

Package ‘methyvim’

October 13, 2019

Title Targeted, Robust, and Model-free Differential Methylation Analysis

Version 1.6.0

Maintainer Nima Hejazi <nhejazi@berkeley.edu>

Author Nima Hejazi [aut, cre, cph], Mark van der Laan [aut, ths], Alan Hubbard [ctb, ths], Rachael Phillips [ctb]

Description This package provides facilities for differential methylation analysis based on variable importance measures (VIMs), a class of statistical target parameters that arise in causal inference. The estimation and inference procedures provided are nonparametric, relying on ensemble machine learning to flexibly assess functional relationships among covariates and the outcome of interest. These tools can be applied to differential methylation at the level of CpG sites, to obtain valid statistical inference even after corrections for multiple hypothesis testing.

Depends R (>= 3.4.0)

License file LICENSE

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

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

Encoding UTF-8

Imports stats, cluster, methods, ggplot2, ggsci, gridExtra, superheat, dplyr, gtools, tmle, future, doFuture, S4Vectors, BiocGenerics, BiocParallel, SummarizedExperiment, GenomeInfoDb, bumphunter, IRanges, limma, minfi

Suggests testthat, knitr, rmarkdown, BiocStyle, SuperLearner, earth, nnet, gam, arm, snow, parallel, BatchJobs, minfiData, methyvimData

VignetteBuilder knitr

RoxygenNote 6.1.1

biocViews Clustering, DNAMethylation, DifferentialMethylation, MethylationArray, MethylSeq

git_url <https://git.bioconductor.org/packages/methyvim>

git_branch RELEASE_3_9

git_last_commit f850cf0

git_last_commit_date 2019-06-10

Date/Publication 2019-10-12

R topics documented:

fdr_msa	2
methyheat	3
methyvim	4
methyvolc	6
plot.methytmle	7
wrap_in_try	8

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

fdr_msa	<i>FDR-MSA correction</i>
---------	---------------------------

Description

Modified FDR Controlling Procedure for Multi-Stage Analyses (MJ van der Laan and C Tuglus, 2009, <doi:10.2202/1544-6115.1397>)

Usage

```
fdr_msa(pvals, total_obs)
```

Arguments

pvals	Numeric vector containing the p-values that result from any chosen statistical hypothesis testing procedure.
total_obs	Numeric indicating the total number of observations that would have been available for testing prior to the selection procedure employed in the multi-stage analysis performed.

Value

A numeric vector of corrected p-values, controlling the False Discovery Rate, using the method of Tuglus and van der Laan.

Examples

```
g <- 1e4
n <- 1e2
p <- abs(rnorm(n, mean = 1e-8, sd = 1e-2))
# treating the vector p as one of p-values, FDR-MSA may be applied
fdr_p <- fdr_msa(pvals = p, total_obs = g)
```

`methyheat`*Heatmap for methytmle objects*

Description

Heatmap for methytmle objects

Usage

```
methyheat(x, ..., n_sites = 25, type = "raw")
```

Arguments

<code>x</code>	Object of class <code>methytmle</code> as produced by an appropriate call to <code>methyvim</code> .
<code>...</code>	Additional arguments passed to <code>superheat</code> . Consult the documentation of the <code>superheat</code> package for a list of options.
<code>n_sites</code>	Numeric indicating the number of CpG sites to be shown in the plot. If the number of sites analyzed is greater than this cutoff, sites to be displayed are chosen by ranking sites based on their raw (marginal) p-values.
<code>type</code>	Whether to plot the original data (M-values or Beta-values) for the set of top CpG sites or to plot the measurements after applying a transformation into influence curve space (with respect to the target parameter of interest). The latter uses the fact that the parameters have asymptotically linear representations to obtain a rotation of the raw data into an alternative space; moreover, in this setting, the heatmap reduces to visualizing a supervised clustering procedure.

Value

Nothing. This function is called for its side-effect of outputting a heatmap to the graphics device. The heatmap is constructed using the `superheat` package.

Examples

```
suppressMessages(library(SummarizedExperiment))
library(methyvimData)
data(grExample)
var_int <- as.numeric(colData(grExample)[, 1])
# TMLE procedure for the ATE parameter over M-values with Limma filtering
methyvim_out_ate <- suppressWarnings(
  methyvim(
    data_grs = grExample, sites_comp = 25, var_int = var_int,
    vim = "ate", type = "Mval", filter = "limma", filter_cutoff = 0.1,
    parallel = FALSE, tmle_type = "glm"
  )
)
methyheat(methyvim_out_ate, type = "raw")
```

Description

Computes the Targeted Minimum Loss-Based Estimate of a specified statistical target parameter, formally defined within models from causal inference. The variable importance measures currently supported are the Average Treatment Effect (ATE) and a Nonparametric Variable Importance Measure (NPVI, formally defined by Chambaz, Neuvial, and van der Laan <doi:10.1214/12-EJS703>).

Usage

```
methyvim(data_grs, var_int, vim = c("ate", "rr", "npvi"),
         type = c("Beta", "Mval"), filter = c("limma"),
         filter_cutoff = 0.05, window_bp = 1000, corr_max = 0.75,
         obs_per_covar = 20, sites_comp = NULL, parallel = TRUE,
         future_param = NULL, bpar_type = NULL, return_ic = FALSE,
         shrink_ic = FALSE, tml_type = c("glm", "sl"),
         tml_args = list(family = "binomial", g_lib = c("SL.mean", "SL.glm",
         "SL.bayesglm", "SL.gam"), Q_lib = c("SL.mean", "SL.glm", "SL.gam",
         "SL.earth"), npvi_cutoff = 0.25, npvi_descr = NULL),
         tml_backend = c("tml", "drtml", "tml.npvi"))
```

Arguments

data_grs	An object of class <code>minfi::GenomicRatioSet</code> , containing standard data structures associated with DNA Methylation experiments. Consult the documentation for <code>minfi</code> to construct such objects.
var_int	A numeric vector containing subject-level measurements of the variable of interest. The length of this vector must match the number of subjects exactly. If argument <code>vim</code> is set to "ate" or "rr", then the variable of interest is treated as an exposure, and the variable must be binary in such cases. If setting <code>vim</code> to target parameters assessing continuous treatment effects, then the variable need not be binary of course.
vim	Character indicating the variable importance measure to be used in the estimation procedure. Currently supported options are the ATE for discretized exposures and NPVI for continuous exposures. ATE and RR are the appropriate choices when the underlying scientific question is of the effect of an exposure on methylation, while NPVI (and other continuous treatment parameters) ought to be used when the effect of methylation on an outcome is sought.
type	Character indicating the particular measure of DNA methylation to be used as the observed data in the estimation procedure, either Beta values or M-values. The data are accessed via <code>minfi::getBeta</code> or <code>minfi::getM</code> .
filter	Character indicating the model to be implemented when screening the <code>data_grs</code> object for CpG sites. The only currently supported option is "limma". Contributions for other methods are welcome.
filter_cutoff	Numeric indicating the p-value cutoff that defines which sites pass through the filter.
window_bp	Numeric indicating the maximum genomic distance (in base pairs) between two sites for them to be considered neighboring sites.

<code>corr_max</code>	Numeric indicating the maximum correlation that a neighboring site can have with the target site.
<code>obs_per_covar</code>	Numeric indicating the number of observations needed for for covariate included in <code>W</code> for downstream analysis. This ensures the data is sufficient to control for the covariates.
<code>sites_comp</code>	A numeric indicating the maximum number of sites for which a variable importance measure is to be estimated post-screening. This is not typically useful in scientific settings, but may be useful when a large number of CpG sites pass the initial screening phase.
<code>parallel</code>	Logical indicating whether parallelization ought to be used. See the documentation of <code>set_parallel</code> for more information, as this argument is passed directly to that internal function.
<code>future_param</code>	Character indicating the type of parallelization to be used from the list available via the <code>future</code> package. See the documentation for <code>set_parallel</code> for more information, as this argument is passed directly to that internal function.
<code>bppar_type</code>	Character specifying the type of backend to be used for parallelization via <code>BiocParallel</code> . See the documentation for <code>set_parallel</code> for more information, as this argument is passed directly to that internal function.
<code>return_ic</code>	Logical indicating whether an influence curve estimate should be returned for each site that passed through the filter.
<code>shrink_ic</code>	Logical indicating whether <code>limma</code> should be applied to reduce the variance in the <code>ic</code> based estimates in <code>return_ic</code> .
<code>tmle_type</code>	Character indicating the general class of regression models to be used in fitting the propensity score and outcome regressions. This is generally a shorthand and is overridden by <code>tmle_args</code> if that argument is changed from its default values.
<code>tmle_args</code>	List giving several key arguments to be passed to one of <code>tmle::tmle</code> or <code>tmle.npvi::tmle.npvi</code> , depending on the particular variable importance measure specified. This overrides <code>tmle_type</code> , which itself provides sensible defaults. Consider changing this away from default settings only if you have sufficient experience with software and the underlying theory for Targeted Learning. For more information, consider consulting the documentation of the <code>tmle</code> and <code>tmle.npvi</code> packages.
<code>tmle_backend</code>	A character indicating the package to be used in the estimation procedure. The user should only set this parameter if they have sufficient familiarity with the backend packages used for estimation. Current choices include <code>tmle</code> , <code>drtmle</code> , and <code>tmle.npvi</code> .

Value

An object of class `methytmle`, with all unique slots filled in, in particular, including indices of CpG sites that pass screening, cluster of neighboring CpG sites, and a matrix of the results of the estimation procedure performed for the given variable importance measure. Optionally, estimates of the propensity score and outcome regressions, as well as the original data rotated into influence curve space may be returned, if so requested.

Examples

```
library(methyvimData)
suppressMessages(library(SummarizedExperiment))
data(grsexample)
var_int <- colData(grsexample)[, 1]
```

```
# TMLE procedure for the ATE parameter over M-values with Limma filtering
methyvim_out_ate <- suppressWarnings(
  methyvim(
    data_grs = grsExample, sites_comp = 1, var_int = var_int,
    vim = "ate", type = "Mval", filter = "limma", filter_cutoff = 0.05,
    parallel = FALSE, tmle_type = "sl"
  )
)
```

methyvolc

Volcano plot for methytmle objects

Description

Volcano plot for methytmle objects

Usage

```
methyvolc(x, param_bound = 2, pval_bound = 0.2)
```

Arguments

x	Object of class methytmle as produced by an appropriate call to methyvim.
param_bound	Numeric for a threshold indicating the magnitude of the size of the effect considered to be interesting. This is used to assign groupings and colors to individual CpG sites.
pval_bound	Numeric for a threshold indicating the magnitude of p-values deemed to be interesting. This is used to assign groupings and colors to individual CpG sites.

Value

Object of class ggplot containing a volcano plot of the estimated effect size on the x-axis and the $-\log_{10}(\text{p-value})$ on the y-axis. The volcano plot is used to detect possibly false positive cases, where a test statistic is significant due to low variance.

Examples

```
suppressMessages(library(SummarizedExperiment))
library(methyvimData)
data(grsexample)
var_int <- as.numeric(colData(grsexample)[, 1])
# TMLE procedure for the ATE parameter over M-values with Limma filtering
methyvim_out_ate <- suppressWarnings(
  methyvim(
    data_grs = grsexample, sites_comp = 25, var_int = var_int,
    vim = "ate", type = "Mval", filter = "limma", filter_cutoff = 0.1,
    parallel = FALSE, tmle_type = "glm"
  )
)
methyvolc(methyvim_out_ate)
```

plot.methytml	<i>Plot p-values of methytml objects</i>
---------------	------------------------------------------

Description

Plot p-values of methytml objects

Usage

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

Arguments

x	Object of class methytml as produced by an appropriate call to methyvim.
...	Additional arguments passed plot as necessary.
type	The type of plot to build: one of side-by-side histograms (type "both") comparing raw p-values to FDR-adjusted p-values (using the FDR-MSA correction) or either of these two histogram separately. Set this argument to "raw_pvals" for a histogram of the raw p-values, and to "fdr_pvals" for a histogram of the FDR-corrected p-values.

Value

Object of class ggplot containing a histogram or side-by-side histograms of the raw (marginal) and corrected p-values, with the latter computed automatically using the method of Tuglus and van der Laan.

Examples

```
suppressMessages(library(SummarizedExperiment))  
library(methyvimData)  
data(grsexample)  
var_int <- as.numeric(colData(grsexample)[, 1])  
# TMLE procedure for the ATE parameter over M-values with Limma filtering  
methyvim_out_ate <- suppressWarnings(  
  methyvim(  
    data_grs = grsexample, sites_comp = 25, var_int = var_int,  
    vim = "ate", type = "Mval", filter = "limma", filter_cutoff = 0.1,  
    parallel = FALSE, tmle_type = "glm"  
  )  
)  
plot(methyvim_out_ate)
```

`wrap_in_try`*Wrap a Function in a Try Statement*

Description

Function factory that generates versions of functions wrapped in try. Originally found in and borrowed from package origami.

Usage

```
wrap_in_try(fun, ...)
```

Arguments

<code>fun</code>	A function to be wrapped in a try statement.
<code>...</code>	Additional arguments passed to the previous argument <code>fun</code> .

Index

fdr_msa, 2

methyheat, 3

methyvim, 4

methyvolc, 6

plot.methytmle, 7

wrap_in_try, 8