# Package 'DrugVsDisease'

November 13, 2025

**Type** Package

**Title** Comparison of disease and drug profiles using Gene set Enrichment Analysis

**Version** 2.53.0

**Date** 2018-11-10

**Author** C. Pacini

**Maintainer** j. Saez-Rodriguez <saezrodriguez@ebi.ac.uk>

**Description** This package generates ranked lists of differential gene expression for either disease or drug profiles. Input data can be downloaded from Array Express or GEO, or from local CEL files. Ranked lists of differential expression and associated p-values are calculated using Limma. Enrichment scores (Subramanian et al. PNAS 2005) are calculated to a reference set of default drug or disease profiles, or a set of custom data supplied by the user. Network visualisation of significant scores are output in Cytoscape format.

**LazyData** yes

**LazyLoad** yes

**License** GPL-3

**Depends** R (>= 2.10), affy, limma, biomaRt, ArrayExpress, GEOquery, DrugVsDiseasedata, cMap2data, qvalue

**Imports** annotate, hgu133a.db, hgu133a2.db, hgu133plus2.db, RUnit, BiocGenerics, xtable

**biocViews** Microarray, GeneExpression, Clustering

**NeedsCompilation** no

**git_url** https://git.bioconductor.org/packages/DrugVsDisease

**git_branch** devel

**git_last_commit** 6483099

**git_last_commit_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2025-11-13

# Contents

DrugVsDisease-package . . . 2
classifyprofile . . . 3
combineProfiles . . . 6
customClust . . . 7
customDB . . . 8
customedge . . . 8
customsif . . . 9
generateprofiles . . . 9
profiles . . . 11
selectrankedlists . . . 12
selprofile . . . 13

**Index** **14**

---

`DrugVsDisease-package` *DrugVsDisease Package Overview*

---

## Description

This package generates ranked lists of differential gene expression for either disease or drug profiles. Input data can be downloaded from Array Express [1] or GEO [2], or from local CEL files. Ranked lists of differential expression and associated p-values are calculated using Limma [3]. Enrichment scores [4] are calculated to a reference set of default drug or disease profiles, or a set of custom data supplied by the user. Significance scores are output in Cytoscape http://www.cytoscape.org/ format.

## Details

| | |
|---|---|
| Package: | DvD |
| Type: | Package |
| Version: | 1.0 |
| Date: | 2012-06-15 |
| License: | GPL-2 |
| LazyLoad: | yes |

Profiles are calculated via generateprofiles and selected profiles are then classified using classifyprofile.

## Author(s)

C. Pacini

Maintainer: J Saez-Rodriguez <saezrodriguez@ebi.ac.uk>

## References

[1]Parkinson et al. (2010) ArrayExpress update an archive of microarray and high-throughput sequencing-based functional genomics experiments. Nucl. Acids Res.,doi: 10.1093/nar/gkq1040.

[2]Barrett T et al. (2011) NCBI GEO: archive for functional genomics data sets-10 years on. Nucl. Acids Res, 39, D1005-D1010.

[3]Smyth et al. (2004). Linear models and empirical Bayes method for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology, Vol. 3, No. 1, Article 3.

[4]Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S. & Mesirov, J. P. (2005) *Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles.* Proc. Natl. Acad. Sci. USA 102, 15545-15550.

## See Also

generateprofiles, selectrankedlists, classifyprofile

## Examples

```
#profileAE<-generateprofiles(input="AE",accession="E-GEOD-22528")
#selprofiles<-selectrankedlists(profileAE,1)
#default<-classifyprofile(data=selprofiles$ranklist, signif.fdr=1.1)
```

---

`classifyprofile` *Classify Profiles*

---

## Description

For a set of ranked gene expression profiles, enrichment scores to a default or custom set of profiles. The ranked gene expression profiles can have been generated from generate profiles or provided by the user. The enrichment scores are assessed for significance using permutations to generate random profiles.

## Usage

```
classifyprofile(data, pvalues = NULL, case = c("disease", "drug"), type = c("fixed", "dynamic", "range"
```

## Arguments

| | |
|---|---|
| `data` | Matrix gene expression profiles. Rows are genes whose names match the genelist from DvDdata and columns are the different profiles. This can also be a path to a txt file containing the data. |
| `pvalues` | Optional numerical matrix of pvalues associated with the differential expression of the ranked lists in data argument. This can also be a path to a txt file containing the data. |

| | |
|---|---|
| case | Character string indicating whether or not the input profiles are disease (default) or drug profiles |
| type | Character string giving the method of gene set sizes, one of fixed (default), dynamic or range |
| lengthtest | Integer giving the number of genes to generate a set size for use with type=fixed, default 100. |
| ranges | A vector of integer values giving a range of gene set sizes, for use with type=range, default 100 to 2000 every 100. |
| adj | Character string to set the method for multiple hypothesis testing, one of qvalue or BH (default) |
| dynamic.fdr | Double giving the false discovery rate to determine the gene set sizes (default 5 |
| signif.fdr | Double giving the false discovery rate to determine significance of enrichment scores (default 5 |
| customRefDB | Optional matrix of reference ranked profiles to compare the input profiles to. Alternatively a string giving the name of the path where the database is stored. |
| noperm | Integer for the number of permutation profiles (default 1000) |
| customClusters | Optional data frame of cluster assignments which relate to customRefDB |
| clustermethod | Character string to give the cluster method one of single (default) or average |
| avgstat | Character string for the statistic used with average cluster method. One of mean (default) or median. |
| cytoout | Logical if Cytoscape SIF and Edge Attribute files should be produced (default is FALSE) |
| customsif | Optional SIF input needed for use with custom clusters. Can be an R object or a character string containing a file path |
| customedge | Optional Edge attribute input needed for use with custom clusters. Can be an R object or a character string containing a file path |
| cytofile | Character string for the filename of the Cytoscape output |
| no.signif | Integer giving the maximum number of significant enrichment scores to return. Default is 10. |
| stat | One of KS (default) or WSR. KS is the Kolmogorov-Smirnov type statistic which equally weights all elements of the gene set. WSR uses both the sign and the position in the ranked list to calculate an enrichment score. |

## Details

The classify profile function contains a default set of drug (from the Connectivity Map [2]) and disease profiles (from various GEO profiles) with corresponding clusters which input profiles of disease and drug respectively are compared to. Enrichment scores [1] are calculated between the input profiles and the corresponding inverted reference profiles such that, the score measures the enrichment of up-regulated genes in the input profiles in the down-regulated genes in the reference set[3]. The gene set sizes to use in the Enrichment scores can be specified using one of three methods - fixed, range or dynamic. With the fixed method the user specifies a fixed number of genes to use with all input profiles. The range option takes a vector of integers - enrichment scores are

calculated for each profile using a gene set size as given by the vector. The dynamic option uses the p-values to determine the gene set size according to the number of significantly differential expressed genes (following multiple hypothesis correction). Multiple hypothesis correction is done using one of two methods, qvalue or Benjamini-Hochberg. Two enrichment scores are calculated for the up and down regulated scores. These contain the scores where the input profiles gene set is compared to the reference profile and where the reference profiles equivalent gene set (as determined by the type method) are compared to the input profile when using the KS option [4]. The WSR option implements the score of [5]. Various optional parameters exist for comparing an input profile to a users own reference set. For using custom reference data the user needs to provide the custom ranked lists of differential expression (customRefDB), the corresponding clusters (or network) between nodes in the reference data set (customClusters) and the SIF and Edge attribute files for Cytoscape option if cytoout=TRUE. Default clusters are provided by the DvDdata package, and include a drug and disease network. Input profiles to classifyprofile can be assigned to clusters using either single or average linkage. With single linkage an edge is drawn between the input profile and any significant scoring (up to a user defined maximum of no.signif) reference profiles. For average linkage either the mean or median (specified through avgstat) of the scores to each member in a cluster is calculated and the profile is assigned to the cluster with the highest average score.

To use your own preprocessed data, make sure the txt files for the data (and optional pvalues) have rownames with genes matching those in the reference data set. The files should have genes as row names in the first column and the header (col names) giving the names of the input profile(s). The input to classifyprofile is then a string of the path to the files.

## Value

Data Frame for each profile with elements:

| | |
|---|---|
| `Node` | Names of the profiles with significant scores to the input profile(s) |
| `ES Distance` | The distance between the input profile and corresponding reference profile. Defined as 1-ES |
| `Cluster` | The cluster number of the node |
| `RPS` | Running sum Peak Sign, taking values -1 to indicate an inverse relationship (potentially) therapeutic, or 1 to indicate a similar profile. |

## Author(s)

C. Pacini

## References

[1]Subramanian A et~al. (2005) Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. PNAS, 102(43), 15545-15550.

[2]Lamb J et~al. (2006) The Connectivity Map: Using Gene-Expression Signatures to Connect Small Molecules, Genes, and Disease. Science, 313(5795), 1929-1935.

[3]Sirota M et~al. (2011) Discovery and Preclinical Validation of Drug Indications Using Compendia of Public Gene Expression Data. Sci Trans Med,3:96ra77.

[4]Iorio et al. (2010) Discovery of drug mode of action and drug repositioning from transcriptional responses. PNAS, 107(33), 14621-14626.

[5]Zhang S et al. (2008) A simple and robust method for connecting small- molecule drugs using gene-expression signatures. BMC Bioinformatics, 9:258.

## See Also

Function for generating profiles for input to classifyprofile: generateprofiles.

## Examples

```
data(selprofile)
classification<-classifyprofile(data=selprofile$ranklist,signif.fdr=1,noperm=20)
```

---

`combineProfiles` *Combine Profiles*

---

## Description

Combine a set of hybridisations using the median rank method as proposed by Warnat

## Usage

```
combineProfiles(data)
```

## Arguments

| | |
|---|---|
| `data` | A list of R objects, each object should contain a matrix of expression values to be merged with each other. The first object will be used as the reference data set. |

## Details

The combine profiles function merges a set of expression values, normalising across different experiments to facilitate the meta-analysis of similar experiments i.e. cell lines treated with the same drug but which may be from different "batches" or platforms.

## Value

Matrix of rank normalised expression data. Rows are genes, columns are hybridisations from the different experiments.

## Author(s)

C. Pacini

## References

R code from CONOR package. Need also reference to the method.

customClust *Custom Clusters*

## Description

Matrix containing a set of drugs and their corresponding clusters which can be used as input to the classifyprofile function.

## Usage

```
data(customClust)
```

## Format

Data frame with 47 rows and 2 columns. Columns are headed "Drug" and "Cluster".

## Details

Each row refers to a compound in the customDB data set `link{customDB}` with its corresponding cluster assignment in the second column. These profiles are a subset of the Connectivity Map data [1] (full set available in the DrugVsDiseasedata package **DrugVsDiseasedata**, data object `drugRL`, for example use. Clusters were generated using affinity propagation clustering [2]

## Source

http://www.broadinstitute.org/cmap/

## References

[1] Lamb J et~al. (2006) The Connectivity Map: Using Gene-Expression Signatures to Connect Small Molecules, Genes, and Disease. Science, 313(5795), 1929-1935. [2] U. Bodenhofer, A. Kothmeier, and S. Hochreiter. APCluster: an R package for affinity propagation clustering. Bioinformatics, 27(17):2463-2464, 2011.

## Examples

```
data(customClust)
```

## customDB *Custom Gene Expression Profiles*

### Description

Matrix containing a set of ranked gene expression profiles which can be used as custom input into classifyprofiles.

### Usage

```
data(customDB)
```

### Details

This is an example subset of the drugRL available in the DrugVsDiseasedata package **DrugVsDiseasedata**, data object drugRL, based on data from the Connectivity Map

### Source

http://www.broadinstitute.org/cmap/

### Examples

```
data(customDB)
## maybe str(profiles) ; plot(profiles) ...
```

## customedge *Edge attributes for example custom network*

### Description

Data frame containing the required format of an edge attribute file input into Cytoscape

### Usage

```
data(customedge)
```

### Details

Example subset of edge attributes taken from the full reference data set of drug compounds in the DvDdata package.

### Source

http://www.broadinstitute.org/cmap/

## Examples

```
data(customedge)
```

---

`customsif` *SIF file for custom clusters*

---

## Description

Data frame containing SIF format file of a network which could be input into the classifyprofile function.

## Usage

```
data(customsif)
```

## Details

Example subset of edge attributes taken from the full reference data set of drug compounds in the DvDdata package.

## Source

http://www.broadinstitute.org/cmap/

## Examples

```
data(customsif)
```

---

`generateprofiles` *Generate Profiles*

---

## Description

Processing Affymetrix data to generate ranked lists of differential gene expression and associated p-values.

## Usage

```
generateprofiles(input = c("AE", "GEO", "localAE", "local"), normalisation = c("rma", "mas5"), accessio
```

## Arguments

| | |
|---|---|
| input | Character string denoting the source of the data. One of AE (default), GEO, localAE or local. |
| normalisation | Character string denoting the normalisation procedure as implemented in the affy package. One of mas5 (default) or rma. |
| accession | Optional character string giving the database reference for use with either the AE or GEO options. |
| customfile | Optional character string giving the path of a file containing the factor values associated with the CEL files specified in folder celfilepath |
| celfilepath | Optional character string giving the path of a folder containing CEL files to analyse. |
| sdrfpath | Optional character string giving path of an sdrf file corresponding to CEL files in celfilepath |
| case | Character string, one of disease (default) or drug denoting whether the input profiles are disease or drug profiles. |
| statistic | Character string, one of coef (default), t or diff. |
| annotation | Optional character string giving the platform of the affymetrix files |
| factorvalue | Optional character string giving the name of the factor value in the GEO database. |
| annotationmap | Optional matrix, or string to text file, containing an annotation map to convert from probes (first column) to HUGO gene symbols (second column). If passing a file path name the text file should have only two columns without rownames or headers. |
| type | The type of statistic to use to combine multiple probes to a single gene. Can be one of average (default) expression values, median polish, maxvar: the single probe to represent the set which has maximum variance or max to use the probe with maximal variance. |
| outputgenedata | Boolean set to default FALSE. Outputs the gene data produced by generate profiles instead of the fitted coefficients from the linear models. |

## Details

Input types of AE and GEO use raw data download from Array Express using the ArrayExpress [1] package or processed GDS files from GEO using the GEOquery package [2]. CEL files and sdrf files downloaded from Array Express and stored locally can be processed using localAE option with the sdrf file path specified in sdrfpath and the path of the folder containing the CEL files contained in celfilepath. Users data stored locally can be processed using the local option with CEL file folders in celfilepath and factors associated with the CEL files in customfile. Where metadata may be missing from the GEO database, platform annotations can be specified using the annotation parameters and the name of main factor value (e.g. disease status, or compound treatment) using factorvalue option. Raw CEL files are normalised (rma or mast)[3] and data is converted from probes to genes using BioMart annotations [4]. Linear models are fitted using the database factor vales or user provided factors for locally stored data [5]. The differential expression is calculated for HUGO genes with the mapping performed automatically for Affymetrix platforms, HGU133A, HGU133Plus2 and HGU133A2 using BioMart. The differential expression statistic is one of coef (default), which corresponds to log (base 2) FC, diff (which is the difference between raw (non-logged) expression values, or t for the t-statistic based on log base 2 expression values.

## Value

List with two elements:

| | |
|---|---|
| `Ranklist` | Matrix containing the ranks of gene expression. Rows containing the genes, columns the different profiles |
| `Pvalues` | Matrix containing the associated p-values to the differential expression profiles in Ranklist |

## Author(s)

C. Pacini

## References

[1]Kauffmann et al. (2009) Importing Array Express datasets into R/Bioconductor. Bioinformatics, 25(16):2092-4.

[2]Davis et al. (2007) GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. Bioinformatics, 14, 1846-1847.

[3]Irizarry et al. (2003) Summaries of Affymetrix GeneChip probe level data. Nucleic Acids Research, 31(4); e15.

[4]Durinck et al. (2009). Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Nature Protocols 4, 1184-1191.

[5]Smyth et al. (2004). Linear models and empirical Bayes method for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology, Vol. 3, No. 1, Article 3.

## See Also

`classifyprofile`

## Examples

```
#profileAE<-generateprofiles(input="AE",accession="E-GEOD-22528")
```

---

`profiles` *Gene Expression Profiles*

---

## Description

List containing ranked lists of gene expression and associated p-values for a set of profiles.

## Usage

```
data(profiles)
```

## Format

List containing rank of differential gene expression and pvalues in a list. Each item in the list contains a matrix. Matrix has number of rows equal number genes and columns for the different profiles.

## Source

http://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-22528

## Examples

```
data(profiles)
## maybe str(profiles) ; plot(profiles) ...
```

---

`selectrankedlists` *Select Ranked Lists*

---

## Description

Given a list with two elements, one containing ranklists for a set of regression models and the second containing the associated p-values. This function is used to extract a subset of the models.

## Usage

```
selectrankedlists(ranklist, colsinc)
```

## Arguments

| | |
|---|---|
| `ranklist` | List with two elements (as output from generate profiles): *Ranklist*: Matrix containing the ranks of gene expression. Rows containing the genes, columns the different profiles.<br>*Pvalues*: Matrix containing the associated p-values to the differential expression profiles in Ranklist. |
| `colsinc` | Vector of integers containing the column references of the profiles to select. |

## Details

Format of list provided to selectrankedlists is the same as is output from generateprofiles `generateprofiles`. The output from selectrankedlists can be used as input to the classify profile function `classifyprofile`.

## Value

| | |
|---|---|
| `Ranklist` | Matrix containing the ranks of gene expression. Rows containing the genes, columns the selected profiles |
| `Pvalues` | Matrix containing the associated p-values to the differential expression profiles in Ranklist |

### Author(s)

C. Pacini

### Source

http://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-22528

### See Also

generateprofiles, classifyprofile.

### Examples

```
data(profiles)
selectprofile<-selectrankedlists(profiles,1)
classification<-classifyprofile(data=selectprofile$ranklist,noperm=10,signif.fdr=1)
```

---

## selprofile *List: Differential gene expression and p-values*

---

### Description

Ranklist which has the rank of each gene according to differential expression. P-values is the second element in the list containing the associated p-value for gene differential expression.

### Usage

```
data(selprofile)
```

### Format

List with two elements, the first ranklist containing the ranked position of the gene according to its differential expression. The second item in the list is the associated pvalues.

### Details

Example ranked list of differential expression. Taken from Array express experiment E-GEOD-22528

### Source

http://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-22528

### Examples

```
data(selprofile)

## maybe str(selprofile) ; plot(selprofile) ...
```

# Index


∗ **GSEA**
    classifyprofile, 3
    combineProfiles, 6
∗ **classify**
    classifyprofile, 3
    combineProfiles, 6
∗ **datasets**
    customClust, 7
    customDB, 8
    customedge, 8
    customsif, 9
    profiles, 11
    selprofile, 13
∗ **package**
    DrugVsDisease-package, 2
∗ **profile**
    classifyprofile, 3
    combineProfiles, 6

classifyprofile, *3*, 3, *11–13*
combineProfiles, 6
customClust, 7
customDB, 8
customedge, 8
customsif, 9

DrugVsDisease *(*DrugVsDisease-package*)*, 2
DrugVsDisease-package, 2

generateprofiles, *3*, *6*, 9, *12, 13*

profiles, 11

selectrankedlists, *3*, 12
selprofile, 13