SOFTWARE TOOL ARTICLE

Easy and efficient ensemble gene set testing with EGSEA [version 1; peer review: 1 approved, 3 approved with reservations]

Monther Alhamdoosh ᵃ¹, Charity W. Law ᵃ²,³, Luyi Tian ᵃ²,³, Julie M. Sheridan ᵃ²,⁴, Milica Ng ᵃ¹, Matthew E. Ritchie ᵃ²,³,⁵

¹CSL Limited, Bio21 Institute, Parkville, Victoria, Australia
²Department of Medical Biology, The University of Melbourne, Parkville, Victoria, Australia
³Molecular Medicine Division, The Walter and Eliza Hall Institute of Medical Research, Parkville, Victoria, Australia
⁴Molecular Genetics of Cancer Division, The Walter and Eliza Hall Institute of Medical Research, Parkville, Victoria, Australia
⁵School of Mathematics and Statistics, The University of Melbourne, Parkville, Victoria, Australia

Abstract
Gene set enrichment analysis is a popular approach for prioritising the biological processes perturbed in genomic datasets. The Bioconductor project hosts over 80 software packages capable of gene set analysis. Most of these packages search for enriched signatures amongst differentially regulated genes to reveal higher level biological themes that may be missed when focusing only on evidence from individual genes. With so many different methods on offer, choosing the best algorithm and visualization approach can be challenging. The EGSEA package solves this problem by combining results from up to 12 prominent gene set testing algorithms to obtain a consensus ranking of biologically relevant results. This workflow demonstrates how EGSEA can extend limma-based differential expression analyses for RNA-seq and microarray data using experiments that profile 3 distinct cell populations important for studying the origins of breast cancer. Following data normalization and set-up of an appropriate linear model for differential expression analysis, EGSEA builds gene signature specific indexes that link a wide range of mouse or human gene set collections obtained from MSigDB, GeneSetDB and KEGG to the gene expression data being investigated. EGSEA is then configured and the ensemble enrichment analysis run, returning an object that can be queried using several S4 methods for ranking gene sets and visualizing results via heatmaps, KEGG pathway views, GO graphs, scatter plots and bar plots. Finally, an HTML report that combines these displays can fast-track the sharing of results with collaborators, and thus expedite downstream biological validation. EGSEA is simple to use and can be easily integrated with existing gene expression analysis pipelines for both human and mouse data.

Keywords
gene expression, RNA-sequencing, microarrays, gene set enrichment, Bioconductor

Open Peer Review

Referee Status:  ?  ?  ?  ✔

Invited Referees

version 1

1 Open Peer Review


1 Robert Castelo ᵃ¹, Universitat Pompeu Fabra, Spain
2 Jenny Drnevich, University of Illinois at Urbana-Champaign, USA
3 Weijun Luo, (University of North Carolina at Charlotte), USA
4 Pekka Kohonen, Karolinska Institutet, Sweden
Roland Grafström, Karolinska Institutet, Sweden

Any reports and responses or comments on the article can be found at the end of the article.
Introduction

Gene set enrichment analysis allows researchers to efficiently extract biological insights from long lists of differentially expressed genes by interrogating them at a systems level. In recent years, there has been a proliferation of gene set enrichment (GSE) analysis methods released through the Bioconductor project to together with a steady increase in the number of gene set collections available through online databases such as MSigDB, GeneSetDB and KEGG. In an effort to unify these computational methods and knowledge-bases, the EGSEA R/Bioconductor package was developed. EGSEA, which stands for Ensemble of Gene Set Enrichment Analyses, combines the results from multiple algorithms to arrive at a consensus gene set ranking to identify biological themes and pathways perturbed in an experiment. EGSEA calculates seven statistics to combine the individual gene set statistics of base GSE methods to rank biologically relevant gene sets. The current version of the EGSEA package utilizes the analysis results of up to twelve prominent GSE algorithms that include: ora, globaltest, plage, safe, zscore, gage, ssgsea, padog, gsve, camera, roast and fry. The ora, gage, camera and gsve methods depend on a competitive null hypothesis which assumes the genes in a set do not have a stronger association with the experimental condition compared to randomly chosen genes outside the set. The remaining eight methods are based on a self-contained null hypothesis that only considers genes within a set and again assumes that they have no association with the experimental condition.

EGSEA provides access to a diverse range of gene signature collections through the EGSEAdata package that includes more than 25,000 gene sets for human and mouse organised according to their database sources (Table 1). For example, MSigDB includes a number of collections (Hallmark (h) and c1–c7) that explore different biological themes ranging from very broad (h, c2, c5) through to more specialised ones focusing on cancer (c4, c6) and immunology (c7). The other main sources are GeneSetDB and KEGG which have similar collections focusing on different biological characteristics (Table 1). The choice of collection/s in any given analysis should of course be guided by the biological question of interest. The MSigDB c2 and c5 collections are the most widely used in our own analysis practice, spanning a wide range of biological processes and can often reveal new biological insights when applied to a given dataset.

The purpose of this article is to demonstrate the gene set testing workflow available in EGSEA on both RNA-seq and microarray data. Each analysis involves four major steps that are summarized in Figure 1: (1) selecting appropriate gene set collections for analysis and building an index that maps between the members of each set and the expression matrix; (2) choosing the base GSE methods to combine and the ranking options; (3) running the EGSEA test and (4) reporting results in various ways to share with collaborators. The EGSEA functions involved in each of these steps are introduced with code examples to demonstrate how they can be deployed as part of a limma differential expression analysis to help with the interpretation of results.

<table>
<thead>
<tr>
<th>Database</th>
<th>Collection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSigDB</td>
<td>h Hallmarks</td>
<td>Gene sets representing well-defined biological states or processes that have coherent expression.</td>
</tr>
<tr>
<td></td>
<td>c1 Positional</td>
<td>Gene sets by chromosome and cytogenetic band.</td>
</tr>
<tr>
<td></td>
<td>c2 Curated</td>
<td>Gene sets obtained from a variety of sources, including online pathway databases and the biomedical literature.</td>
</tr>
<tr>
<td></td>
<td>c3 Motif</td>
<td>Gene sets of potential targets regulated by transcription factors or microRNAs.</td>
</tr>
<tr>
<td></td>
<td>c4 Computational</td>
<td>Gene sets defined computationally by mining large collections of cancer-oriented microarray data.</td>
</tr>
<tr>
<td></td>
<td>c5 GO</td>
<td>Gene sets annotated by Gene Ontology (GO) terms.</td>
</tr>
<tr>
<td></td>
<td>c6 Oncogenic</td>
<td>Gene sets of the major cellular pathways disrupted in cancer.</td>
</tr>
<tr>
<td></td>
<td>c7 Immunologic</td>
<td>Gene sets representing different cell types and stimulations relevant to the immune system.</td>
</tr>
<tr>
<td>KEGG</td>
<td>Signalling</td>
<td>Gene sets obtained from the KEGG database.</td>
</tr>
<tr>
<td></td>
<td>Disease Metabolic</td>
<td></td>
</tr>
<tr>
<td>GeneSetDB</td>
<td>Pathway Disease</td>
<td>Gene sets obtained from various online databases.</td>
</tr>
<tr>
<td></td>
<td>Drug Regulation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO Terms</td>
<td></td>
</tr>
</tbody>
</table>
Gene expression profiling of the mouse mammary gland

The first experiment analysed in this workflow is an RNA-seq dataset from Sheridan et al. (2015) that consists of 3 cell populations (Basal, Luminal Progenitor (LP) and Mature Luminal (ML)) sorted from the mammary glands of female virgin mice. Triplicate RNA samples from each population were obtained in 3 batches and sequenced on an Illumina HiSeq 2000 using a 100 base-pair single-ended protocol. Raw sequence reads from the fastq files were aligned to the mouse reference genome (mm10) using the Rsubread package. Next, gene-level counts were obtained using featureCounts based on Rsubread’s built-in mm10 RefSeq-based annotation. The raw data along with further information on experimental design and sample preparation can be downloaded from the Gene Expression Omnibus (GEO, www.ncbi.nlm.nih.gov/geo) using GEO Series accession number GSE63310 and will be preprocessed according to the RNA-seq workflow published by Law et al. (2016).

The second experiment analysed in this workflow comes from Lim et al. (2010) and is the microarray equivalent of the RNA-seq dataset mentioned above. The same 3 populations (Basal (also referred to as “MaSC-enriched”), LP and ML) were sorted from mouse mammary glands via flow cytometry. Total RNA from 5 replicates of each cell population were hybridised onto 3 Illumina MouseWG-6 v2 BeadChips. The intensity files and chip annotation file available in Illumina’s proprietary formats (IDAT and BGX respectively) can be downloaded from http://bioinf.wehi.edu.au/EGSEA/arraydata.zip. The raw data from this experiment is also available from GEO under Series accession number GSE19446.

Analysis of RNA-seq data with EGSEA

Our RNA-seq analysis follows on directly from the workflow of Law et al. (2016) which performs a differential gene expression analysis on this data set using the Bioconductor packages edgeR, limma and Glimma with gene annotation from the Mus.musculus package. The limma package offers a well-developed suite of statistical methods for dealing with differential expression for both microarray and RNA-seq datasets and will be used in the analyses of both datasets presented in this workflow.

Reading, preprocessing and normalisation of RNA-seq data

To get started with this analysis, download the R data file from http://bioinf.wehi.edu.au/EGSEA/mam.rnaseq.rdata. The code below loads the preprocessed count matrix from Law et al. (2016), performs TMM normalisation on the
raw counts, and calculates voom weights for use in comparisons of gene expression between Basal and LP, Basal and ML, and LP and ML populations.

```r
> library(limma)
> library(edgeR)
> load("mam.rnaseq.rdata")
> names(mam.rnaseq.data)
[1] "samples" "counts" "genes"
> dim(mam.rnaseq.data)
[1] 14165 9
> x = calcNormFactors(mam.rnaseq.data, method = "TMM")
> design = model.matrix(~0+x$samples$group+x$samples$lane)
> colnames(design) = gsub("x\$samples\$group", "", colnames(design))
> colnames(design) = gsub("x\$samples\$lane", "", colnames(design))
> head(design)
Basal LP ML L006 L008
1 0 1 0 0 0
2 0 0 1 0 0
3 1 0 0 0 0
4 1 0 0 1 0
5 0 0 1 1 0
6 0 1 0 1 0
> contr.matrix = makeContrasts(
+ BasalvsLP = Basal-LP,
+ BasalvsML = Basal - ML,
+ LPvsML = LP - ML,
+ levels = colnames(design))
> head(contr.matrix)
Contrasts
Levels  BasalvsLP BasalvsML LPvsML
Basal    1        1       0
LP       -1        0       1
ML       0       -1      -1
L006     0        0       0
L008     0        0       0
```

The `voom` function\(^b\) from the `limma` package converts counts to log-counts-per-million (log-CPM) and calculates observation-level precision weights. The `voom` object (`v`) contains normalized log-CPM values and gene information used by all of the methods in the EGSEA analysis below. The precision weights stored within `v` are also used by the `camera`, `roast` and `fry` gene set testing methods.

```r
> v = voom(x, design, plot=FALSE)
> names(v)
[1] "genes" "targets" "E" "weights" "design"
```

For further information on preprocessing see Law et al. (2016), as a detailed explanation of these steps is beyond the scope of this article.

**Gene set testing**
The EGSEA algorithm makes use of the `voom` object (`v`), a design matrix (`design`) and an optional contrasts matrix (`contr.matrix`). The design matrix describes how the samples in the experiment relate to the coefficients estimated by the linear model\(^c\). The contrasts matrix then compares two or more of these coefficients to allow relative assessment of differential expression. Base methods that utilize linear models such as those from `limma` and GSA (gsva, plight, zscore and ssgsea) make use of the design and contrasts matrices directly. For methods that do not support linear models, these two matrices are used to extract the group information for each comparison.
1. Exploring, selecting and indexing gene set collections

The package **EGSEAdata** includes more than 25,000 gene sets organized in collections depending on their database sources. Summary information about the gene set collections available in **EGSEAdata** can be displayed as follows:

```r
> library(EGSEAdata)
> egsea.data("mouse")
The following databases are available in EGSEAdata for Mus musculus:

Database name: KEGG Pathways
Version: NA
Download/update date: 07 March 2017
Data source: gage::kegg.gsets()
Supported species: human, mouse, rat
Gene set collections: Signaling, Metabolism, Disease
Related data objects: kegg.pathways
Number of gene sets in each collection for Mus musculus:
  Signaling: 132
  Metabolism: 89
  Disease: 67

Database name: Molecular Signatures Database (MSigDB)
Version: 5.2
Download/update date: 07 March 2017
Data source: http://software.broadinstitute.org/gsea
Supported species: human, mouse
Gene set collections: h, c1, c2, c3, c4, c5, c6, c7
Related data objects: msigdb, Mm.H, Mm.c2, Mm.c3, Mm.c4, Mm.c5, Mm.c6, Mm.c7
Number of gene sets in each collection for Mus musculus:
  h Hallmark Signatures: 50
  c2 Curated Gene Sets: 4729
  c3 Motif Gene Sets: 836
  c4 Computational Gene Sets: 858
  c5 GO Gene Sets: 6166
  c6 Oncogenic Signatures: 189
  c7 Immunologic Signatures: 4872

Database name: GeneSetDB Database
Version: NA
Download/update date: 15 January 2016
Data source: http://www.genesetdb.auckland.ac.nz/
Supported species: human, mouse, rat
Gene set collections: gsdbdis, gsdbgo, gsdbdrug, gsdbpath, gsdbreg
Related data objects: gsetdb.human, gsetdb.mouse, gsetdb.rat
Number of gene sets in each collection for Mus musculus:
  GeneSetDB Drug/Chemical: 6019
  GeneSetDB Disease/Phenotype: 5077
  GeneSetDB Gene Ontology: 2202
  GeneSetDB Pathway: 1444
  GeneSetDB Gene Regulation: 201

Type `?<data object name>` to get a specific information about it, e.g., `?kegg.pathways`.
As the output above suggests, users can obtain help on any of the collections using the standard R help (?) command, for instance ?msigdb$c2 will return more information on the mouse version of the c2 collection from MSigDB. The above information can be returned as a list:

```r
info = egsea.data("mouse", returnInfo = TRUE)
> names(info)
[1] "kegg" "msigdb" "gsetdb"
> info$msigdb$info$collections
[1] "h" "c1" "c2" "c3" "c4" "c5" "c6" "c7"
```

To highlight the capabilities of the EGSEA package, the KEGG pathways, c2 (curated gene sets) and c5 (Gene Ontology gene sets) collections from the MSigDB database are selected. Next, an index is built for each gene set collection using the EGSEA indexing functions to link the genes in the different gene set collections to the rows of our RNA-seq gene expression matrix. Indexes for the c2 and c5 collections from MSigDB and for the KEGG pathways are built using the buildIdx function which relies on Entrez gene IDs as its key. In the EGSEAdatagene set collections, Entrez IDs are used as they are widely adopted by the different source databases and tend to be more consistent and robust since there is one identifier per gene in a gene set. It is also relatively easy to convert other gene IDs into Entrez IDs.

```r
> library(EGSEA)
> gs.annots = buildIdx(entrezIDs=v$genes$ENTREZID, species="mouse",
+   msigdb.gsets=c("c2", "c5"), go.part = TRUE)

[1] "Loading MSigDB Gene Sets ... 
[1] "Loaded gene sets for the collection c2 ...
[1] "Indexed the collection c2 ...
[1] "Created annotation for the collection c2 ...
[1] "Loaded gene sets for the collection c5 ...
[1] "Indexed the collection c5 ...
[1] "Created annotation for the collection c5 ...
```

MSigDB c5 gene set collection has been partitioned into c5BP, c5CC, c5MF

```r
> names(gs.annots)
[1] "c2" "c5BP" "c5CC" "c5MF" "kegg"
```

To obtain additional information on the gene set collection indexes, including the total number of gene sets, the version number and date of last revision, the methods `summary`, `show` and `getSetByName` (or `getSetByID`) can be invoked on an object of class `GSCollectionIndex`, which stores all of the relevant gene set information, as follows:

```r
> class(gs.annots$c2)
[1] "GSCollectionIndex"
attr("package")
[1] "EGSEA"
> summary(gs.annots$c2)
c2 Curated Gene Sets (c2): 4726 gene sets - Version: 5.2, Update date: 07 March 2017
> show(gs.annots$c2)
An object of class "GSCollectionIndex"
Number of gene sets: 4726
Annotation columns: ID, GeneSet, BroadUrl, Description, PubMedID, NumGenes, Contributor
Total number of indexing genes: 14165
Species: Mus musculus
Collection name: c2 Curated Gene Sets
Collection unique label: c2
Database version: 5.2
Database update date: 07 March 2017
> s = getSetByName(gs.annots$c2, "SMID_BREAST_CANCER_LUMINAL_A_DN")
ID: M13072
```
GeneSet: SMID_BREAST_CANCER_LUMINAL_A_DN
BroadUrl: http://www.broadinstitute.org/gsea/msigdb/cards/SMID_BREAST_CANCER_LUMINAL_A_DN.html
Description: Genes down-regulated in the luminal A subtype of breast cancer.
PubMedID: 18451135
NumGenes: 23/24
Contributor: Jessica Robertson

> class(s)
[1] "list"
> names(s)
[1] "SMID_BREAST_CANCER_LUMINAL_A_DN"
> names(s$SMID_BREAST_CANCER_LUMINAL_A_DN)
[1] "ID"          "GeneSet"     "BroadUrl"    "Description" "PubMedID"
[6] "NumGenes"    "Contributor"

Objects of class GSCollectionIndex store for each gene set the Entrez gene IDs in the slot original, the indexes in the slot idx and additional annotation for each set in the slot anno.

> slotNames(gs.annots$c2)
[1] "original" "idx"     "anno"     "featureIDs" "species"
[6] "name" "label" "version" "date"

Other EGSEA functions such as buildCustomIdx, buildGMTIdx, buildKEGGIdx, buildMSigDBIdx and buildGeneSetDBIdx can be also used to build gene set collection indexes. The functions buildCustomIdx and buildGMTIdx were written to allow users to run EGSEA on gene set collections that may have been curated within a lab or downloaded from public databases and allow use of gene identifiers other than Entrez IDs. Example databases include, ENCODE Gene Set Hub (available from https://sourceforge.net/projects/encodegenesethub/), which is a growing resource of gene sets derived from high quality ENCODE profiling experiments encompassing hundreds of DNase hypersensitivity, histone modification and transcription factor binding experiments30. Other resources include PathwayCommons (http://www.pathwaycommons.org/31 and the KEGREST32 package that provides access to up-to-date KEGG pathways across many species.

2. Configuring EGSEA
Before an EGSEA test is carried out, a few parameters need to be specified. First, a mapping between Entrez IDs and Gene Symbols is created for use by the visualization procedures. This mapping can be extracted from the genes data.frame of the voom object as follows:

> colnames(v$genes)
[1] "ENTREZID" "SYMBOL" "CHR"
> symbolsMap = v$genes[, c(1, 2)]
> colnames(symbolsMap) = c("FeatureID", "Symbols")
> symbolsMap[, "Symbols"] = as.character(symbolsMap[, "Symbols"])

Another important parameter in EGSEA is the list of base GSE methods (baseMethods in the code below), which determines the individual algorithms that are used in the ensemble testing. The supported base methods can be listed using the function egsea.base as follows:

> egsea.base()
[1] "camera" "roast"    "safe"    "gage"    "padog"    "plage"
[7] "zscore" "gsva"     "ssgsea"  "globaltest" "ora"    "fry"

The plage, zscore and ssgsea algorithms are available in the GSVA package and camera, fry and roast are implemented in the limma package30. The ora method is implemented using the phyper function from the stats package33, which estimates the hypergeometric distribution for a 2 × 2 contingency table. The remaining algorithms are implemented in Bioconductor packages of the same name. A wrapper function is provided for each individual GSE method to utilize this existing R code and create a universal interface for all methods.
Eleven base methods are selected for our EGSEA analysis: camera, safe, gage, padog, plage, zscore, gsva, ssgsea, globaltest, ora and fry. Fry is a fast approximation of roast that assumes equal gene-wise variances across samples to produce similar p-values to a roast analysis run with an infinite number of rotations, and is selected here to save time.

```r
> baseMethods = egsea.base()[-2]
> baseMethods
[1] "camera"   "safe"     "gage"     "padog"     "plage"     "zscore"
[7] "gsva"     "ssgsea"   "globaltest" "ora"      "fry"
```

Although, different combinations of base methods might produce different results, it has been found via simulation that including more methods gives better performance.

Since each base method generates different p-values, EGSEA supports six different methods from the metap package for combining individual p-values (Wilkinson is default), which can be listed as follows:

```r
> egsea.combine()
[1] "fisher"   "wilkinson" "average" "logitp"   "sump"    "sumz"
[7] "votep"    "median"
```

Finally, the sorting of EGSEA results plays an essential role in identifying relevant gene sets. Any of EGSEA’s combined scores or the rankings from individual base methods can be used for sorting the results.

```r
> egsea.sort()
[1] "p.value"  "p.adj"   "vote.rank" "avg.rank" "med.rank"
[6] "min.pvalue" "min.rank" "avg.logfc" "avg.logfc.dir" "direction"
[11] "significance" "camera"  "roast"    "safe"     "gage"
[16] "padog"    "plage"    "zscore"   "gsva"      "ssgsea"
[21] "globaltest" "ora"     "fry"
```

Although p.adj is the default option for sorting EGSEA results for convenience, we recommend the use of either med.rank or vote.rank because they efficiently utilize the rankings of individual methods and tend to produce fewer false positives.

3. Ensemble testing with EGSEA

Next, the EGSEA analysis is performed using the egsea function that takes a voom object, a contrasts matrix, collections of gene sets and other run parameters as follows:

```r
> gsa = egsea(voom.results=v, contrasts=contr.matrix,  
+     gs.annots=gs.annots, symbolsMap=symbolsMap,  
+     baseGSEAs=baseMethods, sort.by="med.rank",  
+     num.threads = 8, report = FALSE)
```

EGSEA analysis has started
```
#------- Fri Jun 16 09:49:11 2017 ------#
Log fold changes are estimated using limma package ...
limma DE analysis is carried out ...
Number of used cores has changed to 3 in order to avoid CPU overloading.
EGSEA is running on the provided data and c2 collection
```

EGSEA is running on the provided data and c2 collection
EGSEA is running on the provided data and c5BP collection
EGSEA is running on the provided data and c5CC collection
EGSEA is running on the provided data and c5MF collection
EGSEA is running on the provided data and kegg collection
```
#------- Fri Jun 16 09:57:56 2017 ------#
EGSEA analysis took 525.812 seconds.
```

EGSEA analysis has completed
In situations where the design matrix includes an intercept, a vector of integers that specify the columns of the design matrix to test using EGSEA can be passed to the `contrasts` argument. If this parameter is NULL, all pairwise comparisons based on `v$targets$group` are created, assuming that `group` is the primary factor in the design matrix. Likewise, all the coefficients of the primary factor are used if the design matrix has an intercept.

EGSEA is implemented with parallel computing features enabled using the `parallel` package at both the method-level and experimental contrast-level. The running time of the EGSEA test depends on the base methods selected and whether report generation is enabled or not. The latter significantly increases the run time, particularly if the argument `display.top` is assigned a large value (> 20) and/or a large number of gene set collections are selected. EGSEA reporting functionality generates set-level plots for the top gene sets as well as collection-level plots.

The EGSEA package also has a function named `egsea.cnt`, that can perform the EGSEA test using an RNA-seq count matrix rather than a `voom` object, a function named `egsea.ora`, that can perform over-representation analysis with EGSEA reporting capabilities using only a vector of gene IDs, and the `egsea.ma` function that can perform EGSEA testing using a microarray expression matrix as shown later in the workflow.

**Classes used to manage the results.** The output of the functions `egsea`, `egsea.cnt`, `egsea.ora` and `egsea.ma` is an S4 object of class `EGSEAResults`. Several S4 methods can be invoked to query this object. For example, an overview of the EGSEA analysis can be displayed using the `show` method as follows:

```r
> show(gsa)
An object of class "EGSEAResults"
Total number of genes: 14165
Total number of samples: 9
Contrasts: BasalvsLP, BasalvsML, LPvsML
Base GSE methods: camera (limma:3.32.2), safe (safe:3.16.0), gage (gage:2.26.0), padog (PADOG:1.18.0), plage (GSVA:1.24.1), zscore (GSVA:1.24.1), gsva (GSVA:1.24.1), ssgsea (GSVA:1.24.1),
P-values combining method: wilkinson
Sorting statistic: med.rank
Organism: Mus musculus
HTML report generated: No
Tested gene set collections:
c2 Curated Gene Sets (c2): 4726 gene sets - Version: 5.2, Update date: 07 March 2017
c5 GO Gene Sets (BP) (c5BP): 4653 gene sets - Version: 5.2, Update date: 07 March 2017
c5 GO Gene Sets (CC) (c5CC): 584 gene sets - Version: 5.2, Update date: 07 March 2017
c5 GO Gene Sets (MF) (c5MF): 928 gene sets - Version: 5.2, Update date: 07 March 2017
KEGG Pathways (kegg): 287 gene sets - Version: NA, Update date: 07 March 2017
EGSEA version: 1.5.2
EGSEAdata version: 1.4.0
Use summary(object) and topSets(object, ...) to explore this object.
```

This command displays the number of genes and samples that were included in the analysis, the experimental contrasts, base GSE methods, the method used to combine the p-values derived from different GSE algorithms, the sorting statistic used and the size of each gene set collection. Note that the gene set collections are identified using the labels that appear in parentheses (e.g. c2) in the output of `show`.

4. Reporting EGSEA results

**Getting top ranked gene sets.** A summary of the top 10 gene sets in each collection for each contrast in addition to the EGSEA comparative analysis can be displayed using the S4 method `summary` as follows:

```r
> summary(gsa)
**** Top 10 gene sets in the c2 Curated Gene Sets collection ****
** Contrast BasalvsLP **
LIM_MAMMARY_STEM_CELL_DN | LIM_MAMMARY_LUMINAL_PROGENITOR_UP
MONTERO_THYROID_CANCER_POOR_SURVIVAL_UP | SMID_BREAST_CANCER_LUMINAL_A_DN
NAKAYAMA_SOFT_TISSUE_TUMORS_PCA2_UP | REACTOME_LATENT_INFECTION_OF_HOMO_SAPIENS...
REACTOME_TRANSFERRIN_ENDOCYTOSIS_AND_RECYCLING | FARMER_BREAST_CANCER_CLUSTER_2
KEGG_EPITHELIAL_CELL_SIGNALING... | LANDIS_BREAST_CANCER_PROGRESSION_UP
```
** Contrast BasalvsML **
LIM_MAMMARY_STEM_CELL_DN | LIM_MAMMARY_STEM_CELL_UP
LIM_MAMMARY_LUMINAL_MATURE_DN | PAPASPYRIDONOS_UNSTABLE_ATEROSCLEROTIC_PLAQUE_DN
NAKAYAMA_SOFT_TISSUE_TUMORS_PCA2_UP | LIM_MAMMARY_LUMINAL_MATURE_UP
CHARAFE_BREAST_CANCER_LUMINAL_VS_MESENCHYMAL_UP | RICKMAN_HEAD_AND_NECK_CANCER_A
YAGUE_PRETUMOR_DRUG_RESISTANCE_DN | BERTUCCI_MEDULLARY_VS_DUCTAL_BREAST_CANCER_UP

** Contrast LPvsML **
LIM_MAMMARY_LUMINAL_MATURE_UP | LIM_MAMMARY_LUMINAL_MATURE_DN
PHONG_TNF_RESPONSE_VIA_P38_PARTIAL | WOTTON_RUNX_TARGETS_UP
WANG_MLL_TARGETS | PHONG_TNF_TARGETS_DN
REACTOME_PEPIDATE_LIGAND_BINDING_RECEPTORS | CHIANG_LIVER_CANCER_SUBCLASS_CTNNB1_DN
GERHOLDR_RESPONSE_TO_TZD_UP | DURAND_STROMA_S_UP

** Comparison analysis **
LIM_MAMMARY_LUMINAL_MATURE_DN | LIM_MAMMARY_STEM_CELL_DN
NAKAYAMA_SOFT_TISSUE_TUMORS_PCA2_UP | LIM_MAMMARY_LUMINAL_MATURE_UP
COLDREN_GEFITINIB_RESISTANCE_DN | LIM_MAMMARY_STEM_CELL_UP
CHARAFE_BREAST_CANCER_LUMINAL_VS_MESENCHYMAL_UP | LIM_MAMMARY_LUMINAL_PROGENITOR_UP
BERTUCCI_MEDULLARY_VS_DUCTAL_BREAST_CANCER_UP | MIKKELSEN_IPS_WITH_HCP_H3K27ME3

**** Top 10 gene sets in the c5 GO Gene Sets (BP) collection ****
** Contrast BasalvsLP **
GO_SYNAPSE_ORGANIZATION | GO_IRON_ION_TRANSPORT
GO_CALCULUM_INDEPENDENT_CELL_CELL_ADHESION_VIA_PLASMA_MEMBRANE_CELL_ADHESION_MOLECULES | GO_PH_REDUCTION
GO_HOMOPHILIC_CELL_ADHESION_VIA_PLASMA_MEMBRANE_ADHESION_MOLECULES | GO_VACUOLAR_ACIDIFICATION
GO_FERRIC_IRON_TRANSPORT | GO_TRIVALENT_INORGANIC_CATION_TRANSPORT
GO_NEURON_PROJECTION_GUIDANCE | GO_MESONEPHROS_DEVELOPMENT

** Contrast BasalvsML **
GO_FERRIC_IRON_TRANSPORT | GO_TRIVALENT_INORGANIC_CATION_TRANSPORT
GO_IRON_TRANSPORT | GO_NEURON_PROJECTION_GUIDANCE
GO_GLIAL_CELL_MIGRATION | GO_SPINAL_CORD_DEVELOPMENT
GO_REGULATION_OF_SYNAPSE_ORGANIZATION | GO_ACTION_POTENTIAL
GO_MESONEPHROS_DEVELOPMENT | GO_NEGATIVE_REGULATION_OF_SMOOTH_MUSCLE_CELL_MIGRATION

** Contrast LPvsML **
GO_NEGATIVE_REGULATION_OF_NECROTIC_CELL_DEATH | GO_PARTURITION
GO_RESPONSE_TO_VITAMIN_D | GO_GPI_ANCHOR_METABOLIC_PROCESS
GO_REGULATION_OF_BLOOD_PRESSURE | GO_DETECTION_OF_MOLECULE_OF_BACTERIAL_ORIGIN
GO_CELL_SUBSTRATE_ADHESION | GO_PROTEIN_TRANSPORT_ALONG_MICROTUBULE
GO_INTRACILIARY_TRANSPORT | GO_CELLULAR_RESPONSE_TO_VITAMIN

** Comparison analysis **
GO_IRON_TRANSPORT | GO_FERRIC_IRON_TRANSPORT
GO_TRIVALENT_INORGANIC_CATION_TRANSPORT | GO_NEURON_PROJECTION_GUIDANCE
GO_MESONEPHROS_DEVELOPMENT | GO_SYNAPSE_ORGANIZATION
GO_REGULATION_OF_SYNAPSE_ORGANIZATION | GO_MESONEPHROS_DEVELOPMENT
GO_HOMOPHILIC_CELL_ADHESION_VIA_PLASMA_MEMBRANE_ADHESION_MOLECULES | GO_NEGATIVE_REGULATION_OF_SMOOTH_MUSCLE_CELL_MIGRATION

**** Top 10 gene sets in the c5 GO Gene Sets (CC) collection ****
** Contrast BasalvsLP **
GO_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX | GO_VACUOLAR_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX

** Contrast BasalvsML **
GO_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX | GO_VACUOLAR_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX

** Contrast LPvsML **
GO_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX | GO_VACUOLAR_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX

** Comparison analysis **
GO_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX | GO_VACUOLAR_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX
** Contrast BasalvsML **

GO_FILOPODIUM_MEMBRANE | GO_LATE_ENDOSOME_MEMBRANE
GO_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX | GO_NEUROMUSCULAR_JUNCTION
GO_COATED_MEMBRANE | GO_ACTIN_FILAMENT_BUNDLE
GO_CLATHRIN_COAT | GO_AP_TYPE_MEMBRANE_COAT_ADAPTOR_COMPLEX
GO_CLATHRIN_ADAPTOR_COMPLEX | GO_CONTRACTILE_FIBER

** Contrast LPvsML **

GO_CILIARY_TRANSITION_ZONE | GO_TCTN_B9D_COMPLEX
GO_NUCLEAR_NUCLEOSOME | GO_INTRINSIC_COMPONENT_OF_ORGANELLE_MEMBRANE
GO_ENDOPLASMIC_RETICULUM_QUALITY_CONTROL_COMPARTMENT | GO_KERATIN_FILAMENT
GO_PROTEASOME_COMPLEX | GO_CILIARY_BASEL_BODY
GO_PROTEASOME_CORE_COMPLEX | GO_CORNIFIED_ENVELOPE

** Comparison analysis **

GO_PROTON_TRANSPORTING_V_TYPE_ATPASE_COMPLEX | GO_ACTIN_FILAMENT_BUNDLE
GO_NEUROMUSCULAR_JUNCTION | GO_AP_TYPE_MEMBRANE_COAT_ADAPTOR_COMPLEX
GO_CONTRACTILE_FIBER | GO_INTERMEDIATE_FILAMENT
GO_LATE_ENDOSOME_MEMBRANE | GO_CLATHRIN_VESICLE_COAT
GO_ENDOPLASMIC_RETICULUM_QUALITY_CONTROL_COMPARTMENT | GO_MICROTUBULE_END

**** Top 10 gene sets in the c5 GO Gene Sets (MF) collection ****

** Contrast BasalvsLP **

GO_HYDROGEN_EXPORTING_ATPASE_ACTIVITY | GO_SIGNALING_PATTERN_RECOGNITION_RECEPTOR_ACTIVITY
GO_LIPID_TRANSPORTER_ACTIVITY | GO_TRIGLYCERIDE_LIPASE_ACTIVITY
GO_AMINE_BINDING | GO_STRUCTURAL_CONSTITUENT_OF_MUSCLE
GO_NEUROPEPTIDE_RECEPTOR_ACTIVITY | GO_WIDE_PORE_CHANNEL_ACTIVITY
GO_CATION_TRANSPORTING_ATPASE_ACTIVITY | GO_LIPASE_ACTIVITY

** Contrast BasalvsML **

GO_G_PROTEIN_COUPLED_RECEPTOR_ACTIVITY | GO_TRANSMEMBRANE_RECEPTOR_PROTEIN_KINASE_ACTIVITY
GO_STRUCTURAL_CONSTITUENT_OF_MUSCLE | GO_VOLTAGE_GATED_SODIUM_CHANNEL_ACTIVITY
GO_CORECEPTOR_ACTIVITY | GO_TRANSMEMBRANE_RECEPTOR_PROTEIN_TYROSINE_KINASE_ACTIVITY
GO_LIPID_TRANSPORTER_ACTIVITY | GO_SULFOTRANSFERASE_ACTIVITY
GO_CATION_TRANSPORTING_ATPASE_ACTIVITY | GO_PEPTIDE_RECEPTOR_ACTIVITY

** Contrast LPvsML **

GO_MANNOSE_BINDING | GO_PHOSPHORIC_DIESTER_HYDROLASE_ACTIVITY
GO_BETA_1_3_GALACTOSYLTRANSFERASE_ACTIVITY | GO_COMPLEMENT_BINDING
GO_ALDEHYDE_DEHYDROGENASE_NAD_ACTIVITY | GO_MANNOSIDASE_ACTIVITY
GO_LIGASE_ACTIVITY_FORMING_CARBON_NITROGEN_BONDS | GO_CARBOHYDRATE_PHOSPHATASE_ACTIVITY
GO_LIPASE_ACTIVITY | GO_PEPTIDE_RECEPTOR_ACTIVITY

** Comparison analysis **

GO_STRUCTURAL_CONSTITUENT_OF_MUSCLE | GO_LIPID_TRANSPORTER_ACTIVITY
GO_CATION_TRANSPORTING_ATPASE_ACTIVITY | GO_CHEMOREPELLENT_ACTIVITY
GO_HEPARAN_SULFATE_PROTEOGLYCAN_BINDING | GO_TRANSMEMBRANE_RECEPTOR_PROTEIN_TYROSINE_KINASE_ACTIVITY
GO_LIPASE_ACTIVITY | GO_PEPTIDE_RECEPTOR_ACTIVITY
GO_CORECEPTOR_ACTIVITY | GO_TRANSMEMBRANE_RECEPTOR_PROTEIN_KINASE_ACTIVITY
**** Top 10 gene sets in the KEGG Pathways collection ****
** Contrast BasalvsLP **
Collecting duct acid secretion | alpha-Linolenic acid metabolism
Synaptic vesicle cycle | Hepatitis C
Vascular smooth muscle contraction | Rheumatoid arthritis
cGMP-PKG signaling pathway | Axon guidance
Progesterone-mediated oocyte maturation | Arrhythmogenic right ventricular cardiomyopathy (ARVC)

** Contrast BasalvsML **
Collecting duct acid secretion | Synaptic vesicle cycle
Other glycan degradation | Axon guidance
Arrhythmogenic right ventricular cardiomyopathy (ARVC) | Glycerophospholipid metabolism
Lysosome | Vascular smooth muscle contraction
Protein digestion and absorption | Oxytocin signaling pathway

** Contrast LPvsML **
Glycosylphosphatidylinositol(GPI)-anchor biosynthesis | Histidine metabolism
Drug metabolism - cytochrome P450 | PI3K-Akt signaling pathway
Proteasome | Sulfur metabolism
Renin-angiotensin system | Nitrogen metabolism
Tyrosine metabolism | Systemic lupus erythematosus

** Comparison analysis **
Collecting duct acid secretion | Synaptic vesicle cycle
Vascular smooth muscle contraction | Axon guidance
Arrhythmogenic right ventricular cardiomyopathy (ARVC) | Oxytocin signaling pathway
Lysosome | Adrenergic signaling in cardiomyocytes
Linoleic acid metabolism | cGMP-PKG signaling pathway

EGSEA's comparative analysis allows researchers to estimate the significance of a gene set across multiple experimental contrasts. This analysis helps in the identification of biological processes that are perturbed in multiple experimental conditions simultaneously. This experiment is the RNA-seq equivalent of Lim et al. (2010)\textsuperscript{22}, who used Illumina microarrays to study the same cell populations (see later), so it is reassuring to observe the LIM gene signatures derived from this experiment amongst the top ranked c2 gene signatures in both the individual contrasts and comparative results.

Another way of exploring the EGSEA results is to retrieve the top ranked \(N\) sets in each collection and contrast using the method `topSets`. For example, the top 10 gene sets in the c2 collection for the comparative analysis can be retrieved as follows:

```r
> topSets(gsa, gs.label="c2", contrast = "comparison", names.only=TRUE)
```

Extracting the top gene sets of the collection
c2 Curated Gene Sets for the contrast comparison
Sorted by med.rank

[1] "LIM_MAMMARY_LUMINAL_MATURE_DN"
[2] "LIM_MAMMARY_STEM_CELL_DN"
[3] "NAKAYAMA_SOFT_TISSUE_TUMORS_PCA2_UP"
[4] "LIM_MAMMARY_LUMINAL_MATURE_UP"
[5] "COLDREN_GEFITINIB_RESISTANCE_DN"
[6] "LIM_MAMMARY_STEM_CELL_UP"
[7] "CHARAFE_BREAST_CANCER_LUMINAL_VS_MESENCHYMAL_UP"
[8] "LIM_MAMMARY_LUMINAL_PROGENITOR_UP"
[9] "BERTUCCI_MEDULLARY_VS_DUCTAL_BREAST_CANCER_DN"
[10] "MIKKELSEN_IPS_WITH_HCP_H3K27ME3"

The gene sets are ordered based on their med.rank as selected when egsea was invoked above. When the argument names.only is set to FALSE, additional information is displayed for each gene set including gene set annotation,
the EGSEA scores and the individual rankings by each base method. As expected, gene sets retrieved by EGSEA included the LIM gene sets that were derived from microarray profiles of analogous mammary cell populations (sets 1, 2, 4, 6 and 8) as well as those derived from populations with similar origin (sets 7 and 9) and behaviour or characteristics (sets 5 and 10).

Next, `topSets` can be used to search for gene sets of interest based on different EGSEA scores as well as the rankings of individual methods. For example, the ranking of the six LIM gene sets from the c2 collection can be displayed based on the `med.rank` as follows:

```
> t = topSets(gsa, contrast = "comparison", 
+             names.only=FALSE, number = Inf, verbose = FALSE)
> t[grep("LIM_", rownames(t)), c("p.adj", "Rank", "med.rank", "vote.rank")]
```

```
p.adj  Rank  med.rank  vote.rank
LIM_MAMMARY_LUMINAL_MATURE_DN 1.646053e-29    1       36         5
LIM_MAMMARY_STEM_CELL_DN           6.082053e-43    2       37         5
LIM_MAMMARY_LUMINAL_MATURE_UP         2.469061e-22    4       92         5
LIM_MAMMARY_STEM_CELL_UP              3.154132e-103    6      134         5
LIM_MAMMARY_LUMINAL_PROGENITOR_UP     3.871536e-30    8      180         5
LIM_MAMMARY_LUMINAL_PROGENITOR_DN  2.033005e-06  178      636       115
```

While five of the LIM gene sets are ranked in the top 10 by EGSEA, the values shown in the median rank (`med.rank`) column indicate that individual methods can assign much lower ranks to these sets. EGSEA’s prioritisation of these gene sets demonstrates the benefit of an ensemble approach.

Similarly, we can find the top 10 pathways in the KEGG collection from the ensemble analysis for the Basal versus LP contrast and the comparative analysis as follows:

```
> topSets(gsa, gs.label="kegg", contrast="BasalvsLP", sort.by="med.rank")
Extracting the top gene sets of the collection
KEGG Pathways for the contrast BasalvsLP
Sorted by med.rank
[1] "Collecting duct acid secretion" "alpha-Linolenic acid metabolism"
[3] "Synaptic vesicle cycle" "Hepatitis C"
[5] "Vascular smooth muscle contraction" "Rheumatoid arthritis"
[7] "cGMP-PKG signaling pathway" "Axon guidance"
[9] "Progesterone-mediated oocyte maturation" "Arrhythmogenic right ventricular cardiomyopathy (ARVC)"

> topSets(gsa, gs.label="kegg", contrast="comparison", sort.by="med.rank")
Extracting the top gene sets of the collection
KEGG Pathways for the contrast comparison
Sorted by med.rank
[1] "Collecting duct acid secretion" "Synaptic vesicle cycle"
[3] "Vascular smooth muscle contraction" "Axon guidance"
[5] "Arrhythmogenic right ventricular cardiomyopathy (ARVC)"
[7] "Lysosome" "Adrenergic signaling in cardiomyocytes"
[9] "Linoleic acid metabolism" "cGMP-PKG signaling pathway"
```

EGSEA highlights many pathways with known importance in the mammary gland such as those associated with distinct roles in lactation like basal cell contraction (Vascular smooth muscle contraction and Oxytocin signalling pathway) and milk production and secretion from luminal lineage cells (Collecting duct acid secretion, Synaptic vesicle cycle and Lysosome).

**Visualizing results at the gene set level.** Graphical representation of gene expression patterns within and between gene sets is an essential part of communicating the results of an analysis to collaborators and other researchers. EGSEA enables users to explore the elements of a gene set via a heatmap using the `plotHeatmap` method. Figure 2 shows...
examples for the LIM_MAMMARY_STEM_CELL_UP and LIM_MAMMARY_STEM_CELL_DN signatures which can be visualized across all contrasts using the code below.

```
> plotHeatmap(gsa, gene.set="LIM_MAMMARY_STEM_CELL_UP", gs.label="c2",
+         contrast = "comparison", file.name = "hm_cmp_LIM_MAMMARY_STEM_CELL_UP")
Generating heatmap for LIM_MAMMARY_STEM_CELL_UP from the collection c2 Curated Gene Sets and for the contrast comparison
> plotHeatmap(gsa, gene.set="LIM_MAMMARY_STEM_CELL_DN", gs.label="c2",
+         contrast = "comparison", file.name = "hm_cmp_LIM_MAMMARY_STEM_CELL_DN")
Generating heatmap for LIM_MAMMARY_STEM_CELL_DN from the collection c2 Curated Gene Sets and for the contrast comparison
```

When using `plotHeatmap`, the `gene.set` value must match the name returned from the `topSets` method. The rows of the heatmap represent the genes in the set and the columns represent the experimental contrasts. The heatmap colour-scale ranges from down-regulated (blue) to up-regulated (red) while the row labels (Gene symbols) are coloured in green when the genes are statistically significant in the DE analysis (i.e. FDR ≤ 0.05 in at least one contrast). Heatmaps can be generated for individual comparisons by changing the `contrast` argument of `plotHeatmap`. The `plotHeatmap` method also generates a CSV file that includes the DE analysis results from `limma::topTable` for all expressed genes in the selected gene set and for each contrast (in the case of `contrast = "comparison"`). This file can be used to create customised plots using other R/Bioconductor packages.

In addition to heatmaps, pathway maps can be generated for the KEGG gene sets using the `plotPathway` method which uses functionality from the `pathview` package. For example, the third KEGG signalling pathway retrieved for the contrast BasalvsLP is Vascular smooth muscle contraction and can be visualized as follows:

```
> plotPathway(gsa, gene.set = "Vascular smooth muscle contraction",
+         contrast = "BasalvsLP", gs.label = "kegg",
+         file.name = "Vascular_smooth_muscle_contraction")
Generating pathway map for Vascular smooth muscle contraction from the collection KEGG Pathways and for the contrast BasalvsLP
```
Pathway components are coloured based on the gene-specific log-fold-changes as calculated in the `limma` DE analysis (Figure 3). Similarly, a comparative map can be generated for a given pathway across all contrasts.

```r
> plotPathway(gsa, gene.set = "Vascular smooth muscle contraction", 
+             contrast = "comparison", gs.label = "kegg", 
+             file.name = "Vascular_smooth_muscle_contraction_cmp")
```

Generating pathway map for Vascular smooth muscle contraction from the collection KEGG Pathways and for the contrast comparison.

The comparative pathway map shows the log-fold-changes for each gene in each contrast by dividing the gene nodes on the map into multiple columns, one for each contrast (Figure 4).

**Figure 3.** Pathway map for *Vascular smooth muscle contraction* (KEGG pathway mmu04270) with log-fold-changes from the Basal vs LP contrast.
Visualizing results at the experiment level. Since EGSEA combines the results from multiple gene set testing methods, it can be interesting to compare how different base methods rank a given gene set collection for a selected contrast. The `plotMethods` command generates a multi-dimensional scaling (MDS) plot for the ranking of gene sets across all the base methods used (Figure 5). Methods that rank gene sets similarly will appear closer together in this plot and we see that certain methods consistently cluster together across different gene set collections. The clustering of methods does not necessarily follow the style of null hypothesis tested though (i.e. self-contained versus competitive).

```r
> plotMethods(gsa, gs.label = "c2", contrast = "BasalvsLP",
+   file.name = "mds_c2_BasalvsLP")
Generating MDS plot for the collection
c2 Curated Gene Sets and for the contrast BasalvsLP
> plotMethods(gsa, gs.label = "c5BP", contrast = "BasalvsLP",
+   file.name = "mds_c5_BasalvsLP")
Generating MDS plot for the collection
c5BP GO Gene Sets and for the contrast BasalvsLP
```

**Figure 4.** Pathway map for **Vascular smooth muscle contraction** (KEGG pathway mmu04270) with log-fold-changes across three experimental contrasts shown for each gene in the same order left to right that they appear in the contrasts matrix (i.e. Basal vs LP, Basal vs ML and LP vs ML).
The significance of each gene set in a given collection for a selected contrast can be visualized using EGSEA’s plotSummary method.

```r
> plotSummary(gsa, gs.label = 3, contrast = 3,
+         file.name = "summary_kegg_LPvsML")
Generating Summary plots for the collection
KEGG Pathways and for the contrast LPvsML
```

The summary plot visualizes the gene sets as bubbles based on the $-\log_{10}(p\text{-value})$ (X-axis) and the average absolute log fold-change of the set genes (Y-axis). The sets that appear towards the top-right corner of this plot are most likely to be biologically relevant. EGSEA generates two types of summary plots: the directional summary plot (Figure 6a), which colours the bubbles based on the regulation direction of the gene set (the direction of the majority of genes), and the ranking summary plot (Figure 6b), which colours the bubbles based on the gene set ranking in a given collection (according to the sort.by argument). The bubble size is based on the EGSEA significance score in the former plot and the gene set size in the latter. For example, the summary plots of the KEGG pathways for the LP vs ML contrast show few significant pathways (Figure 6). The blue colour labels on the ranking plot represents gene sets that do not appear in the top 10 gene sets that are selected based on the sort.by argument, yet their EGSEA significance scores are among the top 5 in the entire collection based on the significance score. This is used to identify gene sets with high significance scores that were not captured by the sort.by score. The gene set IDs and more information about each set can be found in the EGSEA HTML report generated later.

By default, plotSummary uses a gene set’s $p\text{.adj}$ Score for the X-axis. This behaviour can be easily modified by assigning any of the available sort.by scores into the parameter x.axis, for example, med.rank can be used to create an EGSEA summary plot (Figure 7a) as follows:

```r
> plotSummary(gsa, gs.label = 1, contrast = 3,
+         file.name = "summary_c2_LPvsML",
+         x.axis = "med.rank")
Generating Summary plots for the collection
c2 Curated Gene Sets and for the contrast LPvsML
```
The summary plot tends to become cluttered when the size of the gene set collection is very large as in Figure 7a. The parameter x.cutoff can be used to focus in on the significant gene sets rather than plotting the entire gene set collection, for example (Figure 7b):

```r
> plotSummary(gsa, gs.label = 1, contrast = 3,
+     file.name = "summary_sig_c2_LPvsML",
+     x.axis = "med.rank", x.cutoff=300)
```

Generating Summary plots for the collection c2 Curated Gene Sets and for the contrast LPvsML.

Comparative summary plots can be also generated to compare the significance of gene sets between two contrasts, for example, the comparison between Basal vs LP and Basal vs ML (Figure 8a) shows that most of the KEGG
pathways are regulated in the same direction with relatively few pathways regulated in opposite directions (purple coloured bubbles in Figure 8a). Such figures can be generated using the `plotSummary` method as follows:

```r
> plotSummary(gsa, gs.label = "kegg", contrast = c(1,2),
+       file.name = "summary_kegg_1vs2")
```

Generating Summary plots for the collection KEGG Pathways and for the comparison BasalvsLP vs BasalvsML

The `plotSummary` method has two useful parameters: (i) `use.names` that can be used to display gene set names instead of gene set IDs and (ii) `interactive` that can be used to generate an interactive version of this plot.

The c5 collection of MSigDB and the Gene Ontology collection of GeneSetDB contain Gene Ontology (GO) terms. These collections are meant to be non-redundant, containing only a small subset of the entire GO and visualizing how these terms are related to each other can be informative. EGSEA utilizes functionality from the `topGO` package to generate GO graphs for the significant biological processes (BPs), cellular compartments (CCs) and molecular functions (MFs). The `plotGOGraph` method can generate such a display (Figure 9) as follows:

```r
> plotGOGraph(gsa, gs.label="c5BP", contrast = 1, file.name="BasalvsLP-c5BP-top-")
```

Generating GO Graphs for the collection c5 GO Gene Sets (BP) and for the contrast BasalvsLP based on the med.rank

```r
> plotGOGraph(gsa, gs.label="c5CC", contrast = 1, file.name="BasalvsLP-c5CC-top-")
```

Generating GO Graphs for the collection c5 GO Gene Sets (CC) and for the contrast BasalvsLP based on the med.rank

The GO graphs are coloured based on the values of the argument `sort.by`, which in this instance was taken as `med.rank` by default since this was selected when EGSEA was invoked. The top five most significant GO terms are highlighted by default in each GO category (MF, CC or BP). More terms can be displayed by changing the value of the parameter `noSig`. However, this might generate very complicated and unresolved graphs. The colour of the nodes varies between red (most significant) and yellow (least significant). The values of the `sort.by` scoring function are scaled between 0 and 1 to generate these graphs.
The method **Biological Cellula** Processes can be a useful visualization. Figure 9. GO graphs of the top significant GO terms from the c5 gene set collection for the contrast Basal vs LP.

Another way to visualize results at the experiment level is via a summary **bar plot**. The method **plotBars** can be used to generate a bar plot for the top N gene sets in an individual collection for a particular contrast or from a comparative analysis across multiple contrasts. For example, the top 20 gene sets of the comparative analysis carried out on the c2 collection of MSigDB can be visualized in a bar plot (Figure 10) as follows:

```r
> plotBars(gsa, gs.label = "c2", contrast = "comparison", file.name="comparison-c2-bars")
```

Generating a bar plot for the collection c2 Curated Gene Sets and the contrast comparison

The colour of the bars is based on the regulation direction of the gene sets, i.e., red for up-regulated, blue for down-regulated and purple for neutral regulation (in the case of the comparative analysis on experimental contrasts that show opposite behaviours). By default, the \( -\log_2(p\_adj) \) values are plotted for the top 20 gene sets selected and ordered based on the sort.by parameter. The parameters bar.vals, number and sort.by of **plotBars** can be changed to customize the bar plot.

When changes over multiple conditions are of interest, a **summary heatmap** can be a useful visualization. The method **plotSummaryHeatmaps** generates a heatmap of the top N gene sets in the comparative analysis across all experimental conditions (Figure 11). By default, 20 gene sets are selected based on the sort.by parameter and the values plotted are the average log-fold changes at the set level for the genes regulated in the same direction as the set regulation direction, i.e. avg.logfc.dir. The parameters number, sort.by and hm.vals of the **plotSummaryHeatmaps** can be used to customize the summary heatmap. Additionally, the parameter show.vals
can be used to display the values of a specific EGSEA score on the heatmap cells. An example summary heatmap can be generated for the MSigDB c2 collection with the following code:

```r
> plotSummaryHeatmap(gsa, gs.label="c2", hm.vals = "avg.logfc.dir",
+ file.name="summary_heatmaps_c2")
Generating summary heatmap for the collection c2 Curated Gene Sets
sort.by: med.rank, hm.vals: avg.logfc.dir, show.vals:
> plotSummaryHeatmap(gsa, gs.label="kegg", hm.vals = "avg.logfc.dir",
+ file.name="summary_heatmaps_kegg")
Generating summary heatmap for the collection KEGG Pathways
sort.by: med.rank, hm.vals: avg.logfc.dir, show.vals:
```

**Figure 10.** Bar plot of the \(-\log_{10}(p\text{-value})\) of the top 20 gene sets from the comparative analysis of the c2 collection.
We find the heatmap view at both the gene set and summary level and the summary level bar plots to be useful summaries to include in publications to highlight the gene set testing results. The top differentially expressed genes from each contrast can be accessed from the `EGSEAResults` object using the `limmaTopTable` method.

```r
> t = limmaTopTable(gsa, contrast=1)
> head(t)
```

<table>
<thead>
<tr>
<th>ENTREZID</th>
<th>SYMBOL</th>
<th>CHR</th>
<th>logFC</th>
<th>AveExpr</th>
<th>t</th>
<th>P.Value</th>
<th>adj.P.Val</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>19253</td>
<td>19253</td>
<td>1</td>
<td>-5.63</td>
<td>4.13</td>
<td>-34.5</td>
<td>5.87e-10</td>
<td>9.62e-07</td>
<td>13.2</td>
</tr>
<tr>
<td>16324</td>
<td>Inhbb</td>
<td>1</td>
<td>-4.79</td>
<td>6.46</td>
<td>-33.2</td>
<td>7.99e-10</td>
<td>9.62e-07</td>
<td>13.3</td>
</tr>
<tr>
<td>53624</td>
<td>Cldn7</td>
<td>11</td>
<td>-5.51</td>
<td>6.30</td>
<td>-40.2</td>
<td>1.75e-10</td>
<td>9.62e-07</td>
<td>14.5</td>
</tr>
<tr>
<td>218518</td>
<td>Marveld2</td>
<td>13</td>
<td>-5.14</td>
<td>4.93</td>
<td>-34.8</td>
<td>1.52e-10</td>
<td>9.62e-07</td>
<td>14.7</td>
</tr>
<tr>
<td>70350</td>
<td>Baspl</td>
<td>15</td>
<td>-6.07</td>
<td>5.25</td>
<td>-34.3</td>
<td>6.22e-10</td>
<td>9.62e-07</td>
<td>13.3</td>
</tr>
</tbody>
</table>

Creating an HTML report of the results. To generate an EGSEA HTML report for this dataset, you can either set `report=TRUE` when you invoke `egsea` or use the S4 method `generateReport` as follows:

```r
> generateReport(gsa, number = 20, report.dir="./mam-rnaseq-egsea-report")
```

The EGSEA report generated for this dataset is available online at [http://bioinf.wehi.edu.au/EGSEA/mam-rnaseq-egsea-report/index.htm](http://bioinf.wehi.edu.au/EGSEA/mam-rnaseq-egsea-report/index.htm) (Figure 12). The HTML report is a convenient means of organising all of the results generated up to now, from the individual tables to the gene set level heatmaps, pathway maps and summary level plots. It can easily be shared with collaborators to allow them to explore their results more fully. Interactive
tables of results via the DT package (https://CRAN.R-project.org/package=DT) and summary plots from plotly (https://CRAN.R-project.org/package=plotly) are integrated into the report using htmlwidgets (https://CRAN.R-project.org/package=htmlwidgets) and can be added by setting interactive = TRUE in the command above. This option significantly increases both the run time and size of the final report due to the large number of gene sets in most collections.

This example completes our overview of EGSEA’s gene set testing and plotting capabilities for RNA-seq data. Readers can refer to the EGSEA vignette or individual help pages for further details on each of the above methods and classes.

**Analysis of microarray data with EGSEA**

The second dataset analysed in this workflow comes from Lim et al. (2010) and is the microarray equivalent of the RNA-seq data analysed above. Support for microarray data is a new feature in EGSEA, and in this example, we show an express route for analysis according to the steps shown in Figure 1, from selecting gene sets and building indexes, to configuring EGSEA, testing and reporting the results. First, the data must be appropriately preprocessed for an EGSEA analysis and to do this we make use of functions available in limma.
Reading, preprocessing and normalisation of microarray data

To analyse this dataset, we begin by unzipping the files downloaded from http://bioinf.wehi.edu.au/EGSEA/arraydata.zip into the current working directory. Illumina BeadArray data can be read in directly using the `readIDAT` and `readBGX` functions from the `illuminaio` package. However, a more convenient way is via the `read.idat` function in `limma` which uses these `illuminaio` functions and outputs the data as an `EListRaw` object for further processing.

```r
> library(limma)
> targets = read.delim("targets.txt", header=TRUE, sep=" ")
> data = read.idat(as.character(targets$File),
+                 bgxfile="GPL6887_MouseWG-6_V2_0_R0_11278593_A.bgx",
+                 annotation=c("Entrez_Gene_ID","Symbol", "Chromosome"))

Reading manifest file GPL6887_MouseWG-6_V2_0_R0_11278593_A.bgx ... Done
4481850214_B_Grn.idat ... Done
4481850214_C_Grn.idat ... Done
4481850214_D_Grn.idat ... Done
4481850214_F_Grn.idat ... Done
4481850187_A_Grn.idat ... Done
4481850187_B_Grn.idat ... Done
4481850187_D_Grn.idat ... Done
4481850187_E_Grn.idat ... Done
4481850187_F_Grn.idat ... Done
4466975058_A_Grn.idat ... Done
4466975058_B_Grn.idat ... Done
4466975058_C_Grn.idat ... Done
4466975058_D_Grn.idat ... Done
4466975058_E_Grn.idat ... Done
4466975058_F_Grn.idat ... Done
Finished reading data.

> data$other$Detection = detectionPValues(data)
> data$targets = targets
> colnames(data) = targets$Sample

Next the `neqc` function in `limma` is used to carry out `normexp` background correction and quantile normalisation on the raw intensity values using negative control probes. This is followed by log₂-transformation of the normalised intensity values and removal of the control probes.

```r
> data = neqc(data)
```

We then filter out probes that are consistently non-expressed or lowly expressed throughout all samples as they are uninformative in downstream analysis. Our threshold for expression requires probes to have a detection p-value of less than 0.05 in at least 5 samples (the number of samples within each group). We next remove genes without a valid Entrez ID and in cases where there are multiple probes targeting different isoforms of the same gene, select the probe with highest average expression as the representative one to use in the EGSEA analysis. This leaves 7,123 probes for further analysis.

```r
> table(targets$Celltype)
Basal  LP  ML
  5  5  5
> keep.exprs = rowSums(data$other$Detection<0.05)>=5
> table(keep.exprs)
keep.exprs
FALSE TRUE
23638 21643
> data = data[keep.exprs,]
> dim(data)
[1] 21643 15
Setting up the linear model for EGSEA testing

As before, we need to set up an appropriate linear model and contrasts matrix to look for differences between the Basal and LP, Basal and ML and LP and ML populations. A batch term is included in the linear model to account for differences in expression that are attributable to the day the experiment was run.

> head(data$genes)

<table>
<thead>
<tr>
<th>Probe_Id</th>
<th>Array_Address_Id</th>
<th>Entrez_Gene_ID</th>
<th>Symbol</th>
<th>Chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILMN_1219601</td>
<td>2030280</td>
<td>&lt;NA&gt;</td>
<td>C920011N12Rik</td>
<td>6</td>
</tr>
<tr>
<td>ILMN_1252621</td>
<td>1980164</td>
<td>101142</td>
<td>2700050P07Rik</td>
<td></td>
</tr>
<tr>
<td>ILMN_3162407</td>
<td>6220026</td>
<td>&lt;NA&gt;</td>
<td>Zfp36</td>
<td></td>
</tr>
<tr>
<td>ILMN_2514723</td>
<td>2030072</td>
<td>1110067B18Rik</td>
<td>Zfp36</td>
<td></td>
</tr>
<tr>
<td>ILMN_2692952</td>
<td>6040743</td>
<td>329831</td>
<td>4833436C18Rik</td>
<td>4</td>
</tr>
<tr>
<td>ILMN_1257952</td>
<td>7160091</td>
<td>&lt;NA&gt;</td>
<td>B930060K05Rik</td>
<td></td>
</tr>
</tbody>
</table>

> sum(is.na(data$genes$Entrez_Gene_ID))

[1] 11535

> d1 = data[!is.na(data$genes$Entrez_Gene_ID), ]

> ord = order(lmFit(d1)$Amean, decreasing=TRUE)

> ids2keep = d1$genes$Array_Address_Id[ord][!duplicated(d1$genes$Entrez_Gene_ID[ord])]

> d1 = d1[match(ids2keep, d1$genes$Array_Address_Id), ]

> expr = d1$E

> group = as.factor(d1$targets$Celltype)

> probe.annot = d1$genes[, 2:4]

> head(probe.annot)

<table>
<thead>
<tr>
<th>Array_Address_Id</th>
<th>Entrez_Gene_ID</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>39513</td>
<td>4120224</td>
<td>Rps4x</td>
</tr>
<tr>
<td>9062</td>
<td>2260576</td>
<td>Tuba1b</td>
</tr>
<tr>
<td>15308</td>
<td>5720202</td>
<td>Zfp36</td>
</tr>
<tr>
<td>39894</td>
<td>1470600</td>
<td>Atp5b</td>
</tr>
<tr>
<td>24709</td>
<td>2710477</td>
<td>Rps24</td>
</tr>
<tr>
<td>9872</td>
<td>1580471</td>
<td>Atp5g3</td>
</tr>
</tbody>
</table>

Setting up the linear model for EGSEA testing

As before, we need to set up an appropriate linear model and contrasts matrix to look for differences between the Basal and LP, Basal and ML and LP and ML populations. A batch term is included in the linear model to account for differences in expression that are attributable to the day the experiment was run.

> head(d1$targets)

<table>
<thead>
<tr>
<th>File</th>
<th>Sample</th>
<th>Celltype</th>
<th>Time</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-2</td>
<td>4481850214_B_Grn.idat</td>
<td>ML</td>
<td>At1</td>
<td>1</td>
</tr>
<tr>
<td>3-3</td>
<td>4481850214_C_Grn.idat</td>
<td>LP</td>
<td>At1</td>
<td>1</td>
</tr>
<tr>
<td>4-4</td>
<td>4481850214_D_Grn.idat</td>
<td>Basal</td>
<td>At1</td>
<td>1</td>
</tr>
<tr>
<td>6-7</td>
<td>4481850214_F_Grn.idat</td>
<td>ML</td>
<td>At2</td>
<td>1</td>
</tr>
<tr>
<td>7-8</td>
<td>4481850187_A_Grn.idat</td>
<td>LP</td>
<td>At2</td>
<td>1</td>
</tr>
<tr>
<td>8-9</td>
<td>4481850187_B_Grn.idat</td>
<td>Basal</td>
<td>At2</td>
<td>1</td>
</tr>
</tbody>
</table>

> experiment = as.character(d1$targets$Experiment)

> design = model.matrix(~0 + group + experiment)

> colnames(design) = gsub("group", ",", colnames(design))

> design

<table>
<thead>
<tr>
<th>Basal</th>
<th>LP</th>
<th>ML</th>
<th>experiment2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
9      1 0 0     0
10     0 0 1     1
11     0 1 0     1
12     1 0 0     1
13     1 0 0     1
14     0 0 1     1
15     0 1 0     1

attr(,"assign")
[1] 1 1 1 2
attr(,"contrasts")
attr(,"contrasts")$group
[1] "contr.treatment"

attr(,"contrasts")$experiment
[1] "contr.treatment"

> contr.matrix = makeContrasts(
+ BasalvsLP = Basal-LP,
+ BasalvsML = Basal-ML,
+ LPvsML = LP-ML,
+ levels = colnames(design))
> contr.matrix

<table>
<thead>
<tr>
<th>Contrasts</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BasalvsLP BasalvsML LPvsML</td>
</tr>
<tr>
<td>Basal</td>
<td>1</td>
</tr>
<tr>
<td>LP</td>
<td>-1</td>
</tr>
<tr>
<td>ML</td>
<td>0</td>
</tr>
<tr>
<td>experiment2</td>
<td>0</td>
</tr>
</tbody>
</table>

1. Creating gene set collection indexes

We next extract the mouse c2, c5 and KEGG gene signature collections from the EGSEAdata package and build indexes based on Entrez IDs that link between the genes in each signature and the rows of our expression matrix.

> library(EGSEA)
> library(EGSEAdata)
> gs.annots = buildIdx(entrezIDs=probe.annot[, 2],
+ species="mouse",
+ msgidb.gsets=c("c2", "c5"), go.part = TRUE)
[1] "Loading MSigDB Gene Sets ..."
[1] "Loaded gene sets for the collection c2 ..."
[1] "Indexed the collection c2 ..."
[1] "Created annotation for the collection c2 ..."
[1] "Loaded gene sets for the collection c5 ..."
[1] "Indexed the collection c5 ..."
[1] "Created annotation for the collection c5 ..."
MSigDB c5 gene set collection has been partitioned into c5BP, c5CC, c5MF
[1] "Building KEGG pathways annotation object ..."
> names(gs.annots)
[1] "c2" "c5BP" "c5CC" "c5MF" "kegg"

2. Configuring and 3. Testing with EGSEA

The same 11 base methods used previously in the RNA-seq analysis were selected for the ensemble testing of the microarray data using the function egsea.ma. Gene sets were again prioritised by their median rank across the 11 methods.
> baseMethods = egsea.base()[-2]
> baseMethods

[1] "camera"     "safe"       "gage"       "padog"       "plage"    "zscore"
[7] "gsva"       "ssgsea"     "globaltest" "ora"         "fry"

> gsam = egsea.ma(expr=expr, group=group,
+     probe.annot = probe.annot,
+     design = design,
+     contrasts=contr.matrix,
+     gs.annots=gs.annots,
+     baseGSEAs=baseMethods, sort.by="med.rank",
+     num.threads = 8, report = FALSE)

EGSEA analysis has started

Log fold changes are estimated using limma package ...

limma DE analysis is carried out ...

Number of used cores has changed to 3

in order to avoid CPU overloading.

EGSEA is running on the provided data and c2 collection

EGSEA is running on the provided data and c5BP collection

EGSEA is running on the provided data and c5CC collection

EGSEA is running on the provided data and c5MF collection

EGSEA is running on the provided data and kegg collection

EGSEA analysis took 365.359 seconds.

EGSEA analysis has completed

4. Reporting EGSEA results

An HTML report that includes each of the gene set level and summary level plots shown individually for the RNA-seq analysis was then created using the `generateReport` function. We complete our analysis by displaying the top ranked sets for the c2 collection from a comparative analysis across all contrasts.

> generateReport(gsam, number = 20, report.dir="/mam-ma-egsea-report")

EGSEA HTML report is being generated ...

> topSets(gsam, gs.label="c2", contrast = "comparison", names.only=TRUE, number=5)

Sorted by med.rank

[1] "LIM_MAMMARY_STEM_CELL_UP"
[2] "LIM_MAMMARY_LUMINAL_MATURE_DN"
[3] "LIM_MAMMARY_STEM_CELL_DN"
[4] "CHARAFE_BREAST_CANCER_LUMINAL_VS_MESENCHYMAL_DN"
[5] "LIU_PROSTATE_CANCER_DN"

The EGSEA report generated for this dataset is available online at http://bioinf.wehi.edu.au/EGSEA/mam-ma-egsea-report/index.html. Reanalysis of this data retrieves similar c2 gene sets to those identified by analysis of RNA-seq data. These included the LIM gene signatures (sets 1, 2 and 3) as well as those derived from populations with similar cellular origin (set 4).

Discussion

In this workflow article, we have demonstrated how to use the EGSEA package to combine the results obtained from different gene signature databases across multiple GSE methods to find an ensemble solution. A key benefit of an EGSEA analysis is the detailed and comprehensive HTML report that can be shared with collaborators to help them interpret their data. This report includes tables prioritising gene signatures according to the user specified analysis options, and both gene set specific and summary graphics, each of which can be
generated individually using specific R commands. The approach taken by EGSEA is facilitated by the diverse range of gene set testing algorithms and plotting capabilities available within Bioconductor. EGSEA has been tailored to suit a limma-based differential expression analysis which continues to be a very popular and flexible platform for transcriptomic data. Analysts who choose an individual GSE algorithm to prioritise their results rather than an ensemble solution can still benefit from EGSEA’s comprehensive reporting capability.

Software availability

Code to perform this analysis can be found in the EGSEA123 workflow package available from Bioconductor: https://www.bioconductor.org/help/workflows/EGSEA123.

Latest source code is available at: https://github.com/mritchie/EGSEA123.

Archived source code as at the time of publication is available at: https://doi.org/10.5281/zenodo.104343640.

Software license: Artistic License 2.0.

Competing interests

MA and MN are employees of CSL Limited.

Grant information

This work was funded by a National Health and Medical Research Council (NHMRC) Fellowship to MER (GNT1104924), Victorian State Government Operational Infrastructure Support and Australian Government NHMRC IRISS.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgements

This material was first trialled in a workshop at the BioC 2017 conference at the Dana Farber Cancer Institute (Boston, MA) on 28 July 2017. We thank the participants at this workshop for their feedback. The authors also thank Dr Alexandra Garnham (The Walter and Eliza Hall Institute of Medical Research) for feedback on this workflow article.

References

Open Peer Review

Current Referee Status: ? ? ? ✓

Version 1

Referee Report 20 December 2017

https://doi.org/10.5256/f1000research.13583.r27967

Pekka Kohonen 1, Roland Grafström 2
1 Institute of Environmental Medicine, Karolinska Institutet, Stockholm, Sweden
2 IMM Institute of Environmental Medicine, Karolinska Institutet, Stockholm, Sweden

GSEA analysis methods do not all produce same results. Score-based gene set analysis methods like the Broad Institute GSEA tool are considered to perform better than normal Fisher’s exact test (overrepresentation analysis). But analysts often use methods they know to be less than ideal in order to reduce complexity and save time. So it is good to have a unified interface for GSEA analyses with R – it helps save programming time and reduces complexity. In addition EGSEA is a unique method that combines up to 12 gene set analysis methods into a single score. Independent test also corroborate that the tool using the 12 has more specificity and good sensitivity compared to using some of the tests alone.

The EGSEA 1-2-3 workflow is easy to use and generate good-quality figures with the ggplot2 R package. Some of the figures are novel compared to other packages e.g., scatter plots designed to compare different contrasts. It is also very useful that the tool can be applied to multiple contrasts at a time, although if there are too many contrasts then the number of plots becomes unwieldy (increases combinatorically).

Some more technical comments:
The results object is very complicated for retrieving individual method analysis results (although summaries are readily available). I quite like the "biobroom" Bioconductor package that does "tidy" data frames from limma results objects.

All in all a very useful package both for automating the running of lots of methods at the same time and of course for the "ensemble" method. It is recommended to be considered to be part of a standard bioinformatics workflow.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes
Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

We have read this submission. We believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.
4. In addition, there is big difference between the final rank and med.rank (e.g. 1 vs 36). This may suggest inconsistent results came from different base methods. This may also be due to the large number of gene sets being tested. Again, using a smaller gene set collection and a few base methods could make the ranking more consistent.

5. All visualization functions, i.e. plotHeatmap, plotPathway, plotGOGraph, plotMethods, plotSummary and plotBars share largely the same set of arguments, they can have a unified wrapper function like plot.gsa() with an extra argument type to specify the plot type.

6. Functions plotPathway, plotGOGraph are wrapper functions for those in the pathview and topGO package as the author noted in the text. It would be good to explicit show some message like “calling plotting function from pathview or topGO package etc”, just like the message when running egsea().

7. HTML report of the results is a very valuable feature for the users. However, the code can run a long time, it would be helpful to add some progress reminder message to generateReport() function like egsea(). BTW, the KEGG Pathway graphs are not shown properly in the report example at http://bioinf.wehi.edu.au/EGSEA/mam-rnaseq-egsea-report/index.html.

Is the rationale for developing the new software tool clearly explained?  
Yes

Is the description of the software tool technically sound?  
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?  
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?  
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?  
Yes

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.
Jenny Drnevich  
Roy J. Carver Biotechnology Center, University of Illinois at Urbana–Champaign, Urbana, IL, USA

This F1000 software tool article describes the EGSEA package that incorporates many different gene set testing methods from various packages and also allows access to a wide array of gene sets from different databases through the accompanying EGSEAdataset package. These packages will enable researchers to conveniently test many different methods and incorporate their results to get more robust biological insights1, and this article gives a well-written walk-through of how to use the packages.

The biggest limitation I see it that EGSEA is focused only on human and mouse data (and rat? The article does not list rat but the help page for buildIdx() lists rats as one of the species). I understand that many of the gene set collections like MSigDB and GeneSetDB are only available for human/mouse, but KEGG currently lists 429 Eukaryotic organisms (http://www.genome.jp/kegg/catalog/org_list.html) and GO terms are readily available for 19 species using BioC’s pre-built OrgDB packages and hundreds of other through AnnotationHub. It is unclear whether EGSEA functions buildCustomIdx and buildGMTIdx that were "written to allow users to run EGSEA on gene set collections that may have been curated within a lab or downloaded from public databases and allow use of gene identifiers other than Entrez IDs" can be used to run EGSEA on additional species. If so, this should be clearly stated in both the Abstract and in the body of the article, plus an example given on how to use buildCustomIdx for another species. If there is some reason that EGSEA cannot currently extend to other species, this should be acknowledged as a limitation and future versions should strive to allow this (although not required before approval).

Other issues to address before approval:

1. I am unable to create the html report on my Windows machine, getting the following error:

Build GO DAG topology ...........

There are no adj nodes for node: GO:0061857
Error in switch(type, isa = 0, partof = 1, -1) :
EXPR must be a length 1 vector

However, I reported this error to the support site (https://support.bioconductor.org/p/103640/#103748) and got a speedy reply from the author. It hopefully will be resolved soon, although there is a concern of why the error was not found on another Windows machine.

2. I am concerned that as demonstrated in this paper, EGSEA seems to take the place of standard limma differential expression analysis, in that the model fitting takes place within the egsea() function. Certain gene set testing functions do need the individual expression values and not just the fitted values in an MArrayLM object but given the computational time (8 min as shown in the article code block and 19 min on my own computer) you should never run egsea() without first assessing the model fit on your own! Ideally the egsea function could be written to accept MArrayLM, or at least the article should clearly state that users should have first assessed the validity of the model fit through the usual workflow of Law et al. (2016)2 prior to running EGSEA.

3. I also wonder why there are different interfaces for voom-based analysis and microarray data given that both use EList objects. I understand that the voom weights need to be used internally, but limma’s lmFit function handles both without trouble, although it was originally coded for microarray data and the voom functionality came later. Even if there needs to be a separate
function egsea.ma() for non-voom, non-count data, it should still accept an EList object so that the user does not have to pull out the expression data and the grouping info.

4. Back to the computational time required, there are several vague references to removing the roast method "to save time" and that the report generation "significantly" increases run time. It would be nice to have an example of the time required to run roast and the report generation for the computational architecture that created the article.

References

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

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.

Referee Report 27 November 2017
https://doi.org/10.5256/f1000research.13583.r27972

Robert Castelo
Department of Experimental and Health Sciences, Universitat Pompeu Fabra, Barcelona, Spain
This article describes a gene set enrichment analysis (GSEA) workflow for the "Ensembl of GSEA" (EGSEA) R/Bioconductor software package. EGSEA is an ensemble-like method recently published\(^1\) by the authors of this workflow that allows the user to simultaneously apply different GSEA algorithms on a high-throughput molecular profiling data set, by combining p-values associated with each algorithm using classical meta-analysis approaches such as the Fisher's method.

Because the statistical methodology is already described in detail in the corresponding publication, the present software tool article focuses on showing a step-by-step workflow with EGSEA. However, the vignette of the software package already provides a very detailed description about how to use EGSEA through its 39 pages. Therefore, it would be useful for the interested reader to find upfront when he/she should be consulting the vignette and when he/she should be consulting this workflow. Besides this introductory aspects, the following issues should be addressed before approval:

1. The code given in the article breaks, at least in my computer, more concretely, at this line:

```r
gsa = egsea(voom.results=v, contrasts=contr.matrix,
            gs.annots=gs.annots, symbolsMap=symbolsMap,
            baseGSEAs=baseMethods, sort.by="med.rank",
            num.threads = 8, report = FALSE)
```

EGSEA analysis has started

```r
##------ Mon Nov 27 12:37:42 2017 ------##
Log fold changes are estimated using limma package ...
limma DE analysis is carried out ...
Number of used cores has changed to 4 
in order to avoid CPU overloading.
EGSEA is running on the provided data and c2 collection
......camera*.....safe*...gage*...padog*.....gsva*..fry*...plage*...globaltest*...zscore*...ora*...ssgsea*
Error in temp.results[[baseGSEA]][[i]][names(gs.annot@idx), ] :
   incorrect number of dimensions
```

while running it with the latest release version 1.6.0. This is strange since the package builds and runs the vignette without problems. So, this might be related to the different sample data sets. A possible hint may come from the fact that the `buildIdx()` call is not returning the expected class of object, according to the workflow:

```r
class(gs.annots$s2)
## [1] "NULL"
summary(gs.annots$s2)
## Length Class Mode
## 0 NULL NULL
```

2. The workflow contains a rather high amount of code, often with a non-trivial use of externally instantiated objects and nested calls to functions. It would be helpful for the interested reader to be able to easily copy and paste the instructions, but the fact that R commands are given with the R shell '>' and '+' symbols makes it less easy. A non-expert user may even copy those characters and get an error. I would recommend removing those characters from the illustrated code, just as it happens with the vignette.
3. The workflow assumes that the user has a 'DGEList' object with gene metadata including the mapping between Entrez identifiers' and HGNC symbols. This is a rather unrealistic assumption and I would recommend that the workflow starts building that object from scratch and showing how to build that table of gene metadata.

Below I also describe other issues that I would recommend to be considered in future versions of the software but which I do not consider them to be required for approval of this article:

1. The so-called "summary plot" shows the -log10 p-value on the x-axis and average absolute log fold-change of the set genes on the y-axis. Because this is in a way analogous to a rotated volcano plot, I would suggest to use the same arrangement of axes as in the volcano plot, which is a rather standardized display of significance and magnitude of the effects of interest.

2. One of the key features of the Bioconductor project, to which the EGSEA package is contributing to, is enabling software interoperability through sharing the use of common data structures across different software packages. Using specialized data structures, where analogous ones have been already designed by the Bioconductor core team or by a wider community of developers, locks the user into that package and limits the possibilities of using it as a building block in other more complex workflows. I'm making this comment because I have the impression that the EGSEA package would benefit of using the infrastructure provided by the Bioconductor GSEABase package, in which data structures are defined to store and access gene sets and collections of gene sets of different kinds. A salient feature of that infrastructure is the possibility to seamlessly map gene identifiers of different kinds. This would simplify and improve the user experience of EGSEA since mapping between genes coded with a particular kind of identifier, and gene sets defined with another kind, is one of the most common tasks in a GSEA-like analysis.

References

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

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.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com