API Documentation

API Documentation

August 30, 2018

Contents

1 Module enrich

GSEA script. Launches enrich.R script and manages its results.

Requires:

- python 2.7[1] or greater
- R 3.3.3 RC[2]
- Conda 4.4.10[3] or greater with FastGSEA conda environment (fastgsea.yml)

Version: 3.0

Author: Kevin Durimel

License: GPL

1.1 Functions

launchGSEA(*outputDir*, *echFile*, *univFile*)

enrich.R wrapper. Takes as input the paths for output directory, sample GO ids file and universe GO ids file.

Parameters

`outputDir`: path to the output directory

(type=string)

`echFile`: Sample GO ids file path

(type=str)

`univFile`: Universe GO ids file path

(type=str)

1.2 Variables

Name	Description
__doc__	**Value:** `...`
__credits__	**Value:** `['Kevin La', 'Arnaud Felten', 'Meryl Vila-Nova', 'Nicolas...`
__maintainer__	**Value:** `'Kevin Durimel'`
__email__	**Value:** `'k@durimel@gmail.com'`
__status__	**Value:** `'Alpha'`
EXEC_DIR	**Value:** `'/home/wrecker/DUMMY/DNAlogy/FAST_GSEA/src'`
__package__	**Value:** `None`

[1] https://www.python.org/downloads/

[2] https://www.r-project.org/

[3] https://conda.io/

2 Module fastGSEA

FastGSEA workfow. Performs GO-terms enrichment analysis between two gene sets. These gene sets must be provided as files containing one international databank (ncbi, refseq, etc...) gene or protein identifier per line.

Requires:

- python 2.7[4] or greater
- R 3.3.3 RC[5]
- Conda 4.4.10[6] or greater with FastGSEA conda environment (fastgsea.yml)
- map.py
- enrich.py
- trim.py
- managefiles.sh

Version: 3.0

Author: Kevin Durimel

License: GPL

2.1 Functions

get__parser()

Initlializing arguments.

Return Value

arguments values

(type=object)

main()

2.2 Variables

Name	Description
__doc__	**Value:** `...`
__credits__	**Value:** `['Kevin La', 'Arnaud Felten', 'Meryl Vila-Nova', 'Nicolas...`
__maintainer__	**Value:** `'Kevin Durimel'`
__email__	**Value:** `'k@durimel@gmail.com'`
__status__	**Value:** `'Alpha'`
__package__	**Value:** `None`

[4] https://www.python.org/downloads/
[5] https://www.r-project.org/
[6] https://conda.io/

2.3 Class Timer

object

fastGSEA.Timer

Initlialize and manage timers.

2.3.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

2.3.2 Properties

Name	Description
lifetime	
Inherited from object __class__	

3 Module map

```
Id mapping script for a gene set. Takes as input a file (gene set) constitued by Refseq,
Supported ids are (by decreasing quantity of available information):
1. UniProtKB-AC (1st col in gz file)
2. UniProtKB-ID (2)
3. GeneID (EntrezGene) (3)
4. RefSeq (4)
5. GI (5)
6. GO (as output only) (7)
8. UniRef100 (8)
9. UniRef90 (9)
10. UniRef50 (10)
11. UniParc (11)
15. UniGene (15)
17. EMBL (17)
18. EMBL-CDS (18)
19. Ensembl (19)
20. Ensembl_TRS (20)
21. Ensembl_PRO (21)

@requires: U{python 2.7<https://www.python.org/downloads/>} or greater
@requires: U{Conda 4.4.10<https://conda.io/>} or greater with FastGSEA conda environment
@requires: manageFiles.sh
```

Version: 3.0

Author: Kevin Durimel

License: GPL

3.1 Functions

show_progression(*counter*, *total*, *precision*)

Outputs percentage of progression.

Parameters

`counter:` int used as iterator for couting when iterating a object

(type=int)

`total:` object total length

(type=int)

`precision:` number of decimals to show when printing progression

(type=int)

mk_susbet(*idMappingFile*, *idsOptimized*, *showWhichIdIsOptimized*)

Takes as input idMappingfile.gz and the index of the colums we want to generate an subset for.

Parameters

`idMappingFile:` path to the .gz file used for id mapping

(type=string)

`idsOptimized:` column of the identifier in the idMappingFIle

(type=int)

`showWhichIdIsOptimized:` -fastmode value entered by user (ids to optimize)

(type=str)

ids_to_go(*idMappingFile*, *idsFile*, *outputPrefix*)

Takes as input Uniprot or Refseq ids and returns its corresponding GO-terms. Please use this function if you want to retrieve GO-terms using a input file containing Refseq or Uniprot ids. The id mapping file subset is automatically generated if it was not already the case (e.g First script launch , or file deleted.)

Parameters

`idMappingFile`: path to the .gz file used for id mapping

(type=string)

`idsFile`: path to the input file containing all the ids we have to map

(type=string)

`outputPrefix`: file path output prefix

(type=string)

ids_to_go_online(*idMappingFile*, *idsFile*, *outputPrefix*)

This function follows same principles as ids_to_go() but performs id mapping online. It the best choice for reliability and up-to-date GO-terms. Ids_to_to_online() always requires idMappingFile parameter to retrieve refseq ids offline. Takes as input Uniprot or Refseq ids and returns its corresponding GO-terms. Please use this function if you want to retrieve GO-terms using a input file containing Refseq or Uniprot ids.

Parameters

`idMappingFile`: path to the .gz file used for id mapping

(type=string)

`idsFile`: path to the input file containing all the ids we have to map

(type=string)

`outputPrefix`: file path output prefix

(type=string)

any_ids_to_go(*idMappingFile*, *idsFile*, *outputPrefix*)

Takes as input any suppported ids and returns its corresponding GO-terms. Slower than ids_go_to beacause it do not use the idMappingFile subset and requests id per id.

Parameters

`idMappingFile:` path to the .gz file used for id mapping

(type=string)

`idsFile:` path to the input file containing all the ids we have to map

(type=string)

`outputPrefix:` file path output prefix

(type=string)

any_ids_to_go_online(*idMappingFile*, *idsFile*, *outputPrefix*)

This function follows same principles as any_ids_to_go() but performs id mapping online. This function is an experimental feature and support only the ids supported by the Uniprot API[a] Takes as input any suppported ids and returns its corresponding GO-terms. Slower than any_ids_go_to() beacause it do not use the idMappingFile subset and requests id per id.

Parameters

`idMappingFile:` path to the .gz file used for id mapping

(type=string)

`idsFile:` path to the input file containing all the ids we have to map

(type=string)

`outputPrefix:` file path output prefix

(type=string)

[a] https://www.ebi.ac.uk/QuickGO/api/index.html

any_ids_to_any_ids(*idMappingFile*, *idsFile*, *outputPrefix*, *whichDb*)

Takes as input any suppported ids and returns its corresponding GO-terms. Slower than ids_go_to beacause it do not use the idMappingFile subset and requests id per id.

Parameters

`idMappingFile`: path to the .gz file used for id mapping

(type=string)

`idsFile`: path to the input file containing all the ids we have to map

(type=string)

`outputPrefix`: file path output prefix

(type=string)

`whichDb`: the ids database prefix that we want in output (for example, if we want go ids, whichDB = GO. Please refer to "supported ids" at the top section)

(type=string)

3.2 Variables

Name	Description
__doc__	**Value:** `...`
__credits__	**Value:** `['Kevin La', 'Arnaud Felten', 'Meryl Vila-Nova', 'Nicolas...`
__maintainer__	**Value:** `'Kevin Durimel'`
__email__	**Value:** `'k@durimel@gmail.com'`
__status__	**Value:** `'Alpha'`
EXEC_PATH	**Value:** `'/home/wrecker/DUMMY/DNAlogy/FAST_GSEA/src'`
__package__	**Value:** `None`

4 Module trim

GOtrimmer script. Parse object data from a .obo file, and provide a subset of non-prokaryotic and non obsolete GO-terms based on the lightest official Uniprot's gene ontology file (GO-basic.obo). The file generated by this script will serve for non-prokaryotic GO-terms trimming. Obo files are daily updated and can be found here : http://viewvc.geneontology.org/viewvc/GO-SVN/ontology-releases/YYYY-MM-DD/subsets/ (replace YYYY-MM-DD by currrent date) Please notice that 'gosubset_prok' terms are not maitained since 2018/06 because they may be irrelevant. More information here: https://github.com/geneontology/go-ontology/pull/16255

Requires:

- enrich.R output file (hyperesults.csv)
- old GO-basic.obo file (< 2018/06 releases) or gosubset_prok.obo file.

Version: 3.0

Author: Kevin Durimel

License: GPL

4.1 Functions

add_object(*d*, *outputFile*)

Add GO-terms objects only if they are non-prokaryotic and up to date (i.e non obsolete)

mk_subset(*oboFile*, *outputFile*)

Generates a GO-terms file subset based on a obo file.

Parameters

`oboFile`: obo file path

(type=str)

`outputFile`: GO subset output file path

(type=str)

trim(*goprok_txt*, *trimmedFile*)

GO terms trimming based on obo file subset

Parameters

`goprok_txt`: obo file subset path (goprok.txt)

(type=str)

`trimmedFile`: trimmed file path

(type=str)

4.2 Variables

Name	Description
__doc__	**Value:** `...`
__credits__	**Value:** `['PM 2Ring : http://stackoverflow.com/q/32989776/4014959']`
__maintainer__	**Value:** `'Kevin Durimel'`
__email__	**Value:** `'k@durimel@gmail.com'`
__status__	**Value:** `'Alpha'`
all_objects	**Value:** `{}`
__package__	**Value:** `None`

Index