

API Documentation

API Documentation

September 2, 2016

Contents

Contents	1
1 Module EveryGO	2
1.1 Functions	2
1.2 Variables	3
2 Module GOtrimmer	4
2.1 Functions	4
2.2 Variables	4
3 Module GoXML	5
3.1 Functions	5
3.2 Variables	7
4 Module VCFtoGO	8
4.1 Functions	8
4.2 Variables	10

1 Module EveryGO

EveryGO workflow. In order to perform GO-terms enrichment analysis for one or more nodes/comparisons of the phylogeny stored in the XML file.

Requires:

- python 2.7¹ (tested with 2.7.6)
- lxml²
- GOfilter_xml
- GOfilter
- GoView

1.1 Functions

get_parser()

Arguments setting and parsing

Return Value

arguments list

(*type=parser object*)

getcomp(*complist*)

Get comparisons values

Parameters

complist: list of comparisons values (-comp argument)

(*type=string*)

Return Value

list of comparisons values (-comp argument)

(*type=string*)

makeuniverse(*xmlname*)

Make universe of GO-terms from the XML file

Parameters

xmlname: the xml filename

(*type=string*)

¹<https://www.python.org/downloads/>

²<https://github.com/lxml/lxml>

everygo(*compvalues*, *view*, *xmlname*)

Make enrichment analysis for each comparison values

Parameters

compvalues: comparisons values returned by getcomp()

(*type=string*)

view: -view argument -> if view != 0 , graphical representation of enrichment will be processed

(*type=int*)

xmlname: the xml filename

(*type=string*)

main()

1.2 Variables

Name	Description
__doc__	Value: ...
__package__	Value: None

2 Module GOtrimmer

GOtrimmer script. In order to keep prokaryotic terms only in VCFtoGO universe output file, based on GOslim (GO-basic.obo)

Requires:

- VCFtoGO output file (GO-terms universe)
- GO-basic.obo file (GOslim)

2.1 Functions

get_parser()
Arguments setting and parsing
Return Value
arguments list
<i>(type=parser object)</i>

add_object(<i>d</i>)
collect prokaryotic terms in GO-basic.obo
Parameters
d: go-term related data collected from GO-basic.obo
<i>(type=object data)</i>

main()

2.2 Variables

Name	Description
__doc__	Value: ...
__package__	Value: None

3 Module GoXML

GoXML workflow. In order to add GO-terms from a XML file and generate a non-enriched universe of GO-terms.

Requires:

- python 2.7³ (tested with 2.7.6)
- PyVCF⁴
- lxml⁵
- GOtrimmer_xml

3.1 Functions

get_parser()

Arguments setting and parsing

Return Value

arguments list

(*type=parser object*)

readDBFile(fichier)

Read NP-EBI local storage file

Parameters

fichier: NP-EBI_databases.tsv file

(*type=.tsv file*)

Return Value

dictionnary of NP identifiers

(*type=dictionnary*)

readDBGOF(fichier)

Read EBI-GO local storage file

Parameters

fichier: EBI-GO_databases.tsv file

(*type=.tsv file*)

Return Value

dictionnary of GO identifiers

(*type=dictionnary*)

³<https://www.python.org/downloads/>

⁴<https://github.com/jamescasbon/PyVCF>

⁵<https://github.com/lxml/lxml>

extractNP(*path*)

Retrieve NP ids from XML file

Parameters

path: relative path to the XML file
(*type=string*)

Return Value

dictionary of NP identifiers
(*type=dictionary*)

idmapping(*nplist, dicoNP, NPfile*)

Make ID mapping from NP identifiers to Uniprot identifiers requesting Online(Uniprot database) of Offline(NP-EBI_databases.tsv)

Parameters

nplist: list used in ONLINE request -> values = NP identifiers
(*type=list*)

dicoNP: dictionary used in OFFLINE request -> keys = NP identifiers ,
values = uniprot id retrieved
(*type=dictionary*)

NPfile: Storage of uniprot identifiers to NP identifiers (NP-EBI_databases.tsv)
(*type=.tsv file*)

UNIPROTtoGO(*path, output, protlist, dicoGO, GOFile*)

Make ID mapping from Uniprot identifiers to GO identifiers requesting Online(Quick-GO database) of Offline(EBI-GO_databases.tsv)

Parameters

path: relative path to the XML file
(*type=string*)

output: Universe of GO identifiers (univers.txt)
(*type=.txt file*)

protlist: list used in ONLINE request -> values = UNIPROT identifiers
(*type=list*)

dicoGO: dictionary used in ONLINE requests -> keys = Uniprot identifiers ,
values = GO identifiers list
(*type=dictionary*)

GOFile: Storage of uniprot identifiers to GO identifiers
(EBI-GO_databases.tsv)
(*type=.tsv file*)

main()

3.2 Variables

Name	Description
__doc__	Arguments settings Value: ...
__package__	Value: None

4 Module VCFtoGO

VCFtoGO workflow. In order to generate non-enriched universe of GO-terms from a VCF.

Requires:

- python 2.7⁶ (tested with 2.7.6)
- PyVCF⁷
- GOftrimmer

4.1 Functions

get_parser()

Arguments setting and parsing

Return Value

arguments list

(type=parser object)

readDBFile(fichier)

Read NP-EBI local storage file

Parameters

fichier: NP-EBI_databases.tsv file

(type=.tsv file)

Return Value

dictionnary of NP identifiers

(type=dictionnary)

readDBGOFFile(fichier)

Read EBI-GO local storage file

Parameters

fichier: EBI-GO_databases.tsv file

(type=.tsv file)

Return Value

dictionnary of GO identifiers

(type=dictionnary)

⁶<https://www.python.org/downloads/>

⁷<https://github.com/jamescasbon/PyVCF>

extractinfopos(*fichier, pos*)

Parse NP identifiers from VCF file

Parameters

fichier: VCF file
(*type=.vcf file*)

Return Value

dictionary of NP identifiers
(*type=dictionary*)

idmapping(*dico, dicoNP, NPfile*)

Make ID mapping from NP identifiers to Uniprot identifiers requesting Online(Uniprot database) of Offline(NP-EBI_databases.tsv)

Parameters

dico: dictionary used in ONLINE request -> keys = NP identifiers , values = uniprot id retrieved
(*type=dictionary*)

dicoNP: dictionary used in OFFLINE request -> keys = NP identifiers , values = uniprot id retrieved
(*type=dictionary*)

NPfile: Storage of uniprot identifiers to NP identifiers (NP-EBI_databases.tsv)
(*type=.tsv file*)

UNIPROTtoGO(*dico, output, pos, dicoGO, GOfile*)

Make ID mapping from Uniprot identifiers to GO identifiers requesting Online(Quick-GO database) of Offline(EBI-GO_databases.tsv)

Parameters

dico: dictionary used in OFFLINE requests -> keys = Uniprot identifiers , values = GO identifiers list
(*type=dictionary*)

output: Universe of GO identifiers (univers.txt)
(*type=.txt file*)

pos: positions requested if -pos argument was used
(*type=int*)

dicoGO: dictionary used in ONLINE requests -> keys = Uniprot identifiers , values = GO identifiers list
(*type=dictionary*)

GOfile: Storage of uniprot identifiers to GO identifiers (EBI-GO_databases.tsv)
(*type=.tsv file*)

main()

4.2 Variables

Name	Description
<code>__doc__</code>	Value: ...
<code>__package__</code>	Value: None

Index

- EveryGO (*module*), 2–3
 - EveryGO.everygo (*function*), 2
 - EveryGO.get_parser (*function*), 2
 - EveryGO.getcomp (*function*), 2
 - EveryGO.main (*function*), 3
 - EveryGO.makeuniverse (*function*), 2
- GOtrimmer (*module*), 4
 - GOtrimmer.add_object (*function*), 4
 - GOtrimmer.get_parser (*function*), 4
 - GOtrimmer.main (*function*), 4
- GoXML (*module*), 5–7
 - GoXML.extractNP (*function*), 5
 - GoXML.get_parser (*function*), 5
 - GoXML.idmapping (*function*), 6
 - GoXML.main (*function*), 6
 - GoXML.readDBFile (*function*), 5
 - GoXML.readDBGOFfile (*function*), 5
 - GoXML.UNIPROTtoGO (*function*), 6
- VCFtoGO (*module*), 8–10
 - VCFtoGO.extractinfopos (*function*), 8
 - VCFtoGO.get_parser (*function*), 8
 - VCFtoGO.idmapping (*function*), 9
 - VCFtoGO.main (*function*), 9
 - VCFtoGO.readDBFile (*function*), 8
 - VCFtoGO.readDBGOFfile (*function*), 8
 - VCFtoGO.UNIPROTtoGO (*function*), 9