
goenrich Documentation

Release

Author

September 16, 2016

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

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

1. Builds the GO-ontology graph
2. Propagates GO-annotations up the graph
3. Performs enrichment test for all categories
4. Performs multiple testing correction
5. 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_association.goa_ref_human.gz
# Yeast SGD
wget http://downloads.yeastgenome.org/curation/literature/gene_association.sgd.gz -O db/gene_association.sgd.gz
# 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.ontology('db/go-basic.obo')

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

```
gene2go = goenrich.read.gene2go('db/gene2go.gz')
# use values = {k: set(v) for k,v in annot.groupby('go_id')['db_object_symbol']} for uniprot/yeast
values = {k: set(v) for k,v in gene2go.groupby('GO_ID')['GeneID']}

# propagate the background through the ontology
background_attribute = 'gene2go'
goenrich.enrich.propagate(0, values, background_attribute)

# extract some list of entries as example query
# use query = annot['db_object_symbol'].unique()[:20]
query = gene2go['GeneID'].unique()[:20]

# for additional export to graphviz just specify the gvfile argument
# the show argument keeps the graph reasonably small
df = goenrich.enrich.analyze(0, query, background_attribute, gvfile='test.dot')

# generate html
df.dropna().head().to_html('example.html')

# call to graphviz
import subprocess
subprocess.check_call(['dot', '-Tpng', 'test.dot', '-o', 'test.png'])
```

Generate png image using graphviz:

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

or directly from python:

```
import subprocess
subprocess.check_call(['dot', '-Tpng', 'example.dot', '-o', 'example.png'])
```

Check the documentation for all available parameters

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`