# Package 'compendiumdb'

August 29, 2014

**Type** Package

**Title** Tools for Storage and Retrieval of Gene Expression Data

**Version** 0.1.0

**Date** 2014-08-29

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

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

**Description** Package for the systematic collection, storage and retrieval of gene expression data via a MySQL database.

**Depends** Biobase, RMySQL

**Suggests** inSilicoDb, limma, hgu133a.db, GSVA, GSVAdata

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

**License** GPL (>= 2)

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-08-29 19:20:50

## R topics documented:

---

`compendiumdb-package` *A database and R package for storing and analyzing gene expression data*

---

## Description

Public repositories such as Gene Expression Omnibus (GEO) contain thousands of microarray experiment 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 microarray studies and platforms. The package compendiumdb provides a flexible platform for the systematic collection, storage and retrieval of gene expression data downloaded from GEO in the form of a MySQL database accessed via R functions. It provides functions to (a) download data from GEO and load data into the database, (b) store data in the database and (c) retrieve data from the database.

## Details

| | |
|---|---|
| Package: | compendiumdb |
| Type: | Package |
| Version: | 0.1.0 |
| Date: | 2014-02-13 |
| License: | GNU General Public License (GPL) - version 2 |
| LazyLoad: | yes |

## Author(s)

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

---

`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")
```

## 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., "username.userserver.com". |
| dbname | character string defining the name of the compendium database to which one wants to establish a connection. The default value is "compendium". |

## 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 |
| dbname | character string containing the database name |

## Note

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

## Author(s)

Umesh Nandal

## Examples

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

## End(Not run)
```

createESET *Create a Bioconductor ExpressionSet*

## Description

Given the identifier(s) of the GEO series record (GSE) creates an ExpressionSet 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 vector specifying the GSE(s) to be converted to one or more ExpressionSets |
| GPLid | character vector specifying the GPL(s). The default value is "", in which case a separate ExpressionSet will be created for each of the GPLs in a GSE. |
| 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 an ExpressionSet instance for the specified GSE(s) from the data loaded in the compendium database. The ExpressionSet instance contains an assayData slot with all data related to the expression measurement parsed from 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

Object(s) of class ExpressionSet (from the Biobase package). Each object of the ExpressionSet will be named according to the GSEid with its corresponding GPLid(s). If a GSE consists of GSMs with a different number of features, a list of ExpressionSets is returned such that GSMs with the same features are grouped into one ExpressionSet.

## Author(s)

Umesh Nandal

## See Also

GSMdescriptions, updatePhenoData

## Examples

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

 # Create ExpressionSet for the samples in GSE1657 corresponding to GPL96
 createESET(conn,"GSE1657","GPL96")

 # Create ExpressionSet for the samples of both platforms present in GSE1657 (GPL96 &
 # GPL97), i.e, set GPLid to default value
 createESET(conn,"GSE1657") # Default GPLid=""
 # Objects created are "esetGSE1657_GPL96" and "esetGSE1657_GPL97"

## End(Not run)
```

---

# downloadGEOdata *Download a GSE 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 store the SOFT files downloaded from GEO. The default directory is a subdirectory of the current working directory. |

## Details

In the Gene Expression Omnibus (GEO) high-throughput experimental 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 creates 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://www.bioinformaticslaboratory.nl/twiki/bin/view/BioLab/CompendiumDB.

## 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 Nandal

## See Also

loadDatabaseSchema, loadDataToCompendium

## Examples

```
 ## Not run:
  # This will 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 the GDS ID for a given GSE ID*

---

## Description

Retrieve the GDS ID(s) corresponding to a 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 vector specifying the GSE ID(s) |

## Details

The GEO staff manually curates part of the records in GEO and reassembles the biologically and statistically comparable records into a 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

## Author(s)

Umesh Nandal

## See Also

GSEinDB

## Examples

```
 ## Not run:
  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  GDSforGSE(conn,c("GSE1657","GSE1428"))

## End(Not run)
```

---

GSEforGPL *Retrieve the GSE ID for a given GPL ID*

---

## Description

Retrieve the GSE ID(s) corresponding to a given GPL ID

## 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

## Author(s)

Umesh Nandal

## See Also

GSEinDB

## Examples

```
## Not run:
 conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
 # Query to find GSEs for the character vector of GPL(s)
 GSEforGPL(conn,c("GPL96","GPL97","GPL570"))

## End(Not run)
```

---

GSEinDB *Check presence of GSEs in compendium database*

---

## Description

Provide an overview of the GSE IDs 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 and the function then returns information on 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 of the experimental design, iv) GPL ID, v) number of samples, vi) user-specified tag for the experiment, (see tagExperiment), vii) NCBI taxonomy ID, viii) corresponding organism namem, 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 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 in an experiment*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| conn | 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 vector specifying the GPL ID. The default value is "", in which case the phenotypic data will be extracted for each of the GPLs in a GSE |

## 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 of the platform used.

## Author(s)

Umesh 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://www.bioinformaticslaboratory.nl/twiki/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 once, i.e. before filling the database with series record data, or if you want to delete all the records of the database and reload the schema.

## Author(s)

Umesh Nandal

## See Also

link{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) |
| `datadir` | directory where the SOFT files downloaded from GEO are stored. The default directory is the BigMac directory (see `downloadGEOdata`) in 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 `GSEid` have been downloaded to the `BigMac` directory (see `downloadGEOdata`) that 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 Nandal

## 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")
 # Both platforms can 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 series record from compendium database*

---

## Description

Remove a GSE 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 to be removed |

## Details

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

## Author(s)

Umesh 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 experiment |

## Details

This function will update 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. This makes it easy to search for specific experiments based on the tags that were added

## Author(s)

Umesh 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 set of samples*

## Description

Curate and update the phenotypic data of a set of GSMs and store the updated phenotypic data into the compendium database

## Usage

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

## Arguments

| | |
|---|---|
| con | `list` containing a connection object specifying the user name and password to connect or interact with the compendium database (see `connectDatabase`) |
| data | `data.frame` object containing GSM IDs as rownames followed by columns containing updated annotation of the corresponding samples |

## Author(s)

Umesh Nandal

## Examples

```
 ## Not run:
  barcode <- c("GSM28491","GSM28479","GSM30659","GSM30655")
  cellLine <- c("primary culture","primary culture","transduced","transduced")
  tissue <- c("omental","omental","subcutaneous","subcutaneous")
  data <- data.frame(cellLine,tissue)
  rownames(data) <- barcode

  conn <- connectDatabase(user="usrname",password="passwd",dbname="compendium")
  updatePhenoData(conn,data)

## End(Not run)
```

# Index