Package 'biotmle'

April 15, 2020

Title Targeted Learning with Moderated Statistics for Biomarker Discovery

Version 1.10.0

Description This package facilitates the discovery of biomarkers from biological sequencing data (e.g., microarrays, RNA-seq) based on the associations of potential biomarkers with exposure variables by implementing an inferential procedure that combines a generalization of moderated statistics with targeted minimum loss estimates of the average treatment effect whose estimator admits an asymptotically linear representations (in terms of an efficient influence function).

Depends R (>= 3.4)

License file LICENSE

URL https://code.nimahejazi.org/biotmle

BugReports https://github.com/nhejazi/biotmle/issues

Encoding UTF-8

LazyData true

Imports stats, methods, dplyr, tibble, ggplot2, ggsci, superheat, assertthat, future, doFuture, tmle (>= 1.4.0.1), S4Vectors, BiocGenerics, BiocParallel, SummarizedExperiment, limma

Suggests testthat, knitr, rmarkdown, BiocStyle, earth, glmnet, randomForest, SuperLearner, Matrix, DBI, biotmleData (>= 1.1.1)

VignetteBuilder knitr

RoxygenNote 6.1.1

biocViews GeneExpression, DifferentialExpression, Sequencing, Microarray, RNASeq, ImmunoOncology

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

git_branch RELEASE_3_10

git_last_commit d9a0a1f

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

Author Nima Hejazi [aut, cre, cph] (<https://orcid.org/0000-0002-7127-2789>), Alan Hubbard [aut, ths] (<https://orcid.org/0000-0002-3769-0127>), Mark van der Laan [aut, ths] (<https://orcid.org/0000-0003-1432-5511>), Weixin Cai [ctb] (<https://orcid.org/0000-0003-2680-3066>)

Maintainer Nima Hejazi <nh@nimahejazi.org>

R topics documented:

biomarkertmle 2
biomarkerTMLE_exposure 4
bioTMLE-class 5
data.frame_OR_EList-class 5
eif 6
heatmap_ic 6
modtest_ic 7
plot.bioTMLE 8
rnaseq_ic 9
toptable 9
volcano_ic 10

Index **11**

`biomarkertmle` *Biomarker Evaluation with Targeted Minimum Loss Estimation of the ATE*

Description

Computes the causal target parameter defined as the difference between the biomarker expression values under treatment and those same values under no treatment, using Targeted Minimum Loss Estimation.

Usage

```
biomarkertmle(se, varInt, normalized = TRUE, ngscounts = FALSE,
  parallel = TRUE, bppar_type = NULL, future_param = NULL,
  subj_ids = NULL, cv_folds = 5, g_lib = c("SL.mean", "SL.glm",
  "SL.glmnet", "SL.earth"), Q_lib = c("SL.mean", "SL.glm",
  "SL.randomForest"), ...)
```

Arguments

`se`	A `SummarizedExperiment` containing microarray expression or next-generation sequencing data in the `assays` slot and a matrix of phenotype-level data in the `colData` slot.
`varInt`	A `numeric` indicating the column of the design matrix corresponding to the treatment or outcome of interest (in the `colData` slot of the `SummarizedExperiment` argument "se").
`normalized`	(logical) - whether the data included in the `assay` slot of the input `SummarizedExperiment` object has been normalized already. The default is set to `TRUE` since it is expected that most practitioners would apply normalization methods appropriate to the type of assay being analyzed. If set to `FALSE`, median normalization is performed for microarray (i.e., non-RNA-seq) data.
`ngscounts`	(logical) - whether the data are counts generated from a next-generation sequencing (NGS) experiment (e.g., RNA-seq). The default setting assumes continuous expression measures as generated by platforms that are microarray-type (i.e., so-called "targeted" assays).

parallel	(logical) - whether or not to use parallelization in the estimation procedure. Invoking parallelization happens through a combination of calls to **future** and **BiocParallel**. If this argument is set to TRUE, multiprocess is used, and if FALSE, sequential is used, alongside bplapply. Other options for evaluation through futures may be invoked by setting the argument future_param.
bppar_type	(character) - specifies the type of backend to be used with the parallelization invoked by BiocParallel. Consult the manual page for BiocParallelParam for possible types and descriptions on their appropriate uses. The default for this argument is NULL, which silently uses DoparParam.
future_param	(character) - specifies the type of parallelization to be invoked when using futures for evaluation. For a list of the available types, please consult the documentation for plan. The default setting (this argument set to NULL) silently invokes multiprocess. Be careful if changing this setting.
subj_ids	(numeric vector) - subject IDs to be passed directly to the same subject should have the exact same numerical identifier; coerced to class numeric if not provided in the appropriate form.
cv_folds	A numeric scalar indicating how many folds to use in performing targeted minimum loss estimation. Cross-validated estimates are more robust, allowing relaxing of theoretical conditions and construction of conservative variance estimates.
g_lib	(char vector) - library of learning algorithms to be used in fitting the propensity score E[A | W] (the nuisance parameter denoted "g" in the literature on targeted minimum loss-based estimation).
Q_lib	(char vector) - library of learning algorithms to be used in fitting the outcome regression E[Y | A, W] (the nuisance parameter denoted "Q" in the literature on targeted minimum loss-based estimation).
...	Additional arguments to be passed to tmle in fitting the targeted minimum loss estimator of the average treatment effect.

Value

S4 object of class biotmle, generated by sub-classing SummarizedExperiment, with additional slots containing tmleOut and call, among others, containing TMLE-based estimates of the relationship between a biomarker and exposure or outcome variable and the original call to this function (for user reference), respectively.

Examples

```
library(dplyr)
library(biotmleData)
data(illuminaData)
library(SummarizedExperiment)
"%ni%" <- Negate("%in%")

colData(illuminaData) <- colData(illuminaData) %>%
  data.frame() %>%
  dplyr::mutate(age = as.numeric(age > median(age))) %>%
  DataFrame()

varInt_index <- which(names(colData(illuminaData)) %in% "benzene")

biomarkerTMLEout <- biomarkertmle(
  se = illuminaData[1:2, ],
```

```
  varInt = varInt_index,
  parallel = FALSE,
  g_lib = c("SL.mean", "SL.glm"),
  Q_lib = "SL.glm"
)
#
```

`biomarkerTMLE_exposure` *TMLE procedure using ATE for Biomarker Identication from Exposure*

Description

This function performs influence curve-based estimation of the effect of an exposure on biological expression values associated with a given biomarker, controlling for a user-specified set of baseline covariates.

Usage

```
biomarkerTMLE_exposure(Y, W, A, a, subj_ids = NULL, g_lib, Q_lib,
  cv_folds = 5, ...)
```

Arguments

`Y`	A `numeric` vector of expression values for a single biomarker.
`W`	A `Matrix` of `numeric` values corresponding to baseline covariates to be marginalized over in the estimation process.
`A`	A `numeric` vector of discretized exposure vector (e.g., from a design matrix whose effect on expression values is of interest.
`a`	The `numeric` value indicating levels of `A` above against which comparisons are to be made.
`subj_ids`	A `numeric` vector of subject IDs to be passed directly to `tmle` when there are repeated measures; measurements on the same subject should have the exact same numerical identifier. The correction performed utilizes a more conservative estimator of the variance based on the efficient influence function.
`g_lib`	A `character` vector identifying the library of learning algorithms to be used in fitting the propensity score P[A = 1 | W].
`Q_lib`	A `character` vector identifying the library of learning algorithms to be used in fitting the outcome regression E[Y | A, W].
`cv_folds`	A `numeric` scalar indicating how many folds to use in performing targeted minimum loss estimation. Cross-validated estimates are more robust, allowing relaxing of theoretical conditions and construction of conservative variance estimates.
`...`	Additional arguments passed to `tmle` in fitting the targeted minimum loss estimator of the average treatment effect.

Value

TMLE-based estimate of the relationship between biomarker expression and changes in an exposure variable, computed iteratively and saved in the `tmleOut` slot in a `biotmle` object.

bioTMLE-class *Constructor for class bioTMLE*

Description

Constructor for class bioTMLE

Value

class `biotmle` object, sub-classed from SummarizedExperiment.

Examples

```
library(SummarizedExperiment)
library(biotmleData)
data(illuminaData)

example_biotmle_class <- function(se) {
  call <- match.call(expand.dots = TRUE)
  biotmle <- .biotmle(
    SummarizedExperiment(
      assays = assay(se),
      rowData = rowData(se),
      colData = colData(se)
    ),
    call = call,
    tmleOut = as.data.frame(matrix(NA, 10, 10)),
    topTable = as.data.frame(matrix(NA, 10, 10))
  )
  return(biotmle)
}

example_class <- example_biotmle_class(se = illuminaData)
#
```

data.frame_OR_EList-class *S4 class union data.frame_OR_EList*

Description

Virtual class union containing members of both `data.frame` and `limma::Elist`, used internally to handle situations when a returned object has a type that cannot be guessed from the function call.

Value

fusion of classes `data.frame` and `EList`, used within `.biotmle` by class `bioTMLE` to handle uncertainty in the object passed to slot "tmleOut".

eif *Accessor for Table of Raw Efficient Influence Function Values*

Description

Accessor for Table of Raw Efficient Influence Function Values

Usage

```
eif(object)
```

Arguments

object	S4 object of class bioTMLE.

heatmap_ic *Heatmap for class biotmle*

Description

Heatmap of the contributions of a select subset of biomarkers to the variable importance measure changes as assessed by influence curve-based estimation, across all subjects. The heatmap produced performs supervised clustering, in the sense first described in Pollard & van der Laan (2008) <doi:10.2202/1544-6115.1404>.

Usage

```
heatmap_ic(x, ..., design, FDRcutoff = 0.05, type = c("top", "all"),
  top = 25)
```

Arguments

x	Object of class biotmle as produced by an appropriate call to biomarkertmle
...	additional arguments passed to superheat::superheat as necessary
design	A vector providing the contrast to be displayed in the heatmap.
FDRcutoff	Cutoff to be used in controlling the False Discovery Rate.
type	A character describing whether to plot only a top number (as defined by FDR-corrected p-value) of biomarkers or all biomarkers.
top	Number of identified biomarkers to plot in the heatmap.

Value

heatmap (from the superheat package) using hierarchical clustering to plot the changes in the variable importance measure for all subjects across a specified top number of biomarkers.

Examples

```
library(dplyr)
library(biotmleData)
library(SummarizedExperiment)
data(illuminaData)
data(biomarkertmleOut)

colData(illuminaData) <- colData(illuminaData) %>%
  data.frame() %>%
  dplyr::mutate(age = as.numeric(age > median(age))) %>%
  DataFrame()

varInt_index <- which(names(colData(illuminaData)) %in% "benzene")
designVar <- as.data.frame(colData(illuminaData))[, varInt_index]
design <- as.numeric(designVar == max(designVar))

limmaTMLEout <- modtest_ic(biotmle = biomarkerTMLEout)

heatmap_ic(x = limmaTMLEout, design = design, FDRcutoff = 0.05, top = 15)
#
```

`modtest_ic` *Moderated Statistical Tests for Influence Functions*

Description

Performs variance shrinkage via the empirical Bayes procedure of LIMMA on the observed data after a transformation moving the data to influence function space, based on the average treatment effect parameter.

Usage

```
modtest_ic(biotmle, adjust = "BH", pval_type = c("normal", "logistic"),
  ...)
```

Arguments

`biotmle`	`biotmle` object as generated by `biomarkertmle`
`adjust`	the multiple testing correction to be applied to p-values that are generated from the moderated tests. The recommended (and default) method is that of Benjamini and Hochberg. See topTable for a list of appropriate methods.
`pval_type`	The reference distribution to be used for computing the p-value. Those based on the normal approximation tend to provide misleading inference when working with moderately sized (finite) samples. Use of the logistic distribution has been found to empirically improve performance in settings where multiple hypothesis testing is a concern.
`...`	Other arguments to be passed directly to `limma::topTable`.

Value

`biotmle` object containing output from `limma::lmFit` and `limma::topTable`

Examples

```
library(biotmleData)
library(SummarizedExperiment)
data(biomarkertmleOut)

limmaTMLEout <- modtest_ic(biotmle = biomarkerTMLEout)
```

`plot.bioTMLE` *Plot p-values from moderated statistical tests for class biotmle*

Description

Histogram of raw or FDR-adjusted p-values from the moderated t-test.

Usage

```
## S3 method for class 'bioTMLE'
plot(x, ..., type = "pvals_adj")
```

Arguments

`x`	object of class `biotmle` as produced by an appropriate call to `biomarkertmle`
`...`	additional arguments passed `plot` as necessary
`type`	character describing whether to provide a plot of unadjusted or adjusted p-values (adjustment performed via Benjamini-Hochberg)

Value

object of class `ggplot` containing a histogram of the raw or Benjamini-Hochberg corrected p-values (depending on user input).

Examples

```
library(dplyr)
library(biotmleData)
library(SummarizedExperiment)
data(biomarkertmleOut)

limmaTMLEout <- modtest_ic(biotmle = biomarkerTMLEout)

plot(x = limmaTMLEout, type = "pvals_adj")
#
```

rnaseq_ic — *Transformation utility for using "voom" with biomarker TMLE procedure*

Description

This function prepares next-generation sequencing data (counts) for use with the biomarker TMLE procedure by invoking the voom transform of `limma`.

Usage

```
rnaseq_ic(biotmle, weights = TRUE, ...)
```

Arguments

`biotmle`	(bioTMLE) - subclass of `SummarizedExperiment` containing next-generation sequencing (NGS) count data in the "assays" slot.
`weights`	(logical) - whether to return quality weights of samples in the output object.
`...`	- other arguments to be passed to functions `limma::voom` or `limma::voomWithQualityWeights` as appropriate.

Value

`EList` object containing voom-transformed "expression" measures of count data (actually, the mean-variance trend) in the "E" slot, to be passed into the biomarker TMLE procedure.

toptable — *Accessor for Results Table of Moderated Influence Function Hypothesis Test*

Description

Accessor for Results Table of Moderated Influence Function Hypothesis Test

Usage

```
toptable(object)
```

Arguments

`object`	S4 object of class `bioTMLE`.

volcano_ic *Volcano plot for class biotmle*

Description

Volcano plot of the log-changes in the target causal paramter against the log raw p-values from the moderated t-test.

Usage

```
volcano_ic(biotmle, fc_bound = 3, pval_bound = 0.2)
```

Arguments

biotmle	object of class biotmle as produced by an appropriate call to biomarkertmle
fc_bound	(numeric) - indicates the highest magnitude of the fold to be colored along the x-axis of the volcano plot; this limits the observations to be considered differentially expressed to those in a user-specified interval.
pval_bound	(numeric) - indicates the largest corrected p-value to be colored along the y-axis of the volcano plot; this limits observations considered as differentially expressed to those in a user-specified interval.

Value

object of class ggplot containing a standard volcano plot of the log-fold change in the causal target parameter against the raw log p-value computed from the moderated tests in modtest_ic.

Examples

```
library(dplyr)
library(biotmleData)
library(SummarizedExperiment)
data(biomarkertmleOut)

limmaTMLEout <- modtest_ic(biotmle = biomarkerTMLEout)

volcano_ic(biotmle = limmaTMLEout)
#
```

Index

.biotmle *(*bioTMLE-class*)*, 5

BiocParallelParam, *3*
biomarkertmle, 2
biomarkerTMLE_exposure, 4
bioTMLE-class, 5
bplapply, *3*

data.frame_OR_EList-class, 5
DoparParam, *3*

eif, 6

heatmap_ic, 6

modtest_ic, 7
multiprocess, *3*

plan, *3*
plot.bioTMLE, 8

rnaseq_ic, 9

sequential, *3*

tmle, *3*, *4*
topTable, *7*
toptable, 9

volcano_ic, 10