Check for updates

SOFTWARE TOOL ARTICLE

BgeeDB, an R package for retrieval of curated expression datasets and for gene list expression localization enrichment tests [version 1; peer review: 1 approved, 1 approved with reservations, 1 not approved]

Andrea Komljenovic[1,2*], Julien Roux[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

[*] Equal contributors

v1 **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 reprocessed 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: http://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

Referee Status: ✓ ? ✓

	Invited Referees		
	1	**2**	**3**
REVISED **version 2** published 07 Aug 2018			✓ report
version 1 published 23 Nov 2016	✓ report	? report	✗ report

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

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

3 **Leonardo Collado-Torres**, Lieber Institute for Brain Development, 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)

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, Robinson-Rechavi M and Bastian FB. **BgeeDB, an R package for retrieval of curated expression datasets and for gene list expression localization enrichment tests [version 1; peer review: 1 approved, 1 approved with reservations, 1 not approved]** F1000Research 2016, **5**:2748 (https://doi.org/10.12688/f1000research.9973.1)

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

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 access to gene expression datasets programmatically and directly in 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 44,177 RNA array assay experiments stored in ArrayExpress with processed data available as of October 2016, 7,520 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 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 http://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 as supplementary data for publications.

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, a Bioconductor package was recently released to access the Expression Atlas, which includes a selection of microarray and RNA-seq datasets from ArrayExpress that are re-annotated and reprocessed[16,17]. Similarly, the recount Bioconductor package provides access to a dataset of 2,040 reanalyzed human RNA-seq samples from SRA (see https://jhubiostatistics.shinyapps.io/recount/)[18–20].

The Bgee database (http://bgee.org/)[21] is another added-value database, which currently offers access to reprocessed gene expression datasets from 17 animal species. Bgee aims to compare 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). 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. Release 13 of Bgee includes 526 RNA-seq libraries, 12,736 Affymetrix chips, 349,613 results from 46,619 *in situ* hybridization experiments and 3,185 EST libraries. Release 14 of Bgee is in preparation and will notably include 5,746 RNA-seq libraries from 29 animal species, including 4,860 human libraries from the GTEx project[22,23].

Until recently the Bgee database lacked programmatic access to data through an R package, a shortcoming that we have addressed with the release of the BgeeDB Bioconductor package, available at http://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, in BgeeDB we introduce 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[24,25], modified to use Bgee data (A. Alexa, personal communication). TopAnat is similar to the widely used Gene Ontology enrichment test[26–28]. But in our case, the enrichment test is applied to terms from an anatomical ontology, mapped to genes by expression patterns. As a result, TopAnat allows for discovery of tissues where a set of genes is preferentially expressed. This feature is available as a web-tool at http://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.

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

Methods

Requirements

- R >= 3.3
- Bioconductor >= 3.4
- BgeeDB package version >= 2.0.0
- Working internet connection

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 for offline work.

Microarray dataset retrieval. In the following example, we look for a microarray dataset in mouse (*Mus musculus*), spanning multiple early developmental stages, including zygote. At the time of publication the latest Bgee release is 13.2, so if one needs to strictly reproduce the output of the code below in the future, the `release="13.2"` argument needs to be added when creating the Bgee object (see Supplementary file S1 and Supplementary file S2).

```
# specify species and data type
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)
str(annotation.bgee.mouse.affymetrix)
```

This creates a list of two data frames, one including the annotation of experiments, and one including the annotation of each individual sample, i.e., hybridized microarray chip. For mouse, there are 694 Affymetrix experiments and 6,077 samples available in Bgee release 13. Anatomical structures and developmental stages are annotated using the Uberon ontology[29,30]. 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(1,6)])
```

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[31], includes samples from more developmental stages than the other two, so we use this in the next steps. For this experiment, raw data were available from ArrayExpress, so samples were fully normalized with gcRMA[32] version 2.40.0 through 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, although the developmental stages 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 9,017 probesets on the microarray (column “Probeset.ID”), their mapping to Ensembl gene IDs[33] (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 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.

```
# specify species and data type
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"), 1:6]
```

Accessions `GSE41637`[34] and `GSE30352`[35] both include biological replicates for brain and liver. We focus on `GSE41637` for the next steps since it includes three replicates of each tissue, vs. only two for GSE30352. We 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 the RPKM unit (reads per kilobase per million reads), using the option `stats="rpkm"`. In the next Bgee release (release 14), it will be possible to obtain expression levels in the TPM unit (transcript per million)[36,37] from pseudo-mapping of reads computed in Bgee using the Kallisto software[38].

Presence/absence calls retrieval. It is often difficult to compare expression levels across species[39], and even within species, across datasets generated by different experimenters or laboratories[40–42]. 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 triplet "gene / tissue / developmental stage".

A powerful use of these expression calls is the anatomical expression enrichment test "TopAnat". TopAnat uses a similar approach to Gene Ontology enrichment tests[26], but genes are associated to the anatomical structures where they display expression, instead of to their functional classification. These tests allow detecting 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.; unpublished report). 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[43,44].

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[45,46] (version 2.34.0 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)[2:5])]

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

# 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.

Differential expression analysis. Below, we detail a differential expression analysis, with the package `edgeR`[47,48] (version 3.16.1 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.

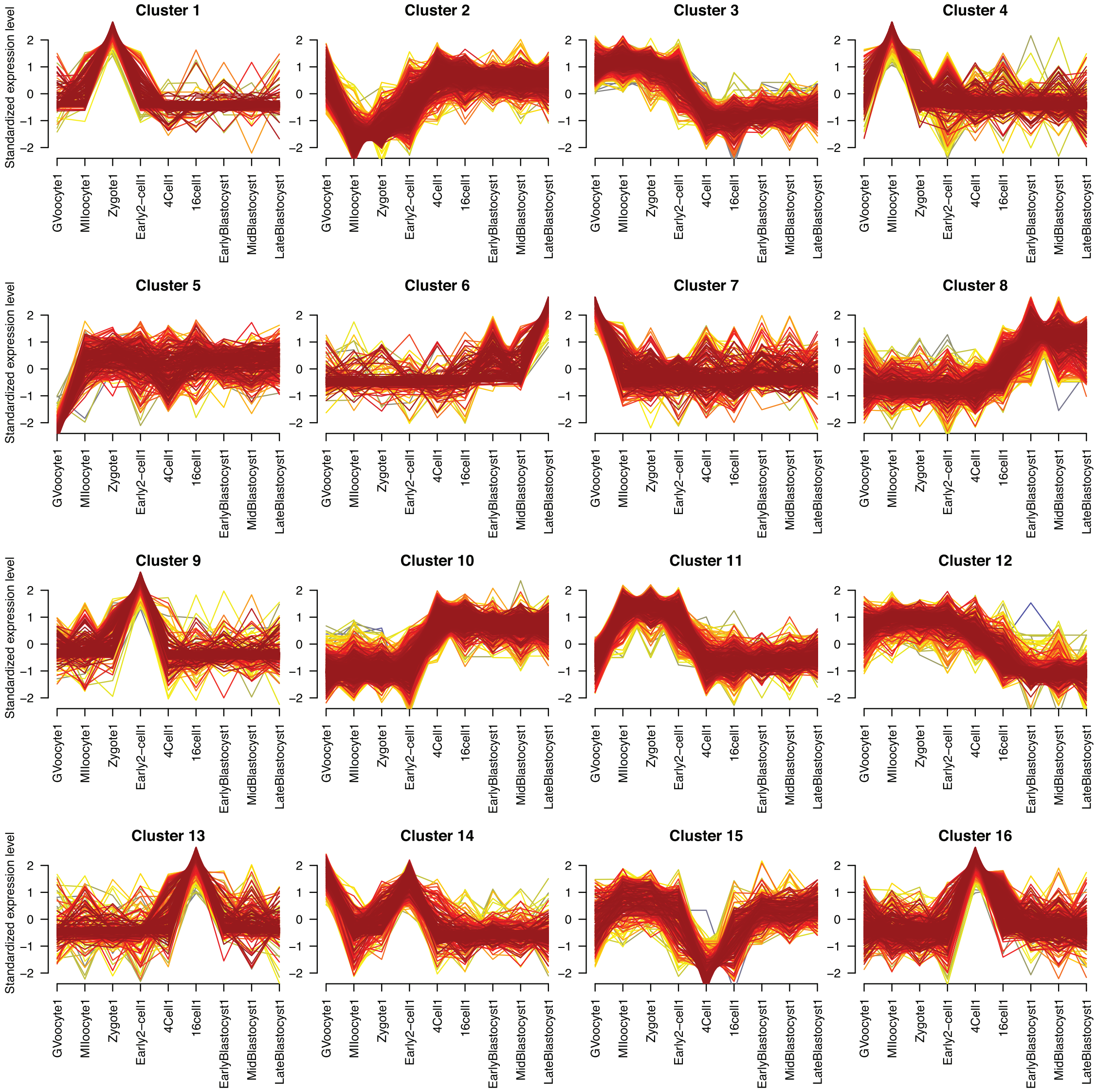

Figure 1. Standardized expression levels of 16 groups of microarray probesets, clustered according to their expression during mouse early development. The x-axis displays sample names (column "Chip.ID" of the `data.E.MEXP.51` object).

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 (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 high and low quality are used, which can be changed with the `confidence` argument of the `loadTopAnatData()` function. 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

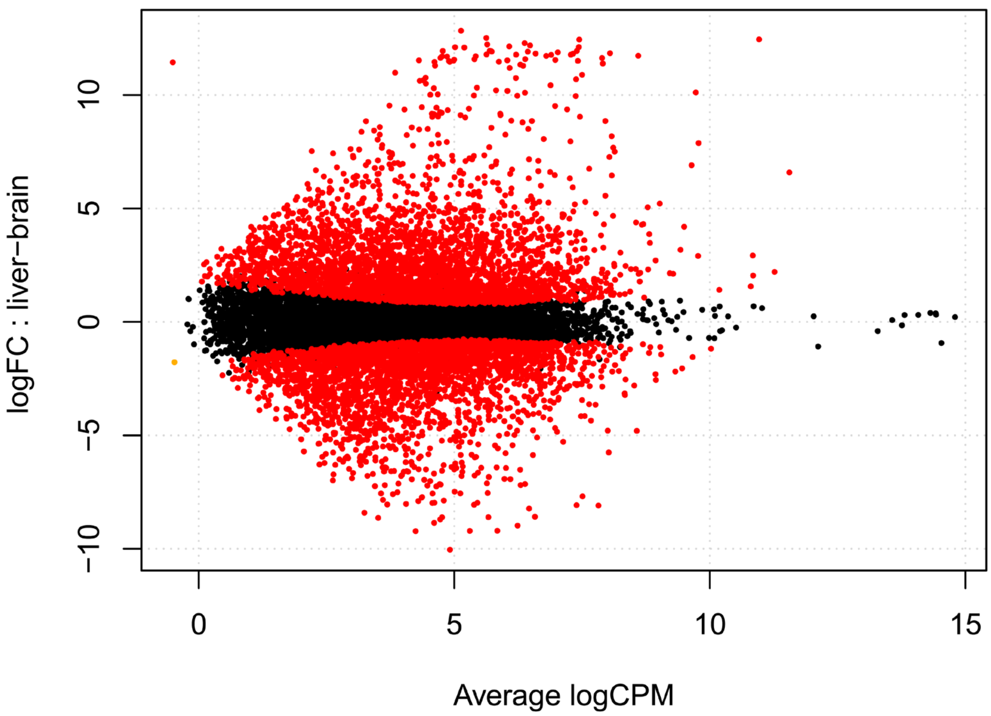

Figure 2. Mean-average (MA) plot of differential gene expression between brain and liver in macaque based on RNA-seq data. Significantly differentially expressed genes (FDR < 1%) are highlighted in red.

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.

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

```
bgee.topanat <- Bgee$new(species="Danio_rerio")
myTopAnatData <- loadTopAnatData(bgee.topanat)
str(myTopAnatData)
```

We 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[49] and integrated into the Ensembl database[50]. We thus retrieve the targeted genes using the `biomaRt`[51] Bioconductor package (version 2.30.0 for this paper).

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

# zebrafish data in Ensembl 85 (stable link)
ensembl <- useMart("ENSEMBL_MART_ENSEMBL",
dataset="drerio_gene_ensembl", host="jul2016.archive.ensembl.org")

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

```
# select phenotypes related to pectoral fin
myPhenotypes <- grep("pectoral fin", unique(genesToPhenotypes$phenotype_description), value=T)

# select the genes annotated to select phenotypes
myGenes <- unique(genesToPhenotypes$ensembl_gene_id[genesToPhenotypes$phenotype_description
%in% myPhenotypes])
```

This gives a list of 150 zebrafish genes implicated in the development and function of pectoral fin. The next step of the analysis relies on the `topGO` Bioconductor package. We 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[52]. Here we choose as background all zebrafish Ensembl genes with an annotated phenotype from ZFIN.

```
# prepare the gene list vector
geneList <- factor(as.integer(unique(genesToPhenotypes$ensembl_gene_id) %in% myGenes))
names(geneList) <- unique(genesToPhenotypes$ensembl_gene_id)
summary(geneList)

# prepare the topAnat object based on topGO
myTopAnatObject <- topAnat(myTopAnatData, geneList)
```

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 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[25]. Here we use a Fisher test, coupled with the “weight” decorrelation algorithm.

```
results <- runTest(myTopAnatObject, algorithm='weight', statistic='fisher')
```

Finally, we implement 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[53]) 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
tableOver <- makeTable(myTopAnatData, myTopAnatObject, results, cutoff=0.01)
```

The 22 anatomical structures displaying a significant enrichment at a FDR threshold of 1% are show in Table 1. The first term is “paired limb/fin bud”, and the second “pectoral fin”. Other terms in the list, especially those with high enrichment folds, are clearly related to pectoral fins (e.g., “pectoral appendage cartilage tissue”), substructures of fins (e.g., “fin bone”), or located next to them (e.g., “ceratohyal cartilage”). This analysis shows that genes with phenotypic effects on pectoral fins are specifically expressed in or next to these structures. More generally, it proves the pertinence of TopAnat analysis for the characterization of lists of genes.

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:0004357	paired limb/fin bud	144	41	7.15	5.7	1.6E-22	1.4E-19
UBERON:0000151	pectoral fin	420	70	20.85	3.4	1.0E-18	4.6E-16
UBERON:2000040	median fin fold	51	18	2.53	7.1	7.2E-12	2.1E-09
UBERON:0003051	ear vesicle	304	41	15.09	2.7	3.1E-10	7.0E-08
UBERON:0005729	pectoral appendage field	16	10	0.79	12.7	4.0E-10	7.1E-08
UBERON:0007390	pectoral appendage cartilage tissue	17	9	0.84	10.7	2.4E-08	3.6E-06
UBERON:0011610	ceratohyal cartilage	29	11	1.44	7.6	4.8E-08	6.1E-06
UBERON:0004376	fin bone	28	9	1.39	6.5	4.3E-06	4.8E-04
UBERON:0003351	pharyngeal epithelium	70	14	3.48	4.0	4.9E-06	4.8E-04
UBERON:0002513	endochondral bone	37	10	1.84	5.4	7.1E-06	6.3E-04
UBERON:0008001	irregular bone	30	9	1.49	6.0	8.2E-06	6.6E-04
UBERON:0008907	dermal bone	46	11	2.28	4.8	8.9E-06	6.6E-04
UBERON:0002539	pharyngeal arch	518	61	25.72	2.4	1.1E-05	7.1E-04
UBERON:0000089	hypoblast (generic)	85	15	4.22	3.6	1.1E-05	7.1E-04
UBERON:0002541	germ ring	94	15	4.67	3.2	3.9E-05	2.3E-03
UBERON:0001003	skin epidermis	109	16	5.41	3.0	6.2E-05	3.5E-03
UBERON:0004375	bone of free limb or fin	23	7	1.14	6.1	8.0E-05	4.2E-03
UBERON:0000165	mouth	97	21	4.82	4.4	1.0E-04	4.8E-03
UBERON:0011152	dorsal hyoid arch skeleton	24	7	1.19	5.9	1.1E-04	4.8E-03
UBERON:0000925	endoderm	126	17	6.26	2.7	1.1E-04	4.8E-03
UBERON:0003128	cranium	283	31	14.05	2.2	1.3E-04	5.5E-03
UBERON:0010312	immature eye	447	41	22.19	1.8	1.9E-04	7.7E-03

Conclusion

In summary, the BgeeDB package serves as a bridge between data from the Bgee database and the R/Bioconductor environment, facilitating 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 that makes 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 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.163768[54]

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. 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

R markdown file including code from the paper.

Click here to access the data.

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. Leinonen R, Sugawara H, Shumway M, *et al.*: **The sequence read archive.** *Nucleic Acids Res.* 2011; **39**(Database issue): D19–21.
PubMed Abstract | Publisher Full Text | Free Full Text
13. **BrainStars Bioconductor package.**
Reference Source
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.**
Reference Source

16. **ExpressionAtlas Bioconductor package.**
Reference Source

17. 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

18. 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

19. 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

20. **recount Bioconductor package.**
Reference Source

21. Bastian F, Parmentier G, Roux J, *et al.*: **Bgee: Integrating and Comparing Heterogeneous Transcriptome Data Among Species.** *Data Integr Life Sci.* 2008; **5109**: 124–31.
Publisher Full Text

22. 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

23. 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

24. 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

25. **topGO Bioconductor package.**
Reference Source

26. Rhee YS, 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

27. 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

28. Dessimoz C, Škunca N editors: **The Gene Ontology Handbook.** Humana Press; 2017; XII, 305.
Publisher Full Text

29. 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

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. 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

32. 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

33. Yates A, Akanni W, Amode MR, *et al.*: **Ensembl 2016.** *Nucleic Acids Res.* 2016; **44**(D1): D710–6.
PubMed Abstract | Publisher Full Text | Free Full Text

34. 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

35. 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

36. 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

37. 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

38. 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

39. 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

40. 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

41. 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

42. 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

43. 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

44. 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

45. 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

46. **Mfuzz Bioconductor package**.
Reference Source

47. 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

48. **eedgeR Bioconductor package**.
Reference Source

49. 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

50. Spudich GM, Fernández-Suárez XM: **Disease and Phenotype Data at Ensembl.** *Curr Protoc Hum Genet.* 2011; **Chapter 6**: Unit 6.11.
PubMed Abstract | Publisher Full Text | Free Full Text

51. **biomaRt Bioconductor package**.
Reference Source

52. 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

53. 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

54. Komljenovic A, Roux J, Robinson-Rechavi M, *et al.*: **BgeeDB/BgeeDB_R: Bgee R package release 2.0.0.** *Zenodo.* 2016.
Data Source

Open Peer Review

Current Referee Status: ✓ ? ✗

Version 1

Referee 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, Laudet V, Robinson-Rechavi M: 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, Taub M, Hansen K, Jaffe A, Langmead B, Leek J: 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 have read this submission. I 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, Swiss Institute of Bioinformatics - University of 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.

Referee 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 have read this submission. I 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, Swiss Institute of Bioinformatics - University of 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.

Referee Report 07 December 2016

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

✓ **Virag Sharma** [1,2]
[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, Duan Q, Wang Z, Koplev S, Jenkins SL, Jagodnik KM, Lachmann A, McDermott MG, Monteiro CD, Gundersen GW, Ma'ayan A: 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 have read this submission. I 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, Swiss Institute of Bioinformatics - University of 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.