compendiumdb: Tools for Retrieval and Storage of Functional Genomics Data

Umesh K. Nandal and Perry D. Moerland

October 9, 2015

Contents

1 Introduction **1**

2 Installation of compendiumdb package **2**

3 Commonly used functions **4**
3.1 Connecting to and creating the compendium database 4
3.2 Downloading data from GEO 4
3.3 Loading data into the compendium database 5
3.4 Creating ExpressionSets 6

4 Sample annotation **7**
4.1 Using *createESET* 7
4.2 Using the inSilicoDb package 8
4.3 Using GEO Datasets 10

5 Querying the compendium database **11**

6 Use case: building a small tissue-resident memory T cell compendium **12**
6.1 Gene set enrichment analysis 14

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, collecting and maintaining datasets for a specific domain of study to extract meaningful biological information from these repositories is often challenging. Several tools and web-based resources have been developed (Bareke et al. 2010; Cheng et al. 2010; Coletta et al. 2012; Kilpinen et al. 2008; Lacson et al. 2010; Liu et al. 2011; Petryszak et al. 2014; Planey and Butte 2013; Taminau et al. 2011; Xia et al. 2009) to facilitate the aggregation of data from functional genomics data repositories. While very useful, most of these resources do not enable easy integration with the rich collection of R/Bioconductor packages available for

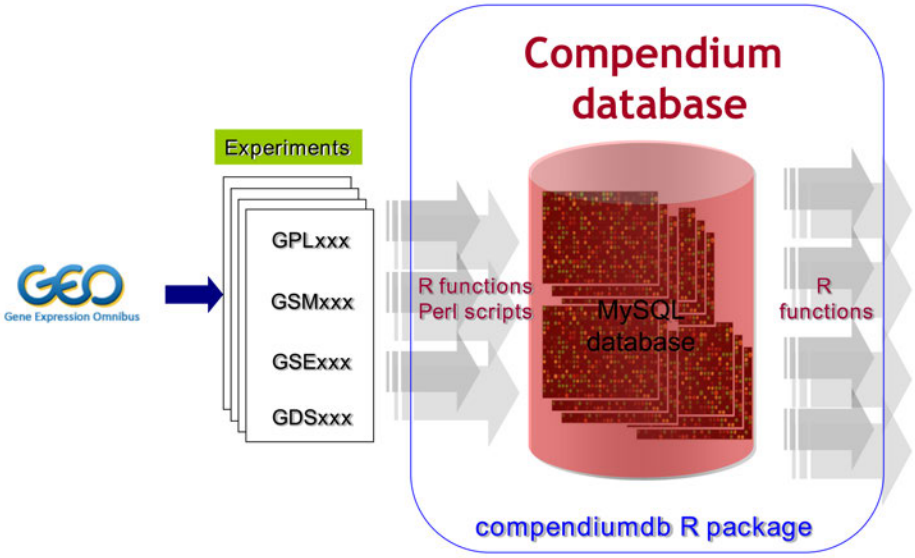

Figure 1: Workflow of the compendiumdb package: GEO records are downloaded from GEO and stored in a MySQL database. Data can be extracted from the compendium database using R functions included in the package.

follow-up analyses. A partial solution is offered by the GEOquery package (Davis and Meltzer 2007) that provides a bridge between GEO and R which enables downloading GEO records and storing expression data as an *ExpressionSet*. However, when using GEOquery systematically maintaining a large collection of *ExpressionSets* remains problematic.

We developed the R package compendiumdb to provide a homogeneous framework to retrieve and store a large number of functional genomics datasets from different studies and profiling platforms using a MySQL database (Figure 1). The compendiumdb package consists of a number of R functions developed to access the database either locally or remotely. The database schema has been designed to be rich enough to store most of the 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 expression data and probe annotation to a relational database, and (*iii*) retrieve the expression data from the compendium database as an *ExpressionSet*. These and addtional functions are described in the rest of this vignette.

2 Installation of compendiumdb package

Before installing compendiumdb itself recent distributions of MySQL and Perl (and some of its modules) have to be installed. Below the steps to take when using a version of the Windows operating system are outlined, but a similar sequence of steps also works for Linux or MacOSX.

1. Install the most recent version of MySQL:

(a) Download the MySQL Installer from `http://dev.mysql.com/downloads/installer/`;

(b) Open the MySQL Installer by clicking on the MSI (MicroSoft Installer) file you just downloaded;

(c) Use the default settings. For a minimal installaton choose 'Server only' under Setup Type;

(d) Create a root account by entering a password under Configuration (Accounts and Roles).

2. Add the path to your MySQL bin directory (e.g., `C:\Program Files\MySQL\MySQL Server 5.6\bin`) to the PATH environment variable (see `http://dev.mysql.com/doc/mysql-windows-excerpt/5.6/en/mysql-installation-windows-path.html`).

3. (a) Open a Command Prompt window and log in to your MySQL account by typing `mysql -u root -p` and entering your password for the root account;

   (b) On the MySQL prompt create a database named `compendium` using `CREATE DATABASE compendium;`. You are free to choose another database name, but `compendium` is the default name assumed by compendiumdb and will therefore save some typing when using the package.

4. Install a recent version of Perl:

   (a) Go to ActiveState's ActivePerl home page `http://www.activestate.com/activeperl`;

   (b) Click on 'Download Now' to download the installer for ActivePerl for Windows. There is no need to fill out any of the contact information on the next page in order to download ActivePerl;

   (c) Install ActivePerl by clicking on the MSI file you just downloaded and accepting the default options (in particular, the option to add Perl to the PATH environment variable);

   (d) Go to the Command Prompt and type `ppm`. This will open the Perl Package Manager (or open it via the Start Menu). Check if the following Perl required modules are already installed: `DateTime-Format-DateManip, DBD-mysql`. If not, install them;

   (e) (Optional) See the INSTALL file provided with the package for some operating system specific recommendations on installing Perl and required modules.

5. Install the most recent version of compendiumdb from CRAN:

```
install.packages("compendiumdb")
```

   Required packages from Bioconductor (Biobase, GEOquery) are automatically installed when also selecting the Bioconductor repository via *setRepositories()*.

6. Add the path to your R installation (e.g., `C:\Program Files\R\R-3.2.2\bin`) to the PATH environment variable and start a new R session.

3 Commonly used functions

3.1 Connecting to and creating the compendium database

We start by loading compendiumdb in the current R session:

```
library(compendiumdb)
```

One first has to connect to the MySQL database using the function *connectDatabase*. Before calling this function, a MySQL server should be running on the host machine and an (empty) database has to be created on the MySQL server (see Section 2, Step 3). We assume that the MySQL database is called `compendium`:

```
# Replace with your username and password
conn <- connectDatabase(user = "root", password = "root", host = "localhost",
    dbname = "compendium")
```

Here we connected to a database running on a local machine, but the *host* argument can also be used to connect to a database on a remote server. Once the connection to the database has been established, load the database schema of the MySQL compendium database using the function *loadDatabaseSchema* (default value *updateSchema=FALSE*):

```
loadDatabaseSchema(conn, updateSchema = TRUE)
```

Note that in general one should set *updateSchema=TRUE* only once, i.e., before filling the database with expression data, or if one explicitly wants to delete all the records of the database and reload the schema.

3.2 Downloading data from GEO

First one has to download the expression datasets of interest from GEO (`http://www.ncbi.nlm.nih.gov/geo/`). For this purpose, the package provides the function *downloadGEOdata*. GEO contains the following types of records (see also `http://www.ncbi.nlm.nih.gov/geo/info/overview.html`):

- Platform record (GPL): describes properties of the microarray, e.g., cDNA or oligonucleotide probesets, or sequencing machine. Each platform has a unique identifier (GPLxxx);
- Sample record (GSM): describes the conditions under which an individual Sample in the experiment was handled, the manipulations it underwent, and the abundance measurement (for example, level of fluorescence) of each element derived from it. It refers to only one platform and can be part of multiple series. Each sample record is assigned a unique identifier (GSMxxx);
- Series record (GSE): links a number of related samples together and provides a description of the whole study, obtained data, analysis and conclusions. Each series has a unique identifer (GSExxx);

- Dataset record (GDS): GEO samples that are processed using the same platform and are biologically and statistically comparable are reassembled by GEO staff into GEO Datasets (GDSxxx). In assembling a GDS, special attention is given to the sample annotation and information about the different experimental factors is provided through DataSet subsets.

The function *downloadGEOdata* downloads SOFT (Simple Omnibus Format in Text) files from GEO to the user's local machine for the GSE, GPLs, GSMs, and GDSs corresponding to the GSE identifier provided by the user. Here we download the series record GSE18290 and its associated GPLs and GSMs from GEO. GSE18290 contains time course expression data from early bovine, human, and mouse embryos (Xie et al. 2010):

```
downloadGEOdata(GSEid = "GSE18290", destdir = getwd())
```

The function *downloadGEOdata* creates a data directory called `BigMac` (BIoinformatics Group MicroArray Compendium) 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 the GSE, 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`.

3.3 Loading data into the compendium database

Data downloaded from GEO to the `BigMac` directory can be loaded in the compendium database using the function *loadDataToCompendium*:

```
loadDataToCompendium(conn, "GSE18290")
```

The current contents of the compendium database can be inspected using the function *GSEinDB*:

```
GSEinDB(conn)

##   id_Compendium Experiment experimentDesign    Chip Samples
## 1             1   GSE18290               SC GPL2112      16
## 2             1   GSE18290               SC  GPL339      18
## 3             1   GSE18290               SC  GPL570      18
##    Tag OrganismNCBIid OrganismName          DateLoaded
## 1 <NA>           9913   Bos taurus 2015-10-09 10:03:18
## 2 <NA>          10090 Mus musculus 2015-10-09 10:03:18
## 3 <NA>           9606 Homo sapiens 2015-10-09 10:03:18
##       GDS
## 1 GDS3960
## 2 GDS3958
## 3 GDS3959
```

Since a different platform was used for each of the three species assayed in GSE18290 the table contains three entries, one for each species.

3.4 Creating ExpressionSets

Once a dataset has been loaded into the compendium database, one would generally like to further analyze the dataset using other packages provided by R/Bioconductor. For this purpose the package provides the function *createESET* that creates an *ExpressionSet* given a GSE identifier:

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

## Creating ExpressionSet for 3 GPL(s): GPL2112,GPL570,GPL339
```

Since in GSE18290 a different platform was used for each of the three species, three different *ExpressionSets* were created and assembled in a list `esets`. The *ExpressionSet* for the bovine microarrays (GPL2112) can be accessed as follows:

```
esets$esetGSE18290_GPL2112

## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 24128 features, 16 samples
##   element names: exprs
## protocolData: none
## phenoData
##   rowNames: GSM456627 GSM456628 ... GSM456642 (16
##     total)
##   varLabels: development stage GPL
##   varMetadata: labelDescription
## featureData
##   featureNames: AFFX-BioB-5_at AFFX-BioB-M_at ...
##     Bt.19900.1.A1_at (24128 total)
##   fvarLabels: ID Gene title ... GenBank Accession (10
##     total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: GPL2112
```

The numerical data contained in the *assayData* slot is identical to the normalized expression data provided by GEO. The *featureData* slot is based upon the most recent (probe) annotation provided by GEO for those platforms listed on `ftp://ftp.ncbi.nlm.nih.gov/pub/geo/DATA/annotation/` (in general those platforms with a GDS). For the other platforms, the annotation is provided by the submitter of the GPL record.

4 Sample annotation

It is a well-known problem that the annotation of individual samples in public expression data repositories is often inconsistent or even non-existent (Pitzer et al. 2009). The compendiumdb package offers various ways to obtain a more consistent sample annotation.

4.1 Using *createESET*

As an example, consider GSE35547 containing gene expression data on the role of Notch in CD4+ T cell response (Helbig et al. 2012):

```
downloadGEOdata("GSE35547")
loadDataToCompendium(conn, "GSE35547")
```

The function *GSMdescriptions* provides a convenient overview of the sample title, sample characteristics, and sample source fields provided by GEO and stored in the compendium database for each sample.

```
head(GSMdescriptions(conn, "GSE35547"), n = 4)

##           sampletitle
## GSM870390 "C-Ig day1_mouse1"
## GSM870391 "C-Ig_day3_mouse1"
## GSM870392 "DLL4_day1_mouse1"
## GSM870393 "C-Ig_day3_mouse1 (technical replicate)"
##           samplesource
## GSM870390 "naive CD4+ T cells, control, day 1"
## GSM870391 "naive CD4+ T cells, control, day 3"
## GSM870392 "naive CD4+ T cells, Delta4-Ig, day 1"
## GSM870393 "naive CD4+ T cells, control, day 3"
##           samplechar
## GSM870390 "strain: C57BL6/NCrl;;tissue: inguinal, axillary and brachial lymph nodes and sp
## GSM870391 "strain: C57BL6/NCrl;;tissue: inguinal, axillary and brachial lymph nodes and sp
## GSM870392 "strain: C57BL6/NCrl;;tissue: inguinal, axillary and brachial lymph nodes and sp
## GSM870393 "strain: C57BL6/NCrl;;tissue: inguinal, axillary and brachial lymph nodes and sp
##           GPL
## GSM870390 "GPL6885"
## GSM870391 "GPL6885"
## GSM870392 "GPL6885"
## GSM870393 "GPL6885"
```

According to GEO guidelines (see `http://www.ncbi.nlm.nih.gov/geo/info/spreadsheet.html#samples_tab`) the sample characteristics field should contain detailed sample annotation. For GSE35547 this is indeed the case, and for each sample the variables strain, tissue, cell type, stimulus, timepoint, and mouse are defined. In the output of *GSMdescriptions* these variables are separated by a semicolon. The function *createESET* with its argument *parsing* set to *TRUE* enables splitting the sample characteristics into separate columns for each of the variables:

```
esets <- createESET(conn, "GSE35547", parsing = TRUE)

## Creating ExpressionSet for 1 GPL(s): GPL6885

head(pData(esets$esetGSE35547_GPL6885), n = 4)

##                strain
## GSM870390 C57BL6/NCrl
## GSM870391 C57BL6/NCrl
## GSM870392 C57BL6/NCrl
## GSM870393 C57BL6/NCrl
##                                                              tissue
## GSM870390 inguinal, axillary and brachial lymph nodes and spleen
## GSM870391 inguinal, axillary and brachial lymph nodes and spleen
## GSM870392 inguinal, axillary and brachial lymph nodes and spleen
## GSM870393 inguinal, axillary and brachial lymph nodes and spleen
##                    cell_type   stimulus timepoint mouse
## GSM870390 naive CD4+ T cells control Ig     day 1     1
## GSM870391 naive CD4+ T cells control Ig     day 3     1
## GSM870392 naive CD4+ T cells  Delta4-Ig     day 1     1
## GSM870393 naive CD4+ T cells control Ig     day 3     1
##                                     sampletitle
## GSM870390                        C-Ig day1_mouse1
## GSM870391                        C-Ig_day3_mouse1
## GSM870392                        DLL4_day1_mouse1
## GSM870393 C-Ig_day3_mouse1 (technical replicate)
##                                    samplesource     GPL
## GSM870390   naive CD4+ T cells, control, day 1 GPL6885
## GSM870391   naive CD4+ T cells, control, day 3 GPL6885
## GSM870392 naive CD4+ T cells, Delta4-Ig, day 1 GPL6885
## GSM870393   naive CD4+ T cells, control, day 3 GPL6885
```

With the resulting *ExpressionSet* it is straightforward to perform follow-up analyses, for example, testing for differential expression using limma.

4.2 Using the inSilicoDb package

Often data uploaded to GEO does not conform with the guidelines. As an example, consider GSE18931 containing gene expression data in human normal mammary stem cells (Pece et al. 2010):

```
downloadGEOdata("GSE18931")
loadDataToCompendium(conn, "GSE18931")
GSMdescriptions(conn, "GSE18931")

##           sampletitle
```

```
## GSM468802 "Mammary epithelial cells, FACS sorted, PKH-Negative, Pool 1"
## GSM468803 "Mammary epithelial cells, FACS sorted, PKH-Positive, Pool 1"
## GSM468804 "Mammary epithelial cells, FACS sorted, PKH-Negative, Pool 2"
## GSM468805 "Mammary epithelial cells, FACS sorted, PKH-Positive, Pool 2"
## GSM468806 "Mammary epithelial cells, FACS sorted, PKH-Negative, Pool 3"
## GSM468807 "Mammary epithelial cells, FACS sorted, PKH-Positive, Pool 3"
##           samplesource
## GSM468802 "Epithelial cells, from dissociated mammospheres obtained from reductive mammopl
## GSM468803 "Epithelial cells, from dissociated mammospheres obtained from reductive mammopl
## GSM468804 "Epithelial cells, from dissociated mammospheres obtained from reductive mammopl
## GSM468805 "Epithelial cells, from dissociated mammospheres obtained from reductive mammopl
## GSM468806 "Epithelial cells, from dissociated mammospheres obtained from reductive mammopl
## GSM468807 "Epithelial cells, from dissociated mammospheres obtained from reductive mammopl
##           samplechar                                GPL
## GSM468802 "cell type: Mammary Epithelial Cells;;" "GPL570"
## GSM468803 "cell type: Mammary Epithelial Cells;;" "GPL570"
## GSM468804 "cell type: Mammary Epithelial Cells;;" "GPL570"
## GSM468805 "cell type: Mammary Epithelial Cells;;" "GPL570"
## GSM468806 "cell type: Mammary Epithelial Cells;;" "GPL570"
## GSM468807 "cell type: Mammary Epithelial Cells;;" "GPL570"
```

Here the essential information of a sample being either PKH positive or PKH negative is not provided in the sample characteristics field but in the sample title field. In this case, one can use curated sample annotation accessible via the package inSilicoDB (Taminau et al. 2011). This package is a command-line front-end to the InSilico DB (`https://insilicodb.com`, accessed March 16, 2015), a web-based database containing close to 250,000 expert-curated Affymetrix and Illumina expression profiles compiled from almost 8,000 GEO series records in human, mouse, and rat (Coletta et al. 2012).

```
library(inSilicoDb)
sample.annot <- getAnnotations(dataset = "GSE18931", platform = "GPL570")
pdata <- pData(sample.annot)
head(pdata, n = 4)

##                          Cell Type  PKH26 Label
## GSM468802 mammary epithelial cells PKH-negative
## GSM468803 mammary epithelial cells PKH-positive
## GSM468804 mammary epithelial cells PKH-negative
## GSM468805 mammary epithelial cells PKH-positive
```

The current annotation of all samples can then easily be updated using the function *updatePhenoData*:

```
updatePhenoData(conn, "GSE18931", data = as.matrix(pdata))
```

```
head(GSMdescriptions(conn, "GSE18931"), n = 4)

##           PKH26 Label    Cell Type
## GSM468807 "PKH-positive" "mammary epithelial cells"
## GSM468806 "PKH-negative" "mammary epithelial cells"
## GSM468805 "PKH-positive" "mammary epithelial cells"
## GSM468804 "PKH-negative" "mammary epithelial cells"
##           GPL
## GSM468807 "GPL570"
## GSM468806 "GPL570"
## GSM468805 "GPL570"
## GSM468804 "GPL570"
```

Here the sample annotation was imported from InSilico DB. Of course, a user can also create a character matrix with curated sample annotation, for example with *fix*, and use *updatePhenoData* to store the updated sample annotation in the compendium database.

4.3 Using GEO Datasets

When downloading a GSE, the function *downloadGEOData* checks whether the GSE has been curated by GEO staff and been made available as a GDS. If this is the case, the corresponding GDS is also downloaded and when loading the data to the compendium database the curated sample annotation provided by the GDS is stored. For GSE18290 loaded above this is indeed the case:

```
GDSforGSE(conn, "GSE18290")

##   id_Compendium Experiment experimentDesign    Chip Samples
## 1             1   GSE18290               SC GPL2112      16
## 2             1   GSE18290               SC  GPL339      18
## 3             1   GSE18290               SC  GPL570      18
##    Tag OrganismNCBIid OrganismName          DateLoaded
## 1 <NA>           9913   Bos taurus 2015-10-09 10:03:18
## 2 <NA>          10090 Mus musculus 2015-10-09 10:03:18
## 3 <NA>           9606 Homo sapiens 2015-10-09 10:03:18
##       GDS
## 1 GDS3960
## 2 GDS3958
## 3 GDS3959
```

GSE18290 has actually been split into three different GDSs, one for each species. If a GDS is available, the individual samples are in general well annotated:

```
head(GSMdescriptions(conn, "GSE18290"))

##           development stage GPL
## GSM456627 "oocyte"          "GPL2112"
## GSM456628 "oocyte"          "GPL2112"
## GSM456629 "1-cell embryo"   "GPL2112"
## GSM456630 "1-cell embryo"   "GPL2112"
## GSM456631 "2-cell embryo"   "GPL2112"
## GSM456632 "2-cell embryo"   "GPL2112"
```

5 Querying the compendium database

Of course, the MySQL compendium database provided with the package can also be queried directly (see `http://wiki.bioinformaticslaboratory.nl/foswiki/bin/view/BioLab/CompendiumDB` for the entity relationship schema of the MySQL database). As a first example, we use a database query to retrieve all GPL IDs and related information corresponding to a given platform provider (Affymetrix in this case) loaded in the compendium database:

```
query <- paste0("select db_platform_id, idorganism, provider,distribution from ",
    "chip where provider = 'Affymetrix'")
res <- dbSendQuery(conn$connect, query)
fetch(res, n = -1)

##   db_platform_id idorganism   provider distribution
## 1        GPL2112        563 Affymetrix   commercial
## 2         GPL570        549 Affymetrix   commercial
## 3         GPL339        576 Affymetrix   commercial
```

Next we limit the query to retrieve all Affymetrix-based GPL IDs corresponding to "Homo sapiens":

```
query <- paste0("select db_platform_id, organism.officialname, provider from ",
    "chip JOIN organism ON chip.idorganism=organism.idorganism WHERE ",
    "chip.provider like 'Affy%' and organism.officialname like 'Homo%'")
res <- dbSendQuery(conn$connect, query)
fetch(res, n = -1)

##   db_platform_id officialname   provider
## 1         GPL570 Homo sapiens Affymetrix
```

Similarly we can also retrieve all Illumina-based GSE IDs and related information from the compendium database:

```
query <- paste0("select distinct db_platform_id, title, experiment.expname, ",
                "provider from chip JOIN expressionset ON ",
                "chip.idchip=expressionset.idchip JOIN experiment ON ",
                "expressionset.idExperiment=experiment.idExperiment WHERE ",
                "chip.provider like 'Illumina%'")
res <- dbSendQuery(conn$connect, query)
fetch (res, n= -1)

##   db_platform_id
## 1        GPL6885
##                                          title  expname
## 1 Illumina MouseRef-8 v2.0 expression beadchip GSE35547
##        provider
## 1 Illumina Inc.
```

6 Use case: building a small tissue-resident memory T cell compendium

The compendiumdb package provides a convenient framework to store and analyze a large number of expression datasets from a specific domain of study. Here we create a small tissue-resident memory CD8 T cell compendium.

We first use the GEOmetadb package to find microarray-based studies of tissue-resident memory CD8 T cell gene expression in mouse:

```
library(GEOmetadb)
if(!file.exists('GEOmetadb.sqlite')) getSQLiteFile()
con.geometadb <- dbConnect(SQLite(),'GEOmetadb.sqlite')
sql <- paste("SELECT DISTINCT gse.title,gse.gse",
             "FROM",
             "  gsm JOIN gse_gsm ON gsm.gsm=gse_gsm.gsm",
             "  JOIN gse ON gse_gsm.gse=gse.gse",
             "  JOIN gse_gpl ON gse_gpl.gse=gse.gse",
             "  JOIN gpl ON gse_gpl.gpl=gpl.gpl",
             "WHERE",
             "  gse.type like '%Expression profiling by array%' AND",
             "  gsm.molecule_ch1 like '%total RNA%' AND",
             "  gse.title LIKE '%resident memory% %CD8+ T cells%' AND",
             "  gpl.organism LIKE '%Mus musculus%'",sep=" ")
rs <- dbGetQuery(con.geometadb,sql)
rs

##                                                          title
## 1       Molecular signature of brain resident memory CD8+ T cells
```

```
## 2 Development pathway for skin resident memory CD103+CD8+ T cells
##        gse
## 1 GSE39152
## 2 GSE47045
```

The listed GSE records were measured using Affymetrix Mouse Gene 1.0 ST Arrays, i.e., the GPL6246 platform. First download the selected datasets from GEO using their GSE identifiers:

```
## GSE39152: Wakim et al., GSE47045: Mackay et al.
gseIDs <- rs$gse
for (i in gseIDs) {
    downloadGEOdata(i)
}
```

Then load the data to the compendium database using *loadDataToCompendium*:

```
for (i in gseIDs) {
    loadDataToCompendium(con = conn, GSEid = i)
}
```

The datasets loaded to the compendium database can be tagged with a specific label such as "tissue resident" using *tagExperiment*:

```
for (i in gseIDs) {
    tagExperiment(con = conn, GSEid = i, tag = "tissue resident")
}

## GSE record GSE39152 has been tagged
## GSE record GSE47045 has been tagged

GSEinDB(conn, gseIDs)

##   id_Compendium Experiment experimentDesign    Chip Samples
## 1             4   GSE39152               SC GPL6246      13
## 2             5   GSE47045               SC GPL6246      24
##               Tag OrganismNCBIid OrganismName
## 1 tissue resident          10090 Mus musculus
## 2 tissue resident          10090 Mus musculus
##            DateLoaded  GDS
## 1 2015-10-09 10:10:42 <NA>
## 2 2015-10-09 10:12:42 <NA>
```

Such a tag can, for example, be used to retrieve datasets of the user's interest from the compendium database. *ExpressionSet*s can be created using the *createESET* function:

```
tab <- GSEinDB(conn)
compendiumDatasets <- tab[which(tab$Tag == "tissue resident"),
    ]
esets <- list()
for (i in compendiumDatasets$Experiment) {
    esets[[i]] <- createESET(con = conn, GSEid = i, GPLid = "GPL6246")[[1]]
    annotation(esets[[i]]) <- "mogene10sttranscriptcluster"
}
```

6.1 Gene set enrichment analysis

To better understand transcriptional differences between tissue-resident memory T cells (T_{RM} cells) in various tissues and circulating T cells, we perform a gene set enrichment analysis.

The *ExpressionSets* generated earlier contain expression data at the probeset level. In order to perform an unbiased enrichment analysis, one has to select a single probe if multiple probes map to the same gene. This can be achieved by using the *nsFilter* from the genefilter package:

```
library(genefilter)
for (i in 1:length(esets)) {
    esets[[i]] <- nsFilter(esets[[i]])$eset
}
```

From GSE39152 (Wakim et al. 2012) we select the brain $CD103^+$ T_{RM} samples and the spleen $CD103^-$ memory samples; from GSE47045 (Mackay et al. 2013) we select the gut, lung, and skin $CD103^+$ T_{RM} samples and the spleen memory samples:

```
trmsets <- list()
trmsets[["Brain"]] <- esets[["GSE39152"]][, -which(pData(esets[["GSE39152"]])$tissue ==
    "Brain" & pData(esets[["GSE39152"]])$cell_type == "CD103-")]
indx <- pData(esets[["GSE47045"]])$cell_type == "memory CD62L high CD8+ T cells" |
    pData(esets[["GSE47045"]])$tissue == "Gut"
trmsets[["Gut"]] <- esets[["GSE47045"]][, indx]
indx <- pData(esets[["GSE47045"]])$cell_type == "memory CD62L high CD8+ T cells" |
    pData(esets[["GSE47045"]])$tissue == "Lung"
trmsets[["Lung"]] <- esets[["GSE47045"]][, indx]
indx <- pData(esets[["GSE47045"]])$cell_type == "memory CD62L high CD8+ T cells" |
    (pData(esets[["GSE47045"]])$tissue == "Skin" & pData(esets[["GSE47045"]])$cell_type ==
        "memory gB-T CD8+. CD103+ T cells")
trmsets[["Skin"]] <- esets[["GSE47045"]][, indx]
```

We then use the GSVA package to transform gene by sample matrices into gene set by sample matrices. Gene sets are based on a mouse version of the C2 collection of the Molecular Signatures Database (MSigDB, `http://www.broadinstitute.org/gsea/msigdb/`) from which we select only those derived from the BioCarta pathway database:

```
download.file("http://bioinf.wehi.edu.au/software/MSigDB/mouse_c2_v4.rdata",
    destfile = "mouse_c2_v4.rdata", mode = "wb")
load("mouse_c2_v4.rdata")
Mm.c2.Biocarta <- Mm.c2[grep("BIOCARTA", names(Mm.c2))]

library(GSVA)
library(mogene10sttranscriptcluster.db)
eset_es = list()
for (i in names(trmsets)) {
    featureNames(trmsets[[i]]) <- select(mogene10sttranscriptcluster.db,
        featureNames(trmsets[[i]]), columns = "ENTREZID")$ENTREZID
    eset_es[[i]] <- gsva(exprs(trmsets[[i]]), Mm.c2.Biocarta,
        min.sz = 10, max.sz = 500, verbose = FALSE)$es.obs
}
```

Now perform a differential expression analysis (T_{RM} cells versus spleen T cells) at the gene set level on the common pathways:

```
indx <- intersect(rownames(eset_es[[1]]), rownames(eset_es[[2]]))
for (i in names(eset_es)) {
    eset_es[[i]] <- eset_es[[i]][indx, ]
}

library(limma)
tstats <- c()
DEgeneSets <- list()
for (i in names(eset_es)) {
    tissue <- factor(pData(trmsets[[i]])$tissue)
    tissue <- relevel(tissue, "Spleen")
    design <- model.matrix(~tissue)
    fit <- lmFit(eset_es[[i]], design)
    tstats <- cbind(tstats, eBayes(fit)$t[, 2])
    DEgeneSets[[i]] <- topTable(eBayes(fit), coef = 2, n = Inf,
        sort = "none")
}
```

A heatmap of the moderated t-statistics of the differentially expressed gene sets illustrates that the molecular signature of tissue resident memory T cells isolated from lung is largely conserved in the other tissues, but less pronounced (Figure 2).

```
library(gplots)
library(RColorBrewer)
adjPvalueCutoff <- 0.025
colors <- colorRampPalette(c("blue", "white", "red"))(255)
indx <- union(rownames(subset(DEgeneSets[["Brain"]], adj.P.Val <
```

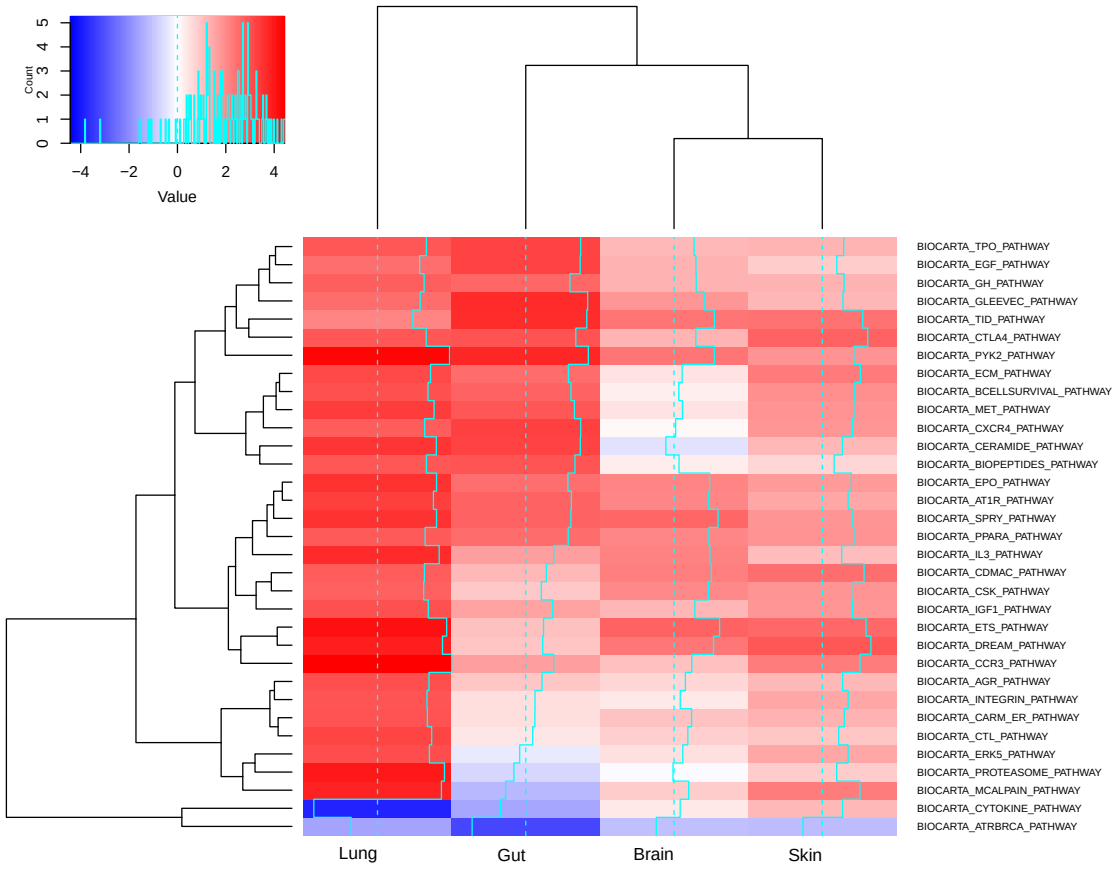

Figure 2: Heatmap of the moderated t-statistics of gene sets differentially expressed (Benjamini-Hochberg adjusted p-value < 0.025) between T_{RM} cells and spleen T cells.

```
    adjPvalueCutoff)), union(rownames(subset(DEgeneSets[["Gut"]],
    adj.P.Val < adjPvalueCutoff)), union(rownames(subset(DEgeneSets[["Lung"]],
    adj.P.Val < adjPvalueCutoff)), rownames(subset(DEgeneSets[["Skin"]],
    adj.P.Val < adjPvalueCutoff)))))
heatmap.2(tstats[indx, ], labCol = c("Brain", "Gut", "Lung",
    "Skin"), cexCol = 1, cexRow = 0.6, scale = "none", margins = c(2,
    11), srtCol = 0, col = colors, key.title = NA)

## NULL
```

Acknowledgements

Many people contributed to (precursors of) compendiumdb. In particular, we would like to thank Raymond Waaijer, Angela Luyf, Marcel Willemsen, Alexander Ivliev, Maria Tsyganova, Peter-Bram 't Hoen, Joris Scharp, Varshna Goelela and Antoine van Kampen.

References

Bareke, E., M. Pierre, A. Gaigneaux, B. Meulder, S. Depiereux, N. Habra, and E. Depiereux (2010). PathEx: a novel multi factors based datasets selector web tool. *BMC Bioinformatics 11*(1), 528.

Barrett, T., S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko, et al. (2013). NCBI GEO: archive for functional genomics data sets – update. *Nucleic Acids Research 41*(D1), D991–D995.

Cheng, W., M. Tsai, C. Chang, C. Huang, C. Chen, W. Shu, Y. Lee, T. Wang, J. Hong, C. Li, et al. (2010). Microarray meta-analysis database (M2DB): a uniformly pre-processed, quality controlled, and manually curated human clinical microarray database. *BMC Bioinformatics 11*(1), 421.

Coletta, A., C. Molter, R. Duque, D. Steenhoff, J. Taminau, V. De Schaetzen, S. Meganck, C. Lazar, D. Venet, V. Detours, 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 Biology 13*(11), R104.

Davis, S. and P. S. Meltzer (2007). GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. *Bioinformatics 23*(14), 1846–1847.

Helbig, C., R. Gentek, R. A. Backer, Y. de Souza, I. A. Derks, E. Eldering, K. Wagner, D. Jankovic, T. Gridley, P. D. Moerland, et al. (2012). Notch controls the magnitude of T helper cell responses by promoting cellular longevity. *Proceedings of the National Academy of Sciences 109*(23), 9041–9046.

Kilpinen, S., R. Autio, K. Ojala, K. Iljin, E. Bucher, H. Sara, T. Pisto, M. Saarela, R. Skotheim, M. Björkman, et al. (2008). Systematic bioinformatic analysis of expression levels of 17,330 human genes across 9,783 samples from 175 types of healthy and pathological tissues. *Genome Biology 9*(9), R139.

Lacson, R., E. Pitzer, J. Kim, P. Galante, C. Hinske, and L. Ohno-Machado (2010). DSGeo: Software tools for cross-platform analysis of gene expression data in GEO. *Journal of Biomedical Informatics 43*(5), 709–715.

Liu, F., J. White, C. Antonescu, D. Gusenleitner, and J. Quackenbush (2011). GCOD-GeneChip oncology database. *BMC Bioinformatics 12*(1), 46.

Mackay, L. K., A. Rahimpour, J. Z. Ma, N. Collins, A. T. Stock, M.-L. Hafon, J. Vega-Ramos, P. Lauzurica, S. N. Mueller, T. Stefanovic, et al. (2013). The developmental pathway for CD103+ CD8+ tissue-resident memory T cells of skin. *Nature Immunology 14*(12), 1294–1301.

Pece, S., D. Tosoni, S. Confalonieri, G. Mazzarol, M. Vecchi, S. Ronzoni, L. Bernard, G. Viale, P. G. Pelicci, and P. P. Di Fiore (2010). Biological and molecular heterogeneity of breast cancers correlates with their cancer stem cell content. *Cell 140*(1), 62–73.

Petryszak, R., T. Burdett, B. Fiorelli, N. A. Fonseca, M. Gonzalez-Porta, E. Hastings, W. Huber, S. Jupp, M. Keays, N. Kryvych, et al. (2014). Expression Atlas update – a database of gene and transcript expression from microarray-and sequencing-based functional genomics experiments. *Nucleic Acids Research 42*(D1), D926–D932.

Pitzer, E., R. Lacson, C. Hinske, J. Kim, P. A. Galante, and L. Ohno-Machado (2009). Towards large-scale sample annotation in gene expression repositories. *BMC Bioinformatics 10*(Suppl 9), S9.

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

Rustici, G., N. Kolesnikov, M. Brandizi, T. Burdett, M. Dylag, I. Emam, A. Farne, E. Hastings, J. Ison, M. Keays, et al. (2013). ArrayExpress update – trends in database growth and links to data analysis tools. *Nucleic Acids Research 41*(D1), D987–D990.

Taminau, J., D. Steenhoff, A. Coletta, S. Meganck, C. Lazar, V. de Schaetzen, R. Duque, C. Molter, H. Bersini, A. Nowé, et al. (2011). inSilicoDb: an R/Bioconductor package for accessing human Affymetrix expert-curated datasets from GEO. *Bioinformatics 27*(22), 3204–3205.

Wakim, L. M., A. Woodward-Davis, R. Liu, Y. Hu, J. Villadangos, G. Smyth, and M. J. Bevan (2012). The molecular signature of tissue resident memory CD8 T cells isolated from the brain. *Journal of Immunology 189*(7), 3462–3471.

Xia, X., M. McClelland, S. Porwollik, W. Song, X. Cong, and Y. Wang (2009). WebArrayDB: cross-platform microarray data analysis and public data repository. *Bioinformatics 25*(18), 2425–2429.

Xie, D., C.-C. Chen, L. M. Ptaszek, S. Xiao, X. Cao, F. Fang, H. H. Ng, H. A. Lewin, C. Cowan, and S. Zhong (2010). Rewirable gene regulatory networks in the preimplantation embryonic development of three mammalian species. *Genome Research 20*(6), 804–815.

```
sessionInfo()

## R version 3.2.2 (2015-08-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices
## [6] utils     datasets  methods   base
##
## other attached packages:
##  [1] RColorBrewer_1.1-2
##  [2] gplots_2.17.0
##  [3] limma_3.24.15
##  [4] GSVA_1.16.0
##  [5] mogene10sttranscriptcluster.db_8.3.1
```

```
##  [6] org.Mm.eg.db_3.1.2
##  [7] AnnotationDbi_1.30.1
##  [8] GenomeInfoDb_1.4.3
##  [9] IRanges_2.2.7
## [10] S4Vectors_0.6.6
## [11] genefilter_1.50.0
## [12] GEOmetadb_1.28.0
## [13] RSQLite_1.0.0
## [14] inSilicoDb_2.4.1
## [15] RCurl_1.95-4.7
## [16] bitops_1.0-6
## [17] rjson_0.2.15
## [18] compendiumdb_1.0.2
## [19] RMySQL_0.10.6
## [20] DBI_0.3.1
## [21] GEOquery_2.34.0
## [22] Biobase_2.28.0
## [23] BiocGenerics_0.14.0
## [24] knitr_1.11
##
## loaded via a namespace (and not attached):
##  [1] formatR_1.2.1      highr_0.5.1
##  [3] tools_3.2.2        annotate_1.46.1
##  [5] evaluate_0.8       graph_1.46.0
##  [7] stringr_1.0.0      caTools_1.17.1
##  [9] gtools_3.5.0       GSEABase_1.30.2
## [11] XML_3.98-1.3       survival_2.38-3
## [13] gdata_2.17.0       magrittr_1.5
## [15] splines_3.2.2      xtable_1.7-4
## [17] KernSmooth_2.23-15 stringi_0.5-5
```