
goenrich Documentation

Release

Author

July 06, 2015

1	goenrich	1
1.1	Installation	1
1.2	Run GO enrichment	1
1.3	GO-slim	2
2	Licence	5
3	Building the documentation	7
4	Indices and tables	9

Convenient GO enrichments from python. For use in python projects.

1. Builds the GO-ontology graph
2. Propagates GO-annotations up the graph
3. Subsetting using `goslim`
4. Performs enrichment test for all categories
5. Performs multiple testing correction
6. Allows for export to `pandas` for processing and `graphviz` for visualization

1.1 Installation

Install package from pypi and download ontology and needed annotations.

```
pip install goenrich
mkdir db
# Ontology
wget http://purl.obolibrary.org/obo/go/go-basic.obo -O db/go-basic.obo
# UniprotACC
wget http://geneontology.org/gene-associations/gene_association.goa_ref_human.gz -O db/gene_associat
# Entrez GeneID
wget ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2go.gz -O db/gene2go.gz
```

1.2 Run GO enrichment

```
import goenrich

# build the ontology
O = goenrich.obo.graph('db/go-basic.obo')

# use all entrez geneid associations form gene2go as background
# use goenrich.read.goa('db/gene_association.goa_ref_human.gz') for uniprot
background = goenrich.read.gene2go('db/gene2go.gz')
```

```
G = goenrich.enrich.set_background(O, background, 'GeneID', 'GO_ID')

# extract some list of entries as example query
query = set(background['GeneID'].unique()[:20])

# for additional export to graphviz just specify the gvfile argument
# the show argument keeps the graph reasonably small
result = goenrich.enrich.analyze(G, query, gvfile='example.dot', show='top20')
```

Generate png image using graphviz

```
dot -Tpng example.dot > example.png
```

1.3 GO-slim

```
# download goslim from ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/goslim/goslim_goa.obo
goa_slim = goenrich.goslim.read('db/goslim_goa.obo')
goenrich.goslim.add(G, 'goslim_goa', (n for n in goa_slim if n in G))
S = goenrich.goslim.subset(G, 'goslim_goa')
result_slim = goenrich.enrich.analyze(S, query, gvfile='example_slim.dot', show='top20')
```

1.3.1 Parameters

Parameters can all be passed to `enrich.analyze` as shown below

```
go_options = {
    'multiple-testing-correction' : 'bonferroni',
    'alpha' : 0.05,
    'node_filter' : lambda x : x.get('significant', False)
}
goenrich.enrich.analyze(G, query, **go_options)

# export results to graphviz
goenrich.enrich.analyze(G, query, gvfile='example.dot', **go_options)
```

Here is an overview over the available parameters

```
read.*:
    experimental = True # don't consider inferred annotations

enrich.analyze:
    node_filter = lambda node : 'p' in node
    show = 'top20' # works for any 'topNUM'

enrich.calculate_pvalues:
    min_hit_size = 2
    min_category_size = 3
    max_category_size = 500
    max_category_depth = 5

enrich.multiple_testing_correction:
    alpha = 0.05
    method = 'benjamin-hochberg' # also supported : 'bonferroni'

export.to_frame:
```

```
node_filter = lambda node: True

export.to_graphviz:
    graph_label = None # if None it is replaced by multiple testing info
```

Licence

This work is licenced under the MIT licence

Contributions are welcome!

Building the documentation

`sphinx-apidoc -f -o docs goenrich goenrich/tests`

Indices and tables

- `genindex`
- `modindex`
- `search`