SOFTWARE TOOL ARTICLE

REVISED BgeeDB, an R package for retrieval of curated expression datasets and for gene list expression localization enrichment tests [version 2; peer review: 2 approved, 1 approved with reservations]

Andrea Komljenovic[1,2*], Julien Roux[2,3*], Julien Wollbrett[1,2], Marc Robinson-Rechavi[1,2], Frederic B. Bastian[1,2]

[1]Department of Ecology and Evolution, University of Lausanne, Lausanne, Switzerland
[2]SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland
[3]Department of Biomedicine, University of Basel, Basel, Switzerland

* Equal contributors

v2 **First published:** 23 Nov 2016, **5**:2748
https://doi.org/10.12688/f1000research.9973.1
Latest published: 07 Aug 2018, **5**:2748
https://doi.org/10.12688/f1000research.9973.2

Abstract

BgeeDB is a collection of functions to import into R re-annotated, quality-controlled and re-processed expression data available in the Bgee database. This includes data from thousands of wild-type healthy samples of multiple animal species, generated with different gene expression technologies (RNA-seq, Affymetrix microarrays, expressed sequence tags, and in situ hybridizations). BgeeDB facilitates downstream analyses, such as gene expression analyses with other Bioconductor packages. Moreover, BgeeDB includes a new gene set enrichment test for preferred localization of expression of genes in anatomical structures ("TopAnat"). Along with the classical Gene Ontology enrichment test, this test provides a complementary way to interpret gene lists.
Availability: https://www.bioconductor.org/packages/BgeeDB/

Keywords

Bioconductor, R Package, Collective Data Access, Gene expression, Gene Enrichment Analysis

This article is included in the Bioconductor gateway.

Open Peer Review

Approval Status ✓ ? ✓

	1	2	3
version 2 (revision) 07 Aug 2018			✓ view
version 1 23 Nov 2016	✓ view	? view	✗ view

1. **Virag Sharma**, Max Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG), Dresden, Germany
Max Planck Institute for the Physics of Complex Systems, Dresden, Germany

2. **Daniel S. Himmelstein**, University of Pennsylvania, Philadelphia, USA

3. **Leonardo Collado-Torres**, Lieber Institute for Brain Development, Baltimore, USA

Any reports and responses or comments on the

article can be found at the end of the article.

Corresponding author: Frederic B. Bastian (frederic.bastian@unil.ch)

Author roles: Komljenovic A: Conceptualization, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Roux J**: Conceptualization, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Wollbrett J**: Software, Validation; **Robinson-Rechavi M**: Funding Acquisition, Project Administration, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Bastian FB**: Project Administration, Resources, Software, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by SIB Swiss Institute of Bioinformatics project Bgee, Swiss National Science Foundation grant 31003A_153341, SystemsX.ch project AgingX, and Etat de Vaud.

How to cite this article: Komljenovic A, Roux J, Wollbrett J *et al.* **BgeeDB, an R package for retrieval of curated expression datasets and for gene list expression localization enrichment tests [version 2; peer review: 2 approved, 1 approved with reservations]** F1000Research 2018, **5**:2748 https://doi.org/10.12688/f1000research.9973.2

First published: 23 Nov 2016, **5**:2748 https://doi.org/10.12688/f1000research.9973.1

REVISED **Amendments from Version 1**

We thank the reviewers for their work, and we feel that, thanks to their comments, the manuscript has been greatly improved.

We have updated considerably the Bgee database, and the corresponding documentation. This addresses the comments made by the reviewers about a lack of transparency of our data processing steps. Moreover, we have set up the documentation is such a manner as to insure that it remains updated with the progress of the database in the future. We have added information about the processing of gene expression data performed at the Bgee database. We notably now link to the source code of our pipeline for data processing. We have added some information about similar tools allowing to perform gene list expression localization enrichment analyses. We also have updated the examples and results based on the use of the latest Bgee release and BgeeDB package version. The code examples have been made more robust to potential future changes to the format of the files used by the package. We felt that it was necessary to link the revised publication of the BgeeDB package to the updated documentation.

We have added an author, Julien Wollbrett, to the author list. Since this manuscript was first submitted, Julien Wollbrett made significant contributions to the development of the package described in this paper, and notably towards the aim of submitting this revised manuscript.

Please also note that we have updated all figures and supplementary files, so that they are based on the latest releases of our database and R package.

See referee reports

Introduction

Gene expression levels influence the behavior of cells, the functionality of tissues, and a wide range of processes from development and aging to physiology or behavior. It is of particular importance that researchers are able to take advantage of the vast amounts of publicly available gene expression datasets to reproduce and validate results, or to investigate new research questions[1–3].

To that purpose, one should be able to easily query and import gene expression datasets generated using different technologies, and their associated metadata. The R environment[4] has now become a standard for bioinformatics and statistical analysis of gene expression data, through the Bioconductor framework and its many open source packages[5,6]. It is thus desirable to provide an access to gene expression datasets programmatically and directly into R. For example, the Bioconductor packages ArrayExpress[7], GEOquery[8] and SRAdb[9] provide access to the reference databases ArrayExpress[10], GEO[11] and SRA[12] respectively.

However, such databases are primary archives aiming at comprehensiveness. They include gene expression datasets and other functional genomics data, generated from diverse experimental conditions, of diverse quality. The data provided are heterogeneous, with some datasets including only unprocessed raw data, and others including only data processed using specific analysis pipelines. For instance, over the 44,481 RNA array assay experiments stored in ArrayExpress with processed data available as of June 2018, 7,544 do not include the raw data. Metadata are often provided as free-text information that is difficult to query. For instance, the GEO database encourages submitters of high-throughput sequencing experiments to provide MINSEQE elements, but does not enforce this practice (see, e.g., GEO submission guidelines, and GEO Excel template for submissions). Unless the user needs to retrieve a specific known dataset from its accession number, it can be difficult to identify relevant available datasets. This can ultimately constitute an obstacle to data reuse.

One response to this diversity of primary archives is topical databases[1]. They can be useful for researchers of specialized fields, and even more so if they propose an R package for data access. For example, the BrainStars Bioconductor package allows access to microarray data of mouse brain regions samples from the BrainStars project[13,14]. The ImmuneSpaceR Bioconductor package allows access to the gene expression data generated by the Human Immunology Project Consortium[15]. Such efforts allow a better control of the data and annotation quality, but by nature they include a limited number of conditions, which only fit the needs of specialized projects. Similarly, numerous "ExperimentData" packages are available on the Bioconductor repository, which each include a single curated and well-formatted expression dataset (see https://www.bioconductor.org/packages/release/BiocViews.html#___ExpressionData). But these packages are rarely updated and are mostly meant to be used as examples in software packages vignettes, for teaching, or to provide supplementary data of publications. The package ExperimentHub[16] also provides access to a central location where this type of single datasets can be retrieved, but it does not address the difficulty of integrating datasets annotated and processed in different ways.

Finally, added-value databases aim at filtering, annotating, and possibly reprocessing all or some of the datasets available from the primary archives[1]. For example, there is a Bioconductor package to access the Expression Atlas, which includes a selection of microarray and RNA-seq datasets from ArrayExpress that are re-annotated and

reprocessed[17,18]. Similarly, the ReCount Bioconductor package provides access to a dataset of over 70,000 reanalyzed human RNA-seq samples from SRA (see https://jhubiostatistics.shinyapps.io/recount/)[19–21].

The Bgee database (https://bgee.org/)[22] is another added-value database, which currently offers access to reprocessed gene expression datasets from 29 animal species. Bgee aims at comparisons of gene expression patterns across tissues, developmental stages, ages and species. It provides manually curated annotations to ontology terms, describing precisely the experimental conditions used. It integrates expression data generated with multiple technologies: RNA-Seq, Affymetrix microarrays, *in situ* hybridization, and expressed sequence tags (ESTs) in release 14. An important characteristic of Bgee is that all datasets are manually curated to retain only "normal" healthy wild-type samples, i.e., excluding gene knock-out, treatments, or diseases. Finally, Bgee datasets are carefully checked for quality issues, and reprocessed to produce normalized expression level, calls of presence/absence of expression, and of differential expression. Bgee thus provides a reference of high-quality and reusable gene expression datasets that are relevant for biological insights into normal conditions of gene expression. The release 14.0 of Bgee covers 29 animal species, and includes 5,745 RNA-seq libraries, 12,996 Affymetrix chips, 360,653 results from 49,241 *in situ* hybridization experiments, and 3,335 EST libraries. This includes 4,860 human RNA-Seq libraries from the GTEx project[23,24].

Until 2016 the Bgee database lacked a programmatic access to data through a R package, a shortcoming that we have addressed with the release of the BgeeDB Bioconductor package, available at https://www.bioconductor.org/packages/BgeeDB/. The package provides functions for fast extraction of data and metadata. The data structures used in the package can be easily incorporated with other Bioconductor packages, offering a wide range of possibilities for downstream analyses.

Moreover, we introduce in BgeeDB the possibility to run TopAnat analyses, i.e., anatomical expression enrichment tests on gene lists provided by the user. This functionality is based on the topGO package[25,26], modified to use Bgee data (A. Alexa, personal communication). TopAnat is similar to the widely used Gene Ontology enrichment test[27–29]. But in our case, the enrichment test is applied to terms from an anatomical ontology, mapped to genes by expression patterns. The reference set of genes in a given species consists of all genes for which at least one "present" expression call is available in Bgee. The expression calls are propagated to parent anatomical structures by part_of and is_a relations, using the Uberon anatomical ontology[30,31] (e.g., a gene expressed in the "hindbrain" is also considered expressed in the parent structure "brain"). Different algorithms, from TopGO, are available in TopAnat to account for the non-independence of anatomical structures, and avoid the over-representation of lowly-informative top-level terms. Enrichment of expression is tested for each anatomical structure independently with a Fisher exact test, and the resulting p-values for all anatomical structures are then corrected using a FDR correction[32]. As a result, TopAnat allows to discover the tissues where a set of genes is preferentially expressed. This feature is available as a web-tool at https://bgee.org/?page=top_anat, but the R package offers more flexibility in the choice of input data and analysis parameters, and possibilities of inclusion within programs or pipelines.

There exist few other tools allowing to perform anatomical expression enrichment tests. For instance, the web-application Tissue Specific Expression Analysis (TSEA[33] based on methods from refs[34,35]) allows to perform such analyses, but only in human and mouse, while TopAnat can be used for any species integrated in Bgee (29 species as of Bgee release 14.0). For human, TSEA is based on the Genotype-Tissue Expression (GTEx) RNA-Seq dataset, while Bgee integrates GTEx data, but also other RNA-Seq datasets, and datasets from different data types, providing a higher diversity of anatomical structures. TSEA was last updated on March 2014. The database wormbase also proposes a similar tool, but for analyses only in C. elegans[36]. There exists another application for expression enrichment analyses, but focused on analyzing gene regulatory networks in human[37].

The pipeline to process the data accessible through the BgeeDB package is documented in detail at https://github.com/BgeeDB/bgee_pipeline. In brief, for RNA-seq experiments: data present in SRA[12] are selected and annotated using information from GEO[11] or from papers, or provided by the Model Organism Database Wormbase[38]. GTF annotation files and genome sequence fasta files are retrieved from Ensembl and Ensembl Genomes Metazoa[39,40]. After quality control steps, the Kallisto software is used to perform a pseudo-mapping of the reads to the transcriptome[41]. TMM normalization[42] is used to normalize TPM and F/RPKM values within each experiment independently. Present/absent expression calls are produced for each library by comparing the level of expression of each gene to the background transcriptional noise in the library (estimated by using the level of expression of intergenic regions; Roux J., Rosikiewicz M., Wollbrett J., Robinson-Rechavi M., Bastian F.B.;

in preparation). In brief, for Affymetrix experiments: data present in ArrayExpress and GEO are selected and annotated using the information available in these repositories, or in papers, or provided by the Model Organism Database Wormbase. Mappings of probesets to genes are retrieved from Ensembl and Ensembl Genomes Metazoa. Quality controls are performed to remove low quality chips and redundant chips[43,44]. When raw data are available, they are normalized using gcRMA (using version 2.42.0 for Bgee release 14.0) within each experiment independently[45]. Present/absent expression calls are generated either from the MAS5 processed data[46], based on the perfect match/mismatch probesets, or using the raw data when available, by comparing the signal of a probeset to a subset of weakly expressed probesets[47].

The BgeeDB package information is available on the Bioconductor website at https://bioconductor.org/packages/BgeeDB/. The source code is available at https://github.com/BgeeDB/BgeeDB_R. The preferred location for filing bug reports and suggestions is the issue tracker on GitHub.

In the following sections we provide some typical examples of usage of the BgeeDB package.

Methods

Requirements

To reproduce the results of examples in this paper, based on Bgee release 14.0:

- R >= 3.5
- Bioconductor >= 3.7
- BgeeDB package version >= 2.6.2
- edgeR = 3.22.2
- Mfuzz = 2.40.0
- biomaRt >= 2.36.1 (with Ensembl release 84 accessible)
- Working internet connection

Please note that an earlier version of Bioconductor and of R (>= 3.3) could be used, but would require to clone our GitHub repository and to use the `R CMD BUILD` command to build the package.

Package installation

```
source("https://bioconductor.org/biocLite.R")
biocLite("BgeeDB")
# load the library
library(BgeeDB)
```

Use cases

Data download and import of normalized expression levels

The first step of data retrieval is to initialize a new `Bgee` reference class object, for a targeted species and data type. Normalized expression levels are currently available in the BgeeDB package for two data types: Affymetrix microarrays and Illumina RNA-seq. The list of species available in the Bgee database for each data type, along with their NCBI taxonomy IDs and common names can be obtained with the `listBgeeSpecies()` function. By default, data will be downloaded from the latest Bgee release, but this can be changed with the `release` argument.

Next, the functions `getAnnotation()`, `getData()`, and `formatData()` can be called to respectively download the annotations of datasets, download the actual expression data, and reformat the expression data for more convenient use. Of note, BgeeDB creates a directory to store the downloaded annotation files and datasets, by default in the user's R working directory, but this can be changed with the `pathToData` argument. These versioned cached files make it faster for the user to return to previously used data and allow to work offline.

Microarray dataset retrieval. In the following example, we will look for a microarray dataset in mouse (*Mus musculus*), spanning multiple early developmental stages, including zygote.

```
# specify species and data type
# the examples in this paper are based on Bgee release 14.0
```

```
# the following line targets the latest Bgee release.
# In order to target specifically the release 14.0,
# add the parameter 'release="14.0"'

bgee.affymetrix <- Bgee$new(species="Mus_musculus",dataType="affymetrix")

# retrieve annotation of all mouse affymetrix datasets in Bgee
annotation.bgee.mouse.affymetrix <- getAnnotation(bgee.affymetrix)
```

This creates a list of two data frames, one including the annotation of experiments, and the other including the annotation of each individual sample, i.e., hybridized microarray chip. For mouse, there are 698 Affymetrix experiments and 6,095 samples available in Bgee release 14.0. Anatomical structures and developmental stages are annotated using the Uberon ontology[30,31]. Sex and strain information is also provided. Below, we are selecting the experiments for which at least one sample is annotated to the zygote stage (`UBERON:0000106`).

```
# retrieve annotations of samples and experiments
sample.annotation <- annotation.bgee.mouse.affymetrix$sample.annotation
experiment.annotation <- annotation.bgee.mouse.affymetrix$experiment.annotation

# list experiments including a zygote sample
selected.experiments <- unique(sample.annotation$Experiment.ID[sample.annotation$Stage.ID == "UBERON:0000106"])
experiment.annotation[experiment.annotation$Experiment.ID %in% selected.experiments,]

# stages sampled in each of these experiments
unique(sample.annotation[sample.annotation$Experiment.ID %in% selected.experiments, c("Experiment.ID", "Stage.name")])
```

This yields three microarray experiments, with accessions `GSE1749`, `E-MEXP-51` and `GSE18290`. Among these, the accession `E-MEXP-51`, submitted to ArrayExpress by Wang and colleagues[48], includes samples from more developmental stages than the other two, so we will choose this one for the next steps. For this experiment, raw data were available from ArrayExpress, so samples were fully normalized with gcRMA[49] trough the Bgee pipeline.

```
# List all samples from E-MEXP-51 in Bgee
sample.annotation[sample.annotation$Experiment.ID == "E-MEXP-51",]
```

The experiment includes 35 samples that passed Bgee quality controls. They originate from 12 developmental stages: primary and secondary oocyte, zygote, early, mid and late 2-cells embryo, 4-cells embryo, 8-cells embryo, 16-cells embryo, early, mid and late blastocyst. The developmental stage ontology used is not precise enough yet to differentiate some of these conditions: the early, mid and late 2-cells stages are annotated as Theiler stage 2 embryo, and the 4-cells and 8-cells stages are annotated as Theiler stage 3 embryo. All samples were hybridized to the `Affymetrix GeneChip Murine Genome U74Av2` microarray. Let us download the normalized probesets intensities measured for all samples.

```
data.E.MEXP.51 <- getData(bgee.affymetrix, experimentId="E-MEXP-51")
head(data.E.MEXP.51)
```

The resulting data frame lists for each sample (column “Chip.ID”), the 8,954 probesets on the microarray (column “Probeset.ID”), their mapping to Ensembl gene IDs (column “Gene.ID”), their logged normalized intensities (column “Log.of.normalized.signal.intensity”), and a presence/absence call and quality (columns “Detection.flag” and “Detection.quality”).

As this format might not be the most convenient for downstream processing of an expression dataset, we offer the `formatData()` function, which creates an `ExpressionSet` object including the expression data matrix, the probesets annotation to Ensembl genes and the samples’ anatomical structure and stage annotation into (`assayData`, `featureData` and `phenoData` slots respectively). This object class is of standard use in numerous Bioconductor packages.

```
data.E.MEXP.51.formatted <- formatData(bgee.affymetrix, data.E.MEXP.51,
callType="all", stats="intensities")
```

```
data.E.MEXP.51.formatted
# matrix of expression intensities
head(exprs(data.E.MEXP.51.formatted))
# annotation of samples
pData(data.E.MEXP.51.formatted)
# annotation of probesets
head(fData(data.E.MEXP.51.formatted))
```

The `callType` option of the `formatData()` function could alternatively be set to `present` or `present high quality` to display only the intensities of probesets detected as actively expressed.

The result is a nicely formatted Bioconductor object including expression data and their annotations, ready to be used for downstream analysis with other Bioconductor packages.

RNA-seq dataset retrieval. We will now search Bgee for a RNA-seq dataset sampling brain and liver tissues (Uberon Ids `UBERON:0000955` and `UBERON:0002107` respectively) in macaque (*Macaca mulatta*), and including multiple biological replicates for each tissue. As for Affymetrix data, Bgee RNA-seq annotations provide information about anatomical structure, developmental stage, sex, and strain.

```
# specify species and data type
# the examples in this paper are based on Bgee release 14.0
# the following line targets the latest Bgee release. In order
# to target specifically the release 14.0, add the parameter
# 'release="14.0"'
bgee.rnaseq <- Bgee$new(species="Macaca_mulatta", dataType="rna_seq")

# retrieve annotations of RNA-seq samples and experiments
annotation.bgee.macaque.rna.seq <- getAnnotation(bgee.rnaseq)
sample.annotation <- annotation.bgee.macaque.rna.seq$sample.annotation
experiment.annotation <- annotation.bgee.macaque.rna.seq$experiment.annotation

# list experiments including both brain and liver samples
selected.experiments <-intersect(unique(sample.annotation$Experiment.ID[sample.annotation$Anatomical.entity.ID == "UBERON:0000955"]),
unique(sample.annotation$Experiment.ID[sample.annotation$Anatomical.entity.ID == "UBERON:0002107"]))
experiment.annotation[experiment.annotation$Experiment.ID %in% selected.experiments,]

# check whether experiments include biological replicates
sample.annotation[sample.annotation$Experiment.ID %in%
selected.experiments & (sample.annotation$Anatomical.entity.ID=="UBERON:0000955"
| sample.annotation$Anatomical.entity.ID == "UBERON:0002107"),
c("Experiment.ID","Library.ID","Anatomical.entity.ID",
"Anatomical.entity.name","Stage.ID")]
```

Accessions `GSE41637`[50] and `GSE30352`[51] both include biological replicates for brain and liver. We will focus on `GSE41637` for the next steps since it includes three replicates of each tissue, vs. only two for `GSE30352`. We will download the dataset and reformat it to obtain an `ExpressionSet` including counts of mapped reads on each Ensembl gene for each sample.

```
data.GSE41637 <- getData(bgee.rnaseq, experimentId="GSE41637")
data.GSE41637.formatted <- formatData(bgee.rnaseq, data.GSE41637, callType="all",
stats="counts")
data.GSE41637.formatted
```

Instead of mapped read counts, it is also possible to fill the data matrix with expression levels in F/RPKMs (fragments/reads per kilobase per million reads) or in TPM (transcript per million)[52,53], using the option stats="fpkm" or stats="tpm".

Presence/absence calls retrieval. It is often difficult to compare expression levels across species[54], and even within species, across datasets generated by different experimenters or laboratories[55–57]. Batch effects have indeed been shown to impact extensively gene expression levels, confounding biological signal differences.

Encoding gene expression as present or absent in a sample allows a more robust comparison across such conditions. In addition to retrieving RNA-seq and Affymetrix quantitative expression levels, BgeeDB also allows to retrieve calls of presence or absence of expression computed in the Bgee database for each gene (RNA-seq) or probeset (Affymetrix), in the column "Detection.flag" of the `data.E.MEXP.51` and `data.GSE41637` objects created above. And interestingly, expression calls are also available in Bgee for ESTs and *in situ* hybridization data, as well as for the consensus of the four data types for each combination "gene / tissue / developmental stage / sex / strain".

A powerful use of these expression calls is the anatomical expression enrichment test "TopAnat". TopAnat uses a similar approach to Gene Ontology enrichment tests[27], but genes are associated to the anatomical structures where they display expression, instead of to their functional classification. These tests allow discovering where a set of genes is preferentially expressed as compared to a background universe (Roux J., Seppey M., Sanjeev K., Rech de Laval V., Moret P., Artimo P., Duvaud S., Ioannidis V., Stockinger H., Robinson-Rechavi M., Bastian F.B.; in preparation). We show an example of such an analysis in the section "Anatomical expression enrichment analysis" below.

Of note, the expression calls imported from BgeeDB can also be used for other downstream analyses. For example, when studying protein-protein interaction datasets, it might be biologically relevant to retain only interactions for which both members are expressed in the same tissues[58,59].

Downstream analysis examples

Clustering analysis. A variety of downstream analyses can be performed on the imported expression data. Below we detail an example of gene expression clustering analysis on the developmental time-series microarray experiment imported above. The analysis, performed with the `Mfuzz` package[60,61] (version 2.40.0 for this paper), aims at uncovering genes with similar expression profiles across development. We can readily start with the `ExpressionSet` object previously created.

```
# for simplicity, keep only one sample per condition
data.E.MEXP.51.formatted <- data.E.MEXP.51.formatted[,!duplicated(pData(data.E.MEXP.51.formatted)[c("Anatomical.
entity.ID","Anatomical.entity.name","Stage.ID","Stage.name")])]

# order developmental stages
stages <- c("GVoocyte1","MIIoocyte1","Zygote1","Early2-cell1","4Cell1","16cell1","EarlyBlastocyst1","MidBlastocyst1",
"LateBlastocyst1")
data.E.MEXP.51.formatted <- data.E.MEXP.51.formatted[, stages]

# filter out rows with no variance
data.E.MEXP.51.formatted <-
data.E.MEXP.51.formatted[apply(exprs(data.E.MEXP.51.formatted),1, sd) != 0, ]

# Mfuzz clustering
biocLite("Mfuzz")
library(Mfuzz)
# standardize matric of expression data
z.mat <- standardise(data.E.MEXP.51.formatted)
# cluster data into 16 clusters
clusters <- mfuzz(z.mat, centers=16, m=1.25)

# visualizing clusters
mfuzz.plot2(z.mat, cl=clusters, mfrow=c(4,4), colo="fancy",
time.labels=row.names(pData(z.mat)), las=2, xlab="", ylab="Standardized expression level", x11=FALSE)
```

The resulting plot can be seen in Figure 1.

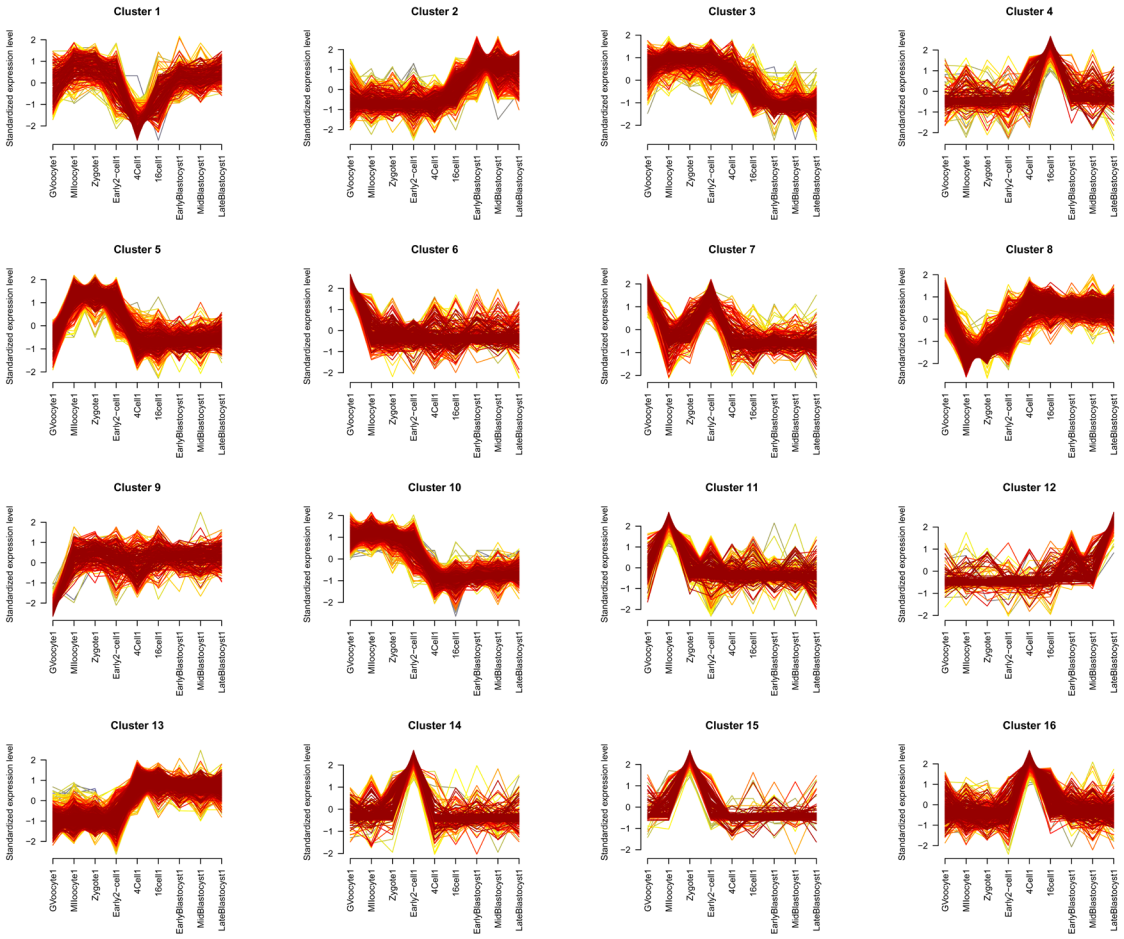

Figure 1. Standardized expression levels of 16 groups of microarray probesets, clustered according to their expression during mouse early development. The x-axis displays sample names (column "Chip.ID" of the `data.E.MEXP.51` object).

Differential expression analysis. Below, we detail a differential expression analysis, with the package `edgeR`[62,63] (version 3.22.2 for this paper), on the previously imported RNA-seq dataset of macaque tissues. We aim at isolating genes differentially expressed between brain and liver.

```
# differential expression analysis with edgeR
biocLite("edgeR")
library(edgeR)

# subset the dataset to brain and liver
brain.liver <- data.GSE41637.formatted[,pData(data.GSE41637.formatted)$Anatomical.entity.name %in%
c("brain", "liver")]

# filter out very lowly expressed genes
brain.liver.filtered <- brain.liver[rowSums(cpm(brain.liver) > 1) > 3, ]

# create edgeR DGElist object
dge <- DGEList(counts=brain.liver.filtered,
group=pData(brain.liver.filtered)$Anatomical.entity.name)
```

```
dge <- calcNormFactors(dge)
dge <- estimateCommonDisp(dge)
dge <- estimateTagwiseDisp(dge)
de <- exactTest(dge, pair=c("brain","liver"))
de.genes <- topTags(de, n=nrow(de))$table

# MA plot with DE genes highlighted
plotSmear(dge, de.tags=rownames(de.genes)[de.genes$FDR < 0.01], cex=0.3)
```

The resulting plot can be seen in Figure 2.

Anatomical expression enrichment analysis

The `loadTopAnatData()` function loads the names of anatomical structures, and relationships between them, from the Uberon anatomical ontology (based on parent-child "is_a" and "part_of" relationships). It also loads a mapping from genes to anatomical structures, based on the presence calls of the genes in the targeted species. These calls come from a consensus of all data types specified in the input Bgee class object. We recommend to use all available data types (in Bgee 14, RNA-seq, Affymetrix, EST and *in situ* hybridization) for both genomic coverage and anatomical precision, which is the default behavior if no `dataType` argument is specified when the Bgee class object is created.

By default, presence calls of both silver and gold quality are used, which can be changed with the `confidence` argument of the `loadTopAnatData()` function (in releases of Bgee up to 13, "high" and "low" confidence were used). Finally, it is possible to specify the developmental stage under consideration, with the `stage` argument. By default expression calls generated from samples of all developmental stages are used, which is equivalent to specifying `stage="UBERON:0000104"` ("life cycle", the root of the stage ontology). Data are stored in versioned tab-separated cached files that will be read again if a query with the exact same parameters is launched later, to save time and server resources, and to work offline.

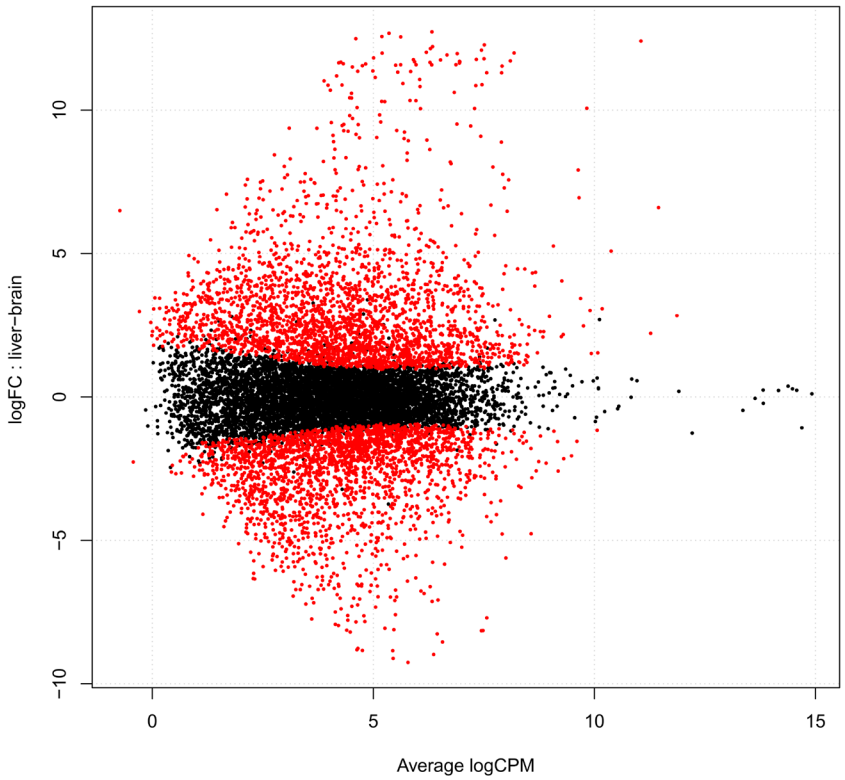

Figure 2. Mean-average (MA) plot of differential gene expression between brain and liver in macaque based on RNA-seq data. Significantly differentially expressed genes (FDR < 1%) are highlighted in red.

In this example, we will use expression calls for zebrafish genes using all sources of expression data.

```
# the examples in this paper are based on Bgee
# release 14.0
# the following line targets the latest Bgee release. In order to target
# specifically the release 14.0, add the parameter 'release="14.0"'
bgee.topanat <- Bgee$new(species="Danio_rerio")
top.anat.data <- loadTopAnatData(bgee.topanat)
```

We will look at the expression localization of the genes with an annotated phenotype related to pectoral fin (i.e., genes which upon knock-out or knock-down led to abnormal phenotypes of pectoral fin or its components). Zebrafish phenotypic data are available from the ZFIN database[64] and integrated into the Ensembl database[39]. We will thus retrieve the targeted genes using the `biomaRt`[65] Bioconductor package (version 2.36.1 for this paper).

```
biocLite("biomaRt")
library(biomaRt)

# zebrafish data in Ensembl 84, that Bgee 14.0 uses (stable link)
ensembl <- useMart("ENSEMBL_MART_ENSEMBL",
dataset="drerio_gene_ensembl", host="mar2016.archive.ensembl.org")

# get the mapping of Ensembl genes to phenotypes
genes.to.phenotypes <- getBM(filters=c("phenotype_source"),value=c("ZFIN"),
attributes=c("ensembl_gene_id","phenotype_description"), mart=ensembl)

# select phenotypes related to pectoral fin
Phenotypes <- grep("pectoral fin", unique(genes.to.phenotypes$phenotype_description), value=T)

# select the genes annotated to select phenotypes
genes <- unique(genes.to.phenotypes$ensembl_gene_id[genes.to.phenotypes$phenotype_
description %in% phenotypes])
```

This gives a list of 147 zebrafish genes implicated in the development and function of pectoral fin. The next step of the analysis relies on the `topGO` Bioconductor package. We will prepare a modified `topGOdata` object allowing to handle the Uberon anatomical ontology instead of the Gene Ontology, and perform a GO-like enrichment test for anatomical terms. As for a classical `topGO` analysis, we need to prepare a vector including all background genes, and with values 0 or 1 depending if genes are part of the foreground or not. The choice of background is very important since the wrong background can lead to spurious results in enrichment tests[66]. Here we choose as background all zebrafish Ensembl genes with an annotated phenotype from ZFIN.

```
# prepare the gene list vector
gene.list  <- factor(as.integer(unique(genes.to.phenotypes$ensembl_gene_id) %in% genes))
names(gene.list) <- unique(genes.to.phenotypes$ensembl_gene_id)
summary(gene.list)

# prepare the topAnat object based on topGO
top.anat.object  <- topAnat(top.anat.data, gene.list)
top.anat.object
```

At this step, expression calls are propagated through the whole ontology (e.g., expression in the forebrain will also be counted as expression in the brain, the nervous system, etc). This can take some time, especially if the gene list is large.

Finally, we can launch an enrichment test for anatomical terms. The functions of the `topGO` package can directly be used at this step. See the vignette of this package for more details[26]. Here we will use a Fisher test, coupled with the "weight" decorrelation algorithm.

```
results <- runTest(top.anat.object, algorithm='weight', statistic='fisher')
results
```

Finally, we implemented a function to display results in a formatted table. By default anatomical structures are sorted by their test *p*-value, which is displayed along with the associated false discovery rate (FDR[32]) and the enrichment fold. Sorting on other columns of the table (e.g., on decreasing enrichment folds) is possible with the `ordering` argument. Of note, it is debated whether a FDR correction is relevant on such enrichment test results, since tests on different terms of the ontologies are not independent. An interesting discussion can be found in the vignette of the `topGO` package.

```
# retrieve anatomical structures enriched at a 1% FDR threshold
table.Over <- makeTable(top.anat.data, top.anat.object, results, cutoff=0.01)
head(table.over)
```

The 27 anatomical structures displaying a significant enrichment at a FDR threshold of 1% are show in Table 1. The first term is "pectoral fin", and the second "paired limb/fin bud". Other terms in the list, especially those with high enrichment folds, are clearly related to pectoral fins (e.g., "pectoral appendage

Table 1. Zebrafish anatomical structures showing a significant enrichment in expression of genes with a pectoral fin phenotype (FDR < 1%). The "weight" algorithm of the topGO package was used to decorrelate the structure of the ontology.

organId	organName	annotated	significant	expected	foldEnrichment	pValue	FDR
UBERON:0000151	pectoral fin	439	79	21.48	3.68	1.36E-27	1.47E-24
UBERON:0004357	paired limb/fin bud	198	48	9.69	4.95	5.19E-23	2.80E-20
UBERON:2000040	median fin fold	59	20	2.89	6.92	9.37E-13	3.38E-10
UBERON:0003051	ear vesicle	391	49	19.13	2.56	5.50E-11	1.49E-08
UBERON:0005729	pectoral appendage field	20	11	0.98	11.22	3.05E-10	6.60E-08
UBERON:0004376	fin bone	34	12	1.66	7.23	2.60E-08	4.69E-06
UBERON:0011004	pharyngeal arch cartilage	66	16	3.23	4.95	4.96E-08	7.65E-06
UBERON:0003406	cartilage of respiratory system	52	14	2.54	5.51	8.61E-08	1.16E-05
UBERON:0004756	dermal skeletal element	55	15	2.69	5.58	9.14E-07	1.10E-04
UBERON:0003108	suspensorium	56	13	2.74	4.74	1.66E-06	1.79E-04
UBERON:0004375	bone of free limb or fin	27	9	1.32	6.82	2.77E-06	2.72E-04
UBERON:0001042	chordate pharynx	417	44	20.40	2.16	3.38E-06	3.04E-04
UBERON:0006068	bone of tail	11	6	0.54	11.11	4.67E-06	3.89E-04
UBERON:0003128	cranium	334	37	16.34	2.26	5.25E-06	4.05E-04
UBERON:4000170	median fin skeleton	26	8	1.27	6.30	2.00E-05	1.44E-03
UBERON:0004117	pharyngeal pouch	51	11	2.49	4.42	2.32E-05	1.57E-03
UBERON:0002533	post-anal tail bud	1447	95	70.79	1.34	2.77E-05	1.76E-03
UBERON:0012275	meso-epithelium	1616	104	79.05	1.32	5.53E-05	3.32E-03
UBERON:0002514	intramembranous bone	23	7	1.13	6.19	7.36E-05	4.01E-03
UBERON:0001708	jaw skeleton	108	24	5.28	4.55	7.77E-05	4.01E-03
UBERON:0001003	skin epidermis	112	16	5.48	2.92	7.79E-05	4.01E-03
UBERON:0002541	germ ring	117	16	5.72	2.80	1.33E-04	6.54E-03
UBERON:0010188	protuberance	598	68	29.25	2.32	1.51E-04	7.01E-03
UBERON:4000163	anal fin	12	5	0.59	8.47	1.57E-04	7.01E-03
UBERON:0010363	endochondral element	58	13	2.84	4.58	1.62E-04	7.01E-03
UBERON:2000555	opercular flap	26	7	1.27	5.51	1.74E-04	7.25E-03
UBERON:0007812	post-anal tail	1452	96	71.03	1.35	2.03E-04	8.13E-03

field"), or substructures of fins (e.g., "fin bone"). This analysis shows that genes with phenotypic effects on pectoral fins are specifically expressed in or next to these structures. More generally, it demonstrates the relevance of TopAnat analysis for the characterization of lists of genes.

Of note, it is possible to retrieve for a particular tissue the significant genes that were mapped to it.

```
# In order to retrieve significant genes mapped to the term "paired limb/fin bud"
term <- "UBERON:0004357"
termStat(top.anat.object, term)

# 198 genes mapped to this term for Bgee 14.0 and Ensembl 84
genesInTerm(top.anat.object, term)
# 48 significant genes mapped to this term for Bgee 14.0
# and Ensembl 84
annotated <- genesInTerm(top.anat.object,
term)[["UBERON:0004357"]]
annotated[annotated %in% sigGenes(top.anat.object)]
```

Conclusion

In summary, the BgeeDB package serves as a bridge between curated data from the Bgee database and the R/Bioconductor environment, facilitating the access to high-quality curated and re-analyzed gene expression datasets, and significantly reducing time for downstream analyses of the datasets. Moreover, it provides access to TopAnat, a new enrichment tool allowing to make sense of lists of genes, by uncovering their preferential localization of expression in anatomical structures. The TopAnat workflow is straightforward; for users already using topGO in their analysis pipelines, performing a TopAnat analysis on the same gene list only requires 6 additional lines of code.

Software and data availability

Software available from: http://www.bioconductor.org/packages/BgeeDB/

Latest source code: https://github.com/BgeeDB/BgeeDB_R

Archived source code as at the time of publication: https://doi.org/10.5281/zenodo.1293418[67]

Author contributions

AK and JR contributed equally to this work. AK developed the initial BgeeDB R package and made it available in Bioconductor. JR implemented the enrichment analyses, and refined the data download part. JW corrected some bugs, updated the package and the files used by the package, to use the latest Bgee release and for better compatibility between operating systems. FBB developed the server-side responses. MRR and FBB tested and commented on the package development. AK and JR wrote the manuscript. All authors discussed the results and implications and commented on the manuscript at all stages.

Competing interests

No competing interests were disclosed.

Grant information

This work was supported by SIB Swiss Institute of Bioinformatics project Bgee, Swiss National Science Foundation grant 31003A_153341, SystemsX.ch project AgingX, and Etat de Vaud.

Supplementary material

File S1. R markdown file including code from the paper.

Click here to access the data.

File S2. PDF file including the results of execution of the code from File S1.

Click here to access the data.

References

1. Rung J, Brazma A: **Reuse of public genome-wide gene expression data.** *Nat Rev Genet.* 2013; **14**(2): 89–99.
PubMed Abstract | Publisher Full Text

2. Ioannidis JP, Allison DB, Ball CA, *et al.*: **Repeatability of published microarray gene expression analyses.** *Nat Genet.* 2009; **41**(2): 149–55.
PubMed Abstract | Publisher Full Text

3. Wan X, Pavlidis P: **Sharing and reusing gene expression profiling data in neuroscience.** *Neuroinformatics.* 2007; **5**(3): 161–75.
PubMed Abstract | Publisher Full Text | Free Full Text

4. R Development Core Team: **R: A Language and Environment for Statistical Computing.** Vienna, Austria: R Foundation for Statistical Computing; 2007.
Reference Source

5. Huber W, Carey VJ, Gentleman R, *et al.*: **Orchestrating high-throughput genomic analysis with Bioconductor.** *Nat Methods.* 2015; **12**(2): 115–21.
PubMed Abstract | Publisher Full Text | Free Full Text

6. Gentleman RC, Carey VJ, Bates DM, *et al.*: **Bioconductor: open software development for computational biology and bioinformatics.** *Genome Biol.* 2004; **5**(10): R80.
PubMed Abstract | Publisher Full Text | Free Full Text

7. Kauffmann A, Rayner TF, Parkinson H, *et al.*: **Importing ArrayExpress datasets into R/Bioconductor.** *Bioinformatics.* 2009; **25**(16): 2092–4.
PubMed Abstract | Publisher Full Text | Free Full Text

8. Davis S, Meltzer PS: **GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor.** *Bioinformatics.* 2007; **23**(14): 1846–7.
PubMed Abstract | Publisher Full Text

9. Zhu Y, Stephens RM, Meltzer PS, *et al.*: **SRAdb: query and use public next-generation sequencing data from within R.** *BMC Bioinformatics.* 2013; **14**(1): 19.
PubMed Abstract | Publisher Full Text | Free Full Text

10. Kolesnikov N, Hastings E, Keays M, *et al.*: **ArrayExpress update--simplifying data submissions.** *Nucleic Acids Res.* 2015; **43**(Database issue): D1113–6.
PubMed Abstract | Publisher Full Text | Free Full Text

11. Barrett T, Wilhite SE, Ledoux P, *et al.*: **NCBI GEO: archive for functional genomics data sets--update.** *Nucleic Acids Res.* 2013; **41**(Database issue): D991–5.
PubMed Abstract | Publisher Full Text | Free Full Text

12. Kodama Y, Shumway M, Leinonen R, *et al.*: **The Sequence Read Archive: explosive growth of sequencing data.** *Nucleic Acids Res.* 2012; **40**(Database issue): D54–D6.
PubMed Abstract | Publisher Full Text | Free Full Text

13. **BrainStars Bioconductor package.**
Publisher Full Text

14. Kasukawa T, Masumoto KH, Nikaido I, *et al.*: **Quantitative expression profile of distinct functional regions in the adult mouse brain.** *PLoS One.* 2011; **6**(8): e23228.
PubMed Abstract | Publisher Full Text | Free Full Text

15. **ImmuneSpaceR Bioconductor package.**
Publisher Full Text

16. Bioconductor Package Maintainer: **ExperimentHub: Client to access ExperimentHub resources.** R package version 1.6.0. 2018.
Publisher Full Text

17. **ExpressionAtlas Bioconductor package.**
Publisher Full Text

18. Petryszak R, Keays M, Tang YA, *et al.*: **Expression Atlas update--an integrated database of gene and protein expression in humans, animals and plants.** *Nucleic Acids Res.* 2016; **44**(D1): D746–52.
PubMed Abstract | Publisher Full Text | Free Full Text

19. Collado-Torres L, Nellore A, Kammers K, *et al.*: **recount: A large-scale resource of analysis-ready RNA-seq expression data.** *bioRxiv.* 2016.
Publisher Full Text

20. Frazee AC, Langmead B, Leek JT: **ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets.** *BMC Bioinformatics.* 2011; **12**(1): 449.
PubMed Abstract | Publisher Full Text | Free Full Text

21. **recount Bioconductor package.**
Publisher Full Text

22. Bastian F, Parmentier G, Roux J, *et al.*: **Bgee: Integrating and Comparing Heterogeneous Transcriptome Data Among Species.** *Data Integr Life Sci.* 2008; 124–31.
Publisher Full Text

23. GTEx Consortium: **Human genomics. The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans.** *Science.* 2015; **348**(6235): 648–60.
PubMed Abstract | Publisher Full Text | Free Full Text

24. Melé M, Ferreira PG, Reverter F, *et al.*: **Human genomics. The human transcriptome across tissues and individuals.** *Science.* 2015; **348**(6235): 660–5.
PubMed Abstract | Publisher Full Text | Free Full Text

25. Alexa A, Rahnenführer J, Lengauer T: **Improved scoring of functional groups from gene expression data by decorrelating GO graph structure.** *Bioinformatics.* 2006; **22**(13): 1600–7.
PubMed Abstract | Publisher Full Text

26. **topGO Bioconductor package.**
Publisher Full Text

27. Rhee SY, Wood V, Dolinski K, *et al.*: **Use and misuse of the gene ontology annotations.** *Nat Rev Genet.* 2008; **9**(7): 509–15.
PubMed Abstract | Publisher Full Text

28. Ashburner M, Ball CA, Blake JA, *et al.*: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nat Genet.* 2000; **25**(1): 25–9.
PubMed Abstract | Publisher Full Text | Free Full Text

29. **The Gene Ontology Handbook.** Dessimoz C, Škunca N, editors: Humana Press; 2016; XII, 305.
Publisher Full Text

30. Haendel MA, Balhoff JP, Bastian FB, *et al.*: **Unification of multi-species vertebrate anatomy ontologies for comparative biology in Uberon.** *J Biomed Semantics.* 2014; **5**(1): 21.
PubMed Abstract | Publisher Full Text | Free Full Text

31. Mungall CJ, Torniai C, Gkoutos GV, *et al.*: **Uberon, an integrative multi-species anatomy ontology.** *Genome Biol.* 2012; **13**(1): R5.
PubMed Abstract | Publisher Full Text | Free Full Text

32. Benjamini Y, Hochberg Y: **Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.** *J R Stat Soc Series B Stat Methodol.* 1995; **57**(1): 289–300.
Reference Source

33. **Tissue Specific Expression Analysis (TSEA) version 1**. [cited 2018 June 14].
Reference Source

34. Dougherty JD, Schmidt EF, Nakajima M, *et al.*: **Analytical approaches to RNA profiling data for the identification of genes enriched in specific cells.** *Nucleic Acids Res.* 2010; **38**(13): 4218–30.
PubMed Abstract | Publisher Full Text | Free Full Text

35. Xu X, Wells AB, O'Brien DR, *et al.*: **Cell type-specific expression analysis to identify putative cellular mechanisms for neurogenetic disorders.** *J Neurosci.* 2014; **34**(4): 1420–31.
PubMed Abstract | Publisher Full Text | Free Full Text

36. Angeles-Albores D, N Lee RY, Chan J, *et al.*: **Tissue enrichment analysis for C. elegans genomics.** *BMC Bioinformatics.* 2016; **17**(1): 366.
PubMed Abstract | Publisher Full Text | Free Full Text

37. Marbach D, Lamparter D, Quon G, *et al.*: **Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases.** *Nat Methods.* 2016; **13**(4): 366–70.
PubMed Abstract | Publisher Full Text | Free Full Text

38. Lee RYN, Howe KL, Harris TW, *et al.*: **WormBase 2017: molting into a new stage.** *Nucleic Acids Res.* 2018; **46**(D1): D869–D874.
PubMed Abstract | Publisher Full Text | Free Full Text

39. Zerbino DR, Achuthan P, Akanni W, *et al.*: **Ensembl 2018.** *Nucleic Acids Res.* 2018; **46**(D1): D754–D761.
PubMed Abstract | Publisher Full Text | Free Full Text

40. Kersey PJ, Allen JE, Allot A, *et al.*: **Ensembl Genomes 2018: an integrated omics infrastructure for non-vertebrate species.** *Nucleic Acids Res.* 2018; **46**(D1): D802–D808.
PubMed Abstract | Publisher Full Text | Free Full Text

41. Bray NL, Pimentel H, Melsted P, *et al.*: **Near-optimal probabilistic RNA-seq quantification.** *Nat Biotechnol.* 2016; **34**(5): 525–7.
PubMed Abstract | Publisher Full Text

42. Robinson MD, Oshlack A: **A scaling normalization method for differential expression analysis of RNA-seq data.** *Genome Biol.* 2010; **11**(3): R25.
PubMed Abstract | Publisher Full Text | Free Full Text

43. Rosikiewicz M, Comte A, Niknejad A, *et al.*: **Uncovering hidden duplicated content in public transcriptomics data.** *Database (Oxford).* 2013; **2013**: bat010.
PubMed Abstract | Publisher Full Text | Free Full Text

44. Rosikiewicz M, Robinson-Rechavi M: **IQRray, a new method for Affymetrix microarray quality control, and the homologous organ conservation score, a new benchmark method for quality control metrics.** *Bioinformatics.* 2014; **30**(10): 1392–9.
PubMed Abstract | Publisher Full Text | Free Full Text

45. Wu Z, Irizarry RA, Gentleman R, *et al.*: **A Model-Based Background**

Adjustment for Oligonucleotide Expression Arrays. *J Am Stat Assoc.* 2004; **99**(468): 909–17.
Publisher Full Text

46. Hubbell E, Liu WM, Mei R: **Robust estimators for expression analysis.** *Bioinformatics.* 2002; **18**(12): 1585–92.
PubMed Abstract | Publisher Full Text

47. Schuster EF, Blanc E, Partridge L, *et al.*: **Correcting for sequence biases in present/absent calls.** *Genome Biol.* 2007; **8**(6): R125.
PubMed Abstract | Publisher Full Text | Free Full Text

48. Wang QT, Piotrowska K, Ciemerych MA, *et al.*: **A genome-wide study of gene activity reveals developmental signaling pathways in the preimplantation mouse embryo.** *Dev Cell.* 2004; **6**(1): 133–44.
PubMed Abstract | Publisher Full Text

49. Wu Z, Irizarry RA, Gentleman R, *et al.*: **A Model-Based Background Adjustment for Oligonucleotide Expression Arrays.** *J Am Stat Assoc.* 2004; **99**(468): 909–17.
Publisher Full Text

50. Merkin J, Russell C, Chen P, *et al.*: **Evolutionary dynamics of gene and isoform regulation in Mammalian tissues.** *Science.* 2012; **338**(6114): 1593–9.
PubMed Abstract | Publisher Full Text | Free Full Text

51. Brawand D, Soumillon M, Necsulea A, *et al.*: **The evolution of gene expression levels in mammalian organs.** *Nature.* 2011; **478**(7369): 343–8.
PubMed Abstract | Publisher Full Text

52. Wagner GP, Kin K, Lynch VJ: **Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples.** *Theory Biosci.* 2012; **131**(4): 281–5.
PubMed Abstract | Publisher Full Text

53. Li B, Dewey CN: **RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome.** *BMC Bioinformatics.* 2011; **12**: 323.
PubMed Abstract | Publisher Full Text | Free Full Text

54. Roux J, Rosikiewicz M, Robinson-Rechavi M: **What to compare and how: Comparative transcriptomics for Evo-Devo.** *J Exp Zool B Mol Dev Evol.* 2015; **324**(4): 372–82.
PubMed Abstract | Publisher Full Text | Free Full Text

55. Gilad Y, Mizrahi-Man O: **A reanalysis of mouse ENCODE comparative gene expression data [version 1; referees: 3 approved, 1 approved with reservations].** *F1000Res.* 2015; **4**: 121.
PubMed Abstract | Publisher Full Text | Free Full Text

56. Leek JT, Scharpf RB, Bravo HC, *et al.*: **Tackling the widespread and critical impact of batch effects in high-throughput data.** *Nat Rev Genet.* 2010; **11**(10): 733–9.
PubMed Abstract | Publisher Full Text | Free Full Text

57. Akey JM, Biswas S, Leek JT, *et al.*: **On the design and analysis of gene expression studies in human populations.** *Nat Genet.* 2007; **39**(7): 807–8.
PubMed Abstract | Publisher Full Text

58. Deane CM, Salwiński Ł, Xenarios I, *et al.*: **Protein Interactions: Two Methods for Assessment of the Reliability of High Throughput Observations.** *Mol Cell Proteomics.* 2002; **1**(5): 349–56.
PubMed Abstract | Publisher Full Text

59. Kotlyar M, Pastrello C, Sheahan N, *et al.*: **Integrated interactions database: tissue-specific view of the human and model organism interactomes.** *Nucleic Acids Res.* 2016; **44**(D1): D536–D41.
PubMed Abstract | Publisher Full Text | Free Full Text

60. Futschik ME, Carlisle B: **Noise-robust soft clustering of gene expression time-course data.** *J Bioinform Comput Biol.* 2005; **3**(4): 965–88.
PubMed Abstract | Publisher Full Text

61. **Mfuzz Bioconductor package.**
Publisher Full Text

62. Robinson MD, McCarthy DJ, Smyth GK: **edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.** *Bioinformatics.* 2010; **26**(1): 139–40.
PubMed Abstract | Publisher Full Text | Free Full Text

63. **edgeR Bioconductor package.**
Publisher Full Text

64. Howe DG, Bradford YM, Conlin T, *et al.*: **ZFIN, the Zebrafish Model Organism Database: increased support for mutants and transgenics.** *Nucleic Acids Res.* 2013; **41**(Database issue): D854–60.
PubMed Abstract | Publisher Full Text | Free Full Text

65. **biomaRt Bioconductor package.**
Publisher Full Text

66. Timmons JA, Szkop KJ, Gallagher IJ: **Multiple sources of bias confound functional enrichment analysis of global -omics data.** *Genome Biol.* 2015; **16**(1): 186.
PubMed Abstract | Publisher Full Text | Free Full Text

67. Komljenovic A, Roux J, Robinson-Rechavi M, *et al.*: **BgeeDB/BgeeDB_R: Bgee R package release 2.6.2.** *Zenodo.* 2018.
http://www.doi.org/10.5281/zenodo.1293418

Open Peer Review

Current Peer Review Status: ✔ ? ✔

Version 2

Reviewer Report 28 August 2018

https://doi.org/10.5256/f1000research.16883.r36881

✔ **Leonardo Collado-Torres**
Lieber Institute for Brain Development, Baltimore, MD, USA

The authors (including the new author J. Wollbrett) have addressed all my comments from version 1 very throughly or plan to address some of the more technical ones in the future.

I want to highlight all the work the authors did ensuring that their paper is reproducible (they mention versions of packages throughout the text) and in clarifying their work (mostly updates on the introduction). I was able to download supplementary file 1 (an .Rmd file) and execute it without any problems or modifications at all using R 3.5.1 with Bioconductor 3.7 on a Mac (BgeeDB 2.6.2, MFuzz 2.40.0, biomaRt 2.36.1). The authors did a great job with https://github.com/BgeeDB/bgee_pipeline which I believe includes all the information needed for anyone who is interested in the Bgee pipeline. If not, the authors seem responsive on https://github.com/BgeeDB/bgee_pipeline/issues and https://github.com/BgeeDB/BgeeDB_R/issues which is a great sign. Thanks to their changes in the introduction I now have a better understanding of their anatomical expression enrichment test.

I look forward to seeing how the community uses BgeeDB in the future and the use cases they apply it to.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 16 December 2016

https://doi.org/10.5256/f1000research.10748.r17980

Leonardo Collado-Torres
Lieber Institute for Brain Development, Baltimore, MD, USA

In this manuscript the authors describe the BgeeDB Bioconductor package and show how to use it (as of Bioconductor 3.4) to interact with Bgee[1] in order to get the data from Bgee into your R session. This allows users to then perform differential expression analyses and integrate Bgee with other data sets such as unpublished data. The manuscript includes code that shows how to use BgeeDB and showcases it's different features including their unique anatomical expression enrichment analysis method.

I find interesting that you can use BgeeDB to get data from different platforms and from different organisms. Most of this can be done using other packages such as GEOquery, but BgeeDB makes it so the user doesn't have to do all the processing of the data and standarization over multiple projects.

My main concern with the manuscript in its current form and the BgeeDB package itself is the lack of clarity on how the data has been processed and how the anatomical expression test works. That is, it could potentially become a black box that produces interesting output but hides information that could be important.

For example, I'm sure some of the Affymetrix data could be downloaded with other packages and I do not know what would be the differences between the raw data and the data downloaded via BgeeDB. Is the data in BgeeDB normalized? If so, how? The help pages of BgeeDB, the package vignette, the original Bgee publication[1] and http://bgee.org/?page=doc did not help me fully answer these questions (I might have missed the information). Maybe the functions in BgeeDB could print a message describing the main steps of how a given data set was processed or this could be added to the help pages. I currently ignore if all data sets were treated the same. For instance, is all the Affymetrix data normalized with the same method and same parameters? I assume that the answer is yes but I don't know. I suggest that the authors describe in more detail the data available in Bgee. The authors might want to consider making the processing code public at https://github.com/BgeeDB or citable via figshare.

With the anatomical expression test it's not clear to me how to interpret the results from BgeeDB::makeTable(). I understand that the authors will describe the details of how their anatomical test works in a future publication, which they did before with Bgee[1] and Homolanto[2]. Ideally, the anatomical expression test would have been described first followed by BgeeDB. Without hindering the current plan, I believe that the authors could provide a summary of how TopAnat works. Then they can explain it fully in the planned future TopAnat publication. I am also curious on how users could use their own data to improve the TopAnat results, although that could be work for the TopAnat paper or future work.

I think that the manuscript is overall well written and will be more appealing if the data and main features (TopAnat) are described in more detail. I hope that the authors are not discouraged by

my report.

Best,
Leonardo

Minor comments:

- I'm an author of recount[3] which is incorrectly cited here. The pre-print version of https://jhubiostatistics.shinyapps.io/recount/ had data from 2040 different projects which together made up more than 60,000 RNA-seq samples. The current version has data from over 70,000 Illumina human RNA-seq samples from SRA, GTEx and TCGA.

- I don't think that it makes sense to include the str() calls in the paper. They do make sense in the supplementary material (the html and pdf rendered versions of the paper code) since those include the output. Also, while str() shows all the details of an object, it can encourage users to write code that depends on the internal structure of the object. You might want to consider adding accessor functions.

- If you added indentation the code that runs over multiple lines would be easier to read. You can use the Bioconductor standard of using 4 spaces at the start of the line. Also make sure that object names don't get split into multiple lines. For example check the line after the "list experiments including both brain and liver samples" comment where "Anatomical.entity.ID" gets split into "Anatomical.e" and "ntity.ID" in the html version of the paper. Copy pasting works fine, but if someone prints the paper they might introduce errors can be avoided with better formatting. F1000Research's team should be able to tell you what is the character limit per line to use so that the PDF and HTML versions look great. The formatR package might be useful here.

- I would not use numerical indexes in the code since the results could change with time in such a way that the current code would not work in the future or worse, it might run without error but change the results in a way a new user would not notice. For example, change the code on the line after the "order developmental stages" comment which currently reads:

  data.E.MEXP.51.formatted <- data.E.MEXP.51.formatted[, c(5,8,9,3,2,1,4,7,6)]

- The comment that reads with "retrieve anatomical structures enriched at a 1% FDR threshold" is mixed with the code. That is, you are missing a new line character.

- Reference 46 is incorrect. It's edgeR, not eedgeR.

- The package's vignette is missing a title as currently shown at http://bioconductor.org/packages/release/bioc/html/BgeeDB.html.

- I recommend adding internal R links to your manual pages. For example, ?topAnat mentions loadTopAnatData(). Those links make it easier for a user to browse the help pages.

I was able to run all the code without any edits (beyond that new line issue I already mentioned)

using Bioconductor 3.4 (current Bioc-release) on R 3.3.1. Here are my session details:

```
> options(width = 120)
> devtools::session_info()
Session info -------------------------------------------------------------------------------------------------------------
 setting  value
 version  R version 3.3.1 (2016-06-21)
 system   x86_64, mingw32
 ui       RStudio (0.99.902)
 language (EN)
 collate  English_United States.1252
 tz       America/Mexico_City
 date     2016-12-15

Packages -----------------------------------------------------------------------------------------------------------------
 package      * version  date       source
 AnnotationDbi * 1.36.0   2016-10-18 Bioconductor
 assertthat     0.1      2013-12-06 CRAN (R 3.3.1)
 BgeeDB       * 2.0.0    2016-10-18 Bioconductor
 Biobase      * 2.34.0   2016-10-18 Bioconductor
 BiocGenerics  * 0.20.0   2016-10-18 Bioconductor
 BiocInstaller * 1.24.0   2016-10-18 Bioconductor
 biomaRt      * 2.30.0   2016-10-18 Bioconductor
 bitops         1.0-6    2013-08-17 CRAN (R 3.3.1)
 class          7.3-14   2015-08-30 CRAN (R 3.3.1)
 data.table     1.10.0   2016-12-03 CRAN (R 3.3.2)
 DBI            0.5-1    2016-09-10 CRAN (R 3.3.1)
 devtools       1.12.0   2016-06-24 CRAN (R 3.3.1)
 digest         0.6.10   2016-08-02 CRAN (R 3.3.1)
 dplyr          0.5.0    2016-06-24 CRAN (R 3.3.1)
 DynDoc       * 1.52.0   2016-10-18 Bioconductor
 e1071        * 1.6-7    2015-08-05 CRAN (R 3.3.1)
 edgeR        * 3.16.4   2016-11-27 Bioconductor
 GO.db        * 3.4.0    2016-10-22 Bioconductor
 graph        * 1.52.0   2016-10-18 Bioconductor
 IRanges      * 2.8.1    2016-11-08 Bioconductor
 lattice        0.20-34  2016-09-06 CRAN (R 3.3.1)
 limma        * 3.30.6   2016-11-29 Bioconductor
 locfit         1.5-9.1  2013-04-20 CRAN (R 3.3.1)
 magrittr       1.5      2014-11-22 CRAN (R 3.3.1)
 matrixStats    0.51.0   2016-10-09 CRAN (R 3.3.1)
 memoise        1.0.0    2016-01-29 CRAN (R 3.3.1)
 Mfuzz        * 2.34.0   2016-10-18 Bioconductor
 R6             2.2.0    2016-10-05 CRAN (R 3.3.1)
 Rcpp           0.12.8   2016-11-17 CRAN (R 3.3.2)
 RCurl          1.95-4.8 2016-03-01 CRAN (R 3.3.1)
 rsconnect      0.6      2016-11-21 CRAN (R 3.3.2)
 RSQLite        1.1-1    2016-12-10 CRAN (R 3.3.2)
```

S4Vectors	* 0.12.1	2016-12-01	Bioconductor
SparseM	* 1.74	2016-11-10	CRAN (R 3.3.2)
tibble	1.2	2016-08-26	CRAN (R 3.3.1)
tidyr	* 0.6.0	2016-08-12	CRAN (R 3.3.1)
tkWidgets	1.52.0	2016-10-18	Bioconductor
topGO	* 2.26.0	2016-10-18	Bioconductor
widgetTools	* 1.52.0	2016-10-18	Bioconductor
withr	1.0.2	2016-06-20	CRAN (R 3.3.1)
XML	3.98-1.5	2016-11-10	CRAN (R 3.3.2)

Regarding Virag Sharma's peer review report[4], I assume that Virag was using an earlier R version (and thus an earlier Bioconductor version). The current development version of BgeeDB uses "dataType" and not "datatype", just like the release version. Check https://github.com/Bioconductor-mirror/BgeeDB/search?utf8=%E2%9C%93&q=datatype. Hopefully the authors won't change the spelling of arguments in the future since that's confusing for users, although that's certainly doable following the deprecated/defunct code cycle.

Regarding Daniel S. Himmelstein's peer review report[5], there is no need to add a license file when the license is specified in the DESCRIPTION file of an R package. See https://github.com/Bioconductor-mirror/BgeeDB/blob/master/DESCRIPTION#L14 where they state that the license is GPL-2. Although the authors should make sure that they correctly specify which license their software is released on: GPL-2 or GPLv3 as Daniel mentioned. Regarding where to place bug reports, the authors could resolve this by specifying the "BugReports" field in their DESCRIPTION file. For example see https://github.com/Bioconductor-mirror/recount/blob/master/DESCRIPTION#L63. I also agree with Daniel that currently BgeeDB has a bit of a messy download structure. I would prefer if the files were downloaded in a single directory (say "bgee_downloads") instead of the current working directory.

References

1. Bastian F, Parmentier G, Roux J, Moretti S, et al.: Bgee: Integrating and Comparing Heterogeneous Transcriptome Data Among Species. 2008. 124-131 Publisher Full Text
2. Parmentier G, Bastian FB, Robinson-Rechavi M: Homolonto: generating homology relationships by pairwise alignment of ontologies and application to vertebrate anatomy.*Bioinformatics*. 2010; **26** (14): 1766-71 PubMed Abstract | Publisher Full Text
3. Collado-Torres L, Nellore A, Kammers K, Ellis S, et al.: recount: A large-scale resource of analysis-ready RNA-seq expression data. 2016. Publisher Full Text
4. Sharma V: Referee Report For: BgeeDB, an R package for retrieval of curated expression datasets and for gene list expression localization enrichment tests [version 1; referees: 1 approved, 1 approved with reservations]. *F1000Research*. 2016; **5** (2748). Publisher Full Text
5. Himmelstein DS: Referee Report For: BgeeDB, an R package for retrieval of curated expression datasets and for gene list expression localization enrichment tests [version 1; referees: 1 approved, 1 approved with reservations]. 2016; **5** (2748). Publisher Full Text

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of

expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 28 Jun 2018

Frederic Bastian, University of Lausanne, Lausanne, Switzerland

My main concern with the manuscript in its current form and the BgeeDB package itself is the lack of clarity on how the data has been processed and how the anatomical expression test works. That is, it could potentially become a black box that produces interesting output but hides information that could be important.
For example, I'm sure some of the Affymetrix data could be downloaded with other packages and I do not know what would be the differences between the raw data and the data downloaded via BgeeDB. Is the data in BgeeDB normalized? If so, how? The help pages of BgeeDB, the package vignette, the original Bgee publication1 and http://bgee.org/?page=doc did not help me fully answer these questions (I might have missed the information).

We have made public the Bgee pipeline source code at https://github.com/BgeeDB/bgee_pipeline. We also have added a paragraph at the end of the "Introduction" section, pointing to the relevant part of the documentation for RNA-Seq and Affymetrix analyses, and describing them in brief.

Maybe the functions in BgeeDB could print a message describing the main steps of how a given data set was processed or this could be added to the help pages.

We have opened an issue on our tracker related to this point, see https://github.com/BgeeDB/BgeeDB_R/issues/22. We will add a function pointing to the relevant documentation in a future release.

I currently ignore if all data sets were treated the same. For instance, is all the Affymetrix data normalized with the same method and same parameters? I assume that the answer is yes but I don't know.

The Affymetrix data are not treated in the same way depending on whether the raw data were available, or only the data processed by using the MAS5 software. This is clarified at the end of the "Introduction" section. Also, in the package, this information about raw data availability can be retrieved in the annotation data frame.

I suggest that the authors describe in more detail the data available in Bgee. The authors might want to consider making the processing code public at https://github.com/BgeeDB or citable via figshare.

We have made public the Bgee pipeline source code at https://github.com/BgeeDB/bgee_pipeline.

With the anatomical expression test it's not clear to me how to interpret the results from BgeeDB::makeTable(). I understand that the authors will describe the details of how their anatomical test works in a future publication, which they did before with Bgee and Homolanto. Ideally, the anatomical expression test would have been described first followed by BgeeDB. Without hindering the current plan, I believe that the authors could provide a summary of how TopAnat works. Then they can explain it fully in the planned future TopAnat publication.

We have added a brief description of how TopAnat works in the "Introduction" section.

I am also curious on how users could use their own data to improve the TopAnat results, although that could be work for the TopAnat paper or future work.

This represents an advanced use of TopAnat that we don't find suitable for the paper. But users can override the association file, mapping genes to anatomical structures in the BgeeDB directory, to use their own data. Also, since the source code of the package is public, users can also modify the mapping files used by modifying the source code.

I'm an author of recount which is incorrectly cited here. The pre-print version of https://jhubiostatistics.shinyapps.io/recount/ had data from 2040 different projects which together made up more than 60,000 RNA-seq samples. The current version has data from over 70,000 Illumina human RNA-seq samples from SRA, GTEx and TCGA.

We have updated the number in our paper. We apologize for the mistake.

I don't think that it makes sense to include the str() calls in the paper. They do make sense in the supplementary material (the html and pdf rendered versions of the paper code) since those include the output. Also, while str() shows all the details of an object, it can encourage users to write code that depends on the internal structure of the object. You might want to consider adding accessor functions.

We have removed str() calls from the paper. For the future, we will think of adding accessor functions, although several are already available thanks to the topGO package.

If you added indentation the code that runs over multiple lines would be easier to read. You can

use the Bioconductor standard of using 4 spaces at the start of the line. Also make sure that object names don't get split into multiple lines. For example check the line after the "list experiments including both brain and liver samples" comment where "Anatomical.entity.ID" gets split into "Anatomical.e" and "ntity.ID" in the html version of the paper. Copy pasting works fine, but if someone prints the paper they might introduce errors can be avoided with better formatting. F1000Research's team should be able to tell you what is the character limit per line to use so that the PDF and HTML versions look great. The formatR package might be useful here.

We didn't know about the formatR package and will have a look at it. In the meantime, we have split such offending lines, as identified by the reviewer.

I would not use numerical indexes in the code since the results could change with time in such a way that the current code would not work in the future or worse, it might run without error but change the results in a way a new user would not notice. For example, change the code on the line after the "order developmental stages" comment which currently reads:
data.E.MEXP.51.formatted <- data.E.MEXP.51.formatted[, c(5,8,9,3,2,1,4,7,6)]

We have replaced all lines using numerical indexes, with use of column names.

The comment that reads with "retrieve anatomical structures enriched at a 1% FDR threshold" is mixed with the code. That is, you are missing a new line character.

This was fixed.

Reference 46 is incorrect. It's edgeR, not eedgeR.

This was fixed.

The package's vignette is missing a title as currently shown at http://bioconductor.org/packages/release/bioc/html/BgeeDB.html.

This was added.

I recommend adding internal R links to your manual pages. For example, ?topAnat mentions loadTopAnatData(). Those links make it easier for a user to browse the help pages.

We thank the reviewer for the suggestion, and we will implement this in a future release.

I was able to run all the code without any edits (beyond that new line issue I already mentioned) using Bioconductor 3.4 (current Bioc-release) on R 3.3.1. Here are my session details:
> options(width = 120)
> devtools::session_info()
[...]
Regarding Virag Sharma's peer review report4, I assume that Virag was using an earlier R version (and thus an earlier Bioconductor version). The current development version of BgeeDB uses "dataType" and not "datatype", just like the release version. Check https://github.com/Bioconductor-mirror/BgeeDB/search?utf8=%E2%9C%93&q=datatype. Hopefully the authors won't change the spelling of arguments in the future since that's confusing for users, although that's certainly doable following the deprecated/defunct code cycle.

This is indeed a change that we introduced in an earlier version, in an effort to name all our arguments in a consistent manner. We will try not to change this in the future.

Regarding Daniel S. Himmelstein's peer review report5, there is no need to add a license file when the license is specified in the DESCRIPTION file of an R package. See https://github.com/Bioconductor-mirror/BgeeDB/blob/master/DESCRIPTION#L14 where they state that the license is GPL-2. Although the authors should make sure that they correctly specify which license their software is released on: GPL-2 or GPLv3 as Daniel mentioned.

We have updated the DESCRIPTION file in the development branch of Bioconductor. The package is now released under the GPL-3.0 license.

Regarding where to place bug reports, the authors could resolve this by specifying the "BugReports" field in their DESCRIPTION file. For example see https://github.com/Bioconductor-mirror/recount/blob/master/DESCRIPTION#L63.

This was done.

I also agree with Daniel that currently BgeeDB has a bit of a messy download structure. I would prefer if the files were downloaded in a single directory (say "bgee_downloads") instead of the current working directory.

While another directory can be specified by using the "pathToData" argument, it is true that the solution proposed by the reviewer would be convenient, and we will try to update the package accordingly in the future.

Competing Interests: No competing interests were disclosed.

Reviewer Report 14 December 2016

https://doi.org/10.5256/f1000research.10748.r18221

?

Daniel S. Himmelstein
Systems Pharmacology and Translational Therapeutics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA

This study describes the BgeeDB R package, which provides a programmatic interface for accessing Bgee gene expression data. Bgee is a valuable resource because it integrates gene expression results across many experiments. Previously, I've used Bgee for its presence/absence of expression calls and its differential expression calls.

In my opinion, Bgee's ability to provide a genome-wide profile of expression for a given species, developmental stage, and anatomical structure is its most powerful capability. It was not clear to me whether BgeeDB provides this functionality. For example, can the user retrieve the normalized expression level across several experiments for the same species-stage-anatomy combination? In general, I think users will be more interested in this high-level functionality than the low level access BgeeDB currently provides. An example here would likely clear things up for me.

Is it possible to integrate expression levels across Affymetrix and RNA-Seq experiments?

The Zenodo archive of the source code specifies GPLv3 as the license. This is great, but it's ideal to also add a LICENSE file to the GitHub.

It looks like there are at least two potential places where bug reports should be filed: on Bioconductor Support and GitHub Issues. It would be nice to clarify the preferred location for filing bug reports go and opening pull requests.

Currently, the GitHub repository BgeeDB/BgeeDB_R mentioned in the manuscript is forked from wirawara/BgeeDB. I expect this may cause some confusion, as BgeeDB/BgeeDB_R should be the upstream repository that users fork and contribute back to. If you make wirawara/BgeeDB private, this should break the relationship. @wirawara can then fork BgeeDB/BgeeDB_R to continue contributions if desired.

Finally, I created some GitHub issues as part of this review:

- Sample annotation variable names

- A less messy default download directory

References

1. Morin A, Urban J, Sliz P: A quick guide to software licensing for the scientist-programmer.*PLoS Comput Biol*. 2012; **8** (7): e1002598 PubMed Abstract | Publisher Full Text

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 28 Jun 2018

Frederic Bastian, University of Lausanne, Lausanne, Switzerland

In my opinion, Bgee's ability to provide a genome-wide profile of expression for a given species, developmental stage, and anatomical structure is its most powerful capability. It was not clear to me whether BgeeDB provides this functionality. For example, can the user retrieve the normalized expression level across several experiments for the same species-stage-anatomy combination? In general, I think users will be more interested in this high-level functionality than the low level access BgeeDB currently provides. An example here would likely clear things up for me.

Indeed there is currently no easy way to do this. As mentioned in https://github.com/BgeeDB/BgeeDB_R/issues/7, it would be nice to have a getDataByCondition function that would return all processed data for chips / libraries matching a queried organ/stage/(sex)/(strain). But it was hard to set priorities for the initial development (should the package complement the web interface, or be orthogonal to it?), and we will likely implement it in the near future.

Is it possible to integrate expression levels across Affymetrix and RNA-Seq experiments?

If the reviewer means to integrate present/absent expression calls, it is relatively easy to get all the genes expressed in one tissue and all sub-tissues from Affymetrix and RNA-Seq data, although a dedicated method could be added, for instance:

```
library(BgeeDB)
bgee_human <- Bgee$new(species='Homo_sapiens', dataType=c('rna_seq', 'affymetrix'))
my_data <- loadTopAnatData(bgee_human)
calls_by_tissue <- reverseSplit(my_data$gene2anatomy)
# pick you favorite tissue, for example liver
calls_by_tissue[["UBERON:0002107"]]
```

And this can be limited by stage too, for example:

my_data <- loadTopAnatData(bgee_human, stage="UBERON:0000068")

We have noted in the issue 7 mentioned above to add a direct function to do this.

If the reviewer means to integrate levels of expression, it is then not the aim of Bgee: Bgee integrate different data types and different experiments, processed and normalized independently (but in a consistent manner).

The Zenodo archive of the source code specifies GPLv3 as the license. This is great, but it's ideal to also add a LICENSE file to the GitHub.

We have added the LICENSE file to GitHub (GPL 3.0).

It looks like there are at least two potential places where bug reports should be filed: on Bioconductor Support and GitHub Issues. It would be nice to clarify the preferred location for filing bug reports go and opening pull requests.

We have added the preferred location for filing bug reports at the end of the "Introduction" section (GitHub), and in the DESCRIPTION file of the source code.

Currently, the GitHub repository BgeeDB/BgeeDB_R mentioned in the manuscript is forked from wirawara/BgeeDB. I expect this may cause some confusion, as BgeeDB/BgeeDB_R should be the upstream repository that users fork and contribute back to. If you make wirawara/BgeeDB private, this should break the relationship. @wirawara can then fork BgeeDB/BgeeDB_R to continue contributions if desired.

We thank the reviewer for the suggestion, we have now made wirawara/BgeeDB private.

Finally, I created some GitHub issues as part of this review:
Sample annotation variable names
https://github.com/BgeeDB/BgeeDB_R/issues/5

We have replied on the issue. Our answer was that is a bit of a controversial topic. For example Google's R Style Guide (https://google.github.io/styleguide/Rguide.xml#identifiers) advise against the use of underscores (although they do not justify why, and we agree that the "words separated with dots" convention can be disturbing for python users).

A less messy default download directory

This point was discussed in https://github.com/BgeeDB/BgeeDB_R/issues/4. We notably mention that another directory can be specified by using the "pathToData" argument. This parameter is mentioned at the end of the section "Data download and import of normalized expression levels". We agree that a default directory should be used in future releases.

Competing Interests: No competing interests were disclosed.

Reviewer Report 07 December 2016

https://doi.org/10.5256/f1000research.10748.r17925

Virag Sharma
[1] Max Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG), Dresden, Germany
[2] Max Planck Institute for the Physics of Complex Systems, Dresden, Germany

In the manuscript, Komljenovic et al. present BgeeDB which is an R package for retrieval of expression datasets which have been curated. Additionally, they also provide a method (TopAnat) to determine tissue-specific enrichments for a given list of genes and species.

The former is a very useful resource because there is clearly a need for a database that provides gene expression datasets which are homogenous in nature and are of comparable quality. The BgeeDB database contains gene expression datasets from 17 species across different tissues and developmental stages, which is impressive. The fact that the database can be queried via a Bioconductor package should ensure that the database will be used by both - wet-lab biologists and computational scientists.
Similarly, the TopAnat method also provides a useful functionality to determine anatomical expression enrichment on a user specified list.

I have a few minor comments regarding the manuscript:

1. The authors should include some details about how they have reprocessed the gene expression datasets that are a part of BgeeDB. At the moment, it is rather unclear how this was achieved. I assume that the authors have an automated pipeline in place but it would be beneficial for readers to know how this was done.

2. The authors state that “TopAnat allows for discovery of tissues where a set of genes is preferentially expressed”. Is TopAnat the only tool that offers such a functionality? A brief background of similar tools that are currently available will be useful for the readers.

3. I was not able to run the workflow that the authors have included in the Supplementary material:

   See below:

   ```
   source("https://bioconductor.org/biocLite.R")
   biocLite("BgeeDB")
   biocLite(c("edgeR", "Mfuzz", "biomaRt"))
   library(BgeeDB)
   listBgeeSpecies()

   bgee_affymetrix <- Bgee$new(species="Mus_musculus", dataType="affymetrix",
   release="13.2")
   Error in envRefSetField(.Object, field, classDef, selfEnv, elements[[field]]) :
   'dataType' is not a field in class "Bgee"

   ## Turns out that I need to use "datatype" instead of "dataType"
   bgee_affymetrix <- Bgee$new(species="Mus_musculus", datatype="affymetrix")
   bgee_affymetrix <- Bgee$new(species="Mus_musculus", datatype="affymetrix",
   release="13.2")
   Error in envRefSetField(.Object, field, classDef, selfEnv, elements[[field]]) :
   'release' is not a field in class "Bgee"

   ####
   ```

   At this moment, I did not try further.
   The authors need to clearly state what version of BgeeDB was used to create this workflow. If something has changed, then this needs to be appropriately addressed. I tried using “release=13.2” but it did not work.

4. I did manage to run an enrichment test for anatomical terms though with some tweaking

   ```
   ## Again an error message
   bgee_topanat <- loadTopAnatData(species="Danio_rerio")
   Error in loadTopAnatData(species = "Danio_rerio") :
   Problem: the specified speciesId is not among the list of species in Bgee.

   ## This works though
   myTopAnatData <- loadTopAnatData(species="7955")

   ####
   ```

   The rest of the work-flow went smoothly and I was able to get a list of anatomical structures sorted by their p-value

   ```
   head(tableOver)
         organId                organName annotated significant
   ```

12	UBERON:0004357	paired limb/fin bud	144	41
2	UBERON:0000151	pectoral fin	420	70
22	UBERON:2000040	median fin fold	51	18
9	UBERON:0003051	ear vesicle	304	41
15	UBERON:0005729	pectoral appendage field	16	10
16	UBERON:0007390	pectoral appendage cartilage tissue	17	9

	expected	foldEnrichment	pValue	FDR
12	7.15	5.734266	1.622480e-22	1.445630e-19
2	20.85	3.357314	1.037552e-18	4.622296e-16
22	2.53	7.114625	7.171001e-12	2.129787e-09
9	15.09	2.717031	3.135769e-10	6.984926e-08
15	0.79	12.658228	4.004917e-10	7.136762e-08
16	0.84	10.714286	2.411891e-08	3.581659e-06

It would be useful if the authors could include a feature that allows the TopAnat method to print the 41 genes which represent the paired limb/fin bud. At some point, the users might want to revisit their gene lists and tag their genes based on the different anatomical structures.

Other tools that perform Enrichment tests, for example Enrichr[1], have this feature and this is extremely useful, in my opinion.

References

1. Kuleshov MV, Jones MR, Rouillard AD, Fernandez NF, et al.: Enrichr: a comprehensive gene set enrichment analysis web server 2016 update.*Nucleic Acids Res*. 2016; **44** (W1): W90-7 PubMed Abstract | Publisher Full Text

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 28 Jun 2018

Frederic Bastian, University of Lausanne, Lausanne, Switzerland

The authors should include some details about how they have reprocessed the gene expression datasets that are a part of BgeeDB. At the moment, it is rather unclear how this was achieved. I assume that the authors have an automated pipeline in place but it would be beneficial for readers to know how this was done.

There is now a complete and updated documentation for the Bgee pipeline: https://github.com/BgeeDB/bgee_pipeline
We have included this information in the manuscript, as well as a brief outline of the analyses we perform, see "Introduction" section.

The authors state that "TopAnat allows for discovery of tissues where a set of genes is preferentially expressed". Is TopAnat the only tool that offers such a functionality? A brief background of similar tools that are currently available will be useful for the readers.

We have added a paragraph describing similar tools, see end of the "Introduction" section.

I was not able to run the workflow that the authors have included in the Supplementary material: [...]
At this moment, I did not try further.
The authors need to clearly state what version of BgeeDB was used to create this workflow. If something has changed, then this needs to be appropriately addressed. I tried using "release=13.2" but it did not work.

We suspect that the reviewer did not use the latest version of the package (maybe the Bioconductor release itself needs to be updated first). The reviewer could maybe uninstall the BgeeDB package and rerun the following steps:

source("https://bioconductor.org/biocLite.R")
biocLite("BgeeDB")
sessionInfo()

With a package version >= 2.6.2, the errors should disappear. The R, Bioconductor, and BgeeDB package version requirements are listed at the beginning of the "Methods" section. If the problem persists, could the reviewer post the sessionInfo() results?

Of note, the "release" argument is used to specify a particular Bgee release, but this is independent of the package version.

I did manage to run an enrichment test for anatomical terms though with some tweaking
Again an error message
bgee_topanat <- loadTopAnatData(species="Danio_rerio")
Error in loadTopAnatData(species = "Danio_rerio") :
Problem: the specified speciesId is not among the list of species in Bgee.
This works though
myTopAnatData <- loadTopAnatData(species="7955")
####

Again, this should be solved by updating to the last BgeeDB version

---

The rest of the work-flow went smoothly and I was able to get a list of anatomical structures sorted by their p-value
[...]
It would be useful if the authors could include a feature that allows the TopAnat method to print the 41 genes which represent the paired limb/fin bud. At some point, the users might want to revisit their gene lists and tag their genes based on the different anatomical structures. Other tools that perform Enrichment tests, for example Enrichr, have this feature and this is extremely useful, in my opinion.

This is a good point. It is possible to cross the *geneList* vector with the expression mapping present in the *myTopAnatData* object. Another approach is to use functions that are inherited from the *topGO* package. For the “paired limb/fin bud” term:

```
myTerm <- "UBERON:0004357"
termStat(myTopAnatObject, myTerm)
# 198 genes mapped to this term for Bgee 14.0 and Ensembl 84
genesInTerm(myTopAnatObject, myTerm)
# 48 significant genes mapped to this term for Bgee 14.0 and Ensembl 84
annotated <- genesInTerm(myTopAnatObject, myTerm)[["UBERON:0004357"]]
annotated[annotated %in% sigGenes(myTopAnatObject)]
```

We have added this example at the end of the "Anatomical expression enrichment analysis" section.

Competing Interests: No competing interests were disclosed.