

Package ‘biotmle’

April 11, 2018

Title Targeted Learning for Biomarker Discovery with Moderated Statistics

Version 1.2.1

Author Nima Hejazi [aut, cre, cph], Alan Hubbard [aut], Weixin Cai [ctb]

Maintainer Nima Hejazi <nhejazi@berkeley.edu>

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 and outcome variables by implementing an estimation procedure that combines a generalization of moderated statistics with asymptotically linear statistical parameters estimated via targeted minimum loss-based estimation (TMLE).

Depends R (>= 3.4)

License file LICENSE

URL <https://github.com/nhejazi/biotmle>

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

Encoding UTF-8

LazyData true

Imports dplyr, magrittr, ggplot2, superheat, wesanderson, doFuture, future, stats, Matrix, methods, DBI, limma, BiocParallel, SummarizedExperiment, biotmleData (>= 1.1.1), SuperLearner, tml

Suggests testthat, knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

RoxygenNote 6.0.1.9000

biocViews GeneExpression, DifferentialExpression, Sequencing, Microarray, RNASeq

NeedsCompilation no

R topics documented:

biomarkertmle	2
biomarkerTMLE_exposure	3
bioTMLE-class	4
data.frame_OR_EList-class	5

heatmap_ic	5
modtest_ic	6
plot.bioTMLE	7
rnaseq_ic	7
volcano_ic	8

Index	9
--------------	----------

biomarkertmle	<i>Biomarker Evaluation with Targeted Minimum Loss-Based Estimation (TMLE)</i>
---------------	--

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-Based Estimation.

Usage

```
biomarkertmle(se, varInt, ngscounts = FALSE, parallel = TRUE,
  bppar_type = NULL, future_param = NULL, family = "gaussian",
  subj_ids = NULL, g_lib = c("SL.glm", "SL.randomForest", "SL.nnet",
  "SL.polymars", "SL.mean"), Q_lib = c("SL.glm", "SL.randomForest", "SL.nnet",
  "SL.mean"))
```

Arguments

se	(SummarizedExperiment) - containing expression or next-generation sequencing data in the "assays" slot and a matrix of phenotype-level data in the "colData" slot.
varInt	(numeric) - indicating the column of the design matrix corresponding to the treatment or outcome of interest (in the "colData" slot of the "se" argument above).
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 microarray-type platforms.
parallel	(logical) - whether or not to use parallelization in the estimation procedure. Invoking parallelization happens through a combination of calls to <code>future</code> and <code>BiocParallel</code> . If this argument is set to <code>TRUE</code> , <code>future::multiprocess</code> is used, and if <code>FALSE</code> , <code>future::sequential</code> is used, alongside <code>BiocParallel::bplapply</code> . Other options for evaluation through futures may be invoked by setting the argument <code>future_param</code> .
bppar_type	(character) - specifies the type of backend to be used with the parallelization invoked by <code>BiocParallel</code> . Consult the manual page for <code>BiocParallel::BiocParallelParam</code> for possible types and descriptions on their appropriate uses. The default for this argument is <code>NULL</code> , which silently uses <code>BiocParallel::DoparParam</code> .
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 <code>future::plan</code> . The default setting (this argument set to <code>NULL</code>) silently invokes <code>future::multiprocess</code> . Be careful if changing this setting.

family	(character) - specification of error family: "binomial" or "gaussian".
subj_ids	(numeric vector) - subject IDs to be passed directly to the same subject should have the exact same numerical identifier; coerced to numeric if not provided in the appropriate form.
g_lib	(char vector) - library of learning algorithms to be used in fitting the "g" step of the standard TMLE procedure.
Q_lib	(char vector) - library of learning algorithms to be used in fitting the "Q" step of the standard TMLE procedure.

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,
  family = "gaussian",
  g_lib = c("SL.mean", "SL.glm"),
  Q_lib = "SL.mean"
)
```

biomarkerTMLE_exposure

TMLE procedure for Biomarker Identification 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, family = "gaussian",
  g_lib, Q_lib)
```

Arguments

Y	(numeric vector) - a vector of expression values for a single biomarker.
W	(numeric matrix) - a matrix of baseline covariates to be controlled in the estimation process.
A	(numeric vector) - a discretized exposure vector (e.g., from a design matrix whose effect on expression values is of interest.
a	(numeric vector) - the levels of A against which comparisons are to be made.
subj_ids	(numeric vector) - subject IDs to be passed directly to the same subject should have the exact same numerical identifier. coerced to numeric if not provided in the appropriate form.
family	(character) - specification of error family: "binomial" or "gaussian"
g_lib	(char vector) - library of learning algorithms to be used in fitting the "g" step of the standard TMLE procedure.
Q_lib	(char vector) - library of learning algorithms to be used in fitting the "Q" step of the standard TMLE procedure.

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	<i>Constructor for class bioTMLE</i>
---------------	--------------------------------------

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

Class union containing `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".

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.

Usage

```
heatmap_ic(x, ..., design, FDRcutoff = 0.05, top = 25)
```

Arguments

<code>x</code>	object of class <code>biotmle</code> as produced by an appropriate call to <code>biomarkertmle</code>
<code>...</code>	additional arguments passed to <code>superheat::superheat</code> as necessary
<code>design</code>	a vector providing the contrast to be displayed in the heatmap.
<code>FDRcutoff</code>	cutoff to be used in controlling the False Discovery Rate
<code>top</code>	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 Curves

Description

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

Usage

```
modtest_ic(biotmle, adjust = "BH")
```

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.

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	<i>Plot p-values from moderated statistical tests for class biotmle</i>
--------------	---

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	<i>Transformation utility for using "voom" with biomarker TMLE procedure</i>
-----------	--

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 <code>limma::voom</code> or <code>limma::voomWithQualityWeights</code> 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.

volcano_ic	<i>Volcano plot for class biotmle</i>
------------	---------------------------------------

Description

Volcano plot of the log-changes in the target causal parameter 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 <code>biomarkertmle</code>
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), 4

`biomarkertmle`, 2

`biomarkerTMLE_exposure`, 3

bioTMLE-class, 4

`data.frame_OR_EList`-class, 5

`heatmap_ic`, 5

`modtest_ic`, 6

`plot.bioTMLE`, 7

`rnaseq_ic`, 7

`topTable`, 6

`volcano_ic`, 8