Package 'BioIDMapper'

September 4, 2009

Title Mapping between BioIDs

Version 1.2

Date 2009-02-16

Depends R (>= 2.7.0), RCurl, XML

Author Xiaoyong Sun

Maintainer Xiaoyong Sun <sunx1@iastate.edu>

Description facilitate mapping between different databases, integrate various ID systems and provide a full view from gene level, mRNA level and functional level regarding one specific ID. The mapping system is based on NCBI and UniProt web service.

License GPL (>= 2)

LazyData true

Collate updateTerm.R global.R bio.convert.R bio.link.R bio.sum.R bio.type.R bio.select.R dbjumper.R geneMap.R parse1.R parse2.R processMessage.R proteinMap.R validate.R

Repository CRAN

Date/Publication 2009-09-04 07:57:27

R topics documented:

`BioIDMapper-package` *Mapping biological ids and linking to external data source*

Description

This package will map ids among different biological ids from NCBI and Uniprot. Ids will be from gene level, protein level, functional level and metabolomic level.

Details

Package:	BioIDMapper
Type:	Package
Version:	1.0
Date:	2007-09-05
License:	What license is it under?

The main interface is bio.convert() function

Author(s)

Xiaoyong Sun Maintainer: johnsunx1<johnsunx1@gmail.com>

Examples

```
data(glist)
bio.convert(glist, 1, 6)
```

`bio.convert` *Main Interface for Mapping*

Description

This function is main interface for mapping service. It can support NCBI mapping and UniProt mapping, from gene level, protein level, functional level, and metabolomic level.

Usage

```
bio.convert(id_list = "character", from = "numeric", to = "numeric")
```

Arguments

id_list	the list of biological ids
from	Biokey number for source biological type. It can be found by bio.type()
to	Biokey number for destination biological type. It can be found by bio.type()

Details

It converts a list of ids from one biological type to another. For detail, please check vegnette "demonstration" section.

Value

a result matrix will be returned with source type in the first column and the destination type in the last column

Author(s)

Xiaoyong Sun

Examples

```
data(glist)
bio.type(1)
bio.type(5)
bio.convert(glist, 1, 5)->done
```

---

bio.link *Link Between BioIDMapper and External Data Sources*

Description

Link to external data sources through specified ID with web browser

Usage

```
bio.link(id, to)
```

Arguments

id	one biolgoical id
to	biokey number. check bio.type()

Details

It opens web browser and links directly to the specified database. For detail, please check vignette "demonstration" section.

Author(s)

Xiaoyong Sun

Examples

```
bio.type(1)
## Not run: bio.link("200529", 1)
```

`bio.select` *select subset from mapped matrix*

Description

use id as paramter, this function can select subste from mapping result

Usage

```
bio.select(result_matrix, colno, myid)
```

Arguments

`result_matrix`	the returned matrix from bio.convert()
`colno`	the column number from returned matrix, which contains the id you are interesed
`myid`	id you are interesed

Author(s)

Xiaoyong Sun

Examples

```
data(glist)
bio.convert(glist, 1, 5)->done
bio.select(done, 1, "200529" )
```

bio.sum *analyze mapped ids*

Description

analyze the mapped ids from bio.convert()

Usage

```
bio.sum(final_matrix, start_matrix, option)
```

Arguments

final_matrix	the returned matrix from bio.convert()
start_matrix	the original vector or matrix for mapping
option	a logical value. If TRUE, all summary results are returned. If FALSE, only basic summary are returned. default value is FALSE

Author(s)

Xiaoyong Sun

Examples

```
data(glist)
bio.convert(glist, 1, 6)->done
bio.sum(done)
bio.sum(done, glist)
bio.sum(done, glist, TRUE)
```

bio.type *Get the biokey number this package support*

Description

show the biokey number this package support

Usage

```
bio.type(type_to_id)
```

Arguments

type_to_id	If no parameter is used, return tables for all biokey numbers; if parameter is number from bio.type(), return the corresponding biological type; if parameter is biological type, return the corresponding biokey number

Author(s)

Xiaoyong Sun

Examples

```
bio.type()
bio.type(5)
bio.type("PDB id")
```

---

dbjumper *Parse mapping pathway*

Description

parse mapping pathway for gene, protein, metabolomic

Usage

```
dbjumper(id_list, from, to, pathMatrix)
```

Arguments

`id_list`

`from`

`to`

`pathMatrix`

Value

return a matrix with mapped result

Author(s)

Xiaoyong Sun

geneMap *Mapping between NCBI*

Description

map between ncbi ids

Usage

```
geneMap(my_list, from_type, to_type)
```

Arguments

```
my_list
from_type
to_type
```

Value

return mapped result

Author(s)

Xiaoyong Sun

glist *Genbank Gi list*

Description

359 GI number from NCBi

Usage

```
data(glist)
```

Format

A data frame with 359 GI numbers.

Examples

```
data(glist)
```

parse1 *parse the returned result from NCBI*

Description

parse the returned ID result from NCBI

Usage

```
parse1(uri)
```

Arguments

uri	the url address for parsing data

Value

retunr parsed id list

Author(s)

Xiaoyong Sun

parse2 *parse the return result from UniProt*

Description

parse the returned id list to matrix

Value

return a matrix with mapped ids

Author(s)

Xiaoyong Sun

plist *PDB id list*

Description

50 protein data bank ids

Usage

```
data(plist)
```

Format

A data frame with 50 observations on the following variable.

Examples

```
data(plist)
```

processMessage *Return a message*

Description

return message

Usage

```
processMessage(message)
```

Arguments

```
message
```

Value

return a message

Author(s)

Xiaoyong Sun

`proteinMap` *map among protein ids*

Description

map among protein ids using ACC id as key

Usage

```
proteinMap(my_list, from_type, to_type)
```

Arguments

`my_list`	id list `my_list` here~~
`from_type`	biological source type `from_type` here~~
`to_type`	destination source type `to_type` here~~

Value

return mapped matrix

Author(s)

Xiaoyong Sun

`ulist` *Uniprot id list*

Description

11 UniProt Accession Numbers

Usage

```
data(ulist)
```

Format

A data frame with 11 UniProt Accession Numbers.

Examples

```
data(ulist)
```

updateTerm *Update Term*

Description

when loading package, automatically get parameters from author's webpage

Usage

```
updateTerm(biourl)
```

Arguments

```
biourl
```

Author(s)

Xiaoyng Sun

validate *Delete extra copies from matrix*

Description

return unique ids

Usage

```
validate(id_list)
```

Arguments

id_list	the list of biological ids

Value

Return matrix of unique id pairs

Author(s)

Xiaoyong Sun

Index