Bioinformatics, 2016, 1–2
doi: 10.1093/bioinformatics/btw335
Advance Access Publication Date: 9 June 2016
Applications Note

OXFORD

Gene expression

compendiumdb: an R package for retrieval and storage of functional genomics data

Umesh K. Nandal[1], Antoine H. C. van Kampen[1,2] and Perry D. Moerland[1,*]

[1]Bioinformatics Laboratory, Department of Clinical Epidemiology, Biostatistics and Bioinformatics, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands and [2]Biosystems Data Analysis Group, University of Amsterdam, Amsterdam, The Netherlands

*To whom correspondence should be addressed.

Associate Editor: Janet Kelso

Received on May 18, 2015; revised on April 14, 2016; accepted on May 23, 2016

Abstract

Summary: Currently, the Gene Expression Omnibus (GEO) contains public data of over 1 million samples from more than 40 000 microarray-based functional genomics experiments. This provides a rich source of information for novel biological discoveries. However, unlocking this potential often requires retrieving and storing a large number of expression profiles from a wide range of different studies and platforms. The compendiumdb R package provides an environment for downloading functional genomics data from GEO, parsing the information into a local or remote database and interacting with the database using dedicated R functions, thus enabling seamless integration with other tools available in R/Bioconductor.

Availability and Implementation: The compendiumdb package is written in R, MySQL and Perl. Source code and binaries are available from CRAN (http://cran.r-project.org/web/packages/compendiumdb/) for all major platforms (Linux, MS Windows and OS X) under the GPLv3 license.

Contact: p.d.moerland@amc.uva.nl

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Public repositories such as the Gene Expression Omnibus (GEO) (Barrett *et al.*, 2013) and ArrayExpress (Rustici *et al.*, 2013) provide a large amount of functional genomics data from a wide range of studies performed in different organisms and on different (microarray) platforms. However, retrieving and systematically storing these datasets to extract novel biological information is often challenging. Several tools and web-based resources have been developed (Bareke *et al.*, 2010; Cheng *et al.*, 2010; Coletta *et al.*, 2012; Lacson *et al.*, 2010; Liu *et al.*, 2011; Petryszak *et al.*, 2014; Planey and Butte, 2013) to facilitate the aggregation of data from functional genomics data repositories. However, often tools are limited to only specific profiling platforms (Coletta *et al.*, 2012; Liu *et al.*, 2011) or a subset of experiments in a specific organism (Cheng *et al.*, 2010). Moreover, most of these resources do not enable easy integration with the rich collection of R/Bioconductor packages available for subsequent analyses. One solution is offered by the GEOquery package (Davis and Meltzer, 2007) providing a bridge between GEO and R. GEOquery enables downloading GEO records and storing expression data as an R/Bioconductor ExpressionSet. However, GEOquery does not provide functionalities for systematically maintaining a large collection of ExpressionSets, which therefore remains a challenge.

We developed the R package compendiumdb to provide a homogeneous framework for constructing large microarray compendia by retrieving preprocessed GEO data from different studies and profiling platforms, and storing and maintaining these using a MySQL database (Fig. 1). The compendiumdb package consists of a number of R functions to access this database either locally or remotely. The database schema has been designed to be rich enough to store information provided by MIAME-compliant expression databases such as GEO. The package provides R functions to (*i*) download data from GEO given the identifier of the experiment, (*ii*) load the

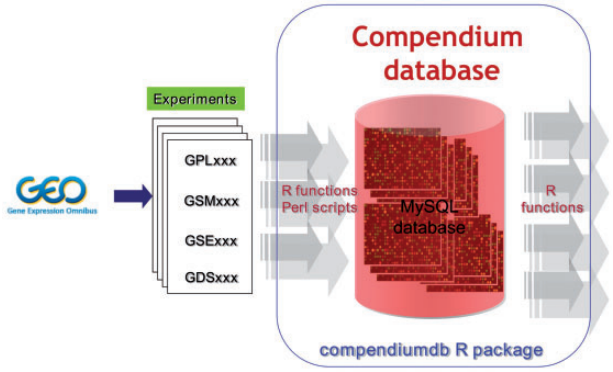

Fig. 1. Workflow of the compendiumdb package: data records are downloaded from GEO and stored in the MySQL compendium database. Data can be extracted from the database using R functions included in the package. Some of the R functions provide convenient wrappers around underlying Perl code used for parsing data and interaction with the database

expression data, sample and probe annotation to the relational database and (*iii*) convert experimental data from the database to an R/Bioconductor ExpressionSet.

2 Description

The compendiumdb package has been developed around a MySQL database designed for storing data from GEO. The database schema is provided with compendiumdb and is described in detail at http://wiki.bioinformaticslaboratory.nl/foswiki/bin/view/BioLab/CompendiumDB. After creating an empty database, one connects to it and loads the database schema:

```
conn<-connectDatabase(user="user",
password="passwd", dbname="compendium")
loadDatabaseSchema(conn,updateSchema = TRUE)
```

By default, this establishes a connection to a database running on a local machine but using the argument `host` of `connectDatabase` one can also connect to a database on a remote server. This way the database can be conveniently deployed in a multi-user environment.

Functional genomics datasets in the form of Simple Omnibus Format in Text (SOFT) files can be downloaded from GEO by specifying the GEO series record (GSE) identifier. The downloaded preprocessed expression data, sample and probe annotation can then be loaded into the database:

```
downloadGEOdata(GSEid="GSE18290")
loadDataToCompendium(conn,GSEid="GSE18290")
```

Probe annotation is automatically retrieved from GEO platform records (GPL) and updated to the most recent annotation for those platforms listed on http://ftp://ftp.ncbi.nlm.nih.gov/pub/geo/DATA/annotation/. Sample annotation is retrieved from the information provided in each GEO sample record (GSM). For those experiments that have been curated by GEO staff into a GEO dataset (GDS), the sample annotation is retrieved automatically from the GDS. Sample annotation as stored in the compendium database can be further curated and updated using the function `updatePhenoData`.

Experimental data stored in the database can be extracted as an R/Bioconductor ExpressionSet enabling straightforward integration with other tools available in R/Bioconductor:

```
esets<- createESET(conn,"GSE18290")
```

The function `createESET` conveniently parses the metadata provided for each sample into separate columns for each of the variables and stores them in the phenoData slot of the ExpressionSet, facilitating down-stream analysis. Using these functions, downloading 39 GSEs from GEO and loading the corresponding 7970 samples in the compendium database took 5.5 h on a single core of a Linux (Red Hat 4.4.7-9, 64-bit) server containing 10 Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz processors having 64 GB of random access memory. Subsequent extraction as ExpressionSets took 0.3 h. For each GSE, a detailed breakdown of the time to download, load and extract data is given in Supplementary File S1.

Next to the main functions described earlier, one can also tag experiments using keywords (`tagExperiment`), enabling easy extraction of a set of related experiments from the database, and extract a succinct overview of the experiments contained in the database (`GSEinDB`). Note that while the package conveniently shields the user from having to write SQL queries, the database can also be queried directly to extract information that is not easily accessible via the functions provided with the package. Further information on how to install compendiumdb, other functionalities and more detailed examples are provided in the package vignette (Supplementary File S2).

3 Conclusions

The package compendiumdb provides a flexible solution for maintaining a database containing a large compendium of microarray-based functional genomics experiments. It can be seamlessly integrated with other R/Bioconductor packages available for follow-up analyses. Since any array-based experiment from GEO can be stored in the database, it provides a useful resource towards an integrative analysis of multiple experiments across platforms, species or measurement modalities.

Funding

This work was supported by the BioRange program of the Netherlands Bioinformatics Centre (NBIC) as part of the Netherlands Genomics Initiative (NGI).

Conflict of Interest: none declared.

References

Bareke,E. *et al.* (2010) PathEx: a novel multi factors based datasets selector web tool. *BMC Bioinformatics*, **11**, 528.

Barrett,T. *et al.* (2013) NCBI GEO: archive for functional genomics data sets – update. *Nucleic Acids Res.*, **41**, D991–D995.

Cheng,W. *et al.* (2010) Microarray meta-analysis database (M2DB): a uniformly pre-processed, quality controlled, and manually curated human clinical microarray database. *BMC Bioinformatics*, **11**, 421.

Coletta,A. *et al.* (2012) InSilico DB genomic datasets hub: an efficient starting point for analyzing genome-wide studies in GenePattern, Integrative Genomics Viewer, and R/Bioconductor. *Genome Biol.*, **13**, R104.

Davis,S. and Meltzer,P.S. (2007) GEOquery: a bridge between the gene expression omnibus (GEO) and BioConductor. *Bioinformatics*, **23**, 1846–1847.

Lacson,R. *et al.* (2010) DSGeo: software tools for cross-platform analysis of gene expression data in GEO. *J. Biomed. Inform.*, **43**, 709–715.

Liu,F. *et al.* (2011) GCOD-GeneChip oncology database. *BMC Bioinformatics*, **12**, 46.

Petryszak,R. *et al.* (2014) Expression Atlas update – a database of gene and transcript expression from microarray-and sequencing-based functional genomics experiments. *Nucleic Acids Res.*, **42**, D926–D932.

Planey,C.R. and Butte,A.J. (2013) Database integration of 4923 publicly-available samples of breast cancer molecular and clinical data. *AMIA Summits Transl. Sci. Proc.*, **2013**, 138–142.

Rustici,G. *et al.* (2013) ArrayExpress update – trends in database growth and links to data analysis tools. *Nucleic Acids Res.*, **41**, D987–D990.