Package 'IdMappingRetrieval'

April 10, 2015

Type Package

Title ID Mapping Data Retrieval

Version 1.12.0

Date 2011-06-30

Author Alex Lisovich, Roger Day

Maintainer Alex Lisovich <all67@pitt.edu>, Roger Day <day01@pitt.edu>

Description Data retrieval for identifier mapping performance analysis

License GPL-2

Depends R.oo, XML, RCurl, rChoiceDialogs, ENVISIONQuery

Imports biomaRt, ENVISIONQuery, DAVIDQuery, AffyCompatible, R.methodsS3, R.oo, utils

LazyLoad yes

Collate 'annotation.R' 'annotationAffx.R' 'annotationDataRoot.R' 'zzz.R' 'annotationDavid.R' 'annotationDavidCsv.R' 'annotationEnsembl.R' 'annotationEnsemblCsv.R' 'annotationEnvision.R' 'annotationNetAffx.R' 'compatibility.R' 'IdMappingRetrieval-package.R' 'serviceManager.R'

biocViews Annotation, MultipleComparison

R topics documented:

IdMappingRetrieval-package

Id Mapping Retrieval.

Description

Id Mapping Retrieval.

Details

Package:	IdMappingRetrieval
Type:	Package
Version:	0.99.0
Date:	2011-06-26
License:	GPL-2
LazyLoad:	yes

Author(s)

Alex Lisovich, Roger Day

Annotation *The Annotation class*

Description

Package:
Class Annotation

Object
~~|
~~+--Annotation

Directly known subclasses:
AnnotationAffx, AnnotationDavid, AnnotationDavidCsv, AnnotationEnsembl, AnnotationEnsemblCsv, AnnotationEnvision, AnnotationNetAffx

public abstract static class **Annotation**
extends Object

This is the base annotation class from which the concrete classes like AnnotationAffx, AnnotationEnvision etc. are derived

Usage

```
Annotation(cacheFolderName="", primaryColumn="From", secondaryColumn="To", swap=FALSE, species="Homo
```

Arguments

cacheFolderName

              The path to a service cashing directory for a given Annotation object. The path is relative to the caching subsystem root directory. Default is 'Affymetrix'

primaryColumn   Primary column to be retrieved from a data frame obtained for a given service when getIdMap() on a given annotation object is called. Default is 'From'.

secondaryColumn

              Secondary column(s) to be retrieved from a data frame obtained for a given service when getIdMap on a given annotation object is called. Default is 'To'.

swap   A logical indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call.Default is FALSE.

species   A character vector or [NA](NA) indicating if filtering of the results on a particular set of species should be performed if a given service provides the species information. If [NA](NA), no filtering is performed. Default is "Homo sapiens".

verbose   if TRUE enables diagnostic messages. Default is FALSE.

...   Additional parameters

Fields and Methods

Methods:

getArrayType	Select the array type.
getArrayTypes	Get all available microarray chip types.
getCredentials	Retrieve credentials for a given service.
getDataFrame	Get the entire data set available from a particular service in a form of a data frame.
getFolderName	Get caching folder name for a given Annotation object.
getIdMap	Get an IdMap object using the data retrieved by a particular service represented by annotation object.
getRoot	Get the path to service caching directory.
getServiceRoot	Get a root directory for a particular annotation object.
init	Init annotation subsystem.
setCredentials	Create and store the credentials information for given service.
setOptions	Set the parameters for an annotation object.

Methods inherited from Object:

$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

AnnotationAffx   *The AnnotationAffx class*

Description

Package:
Class AnnotationAffx

[Object](#)
~~|
~~+--[Annotation](#)
~~~~~~|
~~~~~~+--AnnotationAffx

Directly known subclasses:
AnnotationNetAffx

public static class **AnnotationAffx**
extends *[Annotation](#)*

The AnnotationAffx class encapsulates the functionality allowing to retrieve data from the Affymetrix annotation data online repository through the getIdMap() and getDataFrame() calls.

Usage

    AnnotationAffx(cacheFolderName="Affymetrix", primaryColumn="Probe.Set.ID", secondaryColumn="SwissPro

Arguments

cacheFolderName

                The path to a service cashing directory for a given AnnotationAffx object. The path is relative to the caching subsystem root directory. Default is 'Affymetrix'

primaryColumn   Primary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'Probe.Set.ID'.

secondaryColumn

                Secondary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'SwissProt'.

swap   Logical indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call.Default is [TRUE](#).

...   Additional parameters, see [Annotation](#).

Fields and Methods

Methods:

[getProbesetList](#)   Get Affymetrix probeset ID list for a given array type.
[setCredentials](#)   Create and store the credentials information for Affymetrix service.

Methods inherited from Annotation:
getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

AnnotationDavid   *The AnnotationDavid class*

Description

Package:
Class AnnotationDavid

[Object](#)
~~|
~~+--[Annotation](#)
~~~~~~~|
~~~~~~~+--AnnotationDavid

Directly known subclasses:
[AnnotationDavidCsv](#)

public static class **AnnotationDavid**
extends *[Annotation](#)*

The AnnotationDavid class encapsulates the functionality allowing to retrieve data from the DAVID online query system through the getIdMap() and getDataFrame() calls on this AnnotationDavid object.

Usage

AnnotationDavid(cacheFolderName="DAVID", primaryColumn="AFFYMETRIX_3PRIME_IVT_ID", secondaryColumn="

Arguments

cacheFolderName

        The symbolic name of a service represented by a given AnnotationDavid object.

primaryColumn   Primary column to be retrieved from a data frame obtained from the Affymetrix
annotation repository when getIdMap() is called. Default is 'AFFYMETRIX_3PRIME_IVT_ID'.

secondaryColumn

        Secondary column to be retrieved from a data frame obtained from the Affymetrix
annotation repository when getIdMap() is called. Default is 'UNIPROT_ACCESSION'.

swap   A logical indicating if primary and secondary column(s) need to be swapped
at the end of the IdMap retrieval during the getIdMap() call.Default is TRUE.

...   Additional parameters, see Annotation.

Fields and Methods

Methods:
No public methods defined.

Methods inherited from Annotation:
getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap,
getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll,
load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

Examples

```
## Not run:
 Annotation$init();
 #create DAVID annotation object
 annObj<-AnnotationDavid"DAVID",species="Homo sapiens");

## End(Not run)
```

AnnotationDavidCsv   *The AnnotationDavidCsv class*

Description

Package:
Class AnnotationDavidCsv

```
Object
~~|
~~+--Annotation
~~~~~~~|
~~~~~~~+--AnnotationDavid
~~~~~~~~~~~~|
~~~~~~~~~~~~+--AnnotationDavidCsv
```

Directly known subclasses:

public static class **AnnotationDavidCsv**
extends AnnotationDavid

The AnnotationDavidCsv class encapsulates the functionality allowing to retrieve data from the DAVID backend annotation file system through the Annotation.getIdMap() and Annotation.getDataFrame() calls on this object. The DAVID backend annotation file system requires to submit the request through the e-mail for convertion of one ID type to another. Within a day, the user recieves an email reply containing the link to the resulting conversion file.The results are returned in a form of a tab delimited headerless file containing the match pairs, one pair per line. The AnnotationDavidCsv object encapsulates the functionality allowing to interactively choose the resulting files and convert it into a data frame during the Annotation.getIdMap() and Annotation.getDataFrame() calls on the AnnotationDavidCsv object.

Usage

```
AnnotationDavidCsv(cacheFolderName="DavidCsv", primaryColumn="AFFY_ID", secondaryColumn="UNIPROT_ACC
```

Arguments

cacheFolderName

> The symbolic name of a service represented by a given AnnotationDavidCsv object.

primaryColumn   Primary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'AFFY_ID'.

secondaryColumn

> Secondary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'UNIPROT_ACCESSION'.

swap   Logical indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call.Default is TRUE.

df_filename   Character string or NULL. In the first case the character string contains the name of conversion results file and in the second case the file name is determined

interactively through the Open File dialog during the call to getIdMap() or get-
DataFrame() on the AnnotationDavidCsv object.

...   Additional parameters, see [Annotation](#).

Fields and Methods

Methods:
No public methods defined.

Methods inherited from AnnotationDavid:
getColumns, readDF

Methods inherited from Annotation:
getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, final-ize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

AnnotationEnsembl   *The AnnotationEnsembl class*

Description

Package:
Class AnnotationEnsembl

[Object](#)
~~|
~~+--[Annotation](#)
~~~~~~~|
~~~~~~~+--AnnotationEnsembl

Directly known subclasses:
[AnnotationEnsemblCsv](#)

public static class **AnnotationEnsembl**
extends *[Annotation](#)*

The AnnotationEnsembl class encapsulates the functionality allowing to retrieve data from the En-sembl BioMart online query system using biomaRt R package through the Annotation.getIdMap() and Annotation.getDataFrame() calls on this object.

Usage

```
AnnotationEnsembl(cacheFolderName="Ensembl", primaryColumn=c("uniprot_swissprot_accession", "uniprot
```

Arguments

cacheFolderName

The symbolic name of a service represented by a given AnnotationEnsembl object.

primaryColumn   Primary column(s) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. As the Ensembl returns the match results for SwissProt and Tremb accessions in separate columns, it is possible to retrieve either or them or merge them together by explicetely specifying the set of columns to be merged. Default is c('uniprot_swissprot_accession','uniprot_sptrembl').

secondaryColumn

secondaryColumn Secondary column (containing probeset IDs) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. If [NA] (default), the column name(s) derived automatically from the array type parameter during the getDataFrame() call. It should be noted that the probeset ID column name in Ensembl data format is array specific ('Affy.HG.U133.PLUS.2' for example) and therefore needs to be selected on per array basis if specified explicitly.

swap   A [logical] indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call.Default is [TRUE].

species   Character vector or [NA] indicating if filtering of the results on a particular set of species should be performed if a given service provides the species information. If [NA], no filtering is performed. Default is 'hsapiens_gene_ensembl'.

full.merge   A [logical] indicating which version of primary columns merging algorithm to use. If@ TRUE (default), all unique pairs <probeset ID, SwissProt> and <probeset ID, Trembl> are generated, and if [FALSE], only those pairs from <probeset ID, Trembl> for which Uniprot ID is not present in <probeset ID, SwissProt> pairs are included.

...   Additional parameters, see [Annotation].

Fields and Methods

Methods:
No public methods defined.

Methods inherited from Annotation:
getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

Examples

```
## Not run:
 Annotation$init();
 #create Ensembl annotation object
 annObj<-AnnotationEnsembl(species="hsapiens_gene_ensembl");

## End(Not run)
```

AnnotationEnsemblCsv   *The AnnotationEnsemblCsv class*

Description

Package:
Class AnnotationEnsemblCsv

[Object](#)
~~|
~~+--[Annotation](#)
~~~~~~~|
~~~~~~~+--[AnnotationEnsembl](#)
~~~~~~~~~~~|
~~~~~~~~~~~+--AnnotationEnsemblCsv

Directly known subclasses:

public static class **AnnotationEnsemblCsv**
extends [AnnotationEnsembl](#)

The AnnotationEnsemblCsv class encapsulates the functionality allowing to retrieve data from the Ensembl intercative online query system. The ID matching information fitered on species and the microarray chip type is retrieved as comma delimited csv file. The AnnotationEnsemblCsv object encapsulates the functionality allowing to interactively choose the Ensembl query results csv file and convert it into a data frame during the getIdMap() and getDataFrame() calls on the AnnotationEnsemblCsv object.

Usage

```
AnnotationEnsemblCsv(cacheFolderName="EnsemblCsv", primaryColumn=c("UniProt.SwissProt.Accession", "U
```

Arguments

cacheFolderName

> The symbolic name of a service represented by a given AnnotationEnsembl object.

primaryColumn Primary column(s) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. As the Ensembl returns the match results for SwissProt and Tremb accessions in separate columns, it is possible to retrieve either or them or merge them together by explicetely specifying the set of columns to be merged. Default is c('uniprot_swissprot_accession','uniprot_sptrembl').

secondaryColumn

> secondaryColumn Secondary column (containing probeset IDs) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. If [NA](default), the column name(s) derived automatically from the array type parameter during the getDataFrame() call. It should be noted that the probeset ID column name in Ensembl data format is array specific ('Affy.HG.U133.PLUS.2' for example) and therefore needs to be selected on per array basis if specified explicitely.

swap A [logical](#) indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call.Default is [TRUE](#).

full.merge A [logical](#) indicating which version of primary columns merging algorithm to use. If@ TRUE (default), all unique pairs <probeset ID, SwissProt> and <probeset ID, Tremb> are generated, and if [FALSE](#), only those pairs from <probeset ID, Tremb> for which Uniprot ID is not present in <probeset ID, SwissProt> pairs are included.

df_filename Character string or NULL. In the first case the character string contains the name of conversion results file and in the second case the file name is determined interactively through the Open File dialog during the call to Annotation.getIdMap() or Annotation.getDataFrame() on the AnnotationEnsemblCsv object.

... Additional parameters, see [Annotation](#).

Fields and Methods

Methods:

No public methods defined.

Methods inherited from AnnotationEnsembl:

getColumns, getColumns1, getColumns2, readDF

Methods inherited from Annotation:

getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

Methods inherited from Object:

$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

Examples

```
## Not run:
 Annotation$init();
 #create Ensembl annotation object
 annObj<-AnnotationEnsemblCsv(cacheFolderName="EnsemblCsv");

## End(Not run)
```

AnnotationEnvision   *The AnnotationEnvision class*

Description

Package:
Class AnnotationEnvision

Object
~~|
~~+--Annotation
~~~~~~~|
~~~~~~~+--AnnotationEnvision

Directly known subclasses:

public static class **AnnotationEnvision**
extends *Annotation*

The AnnotationEnvision class encapsulates the functionality allowing to retrieve data from the Envision online query system.The Envision online query system allows to retrieve the ID matching information accompanied by multiple attributes like species and the microarray chip type in the form of the xml file. The AnnotationEnvision class encapsulates the functionality allowing to filter the Envision query results on species and microarray type attributes and convert it into a data frame during the getIdMap() and getDataFrame() calls on the AnnotationEnvision object.

Usage

```
AnnotationEnvision(cacheFolderName="EnVision", primaryColumn=c("UniProt.SwissProt.Accession", "UniPr
```

Arguments

cacheFolderName

> The symbolic name of a service represented by a given AnnotationEnsembl object.

primaryColumn	Primary column(s) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. As the Ensembl returns the match results for SwissProt and Tremb accessions in separate columns, it is possible to retrieve either or them or merge them together by explicetely specifying the set of columns to be merged. Default is c('uniprot_swissprot_accession','uniprot_sptrembl').
secondaryColumn	
	secondaryColumn Secondary column (containing probeset IDs) to be retrieved from a data frame obtained from the Ensembl csv file when getIdMap() is called. If NA (default), the column name(s) derived automatically from the array type parameter during the getDataFrame() call. It should be noted that the probeset ID column name in Ensembl data format is array specific ('Affy.HG.U133.PLUS.2' for example) and therefore needs to be selected on per array basis if specified explicitely.
swap	A logical indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call.Default is TRUE.
species	Character vector or NA indicating if filtering of the results on a particular set of species should be performed if a given service provides the species information. If NA, no filtering is performed. Default is 'Homo sapiens'.
full.merge	A logical indicating which version of primary columns merging algorithm to use. If TRUE (default), all unique pairs <probeset ID, SwissProt> and <probeset ID, Trembl> are generated, and if FALSE, only those pairs from <probeset ID, Trembl> for which Uniparot ID is not present in <probeset ID, SwissProt> pairs are included.
...	Additional parameters, see Annotation.

Fields and Methods

Methods:
No public methods defined.

Methods inherited from Annotation:
getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

Examples

```
## Not run:
 Annotation$init();
 #create Envision annotation object
 annObj<-AnnotationEnvision(species="Homo sapiens");
```

```
## End(Not run)
```

AnnotationNetAffx   *The AnnotationNetAffx class*

Description

Package:
Class AnnotationNetAffx

[Object](#)
~~|
~~+--[Annotation](#)
~~~~~~~|
~~~~~~~+--[AnnotationAffx](#)
~~~~~~~~~~~~|
~~~~~~~~~~~~+--AnnotationNetAffx

Directly known subclasses:

public static class **AnnotationNetAffx**
extends [AnnotationAffx](#)

The AnnotationNetAffx class encapsulates the functionality allowing to retrieve data from the
NetAffx batch query system through the Annotation.getIdMap() and Annotation.getDataFrame()
calls on this object. The NetAffx batch query system requires to submit the probeset IDs by provid-
ing text files in a special format, maximum 10000 IDs per file. The results are returned in a form
of a tab delimited text file, one file per submission, so the query results for a whole array are pre-
sented by a set of such files. The AnnotationNetAffx object encapsulates the functionality allowing
to interactively choose the set of result files and merge them into a single data frame during the
Annotation.getIdMap() and Annotation.getDataFrame() calls on the AnnotationNetAffx object.

Usage

```
AnnotationNetAffx(cacheFolderName="NetAffxCsv", primaryColumn="Probe.Set.ID", secondaryColumn="Swiss
```

Arguments

cacheFolderName

> The symbolic name of a service represented by a given AnnotationNetAffx ob-
> ject.

primaryColumn   Primary column to be retrieved from a data frame obtained from the Affymetrix
                 annotation repository when getIdMap() is called. Default is 'Probe.Set.ID'.

secondaryColumn

> Secondary column to be retrieved from a data frame obtained from the Affymetrix annotation repository when getIdMap() is called. Default is 'SwissProt'.

swap

> A logical indicating if primary and secondary column(s) need to be swapped at the end of the IdMap retrieval during the getIdMap() call.Default is TRUE.

df_filename

> Character vector, character string or NULL. In the first case the character vector contains the names of a resulting file set, in the second, the character string contains the name of directory in which the files are stored, and in the third the file set is determined interactively through the Open File dialog during the call to getIdMap() or getDataFrame() on the AnnotationAffxCsv object.

...

> Additiional parameters, see Annotation.

Fields and Methods

Methods:

createSubmission   Create a set of files for NetAffx batch query.

Methods inherited from AnnotationAffx:
getColumns, getProbesetList, readDF, setCredentials

Methods inherited from Annotation:
getArrayType, getArrayTypes, getColumns, getCredentials, getDataFrame, getFolderName, getIdMap, getRoot, getServiceRoot, init, readDF, setCredentials, setOptions

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

ServiceManager   *The ServiceManager class*

Description

Package:
Class ServiceManager

Object
~~|

```
~~+--ServiceManager
```

Directly known subclasses:

public static class **ServiceManager**
extends [Object](#)

ServiceManager class serves as a container for a set of specialized service objects and provides the means for (optionally interactive) handling of such a set simplifying the process of data retrieval from a variety of resources in a batch mode.

Usage

```
ServiceManager(services=list(), ...)
```

Arguments

services	Annotation services to be contained within the ServiceManager object. Default is [NULL](#).
...	Not used.

Fields and Methods

Methods:

[addServices](#)	Add services to the ServiceManager object.
[getDataFrameList](#)	Collect raw data data from various online query systems utilizing Annotation service functionality
[getDefaultServices](#)	Get the default services available within the annotation system.
[getIdMapList](#)	Collect ID mapping data from various online query systems utilizing Annotation service functiona
[getServices](#)	Get list of annotation services encapsulated within the particular ServiceManager object.
[setServices](#)	Set the list of services for a ServiceManager object.

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

Author(s)

Alex Lisovich, Roger Day

Index