Package ‘compendiumdb’

October 12, 2015

Type Package

Title Tools for Retrieval and Storage of Functional Genomics Data

Version 1.0.3

Date 2015-10-12

Author Umesh K. Nandal <u.k.nandal@amc.uva.nl> and Perry D. Moerland <p.d.moerland@amc.uva.nl>

Maintainer Umesh Nandal <u.k.nandal@amc.uva.nl>

Description Package for the systematic retrieval and storage of functional genomics data via a MySQL database.

URL http://wiki.bioinformaticslaboratory.nl/foswiki/bin/view/BioLab/CompendiumDB

Depends Biobase, GEOquery, RMySQL

Suggests inSilicoDb, genefilter, GEOmetadb, gplots, GSVA, limma, mogene10sttranscriptcluster.db, RColorBrewer

SystemRequirements Perl (>=5), MySQL (>=5.6)

License GPL (>= 2)

Imports methods

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-12 11:29:15

R topics documented:

`compendiumdb-package` *Tools for Retrieval and Storage of Functional Genomics Data*

Description

Public repositories such as the Gene Expression Omnibus (GEO) contain thousands of high-throughput functional genomics datasets. These datasets are a rich source of useful biological information. Extraction of meaningful information often requires the integration of a large number of datasets from different studies and platforms. The package compendiumdb provides a flexible platform for the systematic retrieval and storage of functional genomics data downloaded from GEO in the form of a MySQL database accessed via R functions. It provides functions to (i) download data from GEO, (ii) store data in the database and (iii) retrieve data from the database.

Details

Package:	compendiumdb
Type:	Package
Version:	1.0.3
Date:	2015-10-12
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Umesh K. Nandal <u.k.nandal@amc.uva.nl> and Perry D. Moerland <p.d.moerland@amc.uva.nl>

`checkUpdates` *Check whether GSE records have been updated on GEO*

Description

Check whether GEO series (GSE) records loaded in the compendium database have been updated on GEO

Usage

```
checkUpdates(con, GSEid = NULL)
```

Arguments

con	`list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)
GSEid	character vector specifying the GSE ID(s). The default value is `NULL`, in which case the function performs a check for all GSEs present in the compendium database.

Value

An object of class `data.frame` consisting of the GSE IDs that were updated after having been loaded in the compendium database, their last update date on GEO and the date on which they were loaded in the compendium database

Note

If a GSE record has been updated on GEO, one can first remove the GSE from the compendium database using the function `removeGSE` and then download the updated GSE record and reload the GSE into the compendium database.

Author(s)

Umesh K. Nandal

See Also

`removeGSE`, `downloadGEOdata`, `loadDataToCompendium`

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  ids <- checkUpdates(conn,"GSE18290")


## End(Not run)
```

`connectDatabase` *Create connection with the MySQL compendium database*

Description

Allows the user to create a connection with the compendium database in the `MySQL` server

Usage

```
connectDatabase(user, password, host = "localhost", dbname = "compendium", port = 3306)
```

Arguments

`user` character string defining the `MySQL` user name to login to the database

`password` character string defining the password required to connect to the `MySQL` database

`host` character string defining the host name. The default value is `"localhost"`. One can also connect to a remote server by defining a valid value for the host name, e.g., `"machinename.\domain.\org"`.

`dbname` character string defining the name of the compendium database to which one wants to establish a connection. The default value is `"compendium"`.

`port` port number used to connect to the `MySQL` server. The default port number is 3306.

Details

The compendium database has to be created first, see the package vignette for how to do this from the MySQL prompt.

Value

A `list` with components

`connect` a component of class `MySQLConnection` containing the connection to the `MySQL` database

`user` character string containing the user name

`password` character string containing the password

`host` character string containing the host name

`port` port number used to connect to the `MySQL` server

`dbname` character string containing the database name

Note

Do not check the returned value of this function, since this might abort the current R session. `summary(conn)` can be used to check the returned `list`.

Author(s)

Umesh K. Nandal

Examples

```
 ## Not run:
  # Connect to a database with name "compendium"
  conn <- connectDatabase(user="usrname",password="passwd",host="localhost",dbname="compendium")

## End(Not run)
```

createESET *Create a Bioconductor ExpressionSet*

Description

Given the identifier of a GEO series (GSE) record creates one or more ExpressionSets from the data loaded in the compendium database

Usage

```
createESET(con, GSEid, GPLid = "", parsing = TRUE)
```

Arguments

con
: list containing a connection object specifying the user name and password to connect or interact with the compendium database (see connectDatabase)

GSEid
: character string specifying the GSE ID to be converted to one or more ExpressionSets

GPLid
: character string specifying the GPL ID. The default value is "", in which case a separate ExpressionSet will be created for each of the GPLs in the GSE specified by GSEid.

parsing
: logical, if set to its default value (TRUE) the phenotypic data of the samples as available in the sample characteristics extracted from GEO will be parsed into separate columns.

Details

This function generates one or more ExpressionSets for the specified GSE from the data loaded in the compendium database. Each ExpressionSet contains an assayData slot with all data related to the expression measurements parsed from a GSE SOFT file. Probe annotation is provided in the featureData slot with all data parsed from the most recent annotation file provided for the corresponding GPL (if available at ftp://ftp.ncbi.nlm.nih.gov/pub/geo/DATA/annotation/platforms/). Sample annotation is provided in the phenoData slot and obtained by parsing the output of the function GSMdescriptions.

Value

A list with components of class ExpressionSet (from the Biobase Bioconductor package). Each ExpressionSet is named according to the GSEid with its corresponding GPL ID(s). If a GSE consists of GSMs with a different number of features, multiple ExpressionSets are created such that GSMs with the same features are grouped into one ExpressionSet.

Author(s)

Umesh K. Nandal

See Also

GSMdescriptions, updatePhenoData

Examples

```
 ## Not run:
  conn <- connectDatabase(dbname="compendium")

  # Create ExpressionSet for the samples in GSE1657 corresponding to GPL96
  esets <- createESET(conn,"GSE1657","GPL96")
  # esets contains one component: "esetGSE1657_GPL96_SC"

  # Create ExpressionSet for the samples of both platforms present in GSE1657 (GPL96 &
  # GPL97), i.e, set GPLid to default value
  esets <- createESET(conn,"GSE1657") # Default GPLid=""
  # esets contains two components: "esetGSE1657_GPL96_SC" and "esetGSE1657_GPL97_SC"

## End(Not run)
```

---

downloadGEOdata *Download a GSE record from GEO*

Description

Downloads the SOFT files for the GSE, GPLs, GSMs, and GDSs corresponding to the GSE identifier provided by the user from GEO to the user’s local machine

Usage

```
downloadGEOdata(GSEid, destdir = getwd())
```

Arguments

GSEid	character string specifying the GSE to be downloaded from GEO
destdir	directory where to locate the BigMac directory used for storing the SOFT files downloaded from GEO. The default directory is the current working directory

Details

In the Gene Expression Omnibus (GEO) high-throughput functional genomics data is stored in SOFT (Simple Omnibus Format in Text) file format. Examples are the series record (GSE), the sample record (GSM), the platform record (GPL), and the dataset record (GDS). More information about the different types of SOFT files can be found at http://www.ncbi.nlm.nih.gov/geo/info/overview.html.
The function downloadGEOdata uses (or creates, if it does not exist yet) a data directory called BigMac in a directory destdir specified by the user. The BigMac directory contains several subdirectories: annotation, COMPENDIUM, data and log. The data directory contains further subdirectories to store the downloaded .soft files corresponding to GSEs, GSMs, GPLs, and GDSs downloaded from GEO. More information about the structure of the BigMac directory can be found at http://wiki.bioinformaticslaboratory.nl/foswiki/bin/view/BioLab/CompendiumDB. If an existing BigMac directory is detected that already contains the necessary SOFT files, these files will not be downloaded from GEO again.

Note

If the BigMac directory already exists, the function downloadGEOdata will try to store the downloaded data in the existing directory structure. Therefore, in order to avoid errors, do not change BigMac's directory structure.

Author(s)

Umesh K. Nandal

See Also

loadDatabaseSchema, loadDataToCompendium

Examples

```
 ## Not run:
  # Download the files related to the specified GSE from GEO to the BigMac directory
  # in the user's current working directory
  downloadGEOdata(GSEid="GSE23183")

## End(Not run)
```

GDSforGSE *Retrieve information about a GDS for a given GSE*

Description

Retrieve information about the GDS(s) corresponding to given GSE ID

Usage

```
GDSforGSE(con, GSEid)
```

Arguments

`con` `list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)

`GSEid` character string specifying the GSE ID

Details

The GEO staff manually curates part of the records in GEO and reassembles biologically and statistically comparable records into a GEO dataset (GDS). This function allows the user to check if the series record (GSE) has been manually curated by GEO and has a corresponding GDS ID.

Value

An object of class `data.frame` returned by `GSEinDB` giving detailed information on the corresponding GDS(s).

Author(s)

Umesh K. Nandal

See Also

`GSEinDB`

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  # Retrieve information about GDSs corresponding to GSE1657
  GDSforGSE(conn,c("GSE1657"))

## End(Not run)
```

`GSEforGPL` *Retrieve information about a GSE for a given GPL*

Description

Retrieve information about GSE(s) corresponding to given GPL ID(s)

Usage

```
GSEforGPL(con, GPLid)
```

Arguments

`con` `list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)

`GPLid` character vector specifying the GPL ID(s)

Value

An object of class `data.frame` returned by `GSEinDB` giving detailed information on the corresponding GSE(s).

Author(s)

Umesh K. Nandal

See Also

`GSEinDB`

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  # Retrieve information about GSEs corresponding to three GPLs
  GSEforGPL(conn,c("GPL96","GPL97","GPL570"))

## End(Not run)
```

`GSEinDB` *Retrieve information about a GSE loaded in the compendium database*

Description

Retrieve information about GEO series (GSE) records present in the compendium database

Usage

```
GSEinDB(con, GSEid = NULL)
```

Arguments

`con` `list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)

`GSEid` character vector specifying the GSE ID(s). The default value is `NULL`, in which case the function returns an overview of all GSEs present in the compendium database.

Value

An object of class `data.frame` consisting of ten columns: i) ID of the record in the compendium database, ii) GSE ID, iii) educated guess on the experimental design of the experiment, iv) GPL ID, v) number of samples, vi) user-specified tag for the experiment, (see `tagExperiment`), vii) NCBI taxonomy ID, viii) corresponding organism name, ix) GDS ID and x) date and time on which the data was loaded in the database

Note

The value for the variable `experimentDesign` is determined by parsing the sample information provided by GEO. The variable can take the following values: i) SC: single-channel design, ii) DC: double-channel design, iii) DS: double-channel dye-swap design (if the same source name occurs in both channels) and iv) CR: double-channel common reference design (if the source name is equal for all samples in one of the two channels). The attribution of 'DS' and 'CR' labels makes assumptions on how source names are represented in GEO and should be interpreted with caution.

Author(s)

Umesh K. Nandal

See Also

`GDSforGSE`, `GSEforGPL`, `tagExperiment`

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  GSEinDB(conn,"GSE1657")

## End(Not run)
```

`GSMdescriptions` *List sample annotation of samples for a given GSE*

Description

Extract the phenotypic data of each sample record (GSM) in the specified GSE in a tabular format

Usage

```
GSMdescriptions(con, GSEid, GPLid = "")
```

Arguments

`con` `list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)

`GSEid` character string specifying the GSE ID

`GPLid` character string specifying the GPL ID. The default value is `""`, in which case the phenotypic data will be extracted for each of the GPLs in the GSE specified by `GSEid`.

Details

The function uses the corresponding GDS (if available for that GSE) in order to retrieve the phenotypic data. If a GDS is not available, it generates phenotypic data based on the sample characteristics, sample source, and sample title specified for each GSM. In case of a double-channel experiment, sample characteristics and sample source are given for both channels.

Value

A character `matrix` containing a row for each GSM and columns for the phenotypic data and the GPL ID(s) of the platform used.

Author(s)

Umesh K. Nandal

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  GSMdescriptions(conn,"GSE1657")

## End(Not run)
```

`loadDatabaseSchema` *Load the compendium database schema*

Description

Load a database schema file to the compendium database in the `MySQL` server

Usage

```
  loadDatabaseSchema(con, updateSchema = FALSE , file = "")
```

Arguments

`con`	`list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)
`updateSchema`	logical, default value is `FALSE`
`file`	character string, default value is `""`. In this case the `compendiaSchema.sql` database schema provided with the package is loaded.

Details

See `http://wiki.bioinformaticslaboratory.nl/foswiki/bin/view/BioLab/CompendiumDB` for a detailed description of the database schema.

Note

Execute this function only after having created the database specified in the connection object in the MySQL server. Set the updateSchema value TRUE only before filling the database with series record data for the first time, or if you want to delete all the records of the database and reload the schema. In the latter case the user will be prompted whether (s)he really wants to update the current schema and delete all data in the compendium database.

Author(s)

Umesh K. Nandal

See Also

connectDatabase

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  loadDatabaseSchema(conn,updateSchema=TRUE)

## End(Not run)
```

---

loadDataToCompendium *Load GSE into the compendium database*

Description

Load the data from SOFT files corresponding to the specified GSE and GPL(s) into the tables of the MySQL compendium database

Usage

```
loadDataToCompendium(con, GSEid, GPLid = "", datadir = getwd())
```

Arguments

con — list containing a connection object specifying the user name and password to connect or interact with the compendium database (see connectDatabase)

GSEid — character string specifying the GSE ID to be loaded into the compendium database

GPLid — character vector specifying the GPL ID(s). The default value is "" and will load all the GPL ID(s) corresponding to the GSE specified by GSEid.

datadir — directory where the BigMac directory used for storing the SOFT files downloaded from GEO has been created (see downloadGEOdata). The default directory is the current working directory.

Details

The SOFT files downloaded from GEO using the function `downloadGEOdata` are parsed and loaded into the compendium database. This function can be called once all the SOFT files corresponding to the specified `GSEid` have been downloaded to the `BigMac` directory (see `downloadGEOdata`). The `BigMac` directory should be a subdirectory of the directory specified by the user via the argument `datadir`. The `GPLid` argument provides the option to only load the data for a specific platform.

Author(s)

Umesh K. Nandal

See Also

`downloadGEOdata`

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  downloadGEOdata("GSE1657")

  # GSE1657 has GPL96 and GPL97 platform data. Load only GPL96 data
  loadDataToCompendium(conn,"GSE1657","GPL96")

  # Load both GPL96 and GPL97 as a character vector
  loadDataToCompendium(conn,"GSE1657",c("GPL96","GPL97"))
  # Both platforms can also be loaded using the default value for GPLid

  # Load multiple GSEs to the compendium
  for (i in  c("GSE4251","GSE6495","GSE12597","GSE1657")){
   loadDataToCompendium(con=conn,GSEid=i)
  }

## End(Not run)
```

`removeGSE` *Remove a GSE from the compendium database*

Description

Remove a GEO series (GSE) record and other entries corresponding to it from the compendium database

Usage

```
removeGSE(con, GSEid)
```

Arguments

con	`list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)
GSEid	character string specifying the GSE ID to be removed

Details

A side effect of this function is that the corresponding GPL is also removed from the compendium database if the removed GSE was the only one with this GPL ID.

Author(s)

Umesh K. Nandal

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  removeGSE(conn,"GSE23183")

## End(Not run)
```

`tagExperiment` *Tag an experiment with text labels*

Description

Tag an experiment with text labels

Usage

```
tagExperiment(con, GSEid, tag)
```

Arguments

con	`list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)
GSEid	character string specifying the GSE ID
tag	character string specifying the text labels with which to tag the GSE specified by `GSEid`

Details

This function updates the value of the `tag` record for the specified GSE ID in the compendium database; see the variable `tagExperiment` of the data frame returned by the `link{GSEinDB}` function. Adding tags makes it easy to search for specific experiments.

Author(s)

Umesh K. Nandal

See Also

GSEinDB

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  tagExperiment(conn,"GSE23183","HIV infection")
  GSEinDB(con=conn,"GSE23183")

## End(Not run)
```

updatePhenoData *Update the phenotypic data of a GSE record*

Description

Update the phenotypic data of a GEO series (GSE) record and store the updated phenotypic data into the compendium database

Usage

```
updatePhenoData(con, GSEid, data)
```

Arguments

`con` `list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`)

`GSEid` character string specifying the GSE ID

`data` character `matrix` object containing all GSM IDs for the GSE specified by `GSEid` as rownames followed by columns containing updated annotation of the corresponding samples. Column names may different from those returned by `GMSdescriptions`. This will overwrite the phenotypic data currently stored in the compendium database and the user is prompted to confirm this.

Author(s)

Umesh K. Nandal

Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")

  GSMdescriptions(conn,"GSE18290")
  tab <- GSMdescriptions(conn,"GSE18290")

  # As an example just replace the current annotation by the same annotation
  updatePhenoData(conn,"GSE18290",tab)
  GSMdescriptions(conn,"GSE18290")


## End(Not run)
```

Index