

# Package ‘crossmeta’

October 21, 2021

**Title** Cross Platform Meta-Analysis of Microarray Data

**Version** 1.19.0

**Author** Alex Pickering

**Maintainer** Alex Pickering <alexvpickering@gmail.com>

**Description** Implements cross-platform and cross-species meta-analyses of Affymetrix, Illumina, and Agilent microarray data. This package automates common tasks such as downloading, normalizing, and annotating raw GEO data. The user then selects control and treatment samples in order to perform differential expression analyses for all comparisons. After analysing each contrast separately, the user can select tissue sources for each contrast and specify any tissue sources that should be grouped for the subsequent meta-analyses.

**Depends** R (>= 4.0)

**SystemRequirements** libxml2: libxml2-dev (deb), libxml2-devel (rpm)  
libcurl: libcurl4-openssl-dev (deb), libcurl-devel (rpm)  
openssl: libssl-dev (deb), openssl-devel (rpm), libssl\_dev  
(csw), openssl@1.1 (brew)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** TRUE

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**Suggests** knitr, rmarkdown, lydata, org.Hs.eg.db, testthat,  
tximportData

**Imports** affy (>= 1.52.0), affxparser (>= 1.46.0), AnnotationDbi (>= 1.36.2), Biobase (>= 2.34.0), BiocGenerics (>= 0.20.0), BiocManager (>= 1.30.4), DT (>= 0.2), DBI (>= 1.0.0), DESeq2, data.table (>= 1.10.4), edgeR, fdrtool (>= 1.2.15), GEOquery (>= 2.40.0), limma (>= 3.30.13), matrixStats (>= 0.51.0), metaMA (>= 3.1.2), miniUI (>= 0.1.1), methods, oligo (>= 1.38.0), reader (>= 1.0.6), RColorBrewer (>= 1.1.2), RCurl (>= 1.95.4.11), RSQLite (>= 2.1.1), stringr (>= 1.2.0), sva (>= 3.22.0), shiny (>= 1.0.0), shinyjs (>= 2.0.0), shinyBS (>=

0.61), shinyWidgets ( $\geq 0.5.3$ ), shinypanel ( $\geq 0.1.0$ ), statmod ( $\geq 1.4.34$ ), SummarizedExperiment, tibble, XML ( $\geq 3.98.1.17$ ), readxl ( $\geq 1.3.1$ )

**biocViews** GeneExpression, Transcription, DifferentialExpression, Microarray, TissueMicroarray, OneChannel, Annotation, BatchEffect, Preprocessing, GUI

**git\_url** <https://git.bioconductor.org/packages/crossmeta>

**git\_branch** master

**git\_last\_commit** 7fb4a04

**git\_last\_commit\_date** 2021-05-19

**Date/Publication** 2021-10-21

## R topics documented:

add_adjusted	3
add_sources	3
add_vsd	4
bulkPage	5
bulkPageUI	5
diff_expr	6
es_meta	7
exprs.MA	9
filter_genes	9
fit_ebayes	10
fix_illum_headers	10
get_raw	11
get_sva_mods	11
get_top_table	12
get_vsd	13
gs.names	13
gslist	14
ilmn.nnum	14
iqr_replicates	15
load_agil_plat	15
load_diff	16
load_raw	16
open_raw_illum	17
phenoData.ch2	18
prefix_illum_headers	18
remove_automated	19
run_limma	19
run_limma_setup	20
run_sva	21
setup_prev	21
symbol_annot	22

*add\_adjusted*

3

## Index

[24](#)

---

<code>add_adjusted</code>	<i>Add expression data adjusted for pairs/surrogate variables</i>
---------------------------	---

---

### Description

Add expression data adjusted for pairs/surrogate variables

### Usage

```
add_adjusted(eset, svobj = list(sv = NULL), numsv = 0)
```

### Arguments

<code>eset</code>	ExpressionSet
<code>svobj</code>	surrogate variable object
<code>numsv</code>	Number of surrogate variables to adjust for

### Value

eset with adjusted element added

---

<code>add_sources</code>	<i>Add sample source information for meta-analysis.</i>
--------------------------	---

---

### Description

User selects a tissue source for each contrast and indicates any sources that should be paired. This step is required if you would like to perform source-specific effect-size/pathway meta-analyses.

### Usage

```
add_sources(diff_exprs, data_dir = getwd(), postfix = NULL)
```

### Arguments

<code>diff_exprs</code>	Previous result of <code>diff_expr</code> , which can be reloaded using <code>load_diff</code> .
<code>data_dir</code>	String specifying directory of GSE folders.
<code>postfix</code>	Optional string to append to saved results. Useful if need to run multiple meta-analyses on the same series but with different contrasts.

## Details

The **Sources** tab is used to add a source for each contrast. To do so: click the relevant contrast rows, search for a source in the *Sample source* dropdown box, and then click the *Add* button.

The **Pairs** tab is used to indicate sources that should be paired (treated as the same source for subsequent effect-size and pathway meta-analyses). To do so: select at least two sources from the *Paired sources* dropdown box, and then click the *Add* button.

For each GSE, analysis results with added sources/pairs are saved in the corresponding GSE folder (in `data_dir`) that was created by `get_raw`.

## Value

Same as `diff_expr` with added slots for each GSE in `diff_exprs`:

<code>sources</code>	Named vector specifying selected sample source for each contrast. Vector names identify the contrast.
<code>pairs</code>	List of character vectors indicating tissue sources that should be treated as the same source for subsequent effect-size and pathway meta-analyses.

## Examples

```
library(lydata)

# load result of previous call to diff_expr:
data_dir <- system.file("extdata", package = "lydata")
gse_names <- c("GSE9601", "GSE34817")
anals <- load_diff(gse_names, data_dir)

# run shiny GUI to add tissue sources
# anals <- add_sources(anals, data_dir)
```

---

<code>add_vsd</code>	<i>Add VST normalized assay data element to expression set</i>
----------------------	--

---

## Description

For microarray datasets duplicates `exprs` slot into `vsd` slot.

## Usage

```
add_vsd(eset, rna_seq = TRUE)
```

## Arguments

<code>eset</code>	ExpressionSet with group column in <code>pData(eset)</code>
<code>rna_seq</code>	Is this an RNA-seq eset? Default is TRUE.
<code>pbulk</code>	Is this an pseudobulk single-cell eset? Default is FALSE. Used by package <code>dseqr</code> .
<code>vsd_path</code>	Path to save result to. Allows skipping running transform on each load.

**Value**

eset with 'vsd' assayDataElement added.

---

bulkPage	<i>Logic for Select Contrasts Interface</i>
----------	---

---

**Description**

Logic for Select Contrasts Interface

**Usage**

```
bulkPage(input, output, session, eset, gse_name, prev)
```

**Arguments**

input, output, session	shiny module boilerplate
eset	ExpressionSet
gse_name	GEO accession for the series.
prev	Previous result of diff_expr. Used to allow rechecking previous selections.

---

bulkPageUI	<i>UI for Select Contrasts Interface</i>
------------	--

---

**Description**

UI for Select Contrasts Interface

**Usage**

```
bulkPageUI(id)
```

**Arguments**

id	The id string to be namespaced.
----	---------------------------------

diff\_expr

*Differential expression analysis of esets.***Description**

After selecting control and test samples for each contrast, surrogate variable analysis ([sva](#)) and differential expression analysis is performed.

**Usage**

```
diff_expr(
  esets,
  data_dir = getwd(),
  annot = "SYMBOL",
  prev_anals = list(NULL),
  svanal = TRUE,
  recheck = FALSE,
  postfix = NULL,
  port = 3838
)
```

**Arguments**

esets	List of annotated esets. Created by <a href="#">load_raw</a> .
data_dir	String specifying directory of GSE folders.
annot	String, column name in fData common to all esets. For duplicated values in this column, the row with the highest interquartile range across selected samples will be kept. If meta-analysis will follow, appropriate values are "SYMBOL" (default - for gene level analysis) or, if all esets are from the same platform, "PROBE" (for probe level analysis).
prev_anals	Previous result of <a href="#">diff_expr</a> , which can be reloaded using <a href="#">load_diff</a> . If present, previous selections, names, and pairs will be reused.
svanal	Use surrogate variable analysis? Default is TRUE.
recheck	Would you like to recheck previous group/contrast annotations? Requires <code>prev_anals</code> . Default is FALSE.
postfix	Optional string to append to saved results. Useful if need to run multiple meta-analyses on the same series but with different contrasts.
port	See <a href="#">runApp()</a> .

**Details**

Click the Download icon and fill in the *Group name* column and optionally the *Pairs* column. Then save and upload the filled in metadata csv. After doing so, select a test and control group to compare and click the + icon to add the contrast. Repeat previous step to add additional contrasts.

After control and test samples have been added for all contrasts that you wish to include, click the *Done* button. Repeat for all GSEs.

Paired samples (e.g. the same subject before and after treatment) can be specified by filling out the *Pairs column* before uploading the metadata.

For each GSE, analysis results are saved in the corresponding GSE folder in `data_dir` that was created by `get_raw`. If analyses needs to be repeated, previous results can be reloaded with `load_diff` and supplied to the `prev_anals` parameter. In this case, previous selections, names, and pairs will be reused.

### Value

List of named lists, one for each GSE. Each named list contains:

<code>pdata</code>	data.frame with phenotype data for selected samples. Columns <code>treatment</code> ('ctrl' or 'test'), <code>group</code> , and <code>pair</code> are added based on user selections.
<code>top_tables</code>	List with results of <code>topTable</code> call (one per contrast). These results account for the effects of nuisance variables discovered by surrogate variable analysis.
<code>ebayes_sv</code>	Results of call to <code>eBayes</code> with surrogate variables included in the model matrix.
<code>annot</code>	Value of <code>annot</code> variable.

### Examples

```
library(lydata)

# location of raw data
data_dir <- system.file("extdata", package = "lydata")

# gather GSE names
gse_names <- c("GSE9601", "GSE15069", "GSE50841", "GSE34817", "GSE29689")

# load first eset
esets <- load_raw(gse_names[1], data_dir)

# run analysis (opens GUI)
# anal_old <- diff_expr(esets, data_dir)

# re-run analysis on first eset
prev <- load_diff(gse_names[1], data_dir)
anals <- diff_expr(esets[1], data_dir, prev_anals = prev)
```

---

es\_meta

*Effect size combination meta analysis.*

---

### Description

Performs effect-size meta-analyses across all studies and seperately for each tissue source.

**Usage**

```
es_meta(diff_exprs, cutoff = 0.3, by_source = FALSE)
```

**Arguments**

diff_exprs	Previous result of <code>diff_expr</code> , which can be reloaded using <code>load_diff</code> .
cutoff	Minimum fraction of contrasts that must have measured each gene. Between 0 and 1.
by_source	Should separate meta-analyses be performed for each tissue source added with <code>add_sources</code> ?

**Details**

Builds on `zScores` function from GeneMeta by allowing for genes that were not measured in all studies. This implementation also uses moderated unbiased effect sizes calculated by `effectsize` from metaMA and determines false discovery rates using `fdrtool`.

**Value**

A list of named lists, one for each tissue source. Each list contains two named data.frames. The first, `filt`, has all the columns below for genes present in cutoff or more fraction of contrasts. The second, `raw`, has only `dprime` and `vardprime` columns, but for all genes (NAs for genes not measured by a given contrast).

dprime	Unbiased effect sizes (one column per contrast).
vardprime	Variances of unbiased effect sizes (one column per contrast).
mu	Overall mean effect sizes.
var	Variances of overall mean effect sizes.
z	Overall z score = $\mu / \sqrt{\text{var}}$ .
fdr	False discovery rates calculated from column z using <code>fdrtool</code> .
pval	p-values calculated from column z using <code>fdrtool</code> .

**Examples**

```
library(lydata)

# location of data
data_dir <- system.file("extdata", package = "lydata")

# gather GSE names
gse_names <- c("GSE9601", "GSE15069", "GSE50841", "GSE34817", "GSE29689")

# load previous analysis
anals <- load_diff(gse_names, data_dir)

# add tissue sources to perform separate meta-analyses for each source (optional)
# anals <- add_sources(anals, data_dir)
```



```
# perform meta-analysis
es <- es_meta(anals, by_source = TRUE)
```

---

exprs.MA	<i>Extract Log-Expression Matrix from MAList</i>
----------	--

---

### Description

Converts M and A-values to log-expression values. The output matrix will have two columns for each array, in the order all red then all green. Adapted from [plotDensities.MAList](#) instead of [exprs.MA](#) so that order is same as [phenoData.ch2](#).

### Usage

```
exprs.MA(MA)
```

### Arguments

MA                    an MAList object.

### Value

A numeric matrix with twice the columns of the input.

---

filter_genes	<i>Filter genes in RNA-seq ExpressionSet</i>
--------------	--

---

### Description

Uses [filterByExpr](#) to filter based on 'counts' assay or 'exprs' assay if 'counts' isn't available (for ARCHS4 data).

### Usage

```
filter_genes(eset)
```

### Arguments

eset                    ExpressionSet with 'counts' assayDataElement and group column in pData

### Value

filtered eset

### See Also

[filterByExpr](#)

**Examples**

```
# example ExpressionSet
dds <- DESeq2::makeExampleDESeqDataSet()
eset <- Biobase::ExpressionSet(DESeq2::counts(dds))
eset$group <- dds$condition
eset <- filter_genes(eset)
```

---

fit_ebayes	<i>Fit ebayes model</i>
------------	-------------------------

---

**Description**

Fit ebayes model

**Usage**

```
fit_ebayes(lm_fit, contrasts, robust = TRUE)
```

**Arguments**

lm_fit	Result of call to <a href="#">run_limma</a>
contrasts	Character vector of contrasts to fit.
robust	logical, should the estimation of <code>df.prior</code> and <code>var.prior</code> be robustified against outlier sample variances?

**Value**

result of [eBayes](#)

---

fix_illum_headers	<i>Attempts to fix Illumina raw data header</i>
-------------------	---

---

**Description**

Reads raw data files and tries to fix them up so that they can be loaded by [read.ilmn](#).

**Usage**

```
fix_illum_headers(elist_paths, eset = NULL)
```

**Arguments**

elist_paths	Path to Illumina raw data files. Usually contain patterns: <code>non_normalized.txt</code> , <code>raw.txt</code> , or <code>_supplementary_.txt</code>
eset	ExpressionSet from <a href="#">getGEO</a> .

**Value**

Character vector for annotation argument to [read.ilmm](#). Fixed raw data files are saved with file-name ending in `_fixed.txt`

---

get_raw	<i>Download and unpack microarray supplementary files from GEO.</i>
---------	---

---

**Description**

Downloads and unpacks microarray supplementary files from GEO. Files are stored in the supplied data directory under the GSE name.

**Usage**

```
get_raw(gse_names, data_dir = getwd())
```

**Arguments**

gse_names	Character vector of GSE names to download.
data_dir	String specifying directory for GSE folders.

**Value**

NULL (for download/unpack only).

**See Also**

[load\\_raw](#).

**Examples**

```
get_raw("GSE41845")
```

---

get_sva_mods	<i>Get model matrices for surrogate variable analysis</i>
--------------	---

---

**Description**

Used by `add_adjusted` to create model matrix with surrogate variables.

**Usage**

```
get_sva_mods(pdata)
```

**Arguments**

pdata            data.frame of phenotype data with column 'group' and 'pair' (optional).

**Value**

List with model matrix(mod) and null model matrix (mod0) used for sva.

---

get_top_table	<i>Get top table</i>
---------------	----------------------

---

**Description**

Get top table

**Usage**

```
get_top_table(
  lm_fit,
  groups = c("test", "ctrl"),
  with.es = TRUE,
  robust = FALSE
)
```

**Arguments**

lm_fit	Result of <a href="#">run_limma</a>
groups	Test and Control group as strings.
with.es	Add 'dprime' and 'vardprime' from <a href="#">effectsiz</a> ? Default is TRUE.
robust	logical, should the estimation of df.prior and var.prior be robustified against outlier sample variances?

**Value**

result of [toptable](#)

---

`get_vsd`*Get variance stabilized data for exploratory data analysis*

---

**Description**

Get variance stabilized data for exploratory data analysis

**Usage**

```
get_vsd(eset, rlog_cutoff = 50)
```

**Arguments**

<code>eset</code>	ExpressionSet loaded with <code>load_raw</code> . Requires group column in <code>pData(eset)</code> specifying sample groupings.
<code>rlog_cutoff</code>	Sample number above which will use <code>vst</code> instead of <code>rlog</code> . Default is 50.

**Value**

DESeqTransform with variance stabilized expression data.

---

`gs.names`*Map between KEGG pathway numbers and names.*

---

**Description**

Used to map human KEGG pathway numbers to names. Updated Feb 2017.

**Usage**

```
data(gs.names)
```

**Format**

An object of class character of length 310.

**Value**

A named character vector of human KEGG pathway names. Names of vector are KEGG pathway numbers.

---

gslist	<i>KEGG human pathway genes.</i>
--------	----------------------------------

---

**Description**

Genes for human KEGG pathways. Updated Feb 2017.

**Usage**

```
data(gslist)
```

**Format**

An object of class `list` of length 310.

**Value**

A named list with entrez ids of genes for human KEGG pathways. List names are KEGG pathway numbers.

---

ilmn.nnum	<i>Count numeric columns in raw Illumina data files</i>
-----------	---

---

**Description**

Excludes probe ID cols

**Usage**

```
ilmn.nnum(elist_paths)
```

**Arguments**

`elist_paths` Paths to raw illumina data files

**Value**

Number of numeric columns in `elist_paths` excluding probe ID columns.

---

iqr_replicates	<i>Removes features with replicated annotation.</i>
----------------	---

---

**Description**

For rows with duplicated annot, highested IQR retained.

**Usage**

```
iqr_replicates(eset, annot = "SYMBOL", rm.dup = FALSE)
```

**Arguments**

eset	Annotated eset created by load_raw.
annot	feature to use to remove replicates.
rm.dup	remove duplicates (same measure, multiple ids)? Used for Pathway analysis so that doesn't treat probes that map to multiple genes as distinct measures.

**Value**

Expression set with unique features at probe or gene level.

---

load_agil_plat	<i>Load Agilent raw data</i>
----------------	------------------------------

---

**Description**

Load Agilent raw data

**Usage**

```
load_agil_plat(eset, gse_name, gse_dir, ensql)
```

**Arguments**

eset	ExpressionSet from <a href="#">getGEO</a> .
gse_name	Accession name for eset.
gse_dir	Direction with Agilent raw data.
ensql	For development. Path to sqlite file with ENTREZID and SYMBOL columns created in data-raw/entrezdt.

**Value**

ExpressionSet

---

load_diff	<i>Load previous differential expression analyses.</i>
-----------	--

---

**Description**

Loads previous differential expression analyses.

**Usage**

```
load_diff(gse_names, data_dir = getwd(), annot = "SYMBOL", postfix = NULL)
```

**Arguments**

gse_names	Character vector specifying GSE names to be loaded.
data_dir	String specifying directory of GSE folders.
annot	Level of previous analysis (e.g. "SYMBOL" or "PROBE").
postfix	Optional string to append to saved results. Useful if need to run multiple meta-analyses on the same series but with different contrasts.

**Value**

Result of previous call to [diff\\_expr](#).

**Examples**

```
library(lydata)

data_dir <- system.file("extdata", package = "lydata")
gse_names <- c("GSE9601", "GSE34817")
prev <- load_diff(gse_names, data_dir)
```

---

load_raw	<i>Load and annotate raw data downloaded from GEO.</i>
----------	--

---

**Description**

Loads and annotates raw data previously downloaded with [get\\_raw](#). Supported platforms include Affymetrix, Agilent, and Illumina.

**Usage**

```
load_raw(
  gse_names,
  data_dir = getwd(),
  gpl_dir = "..",
  overwrite = FALSE,
  ensql = NULL
)
```



**Arguments**

gse_names	Character vector of GSE names.
data_dir	String specifying directory with GSE folders.
gpl_dir	String specifying parent directory to search for previously downloaded GPL.soft files.
overwrite	Do you want to overwrite saved esets from previous load_raw?
ensql	For development. Path to sqlite file with ENTREZID and SYMBOL columns created in data-raw/entrezdt.

**Value**

List of annotated esets.

**Examples**

```
library(lydata)
data_dir <- system.file("extdata", package = "lydata")
eset <- load_raw("GSE9601", data_dir = data_dir)
```

---

open\_raw\_illum      *Open raw Illumina microarray files.*

---

**Description**

Helper function to open raw Illumina microarray files in order to check that they are formatted correctly. For details on correct format, please see 'Checking Raw Illumina Data' in vignette.

**Usage**

```
open_raw_illum(gse_names, data_dir = getwd())
```

**Arguments**

gse_names	Character vector of Illumina GSE names to open.
data_dir	String specifying directory with GSE folders.

**Value**

Character vector of successfully formatted Illumina GSE names.

**Examples**

```
library(lydata)

# Illumina GSE names
illum_names <- c("GSE50841", "GSE34817", "GSE29689")

# location of raw data
data_dir <- system.file("extdata", package = "lydata")

# open raw data files with default text editor
# open_raw_illum(illum_names)
```

---

```
phenoData.ch2
```

```
Construct AnnotatedDataFrame from Two-Channel ExpressionSet
```

---

**Description**

Construct AnnotatedDataFrame from Two-Channel ExpressionSet

**Usage**

```
phenoData.ch2(eset)
```

**Arguments**

eset                    ExpressionSet with pData for two-channel Agilent array.

**Value**

AnnotatedDataFrame with twice as many rows as eset, one for each channel of each array in order all red then all green.

---

```
prefix_illum_headers    Run prefix on Illumina raw data files
```

---

**Description**

Run prefix on Illumina raw data files

**Usage**

```
prefix_illum_headers(elist_paths)
```

**Arguments**

elist\_paths            Paths to raw Illumina data files

**Value**

Paths to fixed versions of `elist_paths`

---

<code>remove_automated</code>	<i>Remove columns that are automated by data.table</i>
-------------------------------	--

---

**Description**

Auto-named columns start with 'V' followed by the column number.

**Usage**

```
remove_automated(ex)
```

**Arguments**

`ex` data.frame loaded with [fread](#)

**Value**

`ex` with auto-named columns removed.

---

<code>run_limma</code>	<i>Linear model fitting of eset with limma.</i>
------------------------	---

---

**Description**

After selecting control and test samples for a contrast, surrogate variable analysis ([sva](#)) and linear model fitting with [lmFit](#) is performed.

**Usage**

```
run_limma(  
  eset,  
  annot = "SYMBOL",  
  svobj = list(sv = NULL),  
  numsv = 0,  
  filter = TRUE  
)
```

**Arguments**

eset	Annotated eset created by load_raw.
annot	String, column name in fData. For duplicated values in this column, the row with the highest interquartile range across selected samples will be kept. Appropriate values are "SYMBOL" (default - for gene level analysis) or "ENTREZID_HS" (for probe level analysis).
svobj	Surrogate variable analysis results. Returned from <a href="#">run_sva</a> .
numsv	Number of surrogate variables to model.
filter	For RNA-seq. Should genes with low counts be filtered? dseqr shiny app performs this step separately. Should be TRUE (default) if used outside of dseqr shiny app.

**Details**

If analyses need to be repeated, previous results can be reloaded with [readRDS](#) and supplied to the prev\_anal parameter. In this case, previous selections will be reused.

**Value**

List with:

fit	result of <a href="#">lmFit</a> .
mod	model.matrix used for fit

---

run_limma_setup	<i>Setup ExpressionSet for running limma analysis</i>
-----------------	---

---

**Description**

Setup ExpressionSet for running limma analysis

**Usage**

```
run_limma_setup(eset, prev)
```

**Arguments**

eset	ExpressionSet
prev	previous result of call to diff_expr

**Value**

eset ready for run\_limma

---

run_sva	<i>Run surrogate variable analysis</i>
---------	--

---

**Description**

Run surrogate variable analysis

**Usage**

```
run_sva(mods, eset, svanal = TRUE)
```

**Arguments**

mods	result of <a href="#">get_sva_mods</a>
eset	ExpressionSet
svanal	Should surrogate variable analysis be run? If FALSE, returns dummy result.

---

setup_prev	<i>Setup selections when many samples.</i>
------------	--

---

**Description**

Function is useful when number of samples makes manual selection with [diff\\_expr](#) error prone and time-consuming. This is often true for large clinical data sets.

**Usage**

```
setup_prev(eset, contrasts)
```

**Arguments**

eset	List containing one expression set with pData 'group' and 'pair' (optional) columns. Name of eset should be the GSE name.
contrasts	Character vector specifying contrasts to analyse. Each contrast must take the form "B-A" where both "B" and "A" are present in eset pData 'group' column. "B" is the treatment group and "A" is the control group.

**Value**

List containing necessary information for prev\_anal parameter of [diff\\_expr](#).

**Examples**

```

library(lydata)
library(Biobase)

# location of raw data
data_dir <- system.file("extdata", package = "lydata")

# load eset
gse_name <- c("GSE34817")
eset <- load_raw(gse_name, data_dir)

# inspect pData of eset
# View(pData(eset$GSE34817)) # if using RStudio
head(pData(eset$GSE34817)) # otherwise

# get group info from pData (differs based on eset)
group <- pData(eset$GSE34817)$characteristics_ch1.1

# make group names concise and valid
group <- gsub("treatment: ", "", group)
group <- make.names(group)

# add group to eset pData
pData(eset$GSE34817)$group <- group

# setup selections
sel <- setup_prev(eset, contrasts = "LY-DMSO")

# run differential expression analysis
anal <- diff_expr(eset, data_dir, prev_anal = sel)

```

---

symbol\_annot

*Add hgnc symbol to expression set.*


---

**Description**

Function first maps entrez gene ids to homologous human entrez gene ids and then to hgnc symbols.

**Usage**

```
symbol_annot(eset, gse_name = "", ensql = NULL)
```

**Arguments**

eset	Expression set to annotate.
gse_name	GSE name for eset.
ensql	For development. Path to sqlite file with ENTREZID and SYMBOL columns created in data-raw/entrezdt.

**Details**

Initial entrez gene ids are obtained from bioconductor annotation data packages or from feature data of supplied expression set. Homologous human entrez ids are obtained from homologene and then mapped to hgnc symbols using org.Hs.eg.db. Expression set is expanded if 1:many mappings occur.

**Value**

Expression set with hgnc symbols ("SYMBOL") and row names ("PROBE") added to fData slot.

**See Also**

[load\\_raw](#).

**Examples**

```
library(lydata)

# location of raw data
data_dir <- system.file("extdata", package = "lydata")

# load eset
eset <- load_raw("GSE9601", data_dir)[[1]]

# annotate eset (need if load_raw failed to annotate)
eset <- symbol_annot(eset)
```

# Index

## \* datasets

gs.names, [13](#)  
gslist, [14](#)

add\_adjusted, [3](#)  
add\_sources, [3, 8](#)  
add\_vsd, [4](#)

bulkPage, [5](#)  
bulkPageUI, [5](#)

diff\_expr, [3, 4, 6, 6, 8, 16, 21](#)

eBayes, [7, 10](#)  
effectsize, [8, 12](#)  
es\_meta, [7](#)  
exprs.MA, [9, 9](#)

fdrtool, [8](#)  
filter\_genes, [9](#)  
filterByExpr, [9](#)  
fit\_ebayes, [10](#)  
fix\_illum\_headers, [10](#)  
fread, [19](#)

get\_raw, [4, 7, 11, 16](#)  
get\_sva\_mods, [11, 21](#)  
get\_top\_table, [12](#)  
get\_vsd, [13](#)  
getGEO, [10, 15](#)  
gs.names, [13](#)  
gslist, [14](#)

ilmn.nnum, [14](#)  
iqr\_replicates, [15](#)

lmFit, [19, 20](#)  
load\_agil\_plat, [15](#)  
load\_diff, [3, 6–8, 16](#)  
load\_raw, [6, 11, 13, 16, 23](#)

open\_raw\_illum, [17](#)

phenoData.ch2, [9, 18](#)  
plotDensities.MAList, [9](#)  
prefix\_illum\_headers, [18](#)

read.ilmn, [10, 11](#)  
readRDS, [20](#)  
remove\_autonamed, [19](#)  
rlog, [13](#)  
run\_limma, [10, 12, 19](#)  
run\_limma\_setup, [20](#)  
run\_sva, [20, 21](#)  
runApp(), [6](#)

setup\_prev, [21](#)  
sva, [6, 19](#)  
symbol\_annot, [22](#)

topTable, [7](#)  
toptable, [12](#)

vst, [13](#)

zScores, [8](#)