

API Documentation

API Documentation

December 5, 2017

Contents

Contents	1
1 Module EveryGO	2
1.1 Functions	2
1.2 Variables	3
2 Module GOslimmer_xml	4
2.1 Functions	4
2.2 Variables	4
3 Module GOtrimmer	5
3.1 Functions	5
3.2 Variables	5
4 Module GetGOxml	6
4.1 Functions	6
4.2 Variables	6
5 Module GoXML	7
5.1 Functions	7
5.2 Variables	9
6 Module VCFtoGO	10
6.1 Functions	10
6.2 Variables	12

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
- GOftrimmer_xml
- GOwalker
- GoView

1.1 Functions

get_parser()

Arguments setting and parsing

Return Value

arguments list

(*type=parser object*)

makeuniverse(xmlname)

Make universe of GO-terms from the XML file

Parameters

xmlname: the xml filename

(*type=string*)

everygo(compvalues, view, xmlname, Rpath)

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<https://www.python.org/downloads/>

²<https://github.com/lxml/lxml>

1.2 Variables

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

2 Module GOslimmer_xml

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()
Parse arguments
Return Value
arguments list
<i>(type=parser object)</i>

add_object(<i>d</i>, <i>outputFile</i>)
--

main()

2.2 Variables

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

3 Module GOtrimmer

3.1 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>

3.2 Variables

Name	Description
<code>__doc__</code>	Value: ...
<code>parser</code>	Value: <code>get_parser()</code>
<code>Arguments</code>	Value: <code>parser.parse_args()</code>
<code>fname</code>	Value: <code>"../db/go-basic.obo"</code>
<code>term_head</code>	Value: <code>"[Term]"</code>
<code>all_objects</code>	Value: <code>{}</code>
<code>output</code>	Value: <code>open("goprok.txt", "w")</code>
<code>gofull</code>	Value: <code>str(Arguments.input)</code>
<code>outputuniv</code>	Value: <code>open("VCFtoGOresults/univers.txt", "w")</code>
<code>current</code>	Value: <code>defaultdict(list)</code>

4 Module GetGOxML

GoXML Wrapper.

Requires:

- python 2.7³ (tested with 2.7.6)
- PyVCF⁴
- lxml⁵
- GOslimmer_xml
- GoXML

4.1 Functions

get_parser()

Arguments setting and parsing

Return Value

arguments list

(*type=parser object*)

main()

4.2 Variables

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

³<https://www.python.org/downloads/>

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

⁵<https://github.com/lxml/lxml>

5 Module GoXML

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

Requires:

- python 2.7⁶ (tested with 2.7.6)
- PyVCF⁷
- lxml⁸
- GOslimmer_xml

5.1 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_table.tsv file

(*type=string*)

Return Value

dictionnary of NP identifiers

(*type=dictionnary*)

readDBGOFFile(*fichier*)

Read EBI-GO local storage file

Parameters

fichier: EBI-GO_table.tsv file

(*type=string*)

Return Value

dictionnary of GO identifiers

(*type=dictionnary*)

⁶<https://www.python.org/downloads/>

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

⁸<https://github.com/lxml/lxml>

extractNP(*path, withHomoplasy*)

Retrieve NP ids from XML file

Parameters

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

Return Value

dictionary of NP identifiers
(*type=dictionnary*)

idmapping(*nplist, dicoNP, NPfile*)

Make ID mapping from NP identifiers to Uniprot identifiers requesting Online(Uniprot database) or 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=dictionnary*)

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

Return Value

list of Uniprot identifiers
(*type=list*)

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

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

Parameters

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

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

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=dictionnary*)

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

<code>main()</code>

5.2 Variables

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

6 Module VCFtoGO

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

Requires:

- python 2.7⁹ (tested with 2.7.6)
- PyVCF¹⁰
- GOftrimmer

6.1 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)

⁹<https://www.python.org/downloads/>

¹⁰<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) or 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) or 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()

6.2 Variables

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

Index

- EveryGO (*module*), 2–3
 - EveryGO.everygo (*function*), 2
 - EveryGO.get_parser (*function*), 2
 - EveryGO.main (*function*), 2
 - EveryGO.makeuniverse (*function*), 2
- GetGOxML (*module*), 6
 - GetGOxML.get_parser (*function*), 6
 - GetGOxML.main (*function*), 6
- GOslimmer_xml (*module*), 4
 - GOslimmer_xml.add_object (*function*), 4
 - GOslimmer_xml.get_parser (*function*), 4
 - GOslimmer_xml.main (*function*), 4
- GOtrimmer (*module*), 5
 - GOtrimmer.add_object (*function*), 5
 - GOtrimmer.get_parser (*function*), 5
- GoXML (*module*), 7–9
 - GoXML.extractNP (*function*), 7
 - GoXML.get_parser (*function*), 7
 - GoXML.idmapping (*function*), 8
 - GoXML.main (*function*), 8
 - GoXML.readDBFile (*function*), 7
 - GoXML.readDBGOFfile (*function*), 7
 - GoXML.UNIPROTtoGO (*function*), 8
- VCFtoGO (*module*), 10–12
 - VCFtoGO.extractinfopos (*function*), 10
 - VCFtoGO.get_parser (*function*), 10
 - VCFtoGO.idmapping (*function*), 11
 - VCFtoGO.main (*function*), 11
 - VCFtoGO.readDBFile (*function*), 10
 - VCFtoGO.readDBGOFfile (*function*), 10
 - VCFtoGO.UNIPROTtoGO (*function*), 11