

Enrichment analysis: Determining [over|under]-representation

How do we know that what we are looking at is interesting?

Olivier Dameron, Alban Gaignard, Pierre Larmande



Version 1.0



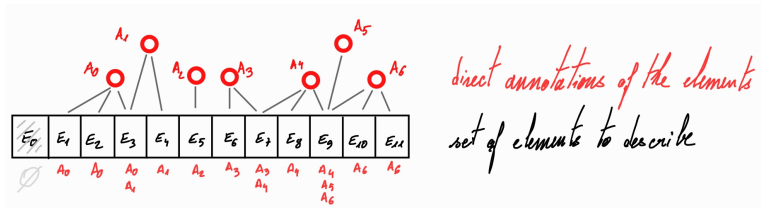
Outline

- 1 What are we trying to do? The big picture
 - Annotations describe elements
 - Ontologies reveal a cascade of new connections
 - Which annotations are relevant?
 - Where does over-representation starts?
- 2 How does it work? Formalizing over/under-representation
 - The good questions
 - Choice of a reference set
 - Choice of a distribution
 - Hypergeometric test
- 3 How to do it? Analyzing a set of proteins
 - Method
 - Example
 - Correcting multiple testings: Bonferroni
- 4 Synthesis

What are we trying to do? The big picture

- 1 What are we trying to do? The big picture
 - Annotations describe elements
 - Ontologies reveal a cascade of new connections
 - Which annotations are relevant?
 - Where does over-representation starts?

Annotations are useful for describing elements

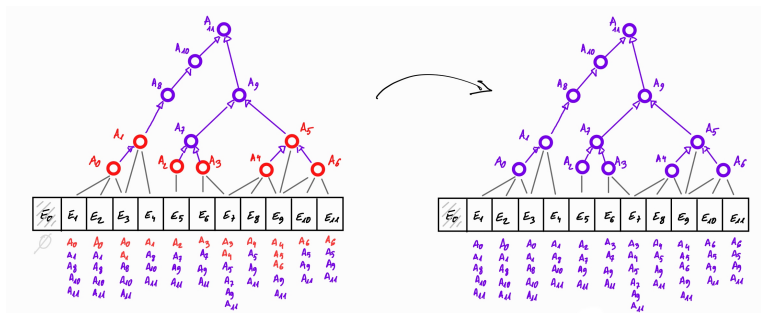


- some elements are not annotated (e.g. E_0)
- some elements have multiple annotations (e.g. E_3)
- some annotations describe multiple elements (e.g. A_0)

Now we can find new relations between some elements

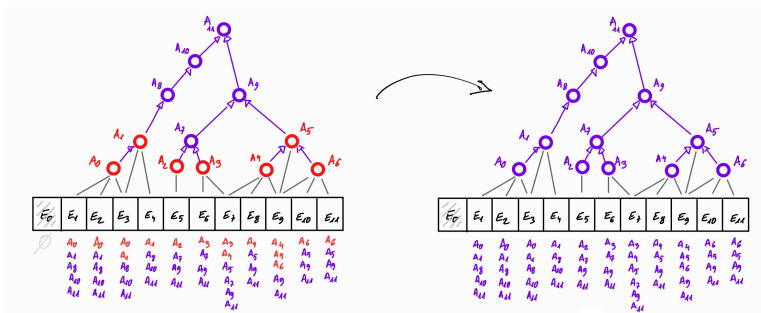
- E_7 and E_8 (and E_9) share A_4
- E_7 and E_6 share A_3
- note that E_6 and E_8 have nothing in common

Ontologies reveal a cascade of new connections



- ontology: explicit hierarchy of (dependent) classes
- the elements associated to an annotation are also implicitly associated to all the ancestors of this annotation
 - some direct annotations (A5) are also indirectly associated to other elements (those of A4 and A6)
 - some new annotations (A7–A11) should also be considered

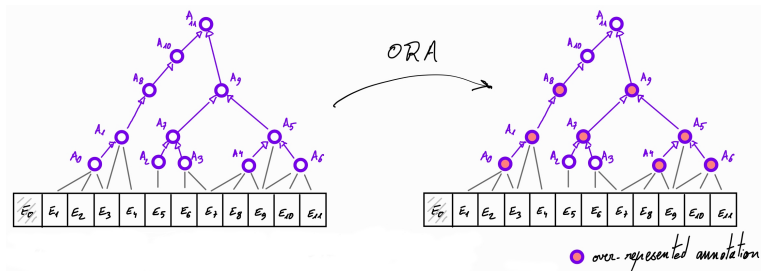
Ontologies reveal a cascade of new connections



Using ontologies, new connections become explicit

- between annotations (e.g. $A0 \rightarrow A1$)
- between annotations and elements ($E1 - A1$)
- between elements (in fact, $E6$ and $E8$ share $A9$ and $A11$)

Which annotations are relevant?



- A1–A11 describe the elements, not the set
- not all the annotations are relevant
 - too precise and only annotate a few elements (e.g. A2, A3)
 - too broad, annotate most of the elements but are not informative (e.g. A10, A11)

Over-representation analysis (ORA)

identifies the annotation that are associated to more elements than we would expect in a random set of the same size

Intuition of over-representation

A species has 20.000 proteins

- 2.000 proteins annotated by A (10%)
- 15.000 proteins annotated by B (75%)

The study S1 identified 12 proteins of interest

- 6 proteins annotated by A (50%)
 - 9 proteins annotated by B (75%)
-
- the proportion of proteins annotated by B corresponds (+/-) to the proportion of proteins observed in the reference
 - the proportion of proteins annotated by A is “obviously” superior to what we would expect

Where is the difference between normal variation and over-representation?

- A species has 20.000 proteins (background or reference)
 - 2.000 proteins annotated by A (10%)
 - 15.000 proteins annotated by B (75%)
- The study S1 identified 12 proteins of interest
 - 6 proteins annotated by A (50%)
 - 9 proteins annotated by B (75%)

The study S2 identified 50 proteins of interest

- 6 proteins annotated by A (12%)
- 36 proteins annotated by B (72%)

Is 12% really different from 10% (annotation A)?
and 72% compared to 75% (annotation B)?

Where do we draw the line?

How does it work? Formalizing over/under-representation

- 2 How does it work? Formalizing over/under-representation
 - The good questions
 - Choice of a reference set
 - Choice of a distribution
 - Hypergeometric test

The good questions...

- for each annotation, is its frequency in the study similar to its background frequency in the species? (χ^2 ?, Student's test?)
- if we find a difference, how likely is this difference to happen by random?
 - **probability**: if we randomly pick a set of N proteins among M , how likely are we to observe n proteins with the characteristics of interest?
 - **p-value**: if we randomly pick a set of N proteins among M , how likely are we to observe **at least** n proteins with the characteristics of interest?
- obviously, the choice of the reference set (M) is critical

Reference set

- **M** items total
- **m** items having the characteristics of interest

Set of interest (\subset set of reference)

- **N** items total
- **n** items having the characteristics of interest

If I randomly select **N** proteins among **M**, what is the probability (**p-value**) of encountering **at least n** proteins having the characteristics of interest?

Choosing a reference set

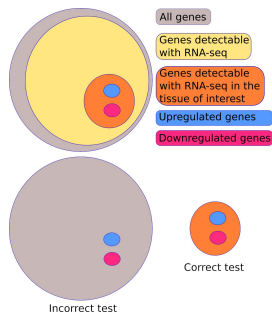
What is your reference set?

- all the species' proteins?
- all the proteins of the study?
- all the proteins known to be involved in the disease, the pathway,... of interest?
- ...

It really depends on what you are doing

Your choice will affect the value of M and m

Choosing the correct reference set



“In 2021, ~9k journal articles used enrichment tests and we estimate only 5% used a background list correctly and describe it in the methods”

<https://twitter.com/mdziemann/status/1532132417188012032>

> [PLoS Comput Biol.](#) 2022 Mar 9;18(3):e1009935. doi: 10.1371/journal.pcbi.1009935.
eCollection 2022 Mar.

Urgent need for consistent standards in functional enrichment analysis

[Kaumadi Wijesooriya](#)¹, [Sameer A Jadaan](#)², [Kaushalya L Perera](#)¹, [Tanuveer Kaur](#)¹,
[Mark Ziemann](#)¹

Affiliations + expand

PMID: 35263338 PMCID: [PMC8936487](#) DOI: [10.1371/journal.pcbi.1009935](#)

[Free PMC article](#)

Abstract

Gene set enrichment tests (a.k.a. functional enrichment analysis) are among the most frequently used methods in computational biology. Despite this popularity, there are concerns that these methods are being applied incorrectly and the results of some peer-reviewed publications are unreliable. These problems include the use of inappropriate background gene lists, lack of false discovery rate correction and lack of methodological detail. To ascertain the frequency of these issues in the literature, we performed a screen of 186 open-access research articles describing functional enrichment results. We find that 95% of analyses using over-representation tests did not implement an appropriate background gene list or did not describe this in the methods. Failure to perform p-value correction for multiple tests was identified in 43% of analyses. Many studies lacked detail in the methods section about the tools and gene sets used. An extension of this survey showed that these problems are not associated with journal or article level bibliometrics. Using seven independent RNA-seq datasets, we show misuse of enrichment tools alters results substantially. In conclusion, most published functional enrichment studies suffered from one or more major flaws, highlighting the need for stronger standards for enrichment analysis.

Choosing a distribution

Binomial distribution

- assumes that the probability of picking an item with the characteristics of interest is fixed (m/M)
- OK for large reference sets

Hypergeometric distribution

- picking an item with the characteristics of interest affects the probability of doing so again in the following picks
- discrete probability distribution that describes the number of successes in a sequence of draws without replacement from a finite population
- OK for small reference sets

Yield similar results, in practice go for hypergeometric

Probability of randomly picking n proteins with the characteristics of interest among N

$$P(n) = \frac{\binom{m}{n} \cdot \binom{M-m}{N-n}}{\binom{M}{N}}$$

Probability: nb of favorable events / nb of events

- Number of favorable events
 - $\binom{m}{n}$ ways of picking n proteins among the m of interest
 - $\binom{M-m}{N-n}$ ways of picking the $N-n$ remaining proteins among the $M-m$ proteins that do not have the characteristic of interest
- Total number of events
 - $\binom{M}{N}$ ways of picking N proteins among M

Reminder: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ ways of picking k items among n

Where is the difference between normal variation and over-representation? **Hypergeometric test**

- A species has 20.000 proteins (background or reference)
 - 2.000 proteins annotated by A (10%)
 - 15.000 proteins annotated by B (75%)
- The study S1 identified 12 proteins of interest
 - 6 proteins annotated by A (50%) → p-value = 0.00053
 - 9 proteins annotated by B (75%) → p-value = 0.64880
- The study S2 identified 50 proteins of interest
 - 6 proteins annotated by A (12%) → p-value = 0.38385
 - 36 proteins annotated by B (72%) → p-value = 0.74831

Where is the difference between normal variation and over-representation? **Hypergeometric test** in Python

- A species has 20.000 proteins (background or reference)
 - 2.000 proteins annotated by A (10%)
 - 15.000 proteins annotated by B (75%)
- The study S1 identified 12 proteins of interest

```
1 >>> import scipy.stats as stats
2 >>> print stats.hypergeom.sf(6-1, 20000, 2000, 12)
3 0.00053836334315
4 >>> print stats.hypergeom.sf(9-1, 20000, 15000, 12)
5 0.648797987327
```

- The study S2 identified 50 proteins of interest

```
1 >>> import scipy.stats as stats
2 >>> print stats.hypergeom.sf(6-1, 20000, 2000, 50)
3 0.38385375643
4 >>> print stats.hypergeom.sf(36-1, 20000, 15000, 50)
5 0.748314759755
```

How to do it? Analyzing a set of proteins

- 3 How to do it? Analyzing a set of proteins
 - Method
 - Example
 - Correcting multiple testings: Bonferroni

Analyzing a set of proteins: method

Let G be a set of proteins

- 1 compute the set of ontology terms A annotating directly or indirectly (the ancestors) at least one of the proteins of G
- 2 for each ontology term a_i in A :
 - determine nb of proteins of G annotated by a_i
 - determine nb of proteins of the reference set annotated by a_i
 - compute an hypergeometric test and determine the p-value
- 3 sort the ontology terms by increasing p-value (the smaller the better)
- 4 determine the p-value cut-off (typically $\frac{0.05}{|A|}$), cf. Bonferroni correction

Example 1: set of proteins (from Heme biosynthesis pathway)

Set 1

- P06132
- P08397
- P10746
- P13196
- P13716
- P22557
- P22830
- P36551
- P50336
- Q12887
- Q7KZN9

Example 1: ask Uniprot what these proteins are

<http://sparql.uniprot.org/>

```
1 PREFIX uniprot: <http://purl.uniprot.org/uniprot/>
2 PREFIX upc:<http://purl.uniprot.org/core/>
3
4 SELECT ?prot ?label
5 WHERE
6 {
7   VALUES ?prot {
8     uniprot:P06132 uniprot:P08397 uniprot:P10746 uniprot:P13196
9     uniprot:P13716 uniprot:P22557 uniprot:P22830 uniprot:P36551
10    uniprot:P50336 uniprot:Q12887 uniprot:Q7KZN9
11  }
12  ?prot upc:mnemonic ?label .
13 }
```

Example 1: protein names according to Uniprot

<http://sparql.uniprot.org/>

P06132	DCUP_HUMAN
P08397	HEM3_HUMAN
P10746	HEM4_HUMAN
P13196	HEM1_HUMAN
P13716	HEM2_HUMAN
P22557	HEM0_HUMAN
P22830	HEMH_HUMAN
P36551	HEM6_HUMAN
P50336	PPOX_HUMAN
Q12887	COX10_HUMAN
Q7KZN9	COX15_HUMAN

Example 1: retrieve the associated biological processes from GOA (and GO for ancestors)

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX uniprot: <http://purl.uniprot.org/uniprot/>
4 PREFIX goavoc: <http://bio2rdf.org/goa_vocabulary:>
5
6 SELECT DISTINCT ?bp ?bpLabel (count(?prot) as ?nbGenes)
7 WHERE {
8   { SELECT DISTINCT ?prot ?bp WHERE {
9     VALUES ?prot {
10      uniprot:P06132 uniprot:P08397 uniprot:P10746 uniprot:P13196
11      uniprot:P13716 uniprot:P22557 uniprot:P22830 uniprot:P36551
12      uniprot:P50336 uniprot:Q12887 uniprot:Q7KZM9
13    }
14    ?prot goavoc:process/
15      (rdfs:subClassOf[](rdfs:subClassOf/owl:someValuesFrom))* ?bp .
16    ?bp rdf:type owl:Class .
17  }
18 }
19 OPTIONAL { ?bp rdfs:label ?bpLabel . }
20 }
21 GROUP BY ?bp ?bpLabel
22 ORDER BY DESC(?nbGenes)
```

Example 1: biological processes annotating the proteins according to GOA (and GO for ancestors)

	?bp	?bpLabel	?nbGenes
1	go:0006725	"cellular aromatic comp. metab. proc."	11
2	go:0006778	"porphyrin-cont. comp. metab. proc."	11
3	go:0006779	"porphyrin-cont. comp. biosynth. proc."	11
4	go:0006783	"heme biosynthetic process"	11
5	go:0006807	"nitrogen compound metabolic process"	11
6	go:0008150	"biological process"	11
7	go:0008152	"metabolic process"	11
8	go:0009058	"biosynthetic process"	11
9	go:0009987	"cellular process"	11
...
382	go:2000112	"regul. of cell. macrom. biosynth. proc."	1

Annotation count alone cannot discriminate the **relevant annotations** from the **boring ones**

Background: Human proteins and their annotations

There are 48.804 human proteins, 38.660 of which annotated by at least one biological process

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX uniprot: <http://purl.uniprot.org/uniprot/>
4 PREFIX uptaxo: <http://purl.uniprot.org/taxonomy/>
5 PREFIX upc: <http://purl.uniprot.org/core/>
6 PREFIX goavoc: <http://bio2rdf.org/goa_vocabulary:>
7
8 SELECT (count(DISTINCT ?prot) as ?nbGenes)
9 WHERE {
10   ?prot rdf:type upc:Protein .
11   ?prot upc:organism uptaxo:9606 .
12   #?prot goavoc:process/
13   #      (rdfs:subClassOf/(rdfs:subClassOf/owl:someValuesFrom))* ?bp .
14 }
```

Result files:

- queryResults_humanGenes_all.csv
- queryResults_humanGenes_annotated.csv

Background: Human proteins and their annotations

For each biological process, we can compute how many human proteins it annotates (directly or indirectly), and add the result to the graph

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX uniprot: <http://purl.uniprot.org/uniprot/>
4 PREFIX upc: <http://purl.uniprot.org/core/>
5 PREFIX goavoc: <http://bio2rdf.org/goa_vocabulary:>
6 PREFIX tf: <http://www.irisa.fr/dyliss/public/odameron/>
7
8 INSERT { ?bp tf:hasBackgroundOccurrences ?nbGenes . }
9 WHERE {
10     SELECT ?bp (count(DISTINCT ?prot) as ?nbGenes)
11     WHERE {
12         ?prot goavoc:process/
13             (rdfs:subClassOf|(rdfs:subClassOf/owl:someValuesFrom))* ?bp .
14     }
15     GROUP BY ?bp
16     ORDER BY DESC(?nbGenes)
17 }
```

Example1: compute annotations frequencies

Comparison of the number of proteins from set1 (among 11) annotated by a GO term VS the number of human proteins annotated by the same GO term (among 38.660)

```
1 SELECT DISTINCT ?bp ?bpLabel (count(?prot) as ?nbGenes) ?nbOccRef
2 WHERE {
3   { SELECT DISTINCT ?prot ?bp WHERE {
4     VALUES ?prot { uniprot:P06132 uniprot:P08397 uniprot:P10746
5       uniprot:P13196 uniprot:P13716 uniprot:P22557 uniprot:P22830
6       uniprot:P36551 uniprot:P50336 uniprot:Q12887 uniprot:Q7KZN9 }
7     ?prot rdf:type upc:Protein .
8     ?prot upc:organism uptaxo:9606 .
9     ?prot goavoc:process/
10      (rdfs:subClassOf|(rdfs:subClassOf/owl:someValuesFrom))* ?bp .
11     ?bp rdf:type owl:Class .
12   }
13 }
14 OPTIONAL { ?bp rdfs:label ?bpLabel . }
15 ?bp tf:hasBackgroundOccurrences ?nbOccRef
16 }
17 GROUP BY ?bp ?bpLabel ?nbOccRef
18 ORDER BY DESC(?nbGenes) ASC(?nbOccRef)
```

Example 1: annotations' frequencies

Comparison of the number of proteins from set1 (among 11) annotated by a GO term VS the number of human proteins annotated by the same GO term (among 38.660)

Result file: queryResults_proteinSet1.csv

	?bp	?bpLabel	?nbGenes	?nbOccRef
1	go:0006783	"heme biosynth. proc."	11	29
2	go:0006779	"porphyrin-cont. biosynth."	11	36
3	go:0042168	"heme metabolic process"	11	47
...	11	...
33	go:0009987	"cellular process"	11	31839
34	go:0008150	"biological process"	11	38660
35	go:0046501	"protoporphyrinogen ..."	9	11
...
382	go:0050789	"regul. of bio. pro."	1	18992

- go:0006783 annotates 11/11 proteins of set1, but only 29/38.660 of the reference proteins ⇒ **over-represented**
- go:0008150 annotates 11/11 proteins of set1, and 38.660/38.660 of the reference proteins ⇒ **not over-represented**

Example 1: annotations' over-representation

File computeOverRepresentation.py

```
1 import csv
2 import scipy.stats as stats
3
4 with open('queryResults_proteinSet1.csv') as fd:
5     reader = csv.reader(fd)
6     currentLine = reader.next()      # skip header
7     for data in reader:
8         goID = data[0].replace('<http://purl.obolibrary.org/obo/GO_', 'GO').replace('>', '')
9         goLabel = data[1].replace('\"', '')
10        goNbOccSet = int(data[2].replace('\"', ''))
11        goNbOccRef = int(data[3].replace('\"', ''))
12        setSize = 11
13        refSize = 38660
14
15        goPvalBinom = stats.binom_test(goNbOccSet, setSize,
16                                       float(goNbOccRef)/refSize, alternative='greater')
17        goPvalHypergeom = stats.hypergeom.sf(goNbOccSet-1, refSize,
18                                              goNbOccRef, setSize)
```

Example 1: annotations' over-representation

Over-representation of the GO terms annotating at least 1 of the proteins from set1 (11 proteins) compared to the reference set (38.660)

File overRepresentation_proteinsSet1.tsv

	GO ident	GO label	Binom. p-val	Hyperg. p-val
1	go:0006783	heme biosynth. proc.	4.23E-35	4.79E-36
2	go:0006779	porphyrin... biosynth. proc.	4.56E-34	8.32E-35
3	go:0042168	heme metab. proc.	8.57E-33	2.41E-33
...
33	go:0009987	cellular proc.	0.11	0.11
34	go:0008150	biological proc.	1.0	1.0
35	go:0046501	protoporphyrinog. met. proc.	6.71E-31	5.69E-33
...
382	go:0050789	regul. biol. proc.	0.99	0.99

- Binomial and hypergeometric tests p-values discriminate **over-expressed** GO terms (255 have p-value < 0.05)
- p-value(heme metab. proc.) < p-value(heme biosynth.) :-)

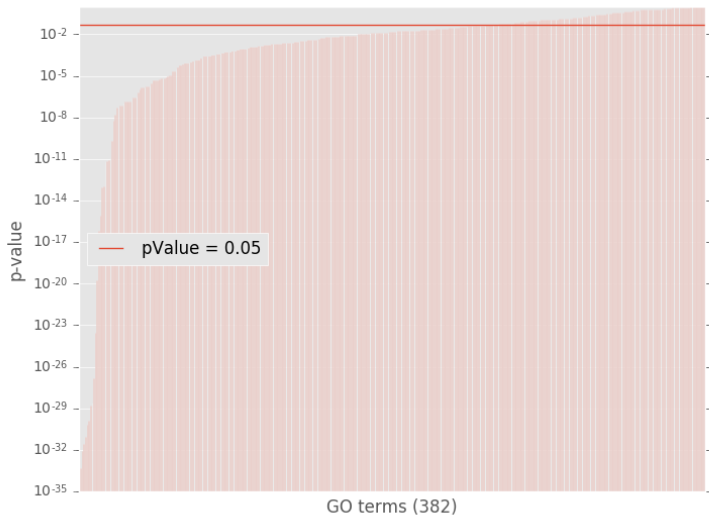
Example 1: annotations' over-representation

Among the 382 GO terms annotating at least one of the 11 proteins of the set of interest:

- Binomial test: 255 have a p-value < 0.05
- Hypergeometric test: 255 have a p-value < 0.05 (the same!)

see Bonferroni correction in a few slides

Example 1: annotations' over-representation



Filtered significant GO terms for Gene set1

File overRepresentedGOterms.pdf



Example 2: set of random proteins

Select 11 at random from the 38.660 annotated ones

Set 2 (proteinsSetRandom.txt)

- H3BS44
- Q15831
- Q9BVM4
- C9J8P9
- P22570
- H0YN65
- O75396
- Q9NX46
- P58400
- Q8TEW6
- G3V4B7

Script `selectRandomGenes.py` on
`queryResults_humanGenes_annotated.csv`

Example 2: ask Uniprot what these proteins are

<http://sparql.uniprot.org/>

C9J8P9	C9J8P9_HUMAN
G3V4B7	G3V4B7_HUMAN
H0YN65	H0YN65_HUMAN
H3BS44	H3BS44_HUMAN
O75396	SC22B_HUMAN
P22570	ADRO_HUMAN
P58400	NRX1B_HUMAN
Q15831	STK11_HUMAN
Q8TEW6	DOK4_HUMAN
Q9BVM4	GGACT_HUMAN
Q9NX46	ARHL2_HUMAN

Example 2: annotations' over-representation

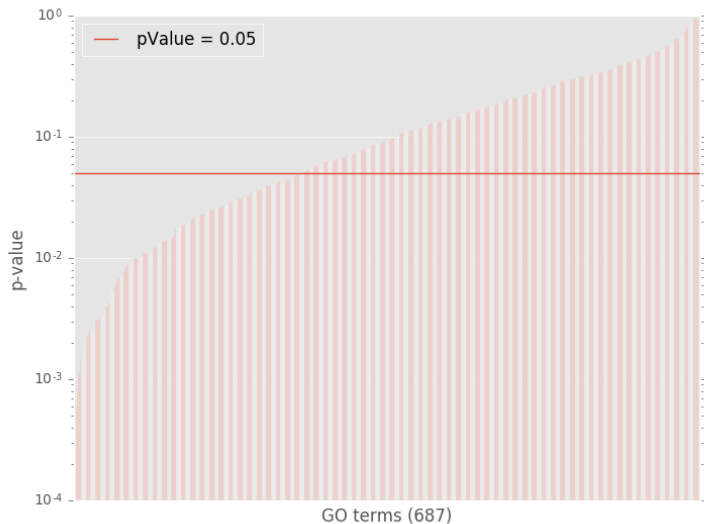
Over-representation of the GO terms annotating at least 1 of the proteins from random set (11 proteins) compared to the reference set (38.660)

File `overRepresentation_proteinsSetRandom.tsv`

	GO ident	GO label	Binom. p-val	Hyperg. p-val
1	go:0036399	TCR signalosome assembly	0.00028449	0.00028453
2	go:0090126	pr. cpx ass. in synap. matur.	0.00028449	0.00028453
...
246	go:0021953	CNS neuron differentiation	0.04922	0.04923
...
687	go:0008150	biological proc.	1.0	1.0

- more GO terms (687 vs. 382)
- P-values are (much) higher than for set1 (min $2.8E - 4$ vs. $4.8E - 36$)
- fewer GO terms with p-value < 5%: 246 (36%) vs. 255 (67%)
- still some terms are (slightly) over-expressed

Example 2: annotations' over-representation



Multiple testings correction: rationale

- When determining whether an annotation associated to k elements among n is over-represented, we compute whether the **likelihood** of observing at least k elements **is low**.
 - low likelihood = **significance level** α : the probability of incorrectly rejecting H_0 is $< \alpha$
 - usually $\alpha = 5\%$: in 5% of the tests, we accept to incorrectly reject H_0 , i.e. to find it significant whereas it is not
- When repeating the test over multiple annotations, we increase the number of hypotheses being tested, which also increases the likelihood of a rare event, and therefore, the likelihood of incorrectly rejecting a null hypothesis.

> Cell J. 2019 Jan;20(4):604-607. doi: 10.22074/cellj.2019.5992. Epub 2018 Aug 1.

Why, When and How to Adjust Your P Values?

Mohieddin Jafari ¹, Naser Ansari-Pour ²

Affiliations + expand

PMID: 30124010 PMID: PMC6099145 DOI: 10.22074/cellj.2019.5992

Free PMC article

Abstract

Currently, numerous papers are published reporting analysis of biological data at different omics levels by making statistical inferences. Of note, many studies, as those published in this Journal, report association of gene(s) at the genomic and transcriptomic levels by undertaking appropriate statistical tests. For instance, genotype, allele or haplotype frequencies at the genomic level or normalized expression level at the transcriptomic level are compared between the case and control groups using the Chi-square/Fisher's exact test or independent (i.e. two-sampled) t-test respectively, with this culminating into a single numeric, namely the P value (or the degree of the false positive rate), which is used to make or break the outcome of the association test. This approach has flaws but nevertheless remains a standard and convenient approach in association studies. However, what becomes a critical issue is that the same cut-off is used when 'multiple' tests are undertaken on the same case-control (or any pairwise) comparison. Here, in brevity, we present what the P value represents, and why and when it should be adjusted. We also show, with worked examples, how to adjust P values for multiple testing in the R environment for statistical computing (<http://www.R-project.org>).

Keywords: Bias; Gene Expression Profiling; Genetic Variation; Research Design; Statistical Data Analyses.

<https://pubmed.ncbi.nlm.nih.gov/30124010/>

Bonferroni correction

Classic method to address multiple testings (there are many others)

Bonferroni correction

- If the desired significance level for a family of n_t tests is α
- then test each individual hypothesis at significance level of $\frac{\alpha}{n_t}$
- assumes that the tests are independent. If we apply the tests on a hierarchy of annotations, this assumption does not hold!
- can be conservative (i.e. increase the number of false negative) if there are a large number of tests or the test statistics are positively correlated \rightarrow we may incorrectly reject some annotations

Still better than no correction at all?

Example: list of random proteins (*without* Bonferroni correction)

No multiple testing correction: for each test $p\text{-value} < 0.05$

set1 (382 annot.)

- 1) 4.79E-36 Heme biosynth.
- 2) 8.32E-35 Porphyrin. biosynth.
- 3) 2.41E-33 Heme metab.
- 4) 5.69E-33 Prototpoph. IX met.
- ...
- 254) 0.0489 Regeneration
- 255) 0.0492 Resp. ionizing rad.

random set (687 annot.)

- 1) 2.8E-4 TCR signalosome
- 2) 2.8E-4 Prot. cplx assembly
- 3) 4.2E-4 Organelle loc.
- 4) 5.7E-4 Guanylate kinase...
- ...
- 245) 0.0492 Resp. ionizing rad.
- 246) 0.0492 CNS neuron diff.

Example: list of random proteins (*with* Bonferroni correction)

Multiple testing correction: for each test p-value $< \frac{0.05}{|annot|}$

set1 $|annot| = 382$

each test p-value $< 1.3E - 4$

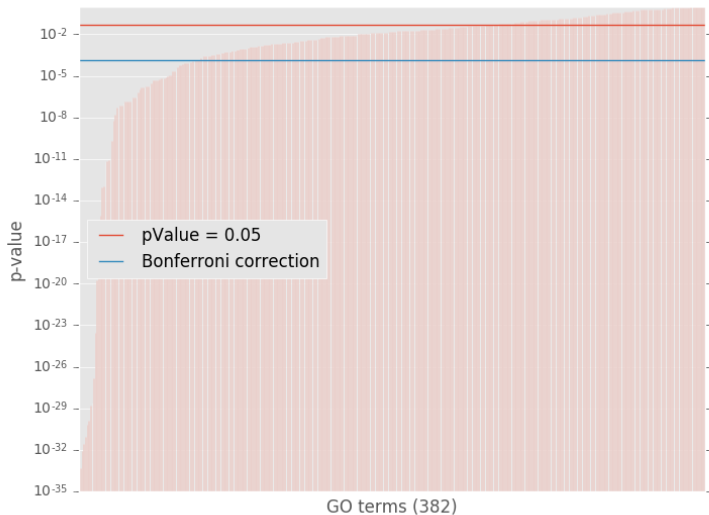
- 1) 4.79E-36 Heme biosynth.
- 2) 8.32E-35 Porphyrin. biosynth.
- 3) 2.41E-33 Heme metab.
- 4) 5.69E-33 Prototpoph. IX met.
- ...
- 71) 1.23E-4 Resp. endog. stim.

random set $|annot| = 687$

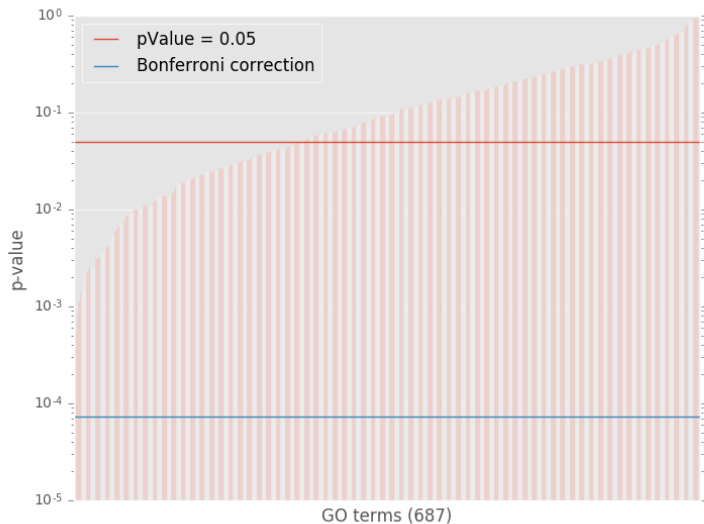
each test p-value $< 7.3E - 5$

∅

Example 1: annotations' over-representation



Example 2: annotations' over-representation



Benjamini-Hochberg correction

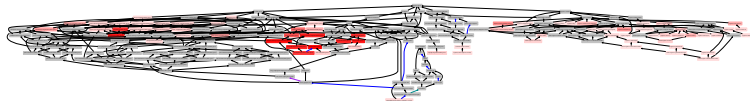
- If the desired significance level for a family of n_t tests is α
- then test each individual hypothesis at a significance level that depends on the rank of its p-value
- Bonferroni: same threshold ($\frac{\alpha}{n_t}$) for all the annotations
- Benjamin-Hochberg: the smaller the p-value, the smaller the threshold (from $\frac{\alpha}{n_t}$ for the annotation with the smallest p-value to α for the annotation with the greatest p-value)

Filtered significant GO terms for Gene set1

File overRepresentedGOtermsWithBonferroni.pdf



File overRepresentedGOtermsWithBonferroniOnly.pdf


























Synthesis

4 Synthesis

- Enrichment analysis is a classical method for identifying the relevant annotations (e.g. GeneOntology terms) that describe a set of elements (e.g. proteins)
- Many tools do the work for you... but most people do not use them correctly
- **Make sure to select the correct reference set!**
- **Do not forget to correct for multiple tests!**

3 Cluster(s)

 Download File

Annotation Cluster 1		Enrichment Score: 17.75	 	Count	P Value	Benjamini
<input type="checkbox"/>	GOTERM_BP_DIRECT	heme biosynthetic process	RT 	11	1.5E-30	1.0E-28
<input type="checkbox"/>	UP_KW_BIOLOGICAL_PROCESS	Heme biosynthesis	RT 	10	5.7E-28	1.7E-27
<input type="checkbox"/>	KEGG_PATHWAY	Porphyrin metabolism	RT 	11	3.0E-24	1.8E-23
<input type="checkbox"/>	KEGG_PATHWAY	Biosynthesis of cofactors	RT 	11	2.2E-18	6.7E-18
<input type="checkbox"/>	KEGG_PATHWAY	Metabolic pathways	RT 	11	3.3E-8	6.5E-8
<input type="checkbox"/>	UP_KW_DISEASE	Disease variant	RT 	10	1.5E-1	3.1E-1
Annotation Cluster 2		Enrichment Score: 4.17	 	Count	P Value	Benjamini
<input type="checkbox"/>	GOTERM_CC_DIRECT	mitochondrial inner membrane	RT 	7	3.5E-8	7.0E-7
<input type="checkbox"/>	UP_KW_CELLULAR_COMPONENT	Mitochondrion	RT 	7	4.6E-6	1.8E-5
<input type="checkbox"/>	UP_KW_DOMAIN	Transit peptide	RT 	5	1.1E-5	4.5E-5
<input type="checkbox"/>	GOTERM_CC_DIRECT	mitochondrion	RT 	7	2.1E-5	2.1E-4
<input type="checkbox"/>	UP_SEQ_FEATURE	TRANSIT:Mitochondrion	RT 	5	9.8E-5	6.1E-3
<input type="checkbox"/>	GOTERM_CC_DIRECT	mitochondrial matrix	RT 	3	1.6E-2	7.8E-2
<input type="checkbox"/>	UP_KW_CELLULAR_COMPONENT	Membrane	RT 	7	1.0E-1	1.4E-1
Annotation Cluster 3		Enrichment Score: 0.14	 	Count	P Value	Benjamini
<input type="checkbox"/>	UP_KW_DOMAIN	Transmembrane helix	RT 	3	6.7E-1	9.0E-1
<input type="checkbox"/>	UP_KW_DOMAIN	Transmembrane	RT 	3	6.8E-1	9.0E-1
<input type="checkbox"/>	GOTERM_CC_DIRECT	integral component of membrane	RT 	3	7.9E-1	1.0E0
<input type="checkbox"/>	UP_SEQ_FEATURE	TRANSMEM:Helical	RT 	3	7.9E-1	1.0E0

18 terms were not clustered.

Set1: Revigo (scatterplot)

