Package: topconfects (via r-universe)

October 30, 2026

Title Top Confident Effect Sizes

Version 1.29.0

Description Rank results by confident effect sizes, while maintaining False Discovery Rate and False Coverage-statement Rate control. Topconfects is an alternative presentation of TREAT results with improved usability, eliminating p-values and instead providing confidence bounds. The main application is differential gene expression analysis, providing genes ranked in order of confident log2 fold change, but it can be applied to any collection of effect sizes with associated standard errors.

Depends R (>= 3.6.0)

Imports methods, utils, stats, assertthat, ggplot2, scales, grid, grDevices

Suggests limma, edgeR, statmod, DESeq2, ashr, NBPSeq, dplyr, testthat, reshape2, tidyr, readr, org.At.tair.db, AnnotationDbi, knitr, rmarkdown, BiocStyle

License LGPL-2.1 | file LICENSE

URL https://github.com/pfh/topconfects

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

BugReports https://github.com/pfh/topconfects/issues

biocViews GeneExpression, DifferentialExpression, Transcriptomics, RNASeq, mRNAMicroarray, Regression, MultipleComparison

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 12:49:28 UTC

RemoteUrl https://github.com/bioc/topconfects

RemoteRef HEAD

RemoteSha 6e9f2dba59bc40ab732a4c48fe15cadef0a3010c

Contents

confects_plot 2
confects_plot_me 3
confects_plot_me2 4
deseq2_confects 5
edger_confects 6
limma_confects 8
nest_confects 9
normal_confects 11
rank_rank_plot 12

Index **13**

`confects_plot` *Top confident effect sizes plot*

Description

Create a ggplot2 object showing the confect, effect, and average expression level of top features in a Topconfects object.

Usage

```
confects_plot(confects, n = 50, limits = NULL)
```

Arguments

`confects`	A "Topconfects" class object, as returned from limma_confects, edger_confects, etc.
`n`	Number if items to show.
`limits`	c(lower, upper) limits on x-axis.

Details

For each gene, the estimated effect is shown as a dot. The confidence bound is shown as a line to positive or negative infinity, showing the set of non-rejected effect sizes for the feature.

Value

A ggplot2 object. Working non-interactively, you must print() this for it to be displayed.

Examples

```
# Generate some random effect sizes with random accuracies
n <- 100
effect <- rnorm(n, sd=2)
se <- rchisq(n, df=3)^-0.5

# Find top confident effect sizes
confects <- normal_confects(effect, se)

# Plot top confident effect sizes
confects_plot(confects, n=30)
```

---

confects_plot_me *Mean-expression vs effect size plot (deprecated)*

Description

Note: I now recommend using `plot_confects_me2` instead of this plot. Like plotMD in limma, plots effect size against mean expression level. However shows "confect" on the y axis rather than "effect" ("effect" is shown underneath in grey). This may be useful for assessing whether effects are only being detected in highly expressed genes.

Usage

```
confects_plot_me(confects)
```

Arguments

`confects`	A "Topconfects" class object, as returned from `limma_confects`, `edger_confects`, or `deseq2_confects`.

Value

The two types of points in this plot make it quite confusing to explain. `plot_confects_me2` is recommended instead.

A ggplot2 object. Working non-interactively, you must print() this for it to be displayed.

Examples

```
library(NBPSeq)
library(edgeR)
library(limma)

data(arab)

# Extract experimental design from sample names
treat <- factor(substring(colnames(arab),1,4), levels=c("mock","hrcc"))
```

```
time <- factor(substring(colnames(arab),5,5))

# Keep genes with at least 3 samples having an RPM of more than 2
y <- DGEList(arab)
keep <- rowSums(cpm(y)>2) >= 3
y <- y[keep,,keep.lib.sizes=FALSE]
y <- calcNormFactors(y)

# Find top confident fold changes by topconfects-limma-voom method
design <- model.matrix(~time+treat)
voomed <- voom(y, design)
fit <- lmFit(voomed, design)
confects <- limma_confects(fit, "treathrcc")

# Plot confident effect size against mean expression
# (estimated effect size also shown as grey dots)
confects_plot_me(confects)
```

`confects_plot_me2` *Mean-expression vs effect size plot (version 2)*

Description

Like plotMD in limma, plots effect size against mean expression level. Confect values are indicated using color. This may be useful for assessing whether effects are only being detected in highly expressed genes.

Usage

```
confects_plot_me2(confects, breaks = NULL, signed_colors = TRUE)
```

Arguments

`confects`	A "Topconfects" class object, as returned from `limma_confects`, `edger_confects`, or `deseq2_confects`.
`breaks`	A vector of non-negative confect thresholds to color points with. Chooses a sensible set of breaks if none are given.
`signed_colors`	Should positive and negative confects be colored distinctly, red and blue.

Value

A ggplot2 object. Working non-interactively, you must print() this for it to be displayed.

Examples

```
library(NBPSeq)
library(edgeR)
library(limma)

data(arab)

# Extract experimental design from sample names
treat <- factor(substring(colnames(arab),1,4), levels=c("mock","hrcc"))
time <- factor(substring(colnames(arab),5,5))

# Keep genes with at least 3 samples having an RPM of more than 2
y <- DGEList(arab)
keep <- rowSums(cpm(y)>2) >= 3
y <- y[keep,,keep.lib.sizes=FALSE]
y <- calcNormFactors(y)

# Find top confident fold changes by topconfects-limma-voom method
design <- model.matrix(~time+treat)
voomed <- voom(y, design)
fit <- lmFit(voomed, design)
confects <- limma_confects(fit, "treathrcc")

# Plot confident effect size against mean expression
# (estimated effect size also shown as grey dots)
confects_plot_me2(confects)
```

`deseq2_confects` *Confident log2 fold changes based on a DESeq2 analysis*

Description

For all possible absolute log2 fold changes, which genes have at least this fold change at a specified False Discovery Rate? This is built by repeatedly calling DESeq2::results with the "greaterAbs" alternative hypothesis.

Usage

```
deseq2_confects(object, ..., fdr = 0.05, step = 0.01)
```

Arguments

`object`	Object produced by the `DESeq2::DESeq` function.
`...`	Further arguments to `DESeq2::results`. At a minimum you should specify either `contrast=` or `name=`.
`fdr`	False Discovery Rate to control for.
`step`	Granularity of log2 fold changes to test.

Details

Results are presented in a table such that for any given LFC, if the reader chooses the genes with abs(confect) less than this they are assured that this set of genes has at least this LFC (with the specified FDR). The confect column may also be viewed as a confidence bound on the LFC of each gene, with a dynamic correction for multiple testing.

Value

See `nest_confects` for details of how to interpret the result.

The `filtered` column in the result indicates whether DESeq2 filtered the gene. Such genes do not count toward the total number of genes when controlling FDR. If your intention is to obtain a ranking of all genes, you should disable this with `deseq2_confects(..., cooksCutoff=Inf,independentFiltering=FALSE)`.

Examples

```
# Generate some random data
n <- 20
folds <- seq(-8,8,length.out=n)
row_means <- runif(n, min=0, max=5)
lib_scale <- c(1,2,3,4)
means <- 2^(outer(folds, c(-0.5,-0.5,0.5,0.5))) *
    row_means * rep(lib_scale,each=n)
counts <- rnbinom(length(means), mu=means, size=1/0.1)
dim(counts) <- dim(means)

group <- factor(c("A","A","B","B"))

# Apply DESeq2
library(DESeq2)
dds <- DESeqDataSetFromMatrix(
    countData = counts,
    colData = data.frame(group=group),
    design = ~group)

dds <- DESeq(dds)

# Find top confident effect sizes
deseq2_confects(dds, name="group_B_vs_A", step=0.1)
```

---

edger_confects *Confident log2 fold change based on the edgeR Quasi-Likelihood method*

Description

For all possible absolute log2 fold changes (LFC), which genes have at least this fold change at a specified False Discovery Rate?

Usage

```
edger_confects(
  fit,
  coef = NULL,
  contrast = NULL,
  fdr = 0.05,
  step = 0.01,
  null = c("worst.case", "interval")
)
```

Arguments

fit	An edgeR DGEGLM object produced using glmQLFit.
coef	Coefficient to test, as per glmTreat. Use either coef or contrast or effect.
contrast	Contrast to test, as per glmTreat. Use either coef or contrast or effect.
fdr	False Discovery Rate to control for.
step	Granularity of log2 fold changes to test.
null	"null" parameter passed through to edger::glmTreat (if coef or contrast given). Choices are "worst.case" or "interval". Note that the default here is "worst.case", to be consistent with other functions in topconfects. This differs from the default for glmTreat.

Details

Results are presented in a table such that for any given LFC, if the reader chooses the genes with abs(confect) less than this they are assured that this set of genes has at least this LFC (with the specified FDR). The confect column may also be viewed as a confidence bound on the LFC of each gene, with a dynamic correction for multiple testing.

Value

See nest_confects for details of how to interpret the result.

Examples

```
# Generate some random data
n <- 100
folds <- seq(-4,4,length.out=n)
row_means <- runif(n, min=0, max=5)
lib_scale <- c(1,2,3,4)
means <- 2^(outer(folds, c(-0.5,-0.5,0.5,0.5))) *
    row_means * rep(lib_scale,each=n)
counts <- rnbinom(length(means), mu=means, size=1/0.1)
dim(counts) <- dim(means)

design <- cbind(c(1,1,0,0), c(0,0,1,1))

# Fit data using edgeR quasi-likelihood
```

```
library(edgeR)
y <- DGEList(counts)
y <- calcNormFactors(y)
y <- estimateDisp(y, design)
fit <- glmQLFit(y, design)

# Find top confident effect sizes
edger_confects(fit, contrast=c(-1,1))
```

`limma_confects` *Confident log2 fold changes based on a limma fit object*

Description

For all possible absolute log2 fold changes (LFC), which genes have at least this fold change at a specified False Discovery Rate (FDR)?

Usage

```
limma_confects(
  fit,
  coef = NULL,
  fdr = 0.05,
  step = 0.001,
  trend = FALSE,
  full = FALSE
)
```

Arguments

`fit`	A limma MArrayLM object.
`coef`	Number or name of coefficient or contrast to test.
`fdr`	False Discovery Rate to control for.
`step`	Granularity of log2 fold changes to test.
`trend`	Should `eBayes(trend=TRUE)` be used?
`full`	Include some further statistics used to calculate confects in the output, and also include FDR-adjusted p-values that effect size is non-zero (note that this is against the spirit of the topconfects approach).

Details

Results are presented in a table such that for any given LFC, if the reader chooses the genes with `abs(confect)` less than this they are assured that this set of genes has at least this LFC (with the specified FDR). Once this set of genes is selected, the confect values provide confidence bounds with False Coverage-statement Rate at the same level as the FDR.

`fit` should be produced using `lmFit`. It is not necessary to use `eBayes`, this function calls `eBayes` itself.

To test contrasts, this function can also be used with the result of `contrasts.fit`, but limma's handling of weights may be approximate (for example if `voom` has been used). For exact results for a contrast, use `contrastToCoef` to adjust the design matrix given to `lmFit`.

Value

See `nest_confects` for details of how to interpret the result.

Examples

```
#Prepare a data set
library(NBPSeq)
library(edgeR)
library(limma)
data(arab)
dgelist <- DGEList(arab)
dgelist <- calcNormFactors(dgelist)
cpms <- cpm(dgelist, log=TRUE)
# Retain genes with more than a geometric mean of 2 RPM
# (about 5 reads per sample)
cpms <- cpms[rowMeans(cpms) >= 1,]

# Fit linear model for each gene
treatment <- c(FALSE,FALSE,FALSE,TRUE,TRUE,TRUE)
batch <- factor(c(1,2,3,1,2,3))
design <- model.matrix(~ treatment + batch)
fit <- lmFit(cpms, design)

# Calculate top confects
# As voom has not been used, it is necessary to use trend=TRUE
limma_confects(fit, "treatmentTRUE", trend=TRUE)
```

---

nest_confects *General purpose function to find confident effect sizes, controlling FDR*

Description

Find sets of discoveries for a range of effect sizes, controlling the False Discovery Rate (FDR) for each set.

Usage

```
nest_confects(n, pfunc, fdr = 0.05, step = 0.001, full = FALSE)
```

Arguments

n	Number of items being tested.
pfunc	A function(indices, effect_size) to calculate p-values. Indices is a subset of 1:n giving the p-values to be computed. Should return a numeric vector of length length(indices). May return NA values. If an NA value is returned, NA must be returned for all effect sizes. NA values are not counted toward the total number of tests performed.
fdr	False Discovery Rate to control for.
step	Granularity of effect sizes to test.
full	If TRUE, also include FDR-adjusted p-value that effect size is non-zero. Note that this is against the spirit of the topconfects approach.

Details

This is a general purpose function, which can be applied to any method of calculting p-values (supplied as a function argument) for the null hypothesis that the effect size is smaller than a given amount.

Value

A "Topconfects" object, containing a table of results and various associated information.

The most important part of this object is the $table element, a data frame with the following columns:

- rank - Ranking by confect and for equal confect by p-value at that effect size.
- index - Number of the test, between 1 and n.
- confect - CONfident efFECT size.

The usage is as follows: To find a set of tests which have effect size greater than x with the specified FDR, take the rows with abs(confect) >= x. Once the set is selected, the confect values provide confidence bounds on the effect size with False Coverage-statement Rate (FCR) at the same level as the FDR.

One may essentially take the top however many rows of the data frame and these will be the best set of results of that size to dependably have an effect size that is as large as possible. However if some genes have the same abs(confect), all or none should be selected.

Some rows in the output may be given the same confect, even if step is made small. This is an expected behaviour of the algorithm. (This is similar to FDR adjustment of p-values sometimes resulting in a run of the same adjusted p-value, even if all the input p-values are distinct.)

Some wrappers around this function may add a sign to the confect column, if it makes sense to do so. They will also generally add an effect column, containing an estimate of the effect size that aims to be unbiassed rather than a conservative lower bound.

Examples

```
# Find largest positive z-scores in a collection,
# and place confidence bounds on them that maintain FDR 0.05.
z <- c(1,2,3,4,5)
pfunc <- function(i, effect_size) {
    pnorm(z[i], mean=effect_size, lower.tail=FALSE)
}
nest_confects(length(z), pfunc, fdr=0.05)
```

`normal_confects` *Confident effect sizes from from normal or t distributions*

Description

A general purpose confident effect size function for where a normal or t distribution of errors can be assumed. Calculates confident effect sizes based on an estimated effect and standard deviation (normal distribution), or mean and scale (t distribution).

Usage

```
normal_confects(
  effect,
  se,
  df = Inf,
  signed = TRUE,
  fdr = 0.05,
  step = 0.001,
  full = FALSE
)
```

Arguments

`effect`	A vector of estimated effects.
`se`	A single number or vector of standard errors (or if t distribution, scales).
`df`	A single number or vector of degrees of freedom, for t-distribution. Inf for normal distribution.
`signed`	If TRUE effects are signed, use TREAT test. If FALSE effects are all positive, use one sided t-test.
`fdr`	False Discovery Rate to control for.
`step`	Granularity of effect sizes to test.
`full`	Include some further statistics used to calculate confects in the output, and also include FDR-adjusted p-values that effect size is non-zero (note that this is against the spirit of the topconfects approach).

Value

See nest_confects for details of how to interpret the result.

Examples

```
# Find largest positive or negative z-scores in a collection,
# and place confidence bounds on them that maintain FDR 0.05.
z <- c(1,-2,3,-4,5)
normal_confects(z, se=1, fdr=0.05, full=TRUE)
```

---

rank_rank_plot *A plot to compare two rankings*

Description

This is useful, for example, when comparing different methods of ranking potentially interesting differentially expressed genes.

Usage

```
rank_rank_plot(
  vec1,
  vec2,
  label1 = "First ranking",
  label2 = "Second ranking",
  n = 40
)
```

Arguments

vec1	A vector of names.
vec2	Another vector of names.
label1	A label to go along with vec1.
label2	A label to go along with vec2.
n	Show at most the first n names in vec1 and vec2.

Value

A ggplot2 object. Working non-interactively, you must print() this for it to be displayed.

Examples

```
a <- sample(letters)
b <- sample(letters)
rank_rank_plot(a,b, n=20)
```

Index

confects_plot, 2
confects_plot_me, 3
confects_plot_me2, 4

deseq2_confects, 5

edger_confects, 6

limma_confects, 8

nest_confects, *6*, *7*, 9, *9*, *12*
normal_confects, 11

rank_rank_plot, 12