

Package ‘GeneListPrioritization’

May 13, 2015

Title Gene List Prioritization and Filtering

Version 0.1

Description What the package does (one paragraph)

Depends R (>= 3.1.3), GSEABase, XML, KEGGREST, RCurl

License MIT + file LICENSE

LazyData true

Encoding UTF-8

Suggests knitr

VignetteBuilder knitr

NeedsCompilation no

Author Paul Anderson [aut, cre],
Benjamin Neely [aut]

Maintainer Paul Anderson <andersonpe2@cofc.edu>

R topics documented:

download.example.data	2
fold.change.filter	2
GeneListPrioritization	3
get.genes.in.data	3
get.genes.kegg	4
get.genes.tf	4
get.genes.wiki	5
list.filter	6
list.filter.inxs	6
load.gene.data	7
load.protein.data	8
load.WebGestalt	8
overall.mean.filter	9
overall.var.filter	10
random.filter	10
Index	12

`download.example.data` *Downloads example data files for examples*

Description

Downloads example data files for examples

Usage

```
download.example.data(dest = ".", allow.cache = T,  
  base.url = "http://freyja.cs.cofc.edu/downloads/GeneListPrioritization")
```

Arguments

<code>dest</code>	The destination path
<code>allow.cache</code>	Allow the files to be cached to prevent downloading multiple times.
<code>base.url</code>	Source URL location

Examples

```
download.example.data()
```

`fold.change.filter` *Fold change filter*

Description

Returns ...

Usage

```
fold.change.filter(data, theta, class1.inxs, class2.inxs)
```

Arguments

<code>data</code>	- transposed gene data
<code>theta</code>	- if > 1, theta is the target number of ranked variables; otherwise, it is the top theta fraction of ranked variables.
<code>class1.inxs</code>	- the row indices of class 1
<code>class2.inxs</code>	- the column indices of class 2

Value

filtered dataset

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
gene.ids = get.genes.wiki(webg.pathways)
Catteno.rna = load.gene.data("Catteno_array.csv", 5)
prioritized.data = list.filter(Catteno.rna$transposed.data, gene.ids)
prioritized.filtered.data = fold.change.filter(prioritized.data, 100,
                                                which(prioritized.data$labels=='adenoma'),
                                                which(prioritized.data$labels=='normal'))
```

GeneListPrioritization	<i>GeneListPrioritization: A package for prioritization and filtering a gene list.</i>
------------------------	--

Description

The foo package provides three categories of important functions: foo, bar and baz.

Foo functions

The foo functions ...

get.genes.in.data	<i>Filter for only those genes in the data.</i>
-------------------	---

Description

Returns a corrected gene ID vector

Usage

```
get.genes.in.data(gene.ids, gene.data)
```

Arguments

- gene.ids A vector of Entrez Gene IDs
- gene.data The gene data loaded with the helper functions.

Value

a vector of Gene IDs

Examples

```
download.example.data()
Catteno.rna = load.gene.data("Catteno_array.csv",5)
webg.pathways = load.WebGestalt("Marra_0_kegg_protein_enrichment.tsv", 'Kegg')
gene.ids = get.genes.kegg(webg.pathways)
genes.in.data = get.genes.in.data(gene.ids,Catteno.rna$data)
```

get.genes.kegg	<i>Downloading Kegg Pathways and returning gene ids</i>
----------------	---

Description

Returns ...

Usage

```
get.genes.kegg(webg.pathways, sleep = 10)
```

Arguments

webg.pathways	dataframe where the second column contains the pathway ids
sleep	Number of seconds to sleep between Kegg requests

Value

an array of gene ids

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_kegg_protein_enrichment.tsv", 'Kegg')
gene.ids = get.genes.kegg(webg.pathways)
```

get.genes.tf	<i>Finds the genes ids from the Transcription Factor Targets database</i>
--------------	---

Description

Returns ...

Usage

```
get.genes.tf(webg.pathways, db = system.file("extdata",
" c3.tft.v4.0.entrez.gmt", package = "GeneListPrioritization"))
```

Arguments

webg.pathways dataframe where the second column contains the pathway ids
db TF database to use

Value

an array of gene ids

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_tf_protein_enrichment.tsv", 'TF')
gene.ids = get.genes.tf(webg.pathways)
```

get.genes.wiki	<i>Downloading WikiPathways and returning gene ids</i>
----------------	--

Description

Returns ... For more information on WikiPathways see <http://webservice.wikipathways.org/> and http://www.wikipathways.org/index.php/Help:WikiPathways_Webservice/API

Usage

```
get.genes.wiki(webg.pathways, url = "http://webservice.wikipathways.org",
               species = "Homo sapiens")
```

Arguments

webg.pathways Dataframe where the second column contains the pathway ids
url WikiPathways base URL for webservice calls.
species Species for pathway analysis

Value

a dataframe

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
gene.ids = get.genes.wiki(webg.pathways)
```

list.filter	Returns a filtered dataset
-------------	----------------------------

Description

Returns ...

Usage

```
list.filter(data, gene.ids)
```

Arguments

data	- transposed gene data
gene.ids	- result of get.genes.*(...)

Value

an array of inxs

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
gene.ids = get.genes.wiki(webg.pathways)
Catteno.rna = load.gene.data("Catteno_array.csv", 5)
filtered.data = list.filter(Catteno.rna$transposed.data, gene.ids)
```

list.filter.inxs	Returns the gene.ixs for filtering
------------------	------------------------------------

Description

Returns ...

Usage

```
list.filter.inxs(data, gene.ids)
```

Arguments

data	- transposed gene data
gene.ids	- result of get.genes.*(...)

Value

an array of inxs

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
gene.ids = get.genes.wiki(webg.pathways)
Catteno.rna = load.gene.data("Catteno_array.csv", 5)
gene.inxs = list.filter.inxs(Catteno.rna$transposed.data, gene.ids)
```

load.gene.data	<i>Loading a correctly formatted gene file.</i>
----------------	---

Description

Returns a list containing two data frames. The second data frame is a formatted and transposed version of the original data with an additional labels column added.

Usage

```
load.gene.data(file, start.data.inx)
```

Arguments

file A string pointing to the location of the data file

start.data.inx An integer ≥ 2 that indicates where the actual data starts.

Value

a list containing two data frames.

Examples

```
download.example.data()
TCGA.rna = load.gene.data("TCGA_rna.csv", 3)
Catteno.rna = load.gene.data("Catteno_array.csv", 5)
```

load.protein.data	<i>Loading a correctly formatted protein file.</i>
-------------------	--

Description

Returns a list containing two data frames. The second data frame is a formatted and transposed version of the original data with an additional labels column added.

Usage

```
load.protein.data(file, start.data.inx)
```

Arguments

file	A string pointing to the location of the data file
start.data.inx	An integer ≥ 2 that indicates where the actual data starts.

Value

a list containing two data frames.

Examples

```
download.example.data()  
TCGA.protein = load.protein.data("TCGA_protein.csv",3)
```

load.WebGestalt	<i>Loading the pathway data from WebGestalt output.</i>
-----------------	---

Description

Returns ...

Usage

```
load.WebGestalt(file, db)
```

Arguments

file	A string pointing to the location of the data file
db	Wiki, Kegg, or TF

Value

a list containing the enrichment results

Examples

```
download.example.data()
pathways = load.WebGestalt("Marra_0_tf_protein_enrichment.tsv", 'TF')
pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
pathways = load.WebGestalt("Marra_0_kegg_protein_enrichment.tsv", 'Kegg')
```

overall.mean.filter	<i>Overall mean filter</i>
---------------------	----------------------------

Description

Returns ...

Usage

```
overall.mean.filter(data, theta)
```

Arguments

- data - transposed gene data
- theta - if > 1, theta is the target number of ranked variables; otherwise, it is the top theta fraction of ranked variables.

Value

an array of inxs

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
gene.ids = get.genes.wiki(webg.pathways)
Catteno.rna = load.gene.data("Catteno_array.csv", 5)
prioritized.data = list.filter(Catteno.rna$transposed.data, gene.ids)
prioritized.filtered.data = overall.mean.filter(prioritized.data, 100)
```

overall.var.filter	<i>Overall variance filter</i>
--------------------	--------------------------------

Description

Returns ...

Usage

```
overall.var.filter(data, theta)
```

Arguments

data	- transposed gene data
theta	- if > 1, theta is the target number of ranked variables; otherwise, it is the top theta fraction of ranked variables.

Value

an array of inxs

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
gene.ids = get.genes.wiki(webg.pathways)
Catteno.rna = load.gene.data("Catteno_array.csv", 5)
prioritized.data = list.filter(Catteno.rna$transposed.data, gene.ids)
prioritized.filtered.data = overall.var.filter(prioritized.data, 100)
```

random.filter	<i>Random filter</i>
---------------	----------------------

Description

Returns ...

Usage

```
random.filter(data, theta)
```

Arguments

data	- transposed gene data
theta	- if > 1, theta is the target number of ranked variables; otherwise, it is the top theta fraction of ranked variables.

Value

an array of inxs

Examples

```
download.example.data()
webg.pathways = load.WebGestalt("Marra_0_wiki_protein_enrichment.tsv", 'Wiki')
gene.ids = get.genes.wiki(webg.pathways)
Catteno.rna = load.gene.data("Catteno_array.csv", 5)
prioritized.data = list.filter(Catteno.rna$transposed.data, gene.ids)
prioritized.filtered.data = random.filter(prioritized.data, 100)
```

Index

`download.example.data`, [2](#)

`fold.change.filter`, [2](#)

`GeneListPrioritization`, [3](#)

`GeneListPrioritization-package`
 (`GeneListPrioritization`), [3](#)

`get.genes.in.data`, [3](#)

`get.genes.kegg`, [4](#)

`get.genes.tf`, [4](#)

`get.genes.wiki`, [5](#)

`list.filter`, [6](#)

`list.filter.inxs`, [6](#)

`load.gene.data`, [7](#)

`load.protein.data`, [8](#)

`load.WebGestalt`, [8](#)

`overall.mean.filter`, [9](#)

`overall.var.filter`, [10](#)

`random.filter`, [10](#)