Package ‘CBEA’

May 15, 2024

Title Competitive Balances for Taxonomic Enrichment Analysis in R

Version 1.5.0

Date 2022-03-03

Description This package implements CBEA, a method to perform set-based analysis for microbiome relative abundance data. This approach constructs a competitive balance between taxa within the set and remainder taxa per sample. More details can be found in the Nguyen et al. 2021+ manuscript. Additionally, this package adds support functions to help users perform taxa-set enrichment analyses using existing gene set analysis methods. In the future we hope to also provide curated knowledge driven taxa sets.

License MIT + file LICENSE

URL https://github.com/qpmnguyen/CBEA, https://qpmnguyen.github.io/CBEA/

BugReports https://github.com/qpmnguyen/CBEA//issues

Depends R (>= 4.2.0)

Imports BiocParallel, BiocSet, dplyr, lmom, fitdistrplus, magrittr, methods, mixtools, Rcpp (>= 1.0.7), stats, SummarizedExperiment, tibble, TreeSummarizedExperiment, tidyr, glue, generics, rlang, goftest

Suggests phyloseq, BiocStyle, covr, knitr, RefManageR, rmarkdown, sessioninfo, testthat (>= 3.0.0), tidyverse, roxygen2, mia, purrr

LinkingTo Rcpp

VignetteBuilder knitr

biocViews Software, Microbiome, Metagenomics, GeneSetEnrichment, DataImport

Config/testthat/edition 3

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

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

git_branch devel

git_last_commit 0234eb6

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-05-15

Author Quang Nguyen [aut, cre] (<https://orcid.org/0000-0002-2072-3279>)

Maintainer Quang Nguyen <quangpmnguyen@gmail.com>

Contents

.cbea — *Internal cbea function*

Description

See main function cbea documentation for more details.

Usage

```
.cbea(
  ab_tab,
  set_list,
  output,
  distr,
  adj = FALSE,
  n_perm = 100,
  parametric = TRUE,
  thresh = 0.05,
  init = NULL,
  control = NULL,
  parallel_backend = NULL,
  ...
)
```

Arguments

ab_tab	(Matrix). Named n by p matrix. This is the OTU/ASV/Strain table where taxa are columns.
set_list	(List). List of length m. This is a list of set membership by column names.
output	See documentation cbea
distr	See documentation cbea
adj	See documentation cbea
n_perm	See documentation cbea
parametric	See documentation cbea
thresh	See documentation cbea
init	See documentation cbea
control	See documentation cbea
parallel_backend	See documentation cbea
...	See documentation cbea

Value

A data.frame of size n by m. n is the total number of samples and m is the total number of sets with elements represented in the data.

cbea *Enrichment analysis using competitive compositional balances (CBEA)*

Description

cbea is used compute enrichment scores per sample for pre-defined sets using the CBEA (Competitive Balances for Enrichment Analysis).

Usage

```
cbea(
  obj,
  set,
  output,
  distr = NULL,
  adj = FALSE,
  n_perm = 100,
  parametric = TRUE,
  thresh = 0.05,
  init = NULL,
  control = NULL,
  parallel_backend = NULL,
  ...
)

## S4 method for signature 'TreeSummarizedExperiment'
cbea(
  obj,
  set,
  output,
  distr = NULL,
  abund_values,
  adj = FALSE,
  n_perm = 100,
  parametric = TRUE,
  thresh = 0.05,
  init = NULL,
  control = NULL,
  parallel_backend = NULL,
  ...
)

## S4 method for signature 'data.frame'
cbea(
  obj,
  set,
```

```
  taxa_are_rows = FALSE,
  id_col = NULL,
  output,
  distr = NULL,
  adj = FALSE,
  n_perm = 100,
  parametric = TRUE,
  thresh = 0.05,
  init = NULL,
  control = NULL,
  parallel_backend = NULL,
  ...
)

## S4 method for signature 'matrix'
cbea(
  obj,
  set,
  taxa_are_rows = FALSE,
  output,
  distr = NULL,
  adj = FALSE,
  n_perm = 100,
  parametric = TRUE,
  thresh = 0.05,
  init = NULL,
  control = NULL,
  parallel_backend = NULL,
  ...
)
```

Arguments

obj	The element of class `TreeSummarizedExperiment`, `data.frame`, or `matrix`. `phyloseq` is not supported due to conflicting dependencies and `TreeSummarizedExperiment` is much more compact.
set	`BiocSet`. Sets to be tested for enrichment in the `BiocSet` format. Taxa names must be in the same format as elements in the set.
output	(String). The form of the output of the model. Has to be either `zscore`, `cdf`, `raw`, `pval`, or `sig`
distr	(String). The choice of distribution for the null. Can be either `mnorm` (2 component mixture normal), `norm` (Normal distribution), or NULL if `parametric` is `TRUE`.
adj	(Logical). Whether correlation adjustment procedure is utilized. Defaults to FALSE.
n_perm	(Numeric). Add bootstrap resamples to both the permuted and unpermuted data set. This might help with stabilizing the distribution fitting procedure, especially if the sample size is low. Defaults to 1.

parametric	(Logical). Indicate whether a parametric distribution will be fitted to estimate z-scores, CDF values, and p-values. Defaults to TRUE
thresh	(Numeric). Threshold for significant p-values if sig is the output. Defaults to 0.05
init	(Named List). Initialization parameters for estimating the null distribution. Default is NULL.
control	(Named List). Additional arguments to be passed to fitdistr and normmixEM. Defaults to NULL.
parallel_backend	See documentation cbea
...	Additional arguments not used at the moment.
abund_values	(Character). Character value for selecting the assay to be the input to cbea
taxa_are_rows	(Logical). Indicate whether the data frame or matrix has taxa as rows
id_col	(Character Vector). Vector of character to indicate metadata columns to keep (for example, sample_id)

Details

This function support different formats of the OTU table, however for best results please use TreeSummarizedExperiment. phyloseq is supported, however CBEA will not explicitly import phyloseq package and will require users to install them separately. If use data.frame or matrix, users should specify whether taxa are rows using the taxa_are_rows option. Additionally, for data.frame, users can specify metadata columns to be kept via the id_col argument.
The output argument specifies what type of values will be returned in the final matrix. The options pval or sig returns either unadjusted p-values or dummy variables indicating whether a set is significantly enriched in that sample (based on unadjusted p-values thresholded at thresh). The option raw returns raw scores computed for each set without any distribution fitting or inference procedure. Users can use this option to examine the distribution of CBEA scores under the null.

Value

R An n by m matrix of enrichment scores at the sample level

Examples

```
data(hmp_gingival)
seq <- hmp_gingival$data
set <- hmp_gingival$set
# n_perm = 10 to reduce runtime
mod <- cbea(obj = seq, set = set, output = "zscore",
    abund_values = "16SrRNA",
    distr = "norm", parametric = TRUE,
    adj = TRUE, thresh = 0.05, n_perm = 10)
```

check_args — *Checking arguments of the function*

Description

This function extracts the parent environment (when called under the cbea function) and then check all the arguments.

Usage

```
check_args()
```

Value

None

check_distr_arg — *This function checks for validity of arguments based on the parameters and the distribution of interest*

Description

This function checks for validity of arguments based on the parameters and the distribution of interest

Usage

```
check_distr_arg(param, distr, .note = NULL)
```

Arguments

`param`	(List). Named list of parameter values
`distr`	(String). String name of the distribution being evaluated
`.note`	(String). Any additional annotation to be put in front of error messages

Value

Returns 0 if there are no errors

combine_distr — *Combining two distributions*

Description

Pass along handling of combining distributions to avoid clogging up the main function

Usage

```
combine_distr(perm, unperm, distr, ...)
```

Arguments

perm	(List). A list of parameters for permuted distribution
unperm	(List). A list of parameters for the unpermuted distribtion
distr	(String). Distribution of choice

Value

A list of the combined distribution form based on the initial distribution of choice

dlst — *Defintions for location-scale t distribution*

Description

Internal functions for defining the t-distribution in terms of location-scale.

Usage

```
dlst(x, df = 1, mu = 0, sigma = 1, log = FALSE)

plst(q, df = 1, mu = 0, sigma = 1, log = FALSE)
```

Arguments

x, q	The data vector
df	Degrees of freedom
mu	The location parameter
sigma	The scale parameter
log	Indicate whether probabilities are return as log

Value

Numeric values representing the density and cumulative probabilitiy values of the location-scale t distribution

Functions

- `dlst`: Probability Density Function
- `plst`: Cumulative distribution function

Examples

```
val <- rnorm(10)
dlst(val, df = 1, mu = 0, sigma = 1)
val <- rnorm(10)
plst(q = val, df = 1, mu = 0, sigma = 1)
```

`estimate_distr` *Estimate distribution parameters from data*

Description

This function takes a numeric vector input and attempts to find the most optimal solution for the parameters of the distribution of choice. Right now only `norm` and `mnorm` distributions are supported.

Usage

```
estimate_distr(data, distr, init = NULL, args_list = NULL)
```

Arguments

`data`	(Numeric Vector). A vector of numbers that can be inputted to estimate the parameters of the distributional forms.
`distr`	(String). The distribution to be fitted. Right now only `norm` or `mnorm` is supported
`init`	(List). Initialization parameters for each distribution. For mixtures, each named element in the list should be a vector with length equal to the number of components
`args_list`	(List). Named list of additional arguments passed onto fitdist and normalmixEM
`...`	Other paremteres passed to fitdistrplus or normalmixEM

Details

The package `fitdistrplus` is used to estimate parameters of the normal distribution while the package `normalmixEM` is used to estimate parameters of the mixture normal distribution. So far we suggest only estimating two components for the mixture normal distribution. For default options, we use mostly defaults from the packages themselves. The only difference was the mixture normal distribution where the convergence parameters were loosened and requiring more iterations to converge.

Value

A named list with all the parameter names and values

---

fit_scores — *Function to compute CBEA scores for each set*

Description

Function to compute CBEA scores for each set

Usage

```
fit_scores(
  index_vec,
  ab_tab,
  adj,
  distr,
  output,
  n_perm,
  parametric,
  thresh,
  init,
  control
)
```

Arguments

index_vec	(Character Vector). A character vector indicating the elements of the set of interest
ab_tab	(Matrix). Named n by p matrix. This is the OTU/ASV/Strain table where taxa are columns.
adj	(Logical). See documentation cbea
distr	(Character). See documentation cbea
output	(Character). See documentation cbea
n_perm	(Numeric). The total number of permutations.
parametric	(Logical). See documentation cbea
thresh	(Numeric). See documentation cbea
init	(List). See documentation cbea
control	(List). See documentation cbea

Value

This function returns a list containing output scores and other diagnostics (as sublists)

get_adj_mnorm — *Function to perform the adjustment for the mixture normal distribution*

Description

Function to perform the adjustment for the mixture normal distribution

Usage

```
get_adj_mnorm(perm, unperm, verbose = FALSE, fix_comp = "none")
```

Arguments

`perm`	(List). Parameter values of the distribution of scores
`unperm`	(List). Parameter values of the distribution of scores computed on unpermuted data
`fix_comp`	(Character). Which component to keep

Value

A List of parameters for the adjusted mixture normal.

get_diagnostics — *Get diagnostic values using parent environment.*

Description

This function is used internally inside fit_scores to grab the relevant objects from the previous parent environment (i.e. the environment from fit_scores) and compute relevant information. The role of this function is break diagnostic component into a different function for maintenance.

Usage

```
get_diagnostics(env = caller_env())
```

Value

This function returns a list of two components: `diagnostic` represent goodness-of-fit statistics for the distribution fitting itself while `lmoment` contains the l-moment comparisons between the computed raw scores, permuted scores, and other fitted distributions.

get_mean — *Get the overall mean of a two component mixture distribution*

Description

Get the overall mean of a two component mixture distribution

Usage

```
get_mean(mu, lambda)
```

Arguments

mu	(Vector). A two value vector of mean values.
lambda	(Vector). A two value vector of component mixing coefficients

Value

A numeric value representing the overall mean

get_raw_score — *Get CBEA scores for a given matrix and a vector of column indices*

Description

Get CBEA scores for a given matrix and a vector of column indices

Usage

```
get_raw_score(X, idx)
```

Arguments

X	(Matrix). OTU table of matrix format where taxa are columns and samples are rows
idx	(Integer vector). Vector of integers indicating the column ids of taxa in a set

Value

A matrix of size n by 1 where n is the total number of samples

Examples

```
data(hmp_gingival)
seq <- hmp_gingival$data
seq_matrix <- SummarizedExperiment::assays(seq)[[1]]
seq_matrix <- t(seq_matrix) + 1
rand_set <- sample(seq_len(ncol(seq_matrix)), size = 10)
scores <- get_raw_score(X = seq_matrix, idx = rand_set)
```

---

get_sd — *Get the overall standard deviation of a two component mixture distribution*

Description

Get the overall standard deviation of a two component mixture distribution

Usage

```
get_sd(sigma, mu, mean, lambda)
```

Arguments

sigma	(Vector). A two value vector of component-wise variances
mu	(Vector). A two value vector of mean values.
mean	(Numeric Value). The overall mean.
lambda	(Vector). A two value vector of component mixing coefficients

Value

A numeric value representing the overall standard deviation

---

glance.CBEAout — *Glance at* CBEAout *object*

Description

This function cleans up all diagnostics of the cbea method (from the CBEAout object) into a nice tibble::tibble()

Usage

```
## S3 method for class 'CBEAout'
glance(x, statistic, ...)
```

Arguments

x	An object of type CBEAout
statistic	What type of diagnostic to return. Users can choose to return fit_diagnostic which returns goodness of fit statistics for the different fitted distributions (e.g. log likelihoods) while fit_comparison returns comparisons across different distributions and raw values (and data) across the 4 l-moments.
...	Unused, kept for consistency with generics

Value

A tibble::tibble() summarizing diagnostic fits per set (as row)

Examples

```
# load the data
data(hmp_gingival)
mod <- cbea(hmp_gingival$data, hmp_gingival$set, abund_values = "16SrRNA",
    output = "sig", distr = "norm", adj = FALSE, n_perm = 5, parametric = TRUE)
glance(mod, "fit_diagnostic")
```

---

gmean *Geometric mean of a vector*

Description

Compute geometric mean of a vector using exp(mean(log(.x))) format

Usage

```
gmean(vec)
```

Arguments

vec	A vector of values with length n

Value

A numeric value of the geometric mean of the vector vec

Examples

```
ex <- abs(rnorm(10))
gmean(ex)
```

gmeanRow — *Geometric mean of rows of a matrix*

Description

This function computes the geometric mean by row of a numeric matrix

Usage

```
gmeanRow(X)
```

Arguments

X	A numeric matrix with n rows and p columns

Value

A numeric vector of the geometric mean of the matrix X with length n

Examples

```
ex <- matrix(rnorm(100), nrow = 10, ncol = 10)
ex <- abs(ex)
gmeanRow(ex)
```

hmp_gingival — *Gingival data set from the Human Microbiome Project*

Description

Gingival data set from the Human Microbiome Project

Usage

```
data(hmp_gingival)
```

Format

A list with two elements

data The microbiome relative abundance data with relevant metadata obtained from the Human Microbiome Project via the HMP16SData package (snapshot: 11-15-2021). The data set is hosted the container of type phyloseq. Using the mia package users can convert it to the TreeSummarizedExperiment type.

set Sets of microbes based on their metabolism annotation at the Genera level. Annotations obtained via Calagaro et al.'s repository on Zenodo (https://doi.org/10.5281/zenodo.3942108)

References

Data can be downloaded directly from https://hmpdacc.org/hmp/

R interface of the data from https://doi.org/doi:10.18129/B9.bioc.HMP16SData

Beghini F, Renson A, Zolnik CP, Geistlinger L, Usyk M, Moody TU, et al. Tobacco Exposure Associated with Oral Microbiota Oxygen Utilization in the New York City Health and Nutrition Examination Study. Annals of Epidemiology. 2019;34:18–25.e3. doi:10.1016/j.annepidem.2019.03.005

Consortium THMP, Huttenhower C, Gevers D, Knight R, Abubucker S, Badger JH, et al. Structure, Function and Diversity of the Healthy Human Microbiome. Nature. 2012;486(7402):207–214. doi:10.1038/nature11234.

Calgaro M, Romualdi C, Waldron L, Risso D, Vitulo N. Assessment of Statistical Methods from Single Cell, Bulk RNA-Seq, and Metagenomics Applied to Microbiome Data. Genome Biology. 2020;21(1):191. doi:10.1186/s13059-020-02104-1

Schiffer L, Azhar R, Shepherd L, Ramos M, Geistlinger L, Huttenhower C, et al. HMP16SData: Efficient Access to the Human Microbiome Project through Bioconductor. American Journal of Epidemiology. 2019;doi:10.1093/aje/kwz006.

`merge_lists` *This function handles the ability to merge supplied and defaults*

Description

This function handles the ability to merge supplied and defaults

Usage

```
merge_lists(defaults, supplied)
```

Arguments

`defaults`	(List). Default options
`supplied`	(List). Supplied options

Value

A merged list

new_CBEAout — *Creating an output object of type CBEAout*

Description

This function takes a list of lists from each object and turns it into a CBEAout type object

Usage

```
new_CBEAout(out, call)
```

Arguments

`out`	A list containing scores for each set
`call`	A list containing all important arguments for printing

Value

A new CBEAout object (which is a cleaner list of lists)

pmnorm — *The Two Component Mixture Normal Distribution*

Description

The Two Component Mixture Normal Distribution

Usage

```
pmnorm(q, mu, sigma, lambda, log = FALSE, verbose = FALSE)

dmnorm(x, mu, sigma, lambda, log = FALSE, verbose = FALSE)
```

Arguments

`q, x`	(Vector). Values to calculate distributional values of.
`mu`	(Vector). A two value vector of mean values.
`sigma`	(Vector). A two value vector of component-wise variances
`lambda`	(Vector). A two value vector of component mixing coefficients
`log`	(Boolean). Whether returning probabilities are in log format
`verbose`	(Boolean). Whether to return component values.

Value

A numeric value representing the probability density value of a two-component mixture distribution

Functions

- pmnorm: Cumulative Distribution Function
- dmnorm: Probability Density Function

Examples

```
library(mixtools)
lambda <- c(0.7,0.3)
mu <- c(1,2)
sigma <- c(1,1)
v <- rnormmix(100, lambda=lambda, mu=mu, sigma=sigma)
pmnorm(v, lambda=lambda,mu=mu,sigma=sigma)
dmnorm(v, lambda=lambda,mu=mu,sigma=sigma)
```

---

print.CBEAout *Print dispatch for CBEAout objects*

Description

Print dispatch for CBEAout objects

Usage

```
## S3 method for class 'CBEAout'
print(x, ...)
```

Arguments

x	The CBEAout object
...	Undefined arguments, keeping consistency for generics

Value

Text for printing

---

reexports *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

generics glance, tidy

scale_scores — *Scaling scores based on estimated null distribution*

Description

Scaling scores based on estimated null distribution

Usage

```
scale_scores(scores, method, param, distr, thresh = 0.05)
```

Arguments

scores	(Numeric Vector). Raw CBEA scores generated without permutations
method	(String). The final form that the user want to return. Options include cdf, zscore, pval and sig.
param	(List). The parameters of the estimated null distribution. Names must match distribution.
thresh	(Numeric). The threshold to decide whether a set is significantly enriched. Only available if method is sig

Value

A vector of size n where n is the sample size

tidy.CBEAout — *Tidy a CBEAout object*

Description

This function takes in a CBEA type object and collects all values across all sets and samples that were evaluated.

Usage

```
## S3 method for class 'CBEAout'
tidy(x, ...)
```

Arguments

x	A CBEAout object.
...	Unused, included for generic consistency only.

Value

A tidy tibble::tibble() summarizing scores per sample per set.

Examples

```
# load the data
data(hmp_gingival)
mod <- cbea(hmp_gingival$data, hmp_gingival$set, abund_values = "16SrRNA",
    output = "sig", distr = "norm", adj = FALSE, n_perm = 5, parametric = TRUE)
tidy(mod)
```

---

var_setup — *Setting up parameter arrays for vectorized call to d/pnorm functions for multi-component mixture distributions*

Description

Setting up parameter arrays for vectorized call to d/pnorm functions for multi-component mixture distributions

Usage

```
var_setup(mu, sigma, lambda, vlen)
```

Arguments

mu	See pmnorm documentation
sigma	See pmnorm documentation
lambda	See pmnorm documentation
vlen	(Integer). Length of the x or p vector to be evaluated

Value

A list containing lambda, mu, and sigma

Index