Package ‘crossmeta’

March 30, 2021

Title Cross Platform Meta-Analysis of Microarray Data

Version 1.16.1

Author Alex Pickering

Maintainer Alex Pickering <alexvpickering@gmail.com>

Description Implements cross-platform and cross-species meta-analyses of Affymentrix, 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 seperately, 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

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), data.table (>= 1.10.4), fdrtool (>= 1.2.15), GEOquery (>= 2.40.0), limma (>= 3.30.13), matrixStats (>= 0.51.0), metaMA (>= 3.1.2), miniUI (>= 0.1.1), oligo (>= 1.38.0), reader(>= 1.0.6), RColorBrewer (>= 1.1.2), RCurl (>= 1.95.4.11), RSQLite (>= 2.1.1), randomcoloR (>= 1.1.0.1), stringr (>= 1.2.0), sva (>= 3.22.0), shiny (>= 1.0.0), shinyjs (>= 2.0.0), shinyBS (>= 0.61), shinyWidgets (>= 0.5.3), shinypanel (>= 0.1.0), statmod (>= 1.4.34), XML (>= 3.98.1.17), readxl (>= 1.3.1)

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

git_url https://git.bioconductor.org/packages/crossmeta

git_branch RELEASE_3_12

git_last_commit 02d63d6

git_last_commit_date 2020-11-02

Date/Publication 2021-03-29

R topics documented:

`add_adjusted` *Add expression data adjusted for pairs/surrogate variables*

Description

Add expression data adjusted for pairs/surrogate variables

Usage

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

Arguments

`eset`	ExpressionSet
`svobj`	surrogate variable object

Value

eset with `adjusted` element added

`add_sources` *Add sample source information for meta-analysis.*

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())
```

Arguments

`diff_exprs`	Previous result of `diff_expr`, which can be reloaded using `load_diff`.
`data_dir`	String specifying directory of GSE folders.

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`:

`sources`	Named vector specifying selected sample source for each contrast. Vector names identify the contrast.
`pairs`	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)
```

`ch2_subset` *Subset for Paired Two-Channel ExpressionSet*

Description

Two-channel esets use intraspotCorrelation and lmscFit so can't use duplicateCorrelation. If not using one channel in contrasts (e.g. because all reference RNA) and have paired design, better to treat as single channel so that can use duplicateCorrelation.

Usage

```
ch2_subset(eset, prev_anal)
```

Arguments

`eset`	ExpressionSet

Value

ExpressionSet. If two-channel, paired and one channel not used will subset to used channel.

`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
)
```

Arguments

`esets` List of annotated esets. Created by `load_raw`.

`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 `diff_expr`, which can be reloaded using `load_diff`. 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 `prev_anals`. 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.

Details

The **Samples** tab is used to select control and test samples for each contrast. To do so: select rows for control samples, type a group name in the *Control group name* text input box and click the *Add Group* button. Repeat for test samples. While adding additional contrasts, a previous control group can be quickly reselected from the *Previous selections* dropdown box. 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 selecting sample rows to pair and then clicking *Pair Samples*. The author does not usually specify paired samples and instead allows surrogate variable analysis to discover these inter-sample relationships from the data itself.

The **Contrasts** tab is used to view and delete contrasts that have already been added.

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:

`pdata` data.frame with phenotype data for selected samples. Columns `treatment` ('ctrl' or 'test'), `group`, and `pair` are added based on user selections.

`top_tables` List with results of `topTable` call (one per contrast). These results account for the effects of nuissance variables discovered by surrogate variable analysis.

`ebayes_sv` Results of call to `eBayes` with surrogate variables included in the model matrix.

`annot` Value of `annot` 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
# anals <- 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 `diff_expr`, which can be reloaded using `load_diff`.

`cutoff` Minimum fraction of contrasts that must have measured each gene. Between 0 and 1.

`by_source` Should seperate meta-analyses be performed for each tissue source added with `add_sources`?

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(var).

fdr False discovery rates calculated from column z using fdrtool.

pval p-values calculated from column z using fdrtool.

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 seperate meta-analyses for each source (optional)
# anals <- add_sources(anals, data_dir)

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

exprs.MA *Extract Log-Expression Matrix from MAList*

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.

`fix_illum_headers` *Attempts to fix Illumina raw data header*

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: non_normalized.txt, raw.txt, or _supplementary_.txt
`eset`	ExpressionSet from getGEO.

Value

Character vector for `annotation` argument to read.ilmn. Fixed raw data files are saved with filename ending in _fixed.txt

`get_ch2_mod` *Get design matrix for two-channel array*

Description

Get design matrix for two-channel array

Usage

```
get_ch2_mod(eset)
```

Arguments

`eset`	ExpressionSet with `colnames` that end in '_red' and '_green' indicating channel and `eset$group` indicating group membership.

Value

model matrix for use by intraspotCorrelation and lmscFit

get_raw *Download and unpack microarray supplementary files from GEO.*

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 *Get model matrices for surrogate variable analysis*

Description

Used by add_adjusted to create model matrix with surrogate variables.

Usage

```
get_sva_mods(pdata)
```

Arguments

eset	Annotated eset with samples selected during add_contrasts.

Value

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

`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` *KEGG human pathway genes.*

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` *Count numeric columns in raw Illumina data files*

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` *Removes features with replicated annotation.*

Description

For rows with duplicated annot, highested IQR retained.

Usage

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

Arguments

`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.
`mod`	Model matrix without surrogate variables. generated by `diff_setup`.
`svobj`	Result from `sva` function called during `diff_setup`.

Value

Expression set with unique features at probe or gene level.

`load_agil_plat` *Load Agilent raw data*

Description

Load Agilent raw data

Usage

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

Arguments

`eset`	ExpressionSet from getGEO.
`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` *Load previous differential expression analyses.*

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 *Load and annotate raw data downloaded from GEO.*

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 formated 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`

`query_ref` *Get correlation between query and reference signatures.*

Description

Determines the pearson correlation between the query and each reference signature.

Usage

```
query_ref(query, ref, sorted = TRUE, ngenes = 200)
```

Arguments

`query`	Named numeric vector of differentual expression values for query genes. Usually 'meta' slot of `get_dprimes` result.
`ref`	A matrix of differential expression to query against (rows are genes, columns are samples).
`sorted`	Would you like the results sorted by decreasing similarity? Default is TRUE.
`ngenes`	The number of top differentially-regulated (up and down) query genes to use.

Value

Vector of pearson correlations between query and reference signatures.

run_select_contrasts *Shiny gadget to upload groups and select contrasts*

Description

Shiny gadget to upload groups and select contrasts

Usage

```
run_select_contrasts(
  eset,
  gse_name,
  prev = NULL,
 app_dir = system.file("select_contrasts", package = "crossmeta", mustWork = TRUE),
  port = 3838
)
```

Arguments

`eset`	ExpressionSet
`gse_name`	GEO accession for the series.
`prev`	Previous result of `diff_expr`. Used to allow rechecking previous selections.
`app_dir`	Directory to shiny app. For local development use 'inst/select_contrasts'. Default is in 'select_contrasts' sub directory of crossmeta package.

Value

result of setup_prev. Used to specify sample groups and contrasts for differential expression analysis.

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

# gather GSE names gse_name <- "GSE9601"

# load previous analysis eset <- load_raw(gse_name, data_dir)[[1]] run_select_contrasts(eset, gse_name)

run_sva *Run surrogate variable analysis*

Description

Run surrogate variable analysis

Usage

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

Arguments

`mods`	result of `get_mods`
`eset`	ExpressionSet
`svanal`	Should surrogate variable analysis be run? If `FALSE`, returns dummy result.

setup_prev *Setup selections when many samples.*

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)
```

`to_eset` *Convert limma object to ExpressionSet*

Description

Convert limma object to ExpressionSet

Usage

```
to_eset(object, eset)
```

Arguments

`object`	an EList of MAList object containing expression data.
`eset`	ExpressionSet from getGEO. Used for annotation.

Value

ExpressionSet using expression data from `object` and annotation from `eset`.

`to_ma` *Covert expression values to MAList*

Description

Covert expression values to MAList

Usage

```
to_ma(y)
```

Arguments

`y`	Expression values from two-channel agilent array in order all red then all green.

Value

MAList

Examples

```
A <- matrix(rnorm(100), ncol = 5)
M <- matrix(rnorm(100), ncol = 5)
MA <- new('MAList', list(M=M, A=A))
colnames(MA) <- letters[1:5]

y <- exprs.MA(MA)
MA2 <- to_ma(y)
all.equal(MA, MA2)
```

which_max_iqr *Get row indices of maximum IQR within annotation groups*

Description

Groups by group_by and determines row with maximum IQR.

Usage

```
which_max_iqr(eset, groub_by, x = exprs(eset))
```

Arguments

eset	ExpressionSet
groub_by	Column in fData(eset) to group by
x	matrix of expression values to use for IQR

Value

Integer vector of row numbers representing rows with the maximum IQR after grouping by group_by

xls_to_txt *Covert .xls files to .txt*

Description

For converting Illumina _Supplementary_.*.xls files to .txt for load_illum_plat.

Usage

```
xls_to_txt(xls_paths)
```

Arguments

xls_paths	Paths to .xls files

Index