

API Documentation

API Documentation

August 27, 2018

Contents

Contents	1
1 Module enrich	2
1.1 Functions	2
1.2 Variables	2
2 Module fastGSEA	3
2.1 Functions	3
2.2 Variables	3
2.3 Class Timer	4
2.3.1 Methods	4
2.3.2 Properties	4
3 Module map	5
3.1 Functions	5
3.2 Variables	8
4 Module trim	9
4.1 Functions	9
4.2 Variables	9

1 Module enrich

GSEA script. Launches Enrich.R script and manage its results.

Requires:

- python 2.7¹ or greater
- R 3.3.3 RC²
- Conda 4.4.10³ or greater with FastGSEA conda environment (fastgsea.yml)

Version: 3.0

Author: Kevin Durimel

License: GPL

1.1 Functions

launchGSEA(*outputDir*)

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/EspaceDeTravailLinux/DNAlogy/aug18/DNAlogy...'
__package__	Value: None

¹<https://www.python.org/downloads/>

²<https://www.r-project.org/>

³<https://conda.io/>

2 Module fastGSEA

FastGSEA workflow. 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 2.7⁴ or greater
- R 3.3.3 RC⁵
- Conda 4.4.10⁶ 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()

Initializing 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

Initialize and manage timers.

2.3.1 Methods

__init__ (<i>self</i>) x. __init__ (...) initializes x; see help(type(x)) for signature Overrides: object. __init__ extit(inherited documentation)

Inherited from object

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

2.3.2 Properties

Name	Description
lifetime	
<i>Inherited from object</i>	
__class__	

3 Module map

Id mapping script for a gene set. Takes as input a file (gene set) constituted 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

<code>show__progression(counter, total, precision)</code>

<code>mk__susbet(idMappingFile)</code>
--

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

outputPrefix: file path output prefix
(*type=string*)

view: (*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

outputPrefix: file path output prefix
(*type=string*)

view: (*type=string*)

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

Takes as input any supported ids and returns its corresponding GO-terms. Slower than `ids_go_to` because it does not use the `idMappingFile` subset and requests id per id.

Parameters

idMappingFile:	path to the .gz file used for id mapping (<i>type=string</i>)
idsFile:	path to the input file containing all the ids we have to map
whichDb:	the ids database prefix that we want in output (for example, if we want go ids, <code>whichDB = GO</code> . Please refer to "supported ids" at the top section) (<i>type=string</i>)
outputPrefix:	file path output prefix (<i>type=string</i>)
view:	(<i>type=string</i>)

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

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 supported ids and returns its corresponding GO-terms. Slower than `any_ids_go_to()` because it does not use the `idMappingFile` subset and requests id per id.

Parameters

idMappingFile:	path to the .gz file used for id mapping (<i>type=string</i>)
idsFile:	path to the input file containing all the ids we have to map
whichDb:	the ids database prefix that we want in output (for example, if we want go ids, <code>whichDB = GO</code> . Please refer to "supported ids" at the top section) (<i>type=string</i>)
outputPrefix:	file path output prefix (<i>type=string</i>)
view:	(<i>type=string</i>)

^a<https://www.ebi.ac.uk/QuickGO/api/index.html>

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

Takes as input any supported ids and returns its corresponding GO-terms. Slower than `ids_go_to` because it does 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

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*)

outputPrefix: file path output prefix
(*type=string*)

view: (*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/EspaceDeTravailLinux/DNAlogy/aug18/DNAlogy.'
__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 current date) Please notice that 'gosubset_prok' terms are not maintained since 2018/06 because they may be irrelevant. More information here: <https://github.com/geneontology/go-ontology/pull/16255>

Requires:

- enrich.R output file (hyperresults.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.

trim(*goprok_txt*, *enrichmentResultsFile*)

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: {}

continued on next page

Name	Description
__package__	Value: None

Index

- enrich (*module*), 2
 - enrich.launchGSEA (*function*), 2
- fastGSEA (*module*), 3–4
 - fastGSEA.get_parser (*function*), 3
 - fastGSEA.main (*function*), 3
 - fastGSEA.Timer (*class*), 3–4
- map (*module*), 5–8
 - map.any_ids_to_any_ids (*function*), 7
 - map.any_ids_to_go (*function*), 6
 - map.any_ids_to_go_online (*function*), 7
 - map.ids_to_go (*function*), 5
 - map.ids_to_go_online (*function*), 6
 - map.mk_susbet (*function*), 5
 - map.show_progression (*function*), 5
- trim (*module*), 9–10
 - trim.add_object (*function*), 9
 - trim.mk_subset (*function*), 9
 - trim.trim (*function*), 9