Package ‘IdMappingRetrieval’

December 21, 2016

Type Package
Title ID Mapping Data Retrieval
Version 1.22.0
Date 2016-01-18
Author Alex Lisovich, Roger Day
Maintainer Alex Lisovich <alex.lisovich@gmail.com>, Roger Day <day01@pitt.edu>
Description Data retrieval for identifier mapping performance analysis
License GPL-2
Depends R.oo, XML, RCurl, rChoiceDialogs
Imports biomaRt, ENVISIONQuery, AffyCompatible, R.methodsS3, utils
LazyLoad yes
Collate 'annotation.R' 'annotationAffx.R' 'annotationDataRoot.R'
   'zzz.R' 'annotationEnsembl.R' 'annotationEnsemblCsv.R'
   'annotationEnvision.R' 'annotationNetAffx.R' 'compatibility.R'
   'IdMappingRetrieval-package.R' 'serviceManager.R'
bioCViews Annotation, MultipleComparison
NeedsCompilation no

R topics documented:

IdMappingRetrieval-package ........................................... 2
Annotation .............................................................. 3
AnnotationAffx .......................................................... 4
AnnotationEnsembl ..................................................... 5
AnnotationEnsemblCsv .................................................. 7
AnnotationEnvision ..................................................... 9
AnnotationNetAffx ....................................................... 11
ServiceManager ......................................................... 12

Index 14
IdMappingRetrieval-package

Id Mapping Retrieval.

Description
Author(s)
Alex Lisovich, Roger Day

The Annotation class

Description
Package:
Class Annotation

Object
```java
|-- Annotation
```

Directly known subclasses:
- AnnotationAffx, AnnotationEnsembl, AnnotationEnsemblCsv, AnnotationEnvision, AnnotationNetAffx

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

Usage
```r
Annotation(cacheFolderName="", primaryColumn="From", secondaryColumn="To", swap=FALSE, species="Homo sapiens")
```

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'.
### Fields and Methods

**Methods:**

- **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.

- **getServiceRoot**
  - Get a root directory for a particular annotation object.

- **setOptions**
  - Set the parameters for an annotation object.

**Methods inherited from Object:**

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

**Author(s)**

Alex Lisovich, Roger Day

---

### 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

```r
AnnotationAffx(cacheFolderName="Affymetrix", primaryColumn="Probe.Set.ID", secondaryColumn="SwissProt", swap=TRUE, ...)
```

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:

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
Directly known subclasses:
AnnotationEnsemblCsv

public static class AnnotationEnsembl
extends Annotation

The AnnotationEnsembl class encapsulates the functionality allowing to retrieve data from the Ensembl 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_sptrembl"), secondaryColumn=NA, swap=FALSE, species="hsapiens_gene_ensembl", full.merge=TRUE, ...)

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 Trembl accessions in separate columns, it is possible to retrieve either or them or merge them together by explicitly specifying the set of columns to be merged. Default is c('uniprot_swissprot_accession', 'uniprot_sptrembl').
- 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, finalize, gc, getEnvironment, getFields, getInstantiationTime, getstaticInstance, hasField, hashCode, ll, load, objectSize, print, registerFinalizer, save

**Author(s)**
Alex Lisovich, Roger Day

**Examples**

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

## End(Not run)
```

---

**AnnotationEnsemblCsv** The AnnotationEnsemblCsv class

**Description**

Package: AnnotationEnsemblCsv

Class AnnotationEnsemblCsv

```r
Object
---
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
Usage

AnnotationEnsemblCsv(cacheFolderName="EnsemblCsv", primaryColumn=c("UniProt.SwissProt.Accession", "UniProt.TrEMBL.Accession"), secondaryColumn=NA, swap=FALSE, full.merge=TRUE, df_filename=NULL, ...)

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 Trembl accessions in separate columns, it is possible to retrieve either or them or merge them together by explicitly setting the specifying the set of columns to be merged. Default is c('uniprot_swissprot_accession', 'uniprot_sptrembl').

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.

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.

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
The AnnotationEnvision class

Description

Package: 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 get DataFrame() calls on the AnnotationEnvision object.

Usage

AnnotationEnvision(cacheFolderName="EnVision", primaryColumn=c("UniProt.SwissProt.Accession", "UniProt.TrEMBL.Accession"), secondaryColumn=NA, ...)

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 Trembl 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

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 ‘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 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:
getAddressType, 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

```r
## 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 providing 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 presented 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="SwissProt", swap=TRUE, df_filename=NULL, ...)
```

Arguments

- cacheFolderName: The symbolic name of a service represented by a given AnnotationNetAffx object.
- 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.

...  Additional parameters, see Annotation.

Fields and Methods

Methods:

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

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

Methods inherited from Object:
S, $<-, [], [[<-, as.character, attach, attachLocally, clearCache, clone, detach, equals, extend, finalize, gc, getEnvironment, getFields, getInstantiationTime, getStaticInstance, hasField, hash, 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:

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.
  getIdMapList Collect ID mapping data from various online query systems utilizing Annotation service functionality.
  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:

$<-, $<-, [l, [l<-, 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

*Topic classes*
- Annotation, 3
- AnnotationAffx, 4
- AnnotationEnsembl, 5
- AnnotationEnsemblCsv, 7
- AnnotationEnvision, 9
- AnnotationNetAffx, 11
- ServiceManager, 12

*Topic package*
- IdMappingRetrieval-package, 2

<table>
<thead>
<tr>
<th>Function</th>
<th>Line Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>addServices</td>
<td>13</td>
</tr>
<tr>
<td>Annotation</td>
<td>3, 4–12</td>
</tr>
<tr>
<td>AnnotationAffx</td>
<td>3, 4, 11</td>
</tr>
<tr>
<td>AnnotationEnsembl</td>
<td>3, 5, 7</td>
</tr>
<tr>
<td>AnnotationEnsemblCsv</td>
<td>3, 6, 7</td>
</tr>
<tr>
<td>AnnotationEnvision</td>
<td>3, 9</td>
</tr>
<tr>
<td>AnnotationNetAffx</td>
<td>3, 4, 11</td>
</tr>
<tr>
<td>character</td>
<td>4</td>
</tr>
<tr>
<td>FALSE</td>
<td>4, 6, 8, 10</td>
</tr>
<tr>
<td>getDataFrame</td>
<td>4</td>
</tr>
<tr>
<td>getDataFrameList</td>
<td>13</td>
</tr>
<tr>
<td>getFolderName</td>
<td>4</td>
</tr>
<tr>
<td>getIdMap</td>
<td>4</td>
</tr>
<tr>
<td>getIdMapList</td>
<td>13</td>
</tr>
<tr>
<td>getServiceRoot</td>
<td>4</td>
</tr>
<tr>
<td>getServices</td>
<td>13</td>
</tr>
<tr>
<td>IdMappingRetrieval</td>
<td>2</td>
</tr>
<tr>
<td>IdMappingRetrieval-package</td>
<td>2</td>
</tr>
<tr>
<td>logical</td>
<td>4, 6, 8, 10, 11</td>
</tr>
<tr>
<td>NA</td>
<td>4, 6, 8, 10</td>
</tr>
<tr>
<td>NULL</td>
<td>12, 13</td>
</tr>
<tr>
<td>Object</td>
<td>3–5, 7, 9, 11, 12</td>
</tr>
<tr>
<td>ServiceManager</td>
<td>12</td>
</tr>
<tr>
<td>setOptions</td>
<td>4</td>
</tr>
<tr>
<td>setServices</td>
<td>13</td>
</tr>
<tr>
<td>TRUE</td>
<td>4–6, 8, 10, 11</td>
</tr>
</tbody>
</table>