Package 'GEOmetadb'

November 12, 2026

Type Package

Title A compilation of metadata from NCBI GEO

Version 1.74.0

Date 2026-03-24

Depends R.utils,RSQLite

Suggests knitr, rmarkdown, dplyr, dbplyr, tm, wordcloud

Author Jack Zhu and Sean Davis

Maintainer Jack Zhu <zhujack@mail.nih.gov>

biocViews Infrastructure

Description The NCBI Gene Expression Omnibus (GEO) represents the largest public repository of microarray data. However, finding data of interest can be challenging using current tools. GEOmetadb is an attempt to make access to the metadata associated with samples, platforms, and datasets much more feasible. This is accomplished by parsing all the NCBI GEO metadata into a SQLite database that can be stored and queried locally. GEOmetadb is simply a thin wrapper around the SQLite database along with associated documentation. Finally, the SQLite database is updated regularly as new data is added to GEO and can be downloaded at will for the most up-to-date metadata. GEOmetadb paper: http://bioinformatics.oxfordjournals.org/cgi/content/short/24/23/2798 .

VignetteBuilder knitr

License Artistic-2.0

git_url https://git.bioconductor.org/packages/GEOmetadb

git_branch RELEASE_3_23

git_last_commit b213e29

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

GEOmetadb-package 2
columnDescriptions 3
geoConvert 4
getBiocPlatformMap 5
getSQLiteFile 6

Index **7**

`GEOmetadb-package` *Query NCBI GEO metadata from a local SQLite database*

Description

The NCBI Gene Expression Omnibus (GEO) represents the largest public repository of microarray data. However, finding data of interest can be challenging using current tools. GEOmetadb is an attempt to make access to the metadata associated with samples, platforms, and datasets much more feasible. This is accomplished by parsing all the NCBI GEO metadata into a SQLite database that can be stored and queried locally. GEOmetadb is simply a thin wrapper around the SQLite database along with associated documentation. Finally, the SQLite database is updated regularly as new data is added to GEO and can be downloaded at will for the most up-to-date metadata.

Details

Package:	GEOmetadb
Type:	Package
Version:	1.1.5
Date:	2008-09-09
License:	Artistic-2.0

Author(s)

Jack Zhu and Sean Davis

Maintainer: Jack Zhu <zhujack@mail.nih.gov>

Examples

```
## Use the demo GEOmetadb database:
if( !file.exists("GEOmetadb.sqlite") ) {
   demo_sqlfile <- getSQLiteFile(destdir = getwd(), destfile = "GEOmetadb.sqlite.gz", type = "demo")
} else {
    demo_sqlfile <- "GEOmetadb.sqlite"
}
columnDescriptions(demo_sqlfile)[1:5,]
```

```
a <- columnDescriptions(demo_sqlfile)[1:5,]
b <- geoConvert('GPL96', out_type='GSM', sqlite_db_name=demo_sqlfile)

## Download the full GEOmetadb database:
## Not run: geometadbfile <- getSQLiteFile()
```

`columnDescriptions` *Get column descriptions for the GEOmetadb database*

Description

Searching the GEOmetadb database requires a bit of knowledge about the structure of the database and column descriptions. This function returns those column descriptions for all columns in all tables in the database.

Usage

```
columnDescriptions(sqlite_db_name='GEOmetadb.sqlite')
```

Arguments

`sqlite_db_name`	The filename of the GEOmetadb sqlite database file

Value

A three-column data.frame including TableName, FieldName, and Description.

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

Examples

```
## Use the demo GEOmetadb database:
if( !file.exists("GEOmetadb.sqlite") ) {
   demo_sqlfile <- getSQLiteFile(destdir = getwd(), destfile = "GEOmetadb.sqlite.gz", type = "demo")
} else {
    demo_sqlfile <- "GEOmetadb.sqlite"
}
columnDescriptions(demo_sqlfile)[1:5,]

## Download the full GEOmetadb database:
## Not run: geometadbfile <- getSQLiteFile()
```

geoConvert *Cross-reference between GEO data types*

Description

A common task is to find all the GEO entities of one type associated with another GEO entity (eg., find all GEO samples associated with GEO platform 'GPL96'). This function provides a very fast mapping between entity types to facilitate queries of this type.

Usage

```
geoConvert(in_list, out_type = c("gse", "gpl", "gsm", "gds", "smatrix"), sqlite_db_name = "GEOmetadb.sq
```

Arguments

in_list	Character vector of GEO entities to convert from.
out_type	Character vector of GEO entity types to which to convert.
sqlite_db_name	The filename of the GEOmetadb sqlite database file

Value

A list of data.frames.

Author(s)

Jack Zhu <zhujack@mail.nih.gov>

Examples

```
## Use the demo GEOmetadb database:
if( !file.exists("GEOmetadb.sqlite") ) {
   demo_sqlfile <- getSQLiteFile(destdir = getwd(), destfile = "GEOmetadb.sqlite.gz", type = "demo")
} else {
    demo_sqlfile <- "GEOmetadb.sqlite"
}
ls = geoConvert('GPL96', out_type=c("GSE", 'GSM'), sqlite_db_name=demo_sqlfile)
names(ls)
head(ls[[1]])

## Download the full GEOmetadb database:
## Not run: geometadbfile <- getSQLiteFile()
```

getBiocPlatformMap *Get mappings between GPL and Bioconductor microarry annotation packages*

Description

Query the gpl table and get GPL information of a given list of Bioconductor microarry annotation packages. Note currently the GEOmetadb does not contains all the mappings, but we are trying to construct a relative complete list.

Usage

```
getBiocPlatformMap(con, bioc='all')
```

Arguments

con	Connection to the GEOmetadb.sqlite database
bioc	Character vector of Biocondoctor microarry annotation packages, e.g. c('hgu133plus2','hgu95av2'). 'all' returns all mappings.

Value

A six-column data.frame including GPL title, GPL accession, bioc_package, manufacturer, organism, data_row_count.

Author(s)

Jack Zhu <zhujack@mail.nih.gov>, Sean Davis <sdavis2@mail.nih.gov>

Examples

```
## Use the demo GEOmetadb database:
if( !file.exists("GEOmetadb.sqlite") ) {
   demo_sqlfile <- getSQLiteFile(destdir = getwd(), destfile = "GEOmetadb.sqlite.gz", type = "demo")
} else {
    demo_sqlfile <- "GEOmetadb.sqlite"
}

con <- dbConnect(SQLite(), demo_sqlfile)
getBiocPlatformMap(con)[1:5,]
getBiocPlatformMap(con, bioc=c('hgu133a','hgu95av2'))
dbDisconnect(con)

## Download the full GEOmetadb database:
## Not run: geometadbfile <- getSQLiteFile()
```

getSQLiteFile *Download and unzip the most recent GEOmetadb SQLite file*

Description

This function is the standard method for downloading and unzipping the most recent GEOmetadb SQLite file from the server. Note: size of the full GEOmetadb.sqlite.gz could be over 10GB and the demo database is 25MB (use type="demo")

Usage

```
getSQLiteFile(destdir = getwd(), destfile = "GEOmetadb.sqlite.gz", type = "normal")
```

Arguments

destdir	The destination directory of the downloaded file
destfile	The filename of the downloaded file. This filename should end in ".gz" as the unzipping assumes that is the case
type	type of GEOmetadb.sqlite to download, if it is 'normal', a full database will be downloaded, otherwise a demo database will be downloaded, which is 25MB.

Value

Prints some diagnostic information to the screen.

Returns the local filename for use later.

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

Examples

```
## Download the demo GEOmetadb database:
if( !file.exists("GEOmetadb.sqlite") ) {
   demo_sqlfile <- getSQLiteFile(destdir = getwd(), destfile = "GEOmetadb.sqlite.gz", type = "demo")
} else {
    demo_sqlfile <- "GEOmetadb.sqlite"
}

## Download the full GEOmetadb database:
## Not run: geometadbfile <- getSQLiteFile()
```

Index

∗ **package**
GEOmetadb-package, 2

columnDescriptions, 3

geoConvert, 4
GEOmetadb *(*GEOmetadb-package*)*, 2
GEOmetadb-package, 2
getBiocPlatformMap, 5
getSQLiteFile, 6