bioinformatics
User Manual
January 2017

The **Fu**nctional **En**richment **To**ol *fuento* compares a subset of genes against its background. *fuento* outputs only relevant enrichment of gene functions in the set; i.e. it tells you the probability of picking the set at random if it shares its distribution of gene functions with the background. Below is an example of an enrichment analysis on a subset of the 100 shortest proteins against a background of 500 yeast proteins. Although the distribution of ribosomal proteins between the two sets should seem random for most naïve observers, to pick from these 500 proteins a set with a similar or higher ratio of ribosomal proteins is highly unlikely.



Fuento was made with efficient commandline-based analysis in mind. It provides you with the information you need, quickly and without uploading your sets to a web application. All files can be dynamically generated from online databases and updated with minimal effort.

1 Quickstart

Fuento was developed for a UNIX-style operating system. Please contact us [a] with any compatibility problems, bugs or with a request for a function not implemented yet.

[a] david.weichselbaum@univie.ac.at, anton.polyansky@univie.ac.at, bojan.zagrovic@univie.ac.at

1.1 Installation

Note: *fuento* depends on the *cURL* and *boost* libraries.

1.1.1 Linux

Run:

```
apt-get install libcurl4-gnutls-dev libboost-all-dev
```

To compile the project, go to the unpacked *fuento* folder and run:

```
make
sudo make install
```

To uninstall:

```
sudo make uninstall
```

1.1.2 OSX

To run *fuento* on Mac OS you should have CURL and BOOST library installed. If you do not have them, this can simply be done using MacPort - an open-source system for compiling, installing and upgrading. Please, follow the instruction below.

1. Install *Xcode* (Apple developer tools), and with it *clang++*
2. Install *MacPort* that matches to your Mac OS version (https://www.macports.org/install.php)
3. Install *BOOST libraries*: sudo port install boost +universal
4. Install *CURL libraries*: sudo port install curl

The provided binary file of *fuento* is complied for OS 10.9.6.

Fuento on Mac can be compiled as follows:

```
clang++  fuento.cpp -I [headers path]  -L[library path] -lcurl
    -lboost_regex-mt -lboost_filesystem-mt -std=c++0x
    --stdlib=libc++ -O2 -w -o fuento
```

defaults:

```
[headers path] = /opt/local/include/
[library path]= /opt/local/lib/
```

1.2 Gene Ontology Database

To use *fuento*, first get the newest Gene Ontology (GO) database:

```
fuento -g
```

```
david ~/fuento_demo fuento -g
Downloading from:       http://www.geneontology.org/ontology/go.obo ...
Database file created:  /home/david/.fuento/go.obo
Gene ontology slim file "Aspergillus GO slim" created: /home/david/.fuento/goslim_aspergillus.obo
Gene ontology slim file "Candida GO slim" created: /home/david/.fuento/goslim_candida.obo
Gene ontology slim file "ChEMBL protein targets summary" created: /home/david/.fuento/goslim_chembl.obo
Gene ontology slim file "Generic GO slim" created: /home/david/.fuento/goslim_generic.obo
Gene ontology slim file "GOA and proteome slim" created: /home/david/.fuento/goslim_goa.obo
Gene ontology slim file "Metagenomics GO slim" created: /home/david/.fuento/goslim_metagenomics.obo
Gene ontology slim file "PIR GO slim" created: /home/david/.fuento/goslim_pir.obo
Gene ontology slim file "Plant GO slim" created: /home/david/.fuento/goslim_plant.obo
Gene ontology slim file "Fission yeast GO slim" created: /home/david/.fuento/goslim_pombe.obo
Gene ontology slim file "synapse GO slim" created: /home/david/.fuento/goslim_synapse.obo
Gene ontology slim file "Viral GO slim" created: /home/david/.fuento/goslim_virus.obo
Gene ontology slim file "Yeast GO slim" created: /home/david/.fuento/goslim_yeast.obo
Added gene ontology slim file /home/david/.fuento/goslim_aspergillus.obo to slim list file.
Added gene ontology slim file /home/david/.fuento/goslim_candida.obo to slim list file.
Added gene ontology slim file /home/david/.fuento/goslim_chembl.obo to slim list file.
Added gene ontology slim file /home/david/.fuento/goslim_goa.obo to slim list file.
Added gene ontology slim file /home/david/.fuento/goslim_metagenomics.obo to slim list file.
Added gene ontology slim file /home/david/.fuento/goslim_pir.obo to slim list file.
Added gene ontology slim file /home/david/.fuento/goslim_yeast.obo to slim list file.
```

This adds a folder to your home directory containing the database and Gene Ontology slim files[b].

1.3 Backgrounds

Then build a background from a file of UniProt IDs, separated by whitespace.

```
fuento -b FILE
```

```
david[11.8] ~/fuento_demo fuento -b human102013.IDs
Downloading geneontology annotation...
Creating file: human102013.IDs_2016-12-15.bkg
707  element(s) have zero functions annotated. Not UniProt IDs?
Q9UGB4  P0CW21  A6NGU7  Q5JQF7  Q8TAD7  Q9Y2S6  H3BMG3  Q5VT28  A6NKN8
```

The *background* option downloads Gene Ontology[1,2] annotation for each gene from an online database[c]. If there are no functions annotated to a given gene, it will display the ID for you to check.

If you used a different gene ID, build a background by setting the *map*ping option to the standard you use:

[b] A GO slim is a pruned Gene Ontology tree. See section Theory.
[c] www.geneontology.org/ontology/go.obo

```
fuento -M GENE_ID -b FILE
```

```
david ~/fuento_demo fuento -M P_ENTREZGENEID -b escherichia102013.EGID
Mapping P_ENTREZGENEID entries from escherichia102013.EGID to UniProt Accession (ACC)
 3997/3997      100.00%
Creating file: escherichia102013.EGID_P_ENTREZGENEID
Downloading geneontology annotation...
Creating file: escherichia102013.EGID_2016-12-15.bkg
303  element(s) have zero functions annotated. Not UniProt IDs?
P0AD72  Q6BF86  Q6BF87  P76061  P0ADD9  Q47272  P0ACW0  P0ACW8  P76157  P39390  P42625
```

GENE_ID can be any of the 99 gene IDs supported by UniProt's mapping API[d]. This also works with the '*-B*' option described below.

Using the UniProt API[3] it is possible to generate backgrounds using any of the 172 annotations available as a UniProt column[e]. To download protein family annotation do:

```
fuento -B FILE families
```

Fuento generates files in the path "*/home/USER/.fuento*" storing paths to all backgrounds and GO slims. Those can be listed by setting the following flag:

```
fuento -l
```

```
david ~/fuento_demo fuento -l
Gene ontology version date: 02:09:2016 16:53
Available backgrounds:
# 1:     17856 entries                    geneontology              human   /home/david/fuento_demo/human102013.IDs_2016-09-06.bkg
# 2:     17856 entries                        families           humanFam   /home/david/fuento_demo/human102013.IDs_families_2016-07-11.bkg
# 3:     17856 entries                 feature(MOTIF)          humanMotif   /home/david/fuento_demo/human102013.IDs_featureMOTIF_2016-07-19.bkg
# 4:      5897 entries                    geneontology            saccharo   /home/david/fuento_demo/saccharomyces032014.IDs_2016-06-20.bkg
# 5:      5897 entries                        families       saccharoFam   /home/david/fuento_demo/saccharomyces032014.IDs_families_2016-06-20.bkg
# 6:      5897 entries          geneontology, families            saccharo   /home/david/fuento_demo/saccharomyces032014.IDs_union_2016-09-07.bkg
Available GO slims:
# 1:       85 entries            Aspergillus GO slim         Aspergillus   /home/david/.fuento/goslim_aspergillus.obo
# 2:       89 entries              Candida GO slim             Candida   /home/david/.fuento/goslim_candida.obo
# 3:      309 entries  ChEMBL protein targets summary              ChEMBL   /home/david/.fuento/goslim_chembl.obo
# 4:        0 entries           GOA and proteome slim                 GOA   /home/david/.fuento/goslim_goa.obo
# 5:      116 entries         Metagenomics GO slim         Metagenomics   /home/david/.fuento/goslim_metagenomics.obo
# 6:      462 entries                  PIR GO slim                 PIR   /home/david/.fuento/goslim_pir.obo
# 7:      169 entries                Yeast GO slim               Yeast   /home/david/.fuento/goslim_yeast.obo
```

Numbers and labels in cyan can be used to refer to files alternatively to the actual file names. Labels can be added with the '-L' option: they are automatically generated for GO slims. To remove a file from the list, delete or rename it. To add a background or slim file manually, use:

```
fuento -L FILE LABEL
```

Slim files must have the extension '*.obo*'.

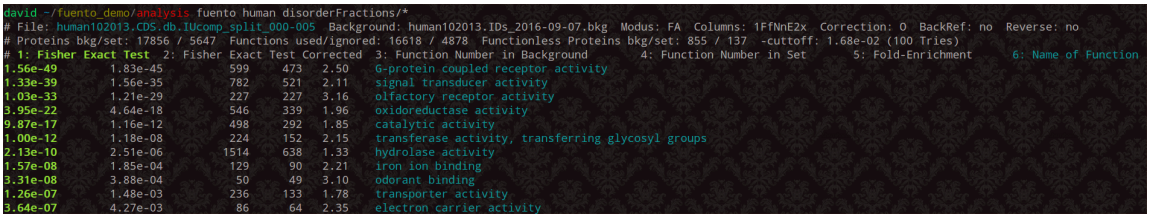
[d] www.uniprot.org/help/programmatic_access#id_mapping_examples
[e] www.uniprot.org/help/uniprotkb_column_names

2 Analysis

2.1 Simple Analysis

You will likely not find a functional enrichment tool with a simpler analysis API. Just do:

```
fuento BKG SET(S)
```



'BKG' can be an actual background file or a number or label pointing to it (fuento -l). The 'set' file or files are a collection of whitespace separated UniProt IDs.

Output columns can be modified to display a range of tests, correction and information about the functions annotated. Following the "*--columns / -c*" argument, a string of the following characters modifies output behaviour:

- F: Fisher's exact test
- f: Fisher's exact test with multiple-hypothesis correction
- B: binomial test
- b: binomial test with multiple-hypothesis correction
- H: hypergeometric test
- h: hypergeometric test with multiple-hypothesis correction
- E : fold-enrichment
- N: number of functions in Background
- n: number of functions in set
- x: short explanation of function (function name)
- X: long explanation of function
- P : protein IDs
- G : Gene Ontology IDs
- 1-5: highlight next column: 1:green 2:cyan 3:magenta 4:yellow 5:red

Standard output as in the image above uses the format string "*1FfNnE2x*"

2.2 Elaborate Analysis

There are several options to modify the output. This is an example for an elaborate analysis:

```
fuento --namespace "FC" --columns "bFG" --correction C 0.01
  --filter ".*[RD]NA.*" --cutoff 0.05 --sep "," --auto_output BKG SETS
```

Here a .csv file was generated, containing only enrichment in functions annotated as "molecular function" or "cellular component", containing either "RNA" or "DNA" in their function definition and enriched with a p-value <0.05 (generated from a binomial test with a Benjamini-Hochberg correction at a false-discovery rate of 0.01). In addition to the sorted column, columns are saved with uncorrected p-values using Fisher's exact test as well as the Gene Ontology IDs. It is automatically saved in a .csv file named after the consensus-name of the sets. All options are covered below.

3 Utility

3.1 Customization

Fuento ships with a set of defaults, designed for having a quick look at a given gene set. As demonstrated above, its behaviour can be heavily modified, according to usage. Setting multiple command line options for every job however is cumbersome. Therefore, we included the *--defaults* argument, which lets you set standard settings fitting your needs (for instance, that of the section above). To set a setting do:

```
fuento --namespace "FC" --columns "bFG" --correction C 0.01
  --filter ".*[RD]NA.*" --cutoff 0.05 --sep "," --auto_output --defaults
```

and use it to generate the same results as in the above section with:

```
fuento BKG SETS
```

Factory defaults can be restored using the *--defaults* argument on its own:

```
fuento --defaults
```

Default files are saved in the *fuento* home directory ("/home/USER/.fuento/defaults.dat") and can be saved for later use.

3.2 Custom Backgrounds

As mentioned earlier, *fuento* can use any background consisting of lines of identifiers (gene IDs or any strings) followed by tab-separated functions (Gene Ontology IDs or any strings). This makes it easy to generate custom backgrounds for non-standard enrichment analysis. For example, it is easy to create a background for stop-codon usage. It can be generated from a file of gene IDs and their

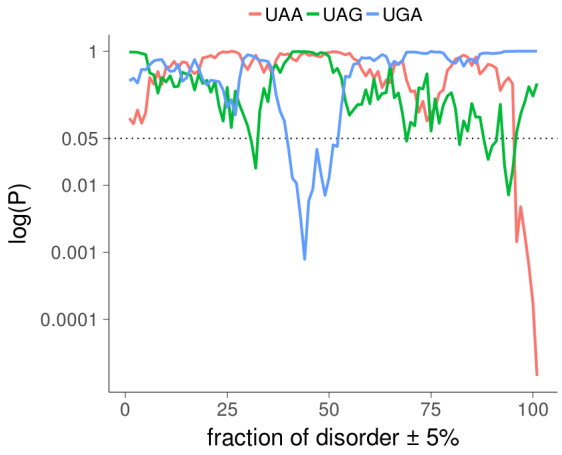
coding sequences using simple shell-commands and added to the *fuento* repository using the "*--add / -L* " argument. For easy processing, it is useful to display only the actual functions, save the set description. Set descriptions can be removed in downstream processing, since they are headed by the comment symbol *#*. Sorting by function name gives a consistent order to function enrichment values and further eases downstream processing. To alphabetically sort by function explanation (column-character: *x*), display all functions and remove the header, do:

```
fuento --columns "xF" --all --no_header SP_CODON_BKG SETS
```

Below a typical output is shown, generated on sets of human genes divided by fraction of disorder into 100 groups with ± 5% disordered residues:

```
# human102013.CDS.db.IUcomp_split_000-010
UAA     4.03e-01
UAG     4.74e-01
UGA     6.25e-01
# human102013.CDS.db.IUcomp_split_001-011
UAA     3.23e-01
UAG     4.44e-01
UGA     7.17e-01
# human102013.CDS.db.IUcomp_split_002-012
UAA     4.97e-01
UAG     2.37e-01
UGA     7.43e-01
# human102013.CDS.db.IUcomp_split_003-013
UAA     3.20e-01
UAG     4.05e-01
UGA     7.49e-01
```

Lines without enrichment columns are commented by a number sign (*#*) to facilitate downstream processing using shell scripts. A visualisation of the resulting p-value plot is shown below.



3.3 ID Mapping

Analysis does not need to be done exclusively on gene IDs. When using custom files, the identifier must not even be a protein ID – one can run an analysis of e.g. protein motives as identifier with phenotypes as functions. In this case, specify your own background as a file of lines headed by your identifier, followed by tab-separated functions.

When analysing gene IDs with of a different type than UniProt IDs, it is useful to map set-files before running analysis. When using the "*--map / -M*" option and no background option is set, all arguments following GENE_ID will be assumed to be set files of type GENE_ID to be mapped to UniProt accession ID for analysis:

```
fuento -M GENE_ID SET(S)
```

```
david ~/fuento_demo/analysis fuento -M P_ENTREZGENEID escherichia102013.EGID_0*
Mapping P_ENTREZGENEID entries from escherichia102013.EGID_00 to UniProt Accession (ACC)
 1332/1332      100.00%
Creating file: escherichia102013.EGID_00_P_ENTREZGENEID
Mapping P_ENTREZGENEID entries from escherichia102013.EGID_01 to UniProt Accession (ACC)
 1270/1270      100.00%
Creating file: escherichia102013.EGID_01_P_ENTREZGENEID
1  element(s) could not be mapped to UniProt IDs (ACC):
7910
Mapping P_ENTREZGENEID entries from escherichia102013.EGID_02 to UniProt Accession (ACC)
 1396/1396      100.00%
Creating file: escherichia102013.EGID_02_P_ENTREZGENEID
```

GENE_ID can be omitted and replaced with an "+" if the user is not sure about the ID type. *Fuento* will complain if IDs cannot be mapped to UniProt accession IDs, or if mapping is ambiguous, i.e. yielding more than one valid UniProt ID. This, for instance is the case when using gene names, which are shared over a variety of species. In this case, *fuento* pics the first valid ID. To prevent this, filters can be applied to the mapping via the header of the ID file, in the style of a UniProt query[f]. To obtain entries for *Escherichia coli (strain K12)* only, one would head the ID file with:

```
# organism:83333
```

or simply:

```
# ECOLI
```

3.4 Updates

Databases change and proteins are newly annotated. Therefore it comes in handy to update everything once in a while with:

```
fuento -u
```

[f] www.uniprot.org/help/text-search

This generates new backgrounds with new date-labled names. Of course, your old files are kept for consistency.

3.5 Merge Backgrounds

In case you want to use functions which are not part of the Gene Ontology in-line with common ones, you can merge backgrounds with your custom file. The file should contain lines headed by an identifier and followed by tab-separated strings which can contain spaces.

```
fuento -m bkg1 bkg2 operator out
```

The merge can result in the union (U) of backgrounds 1 and 2, their intersection (I) or the extension (E) of background 1 by 2. The 'E' operator can also be used to cut down an existing background by specifying a file of IDs that you want to keep as background 1. If the operator is in lowercase, the new background is not added to the list, so your list file does not explode when background generation is automated.

4 Exhaustive List of Options

```
usage: fuento [OPTION...] [ BACKGROUND [ SET... ] ]
options:
 -n --namespace <STRING>    Gene Ontology namespace(s) displayed.
    default: '--namespace FA'. Components of STRING:
      F:  molecular function
      C:  cellular component
      P:  biological process
      A:  aberrant function
 -c --columns <STRING>      columns to display and statistical test(s) to perform.
    Results will be sorted by first test column.
    default: '--columns 1FfNnE2x'. Components of STRING:
      F/f:  Fisher's exact test / with multiple-hypothesis correction
      B/b:  binomial test       / with multiple-hypothesis correction
      H/h:  hypergeometric test / with multiple-hypothesis correction
       E :  fold-enrichment
      N/n:  number of functions in Background / number of functions in set
      x/X:  short/long explanation of function
       P :  protein IDs
       G :  Gene Ontology IDs
      1-5:  highlight next column: 1:green 2:cyan 3:magenta 4:yellow 5:red
 -C --correction <STRING>   multiple-hypothesis correction
    default: '--correction B'. Possible values for STRING:
       B:           Bonferroni correction, returns corrected p-values
       C <NUMBER>:  Benjamini-Hochberg FDR (False Discovery rate) correction,
                    returns corrected p-values for a given FDR=NUMBER
       A <NUMBER>:  Benjamini-Hochberg-Yekutieli FDR adjustment,
                    returns adjusted p-values for a given FDR=NUMBER
       T <NUMBER>:  Benjamini-Hochberg test, returns only 0=significant,
                    1=insignificant for a given FDR=NUMBER.
                    If a Benjamini-Hochberg test corrected column leads,
                    it disables permutation tests, and displays
                    all significant numbers.
 -f --filter <STRING>       filter for GOIDs or plaintext functions
                    separated by semicolon.
    If FILTER(s) contains whitespace, enclose in quotes. Set '--all' to see all
    possible functions. Plaintext functions can be filtered by regex.
 -s --slim <FILE/NUMBER/LABEL>
    use GO slim instead of full go.obo. Slims can be viewed with '-l'
 -x --cutoff <NUMBER>       cutoff for first column (p-value, fold-enrichment,
    function number). Can be supplied in scientific or standard notation.
    Applied only to first test column
 -r --reverse               reverse enrichment, display
    functional depletion for all columns.
 -a --all                   display all enrichments.
```

```
-m --max <NUMBER>          max number of functions displayed (overrides -t)
-t --trial_number <NUMBER> random trial number (default: '--trial_number 100').
   If all functions should be displayed: '--trial_number 0'.
-G --global_minimum        print only minimum p-value of each set
-H --no_header             do not display header explaining columns.
   Will print only filename of set instead.
-e --obsolete              do not ignore obsolete go entries
-A --back_reference        back-reference functions ('is_a:' marker in go.obo)
   Only recommended if background was made by hand, since '-b' accounts for that.
-S --sep <STRING>          specify separator (default: '-sep "\t"')
-d --defaults              set current options as custom defaults.
   Use without options to restore defaults. Defaults are setable for
   all arguments listed above, but not for the ones listed below.
   To view currently set defaults do '-h'.
-b --background <FILE>     create background file from uniprot-accid file.
   Downloads newest annotation from ebi server. This may take a while.
-B --background_function <FILE> <STRING>
   Create background file from uniprot-accid file with function type STRING.
   STRING should be a uniprotkb column name explained here:
   'www.uniprot.org/help/uniprotkb_column_names'.
-M --map <STRING> [<FILE(S)>]
   maps gene IDs of type STRING to uniprot-accid when creating backgrounds
   with '-b' or '-B', or maps FILE(S) to
   FILES_STRING. Available gene id types are found here:
   'http://www.uniprot.org/help/programmatic_access#id_mapping_examples'
   If several ids can be mapped to, the first in the list is used.
-l --list                  lists available background files and slims.
   columns: number, entries, explanation, [label], file
-L --add <FILE> <LABEL>    adds FILE to list of background/slim files
   with custom LABEL.
   Slim FILEs must have the ending '.obo'. to remove entry, delete or rename file.
-R --merge <FILE_A> <FILE_B> <OPERATOR> <STRING>
   merge two backgrounds FILEs A/B and save as file STRING.
   Possible values for OPERATOR:
       'u': union of a and b, keep all IDs and functions.
       'i': intersection of a and b, keep IDs that occur in both sets.
       'e': extend a by b, keep only ids of a, extend functions from b.
   If OPERATOR is lowercase, background is not added to list
   (useful for scripting).
-g --get                   get newest Gene Ontology database (go.obo)
   from uniprot server. saves to "~/.fuento"
-u --update                update everything, including Gene Ontology
   database and backgrounds.
   Generates new background files in the same folder with a new timestamp.
-o --output <FILE>         specify output FILE instead of stdout.
   Will be saved as ".enr"
```

```
  -O --auto_output          generate output file of the longest common prefix
     of set names.
  -v --version              display version and exit
  -h --help                 display this and exit
background: <FILE/NUMBER/LABEL>
set: <FILES/FILES*>
```

Background files can be specified by a filename, number of the background in the list or label (see with -l). Labels can be added to backgrounds and GO slims with the '-L' option or by heading the file with: '# LAB:[tab]label'. Using wildcards and globbing is possible since all strings following the background will be assumed sets. Piping whitespace separated Uniprot IDs into *fuento* is an alternative to specifying protein sets. If piping is used, sets can be divided by the word 'END'. For each protein, *fuento* counts a function exactly once, even if the back-reference-flag is set. By default, *fuento* limits the output to results, which are significant in a permutation test using 100 random protein sets.

Using wildcards and globbing is possible since all strings following the background will be assumed sets. Piping whitespace separated Uniprot IDs into *fuento* is an alternative to specifying protein sets. If piping is used, sets can be divided by the word 'END'.

5 Theory

Gene-sets are compared with their corresponding proteomes by their gene's functional annotations. These annotations are curated and compiled by the Gene Ontology project.[1,2] The Gene Ontology is a tree-graph of gene functions in which nodes are related to each other by order of their functions (Figure 1). To cite the GO website: "[...] the biological process term hexose biosynthetic process has two parents, hexose metabolic process and monosaccharide biosynthetic process. This is because biosynthetic process is a subtype of metabolic process and a hexose is a subtype of monosaccharide.".[4]

For comparison of such annotations and their relative frequency, there already is software published, but none of the available tools satisfy the need for high-throughput exploratory calculations tightly interwoven with shell scripts. Thus we defaulted do developing our own tool described here. Below, the statistical tests and corrections used in *fuento* as well as their implementation are discussed.

5.1 Hypergeometric Test

The *hypergeometric distribution* describes the probability of drawing k successes in n draws without replacement from a population with size N containing K successes.

$$P(n,k) \;=\; \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{1}$$

Binomial coefficients expand to the number of ways to choose k elements out of a total of n elements disregarding their order.

$$\binom{n}{k} \;=\; \frac{n!}{k!(n-k)} \tag{2}$$

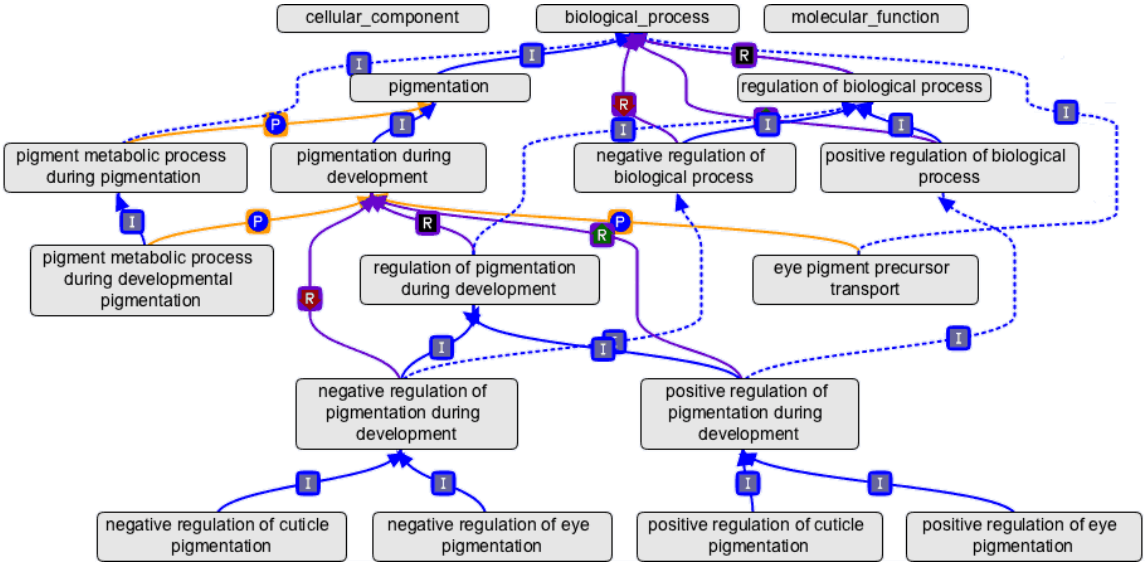


Figure 1: **Example for a subset of the Gene Ontology database.** The following diagram is a screenshot from the ontology editing software *OBO-Edit*, showing a small set of terms from the ontology. A set of terms under the biological process node pigmentation. In the diagram, relations between the terms are represented by the colored arrows; the letter in the box midway along each arrow is the relationship type. Note that the terms get more specialized going down the graph, with the most general terms – the root nodes, cellular component, biological process and molecular function – at the top of the graph. Terms may have more than one parent, and they may be connected to parent terms via different relations. — figure and text adapted from *geneontology.org*[1, 2, 4]

For a 2×2 contingency table, the hypergeometric distribution can be abstracted in the following way:

	Drawn	Not drawn	Margin
Success	k	K-k	K
Failure	n-k	N+k-n-K	N-K
Margin	n	N-n	N

OR

	Set 1	Set 2	Margin
Category 1	a	b	a+b
Category 2	c	d	c+d
Margin	a+c	b+d	t

Using Equation 1, one is able to define the probability mass function (PMF) for the table above:

$$PMF(a,b,c,d) \;=\; \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{t}{a+c}} \;=\; \frac{(a+b)!\;(c+d)!\;(a+c)!\;(b+d)!}{a!\;b!\;c!\;d!\;t!} \tag{3}$$

Assuming that items of Set 1 and Set 2 are equally likely to fall into category 1 or 2, one can calculate the probability of a given contingency table. In the case of functional enrichment, the probability of observing a given outcome for the division of a function X between a protein background and a subset thereof is calculated using such a table:

	subset	Background	Margin
Function	f_s	f_b	f_s+f_b
Not-Function	$n_s - f_s$	$n_b - f_b$	$n_s - f_s + n_b - f_b$
Margin	n_s	n_b	$n_s + n_b$

n_s	number of genes in set
n_b	number of genes in background
f_s	number of genes with function X in set
f_b	number of genes with function X in background

To efficiently generate all these contingency tables, a list of logarithmic factorials is dynamically generated to the necessary extent. This list is used to efficiently sum and subtract factorials in place of the multiplications and divisions in Equation 3. The reason for this is not only speedup, but also a need to avoid the danger of overflowing 64-bit floating point numbers. The algorithm was inspired by SAVI (statistical algorithm for variant identification).[5]

5.2 Fisher's Exact Test

Fisher's exact test[6] gives the probability of divergence from a hypergeometric distribution of functions and is most widely used in functional enrichment. It can be understood as the probability to find a similar or better outcome by drawing proteins randomly from the background. As above, *fuento* builds a contingency table for each function using the function's counts for the background and the subset and the sum of all other functions in both background and subset. The significance level for a given table is generated by adding the hypergeometric probabilities of all contingency tables with the same margin totals as the observed one and similar or more extreme outcomes. To keep margin totals constant, all fields of the table have to be changed according to a change x in the first field:

	Subset	Background	Margin
Function	$A = x$	$B = a + b - x$	$a + b$
Not-Function	$C = a + c - x$	$D = d - a + x$	$c + d$
Margin	$a + c$	$b + d$	n

$$n = a + b + c + d$$

The change in field D can be explained as follows:

$$\begin{aligned} \Delta a &= -a + x \\ \Delta b &= a - x \\ \Delta c &= a - x \\ \Delta d &= -\Delta b = -\Delta c = -a + x \end{aligned} \tag{4}$$

P-values are generated by summation of hypergeomteric probabilities of different tables more extreme than the case tested. For functional enrichment, all tables are added up in which $a = f_s$ is less or equal to $A = f_s$ for the case tested. For functional depletion it is the other way around:

$$\begin{aligned} enrichment: \quad P(a \leq A) &= \sum_{x=0}^{x \leq A} PMF(x, a+b-x, a+c-x, d-a+x) \\ depletion: \quad P(a \geq A) &= \sum_{x=A}^{x \leq n} PMF(x, a+b-x, a+c-x, d-a+x) \\ &\forall x \in \mathbb{N}(a+b-x \geq 0, a+c-x \geq 0, d-a+x \geq 0) \end{aligned} \tag{5}$$

Implemented is also a buffer for already calculated contingency-table/p-value pairs, effectively speeding up bulk analysis.

5.3 Binomial Test

Like in the above test, the *binomial test* calculates p-values as sums of probabilities, replacing the hypergeometric with the binomial distribution. Its probability mass function is defined as the probability of drawing k successes in n draws if each draw has a probability of success of p:

$$PMF(n, k, p) = \binom{n}{k} p^k (1-p)^{n-k} \tag{6}$$

Using the table of precalculated log-factorials, the logarithm of the binomial coefficient (Equation 1) is calculated in a computationally inexpensive and overflow-safe way:

$$log\binom{n}{k} = log!(n) - log!(k) - log!(n-k) \tag{7}$$

Similarly, the binomial PMF (Equation 6) can be calculated using the logarithm of the binomial coefficient.

$$PMF(n, k, p) = exp\left(log\binom{n}{k} + log(p) * k + log(1-p) * (n-k)\right) \tag{8}$$

Similar to Fisher's exact test, the binomial test's p-value is the sum of binomial probabilities of equal or better outcomes when testing for functional enrichment, and the sum of binomial probabilities

of equal or worse outcome when testing for functional depletion. With N being the number of genes in the set, K being the number of genes with the function tested and P being the ratio of genes with the function and without it in the background, the p-value is calculated as:

$$\begin{aligned} enrichment: \quad P(X \leq P) &= \sum_{k=0}^{k<=K} PMF(N,k,P) \\ depletion: \quad P(X \geq P) &= \sum_{k=K}^{k<=N} PMF(N,k,P) \end{aligned} \tag{9}$$

5.4 Bonferroni Correction

Since in functional enrichment analysis several thousands of comparisons are done per set, to prevent *false discoveries*, one has to apply *multiple hypothesis correction*. The significance level α for a single observation with p-value p when testing n hypothesis can be corrected to α' like this:

$$\alpha' = \frac{\alpha}{n} \tag{10}$$

Alternatively to adjusting the α-level, p-values themselves can be adjusted to be rejectable with a given α:

$$p' = pn \tag{11}$$

The Bonferroni correction is known to be conservative in the case of a large number of tests performed or the tested statistics are correlated.

5.5 Benjamini-Hochberg Test

A less conservative alternative to the Bonferroni method is the Benjamini-Hochberg correction.[7] This significance test is implemented in *fuento* as a step-up procedure, since it depends on sorting p-values, and most p-values are in the less significant range in a regular functional enrichment analysis.

P-values p_1 - p_n are sorted in ascending order and a cutoff defined as the smallest p-value p_i $(0 < i \leq n)$ observing the following inequality:

$$p_i \leq \frac{i}{n}\alpha \tag{12}$$

For a given α-level, *fuento* annotates functions as 1 for non-significant and 0 for significant.

5.6 Benjamini-Hochberg FDR Correction

Alternatively to a significance test, one could want to find the minimal α-level, at which an original p-value would still be rejected. The corrected q-value is defined like this:

$$q_i = \frac{p_i n}{i} \tag{13}$$

5.7 Benjamini-Hochberg-Yekutieli FDR Adjustment

The above FDR correction is, however, not a monotonic function of p_i. Yekutieli et al.[8] solved this by defining each adjusted q-value, so that:

$$\begin{aligned} q'_i &= min(q_j) \\ \forall i \in \mathbb{N}(j \geq i) \end{aligned} \tag{14}$$

5.8 Permutation Test

The p-value cutoff for a function displayed by *fuento*, if not otherwise specified, is calculated from randomly choosing N protein sets of the same size as the subset in question and calculating their p-values using the specified test with the specified correction. The average of the lowest p-value generated is used as the cutoff for the real subset. If not otherwise specified, N=100.

6 Remarks

If you are using *fuento* in your work please cite the publication.[9]
Contact us at david.t.weichselbaum@gmail.com, newant@gmail.com, bojan.zagrovic@univie.ac.at.

References

[1] J. A. Blake, K. R. Christie, M. E. Dolan, et al. Gene Ontology consortium: Going forward. *Nucleic Acids Research*, 43(D1):D1049–D1056, 2015.

[2] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(may):25–29, 2000.

[3] T. U. Consortium. Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Research*, 40(D1):D71–D75, 2012.

[4] G. O. Consortium. Ontology Structure, geneontology.org/page/ontology-structure, 2015.

[5] V. Trifonov, L. Pasqualucci, E. Tiacci, B. Falini, and R. Rabadan. SAVI: a statistical algorithm for variant frequency identification. *BMC Systems Biology*, 7 Suppl 2(Suppl 2):S2, 2013.

[6] R. A. Fisher. On the Interpretation of χ^2 from Contingency Tables, and the Calculation of P. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

[7] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57(1):289–300, 1995.

[8] D. Yekutieli and Y. Benjamini. Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics. *Journal of Statistical Planning and Inference*, 82:171–196, 1999.

[9] D. Weichselbaum, B. Zagrovic, and A. A. Polyansky. Fuento: functional enrichment for bioinformatics. *TBA.*, 2016.