`methyvim`: Variable Importance Measures for Differential Methylation

Nima Hejazi
Division of Biostatistics
University of California, Berkeley
Berkeley, CA, 94720, USA

2017-09-26

Abstract

We present a general algorithm for the nonparametric estimation of effects of DNA methylation at CpG sites scattered across the genome, complete with honest statistical inference for such estimates. This approach leverages variable importance measures, a class of parameters that arise in the study of causal inference. The parameters we persent are defined in such a manner that they provided targeted estimates of the relative importance of CpG sites in the case of binary exposures/treatments assigned at the level of subjects. Such parameters come equipped with rich scientific interpretations, providing an avenue to move beyond linear models, applying modern developments in machine learning to estimating quantities of scientific interest in computational biology.

Contents

0.1 Introduction

DNA methylation is a fundamental epigenetic process known to play an important role in the regulation of gene expression. DNA methylation occurrs as a chemical modification of CpG sites in which a methyl group (CH_3) is added to a carbon of the cytosine ring structure; in humans, this process is carried out by a set of eznymes known as the the DNA methyltransferases. Numerous biological and medical studies have implicated DNA methylation as playing a role in disease processes with a molecular signature. Perhaps unsurprisingly then, numerous biotechnologies have been developed to probe the molecular mechanisms of this epigenetic process. Modern assays, like the Illumina *Infinium* arrays, allow for DNA methylation signatures at nearly $850,000$ CpG sites scattered across the genome to be measured simultaneously; moreover, much effort has been invested, by the bioinformatics community, in the development of tools for properly removing technological effects may contaminate biological signatures measured by such assays. Despite these leaps and bounds in both biological and bioninformatical techniques, most statistical methods available for the analysis of data produced by such assays rely on (generalized) linear models.

Here, we present an alternative to such approaches, in the form of nonparametric estimation procedures inspired by machine learning and causal inference. Specifically, we provide a technique for obtaining estimates of nonparametric *variable importance measures* (**VIM**), parameters with rich scientific interpretations under the standard (untestable) assumptions used in statistical causal inference, defining a limited set of VIMs specifically with respect to the type of data commonly produced by DNA methylation assays. For VIMs defined in such a manner, targeted minimum loss-based estimates may be readily computed based on the data made available by DNA methylation assays. Our contribution, `methyvim` is an R package that provides facilities for performing differential methylation analyses within exactly this scope.

As the substantive contribution of our work is an estimation procedure, we focus on data produced by 450k and 850k (EPIC) arrays made by Illumina and assume that data has been subjected to proper quality control and normalizaton procedures, as outlined by others in the computational biology community (Fortin et al. 2014, Dedeurwaerder et al. (2013)). For a general discussion of the framework of targeted minimum loss-based estimation and the role this approach plays in statistical causal inference, the interested reader is invited to consult M. J. van der Laan and Rose (2011) and M. J. van der Laan and Rose (2017). For a more general introduction to (statistical) causal inference, Pearl (2009) and Hernan and Robins (2018, forthcoming) may be of interest.

0.2 Methodology

The core functionality of this package is made available via the eponymous `methyvim` function, which implements a statistical algorithm designed to compute targeted estimates of VIMs, defined in such a way that the VIMs represent parameters of scientific interest in computational biology experiments; moreover, these VIMs are defined such that they may be estimated in a manner that is very nearly assumption-free, that is, within a *fully nonparametric statistical model*. **The statistical algorithm consists in several major steps:**

1. Pre-screening of genomic sites is used to isolate a subset of sites for which there is cursory evidence of differential methylation. For the sake of computational feasibility, targeted minimum loss-based estimates of VIMs are computed only for this subset of sites. Currently, the available screening approach adapts core routines from the `limma` R package. Future releases will support functionality from other packages (e.g., `randomForest`, `tmle.npvi`). Following the style of the function for performing screening via `limma`, users may write their own screening functions and are invited to contribute such functions to the core software package by opening pull requests at the GitHub repository.
2. Nonparametric estimates of VIMs, for the specified target parameter, are computed at each of the CpG sites passing the screening step. The VIMs are defined in such a way that the estimated effects is of an exposure/treatment on the methylation status of a target CpG site, controlling for the observed methylation status of the neighbors of that site. Currently, routines are adapted from the `tmle` R package. Future releases will support doubly-robust estimates of these VIMs (via the `drtmle` package) and add parameters for continuous treatments/exposures (via the `tmle.npvi` package).
3. Since pre-screening is performed prior to estimating VIMs, we make use of a multiple testing correction uniquely suited to such settings. Due to the multiple testing nature of the estimation problem, a variant of the Benjamini & Hochberg procedure for controlling the False Discovery Rate (FDR) is applied (Benjamini and Hochberg 1995). Specifically, we apply the "modified marginal Benjamini & Hochberg step-up False Discovery Rate controlling procedure for multi-stage analyses" (FDR-MSA), which is guaranteed to control the FDR as if all sites were tested (i.e., without screening) (Tuglus and van der Laan 2009).

0.3 Parameters of Interest

For discrete-valued treatments or exposures:

- The *average treatment effect* (ATE): The effect of a binary exposure or treatment on the observed methylation at a target CpG site is estimated, controlling for the observed methylation at all other CpG sites in the same neighborhood as the target site, based on an additive form. In particular, the parameter estimate represents the **additive difference** in methylation that would have been observed at the target site had all observations received the treatment versus the counterfactual under which none received the treatment.
- The *relative risk* (RR): The effect of a binary exposure or treatment on the observed methylation at a target CpG site is estimated, controlling for the observed methylation at all other CpG sites in the same neighborhood as the target site, based on an geometric form. In particular, the parameter estimate represents the **multiplicative difference** in methylation that would have been observed at the target site had all observations received the treatment versus the counterfactual under which none received the treatment.

Support for continuous-valued treatments or exposures is *planned but not yet available*, though work is underway to incorporate into our methodology the following

- A *nonparametric variable importance measure* (NPVI) (Chambaz, Neuvial, and van der Laan 2012): The effect of continous-valued exposure or treatment (the observed methylation at a target CpG site) on an outcome of interest is estimated, controlling for the observed methylation at all other CpG sites in the same neighborhood as the target (treatment) site, based on a parameter that compares values of the treatment against a reference value taken to be the null. In particular, the implementation provided is designed to assess the effect of differential methylation at the target CpG site on a (typically) phenotype-level outcome of interest (e.g., survival), in effect providing an nonparametric evaluation of the impact of methylation at the target site on said outcome.

As previously noted, in all cases, an estimator of the target parameter is constructed via targeted minimum loss-based estimation.

Having now discussed the foundational principles of the estimation procedure employed and the statistical algorithm implemented, it is best to proceed by examining `methyvim` by example.

0.4 Preliminaries: Setting up the Data

First, we'll load the `methyvim` package and the example data contained in the `methyvimData` package that accompanies it:

```
set.seed(479253)
library(methyvim)
## methyvim: Nonparametric Differential Methylation Analysis
##  with Targeted Estimates of Variable Importance Measures
## Version: 0.99.2
library(methyvimData)
```

Now, let's load the data set and seed the RNG:

```
data(grsExample)
grsExample
## class: GenomicRatioSet
## dim: 400 210
## metadata(0):
## assays(2): Beta M
## rownames(400): cg23578515 cg06747907 ... cg01715842 cg09895959
## rowData names(0):
## colnames(210): V2 V3 ... V397 V398
## colData names(2): exp outcome
## Annotation
##   array: IlluminaHumanMethylationEPIC
## Preprocessing
##   Method: NA
##   minfi version: NA
##   Manifest version: NA
var_int <- as.numeric(colData(grsExample)[, 1])
```

The example data object is of class `GenomicRatioSet`, provided by the `minfi` package. The summary provided by the `print` method gives a wealth of information on the experiment that generated the data – since we are working with a simulated data set, we need not concern ourselves with much of this information.

We can create an object of class `methytmle` from any `GenomicRatioSet` object simply invoking the S4 class constructor:

```
mtmle <- .methytmle(grsExample)
```

Since the methytmle class builds upon the GenomicRatioSet class, it contains all of the slots of GenomicRatios. The new class introduced in the mehtyvim package includes several new slots:

- call - the form of the original call to the methyvim function.
- screen_ind - indices identifying CpG sites that pass the screening process.
- clusters - non-unique IDs corresponding to the manner in wich sites are treated as neighbors. These are assigned by genomic distance (bp) and respect chromosome boundaries (produced via a call to bumphunter::clusterMaker).
- var_int - the treatment/exposure status for each subject. Currently, these must be binary, due to the definition of the supported targeted parameters.
- param - the name of the target parameter from which the estimated VIMs are defined.
- vim - a table of statistical results obtained from estimating VIMs for each of the CpG sites that pass the screening procedure.
- ic - the measured array values for each of the CpG sites passing the screening, transformed into influence curve space based on the chosen target parameter.

The interested analyst might consider consulting the documentation of the minfi package for an in-depth description of all of the other slots that appear in this class (Aryee et al. 2014). Having examined the core structure of the package, it is time now to discuss the analytic capabilities implemented.

0.5 Differential Methylation Based on a Binary Treatment or Exposure

0.5.1 The Average Treatment Effect as Variable Importance Measure

The average treatment effect (ATE) is a canonical parameter that arises in statistical causal inference, often denoted $\psi_0 = \psi_0(1) - \psi_0(0)$, representing the difference in an outcome between the counterfactuals under which all subjects received the treatment/exposure and under which none received such treatment/exposure. Under additional (untestable) assumptions, this parameter has a richer interpretation as a mean counterfactual outcome, wherein the counterfactuals used in this definition define causal effects. When causal assumptions remain unfulfilled or untested, this parameter may still be estimated in the form of a nonparametric VIM.

Estimating such the VIM corresponding to such a parameter requires two separate regression steps: one for the treatment mechanism (propensity score) and one for the outcome regression. Technical details on the nature of these regressions are discussed in Hernan and Robins (2018, forthcoming), and details for estimating these regressions in the framework of targeted minimum loss-based estimation are discussed in M. J. van der Laan and Rose (2011).

0.5.1.1 Super Learning for nonparametric parameter estimation

Nonparametric and data-adaptive regressions (i.e., machine learning) may be used in the two regression steps outlined above. This is implemented using the Super Learner algorithm, which produces optimal combinations of such regression functions (i.e., stacked regressions) using cross-validation (M. J. van der Laan, Polley, and Hubbard 2007, Breiman (1996)).

methyvim makes performing such estimation for CpG sites using a given VIM essentially trivial:

```
suppressMessages(
  methyvim_ate_sl <- methyvim(data_grs = grsExample, sites_comp = 25,
                              var_int = var_int, vim = "ate", type = "Mval",
                              filter = "limma", filter_cutoff = 0.10,
                              parallel = FALSE, tmle_type = "sl"
                             )
)
## Warning in set_parallel(parallel = parallel, future_param = future_param, : Sequential evaluation is st
##  Proceed with caution.
```

```
methyvim_ate_sl
## class: methytmle
## dim: 400 210
## metadata(0):
## assays(2): Beta M
## rownames(400): cg23578515 cg06747907 ... cg01715842 cg09895959
## rowData names(0):
## colnames(210): V2 V3 ... V397 V398
## colData names(2): exp outcome
## Annotation
##   array: IlluminaHumanMethylationEPIC
## Preprocessing
##   Method: NA
##   minfi version: NA
##   Manifest version: NA
```

As is clear from examining the object `methyvim_ate_sl`, the output resembles exactly that returned when examining objects of class `GenomicRatioSet` from the `minfi` R package. In particular, the returned `methytmle` object is merely a modified form (in particular, a subclass) of the input `GenomicRatioSet` object – thus, it contains all of the original slots, with all experimental data intact. Several extra pieces of information are contained within the output object as well_.

We can take a look at the results produced from the estimation procedure by examining the `"vim"` `slot` of the `methytmle` object:

```
head(slot(methyvim_ate_sl, "vim"))
##            lowerCI_ATE     est_ATE upperCI_ATE    Var_ATE      pval
## cg22913481  -0.3790600 -0.10205080  0.17495839 0.01997451 0.4702524
## cg15131207  -0.3121142 -0.10416890  0.10377642 0.01125605 0.3261739
## cg10613282  -0.4386948 -0.17460160  0.08949158 0.01815525 0.1950350
## cg15857610  -0.3239377 -0.08180582  0.16032610 0.01526131 0.5078440
## cg24775884  -0.4710473  0.03939662  0.54984050 0.06782407 0.8797588
## cg22954484  -0.3307531  0.08234980  0.49545274 0.04442265 0.6960077
##            n_neighbors_all n_neighbors_w max_corr_w
## cg22913481               0             0         NA
## cg15131207               0             0         NA
## cg10613282               0             0         NA
## cg15857610               5             4  0.7720705
## cg24775884               5             2  0.8897566
## cg22954484               5             2  0.8643936
```

From the table displayed, we note that we have access to point estimates of the ATE ("est_ATE") as well as lower and upper confidence interval bounds for each estimate ("lowerCI_ATE" and "upperCI_ATE", respectively). Additional statistical information we have access to include the variance ("Var_ATE") of the estimate as well as the p-value ("pval") associated with each estimate (based on Wald-style testing procedures). Beyond these, key bioinformatical quantities (with respect to the algorithm outlined above) are also returned; these include the total number of neighbors of the target site, the number of neighboring sites controlled for when estimating the effect of exposure on DNA methylation, and, finally, the maximum correlation between the target site and any given site in its full set of neighbors.

0.5.1.2 Generalized linear models for parameter estimation

In cases where nonparametric regressions may not be preferred (e.g., where time constraints are of concern), generalized linear models (GLMs) may be used to fit the two regression steps required for estimating a VIMs for the ATE.

`methyvim` makes performing such estimation for CpG sites using a given VIM essentially trivial:

```
suppressMessages(
  methyvim_ate_glm <- methyvim(data_grs = grsExample, sites_comp = 25,
                               var_int = var_int, vim = "ate", type = "Mval",
                               filter = "limma", filter_cutoff = 0.10,
                               parallel = FALSE, tmle_type = "glm"
                              )
)
## Warning in set_parallel(parallel = parallel, future_param = future_param, : Sequential evaluation is st
##  Proceed with caution.
methyvim_ate_glm
## class: methytmle
## dim: 400 210
## metadata(0):
## assays(2): Beta M
## rownames(400): cg23578515 cg06747907 ... cg01715842 cg09895959
## rowData names(0):
## colnames(210): V2 V3 ... V397 V398
## colData names(2): exp outcome
## Annotation
##   array: IlluminaHumanMethylationEPIC
## Preprocessing
##   Method: NA
##   minfi version: NA
##   Manifest version: NA
```

Just as before, we can take a look at the results produced from the estimation procedure by examining the `"vim"` slot of the `methytmle` object:

```
head(slot(methyvim_ate_glm, "vim"))
##            lowerCI_ATE     est_ATE upperCI_ATE    Var_ATE      pval
## cg22913481  -0.3790600 -0.10205080   0.1749584 0.01997451 0.4702524
## cg15131207  -0.3121142 -0.10416890   0.1037764 0.01125605 0.3261739
## cg10613282  -0.4272500 -0.16315679   0.1009364 0.01815525 0.2259382
## cg15857610  -0.3548160 -0.07921234   0.1963913 0.01977233 0.5732093
## cg24775884  -0.3960505  0.12413831   0.6443272 0.07043847 0.6399733
## cg22954484  -0.2767069  0.13969139   0.5560897 0.04513420 0.5108390
##            n_neighbors_all n_neighbors_w max_corr_w
## cg22913481               0             0         NA
## cg15131207               0             0         NA
## cg10613282               0             0         NA
## cg15857610               5             4  0.7720705
## cg24775884               5             2  0.8897566
## cg22954484               5             2  0.8643936
```

Remark: Here, the estimates are obtained via GLMs, making each of the regression steps less robust than if nonparametric regressions were used. It is expected that these estimates differ from those obtained previously.

0.5.2 The Risk Ratio as Variable Importance Measure

The risk ratio (RR) is another popular parameter that arises in statistical causal inference, denoted $\psi_0 = \frac{\psi_0(1)}{\psi_0(0)}$, representing the multiplicative contrast of an outcome between the counterfactuals under which all subjects received the treatment/exposure and under which none received such treatment/exposure. Under additional (untestable) assumptions,

this parameter has a richer interpretation as a mean counterfactual outcome, wherein the counterfactuals used in this definition define causal effects. When causal assumptions remain unfulfilled or untested, this parameter may still be estimated in the form of a nonparametric VIM.

Just as before (in the case of the ATE), there are two regression steps required for estimating VIMs based on this parameter. We do so in a manner analogous to that described previously.

0.5.2.1 Super Learning for nonparametric parameter estimation

Nonparametric and data-adaptive regressions (i.e., machine learning) may be used in the two regression steps required for estimating a VIM based on the RR. This is implemented using the Super Learner algorithm.

`methyvim` makes performing such estimation for CpG sites using a given VIM essentially trivial:

```
methyvim_rr_sl <- methyvim(data_grs = grsExample, sites_comp = 25,
                           var_int = var_int, vim = "rr", type = "Mval",
                           filter = "limma", filter_cutoff = 0.10,
                           parallel = FALSE, tmle_type = "sl"
                          )
## Warning in set_parallel(parallel = parallel, future_param = future_param, : Sequential evaluation is st
##  Proceed with caution.
methyvim_rr_sl
## class: methytmle
## dim: 400 210
## metadata(0):
## assays(2): Beta M
## rownames(400): cg23578515 cg06747907 ... cg01715842 cg09895959
## rowData names(0):
## colnames(210): V2 V3 ... V397 V398
## colData names(2): exp outcome
## Annotation
##   array: IlluminaHumanMethylationEPIC
## Preprocessing
##   Method: NA
##   minfi version: NA
##   Manifest version: NA
```

We can take a look at the results produced from the estimation procedure by examining the `"vim"` slot of the `methytmle` object:

```
head(slot(methyvim_rr_sl, "vim"))
##            lowerCI_logRR    est_logRR upperCI_logRR    Var_logRR      pval
## cg22913481   -0.16032506 -0.043923643    0.07247777 0.0035269911 0.4595435
## cg15131207   -0.10832796 -0.036151215    0.03602553 0.0013560711 0.3262445
## cg10613282   -0.18416020 -0.070114980    0.04393024 0.0033856500 0.2282004
## cg15857610   -0.07554678 -0.030125512    0.01529575 0.0005370396 0.1936134
## cg24775884   -0.07990909  0.007717539    0.09534417 0.0019987575 0.8629477
## cg22954484   -0.06531915  0.018479664    0.10227848 0.0018279469 0.6655762
##            n_neighbors_all n_neighbors_w max_corr_w
## cg22913481               0            0         NA
## cg15131207               0            0         NA
## cg10613282               0            0         NA
## cg15857610               5            4  0.7720705
## cg24775884               5            2  0.8897566
## cg22954484               5            2  0.8643936
```

0.5.2.2 Generalized linear models for parameter estimation

In cases where nonparametric regressions may not be preferred (e.g., where time constraints are of concern), generalized linear models (GLMs) may be used to fit the two regression steps required for estimating a VIMs for the ATE.

`methyvim` makes performing such estimation for CpG sites using a given VIM essentially trivial:

```
methyvim_rr_glm <- methyvim(data_grs = grsExample, sites_comp = 25,
                            var_int = var_int, vim = "rr", type = "Mval",
                            filter = "limma", filter_cutoff = 0.10,
                            parallel = FALSE, tmle_type = "glm"
                           )
## Warning in set_parallel(parallel = parallel, future_param = future_param, : Sequential evaluation is st
##  Proceed with caution.
methyvim_rr_glm
## class: methytmle
## dim: 400 210
## metadata(0):
## assays(2): Beta M
## rownames(400): cg23578515 cg06747907 ... cg01715842 cg09895959
## rowData names(0):
## colnames(210): V2 V3 ... V397 V398
## colData names(2): exp outcome
## Annotation
##   array: IlluminaHumanMethylationEPIC
## Preprocessing
##   Method: NA
##   minfi version: NA
##   Manifest version: NA
```

Just as before, we can take a look at the results produced from the estimation procedure by examining the `"vim"` slot of the `methytmle` object:

```
head(slot(methyvim_rr_glm, "vim"))
##            lowerCI_logRR   est_logRR upperCI_logRR    Var_logRR      pval
## cg22913481   -0.15978435 -0.04298097    0.07382242 0.0035513930 0.4707649
## cg15131207   -0.10832796 -0.03615121    0.03602553 0.0013560711 0.3262445
## cg10613282   -0.18349779 -0.06986673    0.04376433 0.0033611044 0.2281579
## cg15857610   -0.07868970 -0.01713262    0.04442445 0.0009863789 0.5854034
## cg24775884   -0.06952379  0.01867609    0.10687598 0.0020249948 0.6781237
## cg22954484   -0.05704721  0.02698683    0.11102087 0.0018382237 0.5290626
##            n_neighbors_all n_neighbors_w max_corr_w
## cg22913481               0             0         NA
## cg15131207               0             0         NA
## cg10613282               0             0         NA
## cg15857610               5             4  0.7720705
## cg24775884               5             2  0.8897566
## cg22954484               5             2  0.8643936
```

Remark: Here, the estimates are obtained via GLMs, making each of the regression steps less robust than if nonparametric regressions were used. It is expected that these estimates differ from those obtained previously.

0.6 Data Analysis with methyvim

In order to explore practical applications of the methyvim package, as well as the full set of utilities it provides, our toy example (of just 10 CpG sites) is unfortunately insufficient. To proceed, we will use a publicly available example data set produced by the Illumina 450K array, from the minfiData R package. Now, let's load the package and data set, and take a look

```
suppressMessages(library(minfiData))
data(MsetEx)
mset <- mapToGenome(MsetEx)
grs <- ratioConvert(mset)
grs
## class: GenomicRatioSet
## dim: 485512 6
## metadata(0):
## assays(2): Beta CN
## rownames(485512): cg13869341 cg14008030 ... cg08265308 cg14273923
## rowData names(0):
## colnames(6): 5723646052_R02C02 5723646052_R04C01 ...
##   5723646053_R05C02 5723646053_R06C02
## colData names(13): Sample_Name Sample_Well ... Basename filenames
## Annotation
##   array: IlluminaHumanMethylation450k
##   annotation: ilmn12.hg19
## Preprocessing
##   Method: Raw (no normalization or bg correction)
##   minfi version: 1.21.2
##   Manifest version: 0.4.0
```

After loading the data, which comes in the form of a raw MethylSet object, we perform some further processing by mapping to the genome (with mapToGenome) and converting the values from the methylated and unmethylated channels to Beta-values (via ratioConvert). These two steps together produce an object of class GenomicRatioSet, like what we had worked with previously.

For this example analysis, we'll treat the condition of the patients as the exposure/treatment variable of interest. The methyvim function requires that this variable either be numeric or easily coercible to numeric. To facilitate this, we'll simply convert the covariate (currently a character):

```
var_int <- (as.numeric(as.factor(colData(grs)$status)) - 1)
```

n.b., the re-coding process results in "normal" patients being assigned a value of 1 and cancer patients a 0.

Now, we are ready to analyze the effects of cancer status on DNA methylation using this data set. To do this with a targeted minimum loss-based estimate of the Average Treatment Effect, we may proceed as follows:

```
suppressMessages(
  methyvim_cancer_ate <- methyvim(data_grs = grs, var_int = var_int,
                                  vim = "ate", type = "Beta", filter = "limma",
                                  filter_cutoff = 0.20, obs_per_covar = 2,
                                  parallel = FALSE, sites_comp = 125,
                                  tmle_type = "glm"
                                 )
)
## Warning in set_parallel(parallel = parallel, future_param = future_param, : Sequential evaluation is st
##  Proceed with caution.
```

Note that we set the obs_per_covar argument to a relatively low value (2, where the recommended default is 20) for the purposes of this example. We do this only to exemplify the estimation procedure and would point out here that such

low values will compromise the quality of inference obtained as this setting directly affects the definition of the target parameter.

Further, note that here we apply the `glm` flavor of the `tmle_type` argument, which produces faster results by fitting models for the propensity score and outcome regressions using a limited number of parametric models. By contrast, the `sl` (for "Super Learning") flavor fits these two regressions using highly nonparametric and data-adaptive procedures (i.e., via machine learning).

Just as before, we can view a table of results by examining the `vim` slot of the produced `methytmle` object:

```
head(slot(methyvim_cancer_ate, "vim"))
##            lowerCI_ATE      est_ATE  upperCI_ATE      Var_ATE         pval
## cg14008030 -0.11956597 -0.031415962  0.056734049 2.022705e-03 4.848468e-01
## cg20253340 -0.08850637 -0.058866142 -0.029225917 2.286919e-04 9.917426e-05
## cg21870274 -0.09499057 -0.029118982  0.036752609 1.129495e-03 3.862537e-01
## cg17308840 -0.04626018 -0.007152452  0.031955277 3.981191e-04 7.199943e-01
## cg00645010 -0.02697515 -0.013809263 -0.000643373 4.512199e-05 3.980386e-02
## cg27534567  0.06745648  0.115711854  0.163967223 6.061486e-04 2.602936e-06
##            n_neighbors_all n_neighbors_w max_corr_w
## cg14008030               0             0         NA
## cg20253340               0             0         NA
## cg21870274               2             1  0.9443580
## cg17308840               2             1  0.9443580
## cg00645010               2             2  0.5236810
## cg27534567               1             0  0.9362968
```

Finally, we may compute FDR-corrected p-values, by applying a modified procedure for controlling the False Discovery Rate for multi-stage analyses (FDR-MSA) (Tuglus and van der Laan 2009). We do this by simply applying the `fdr_msa` function:

```
fdr_p <- fdr_msa(pvals = slot(methyvim_cancer_ate, "vim")$pval,
                 total_obs = nrow(methyvim_cancer_ate))
```

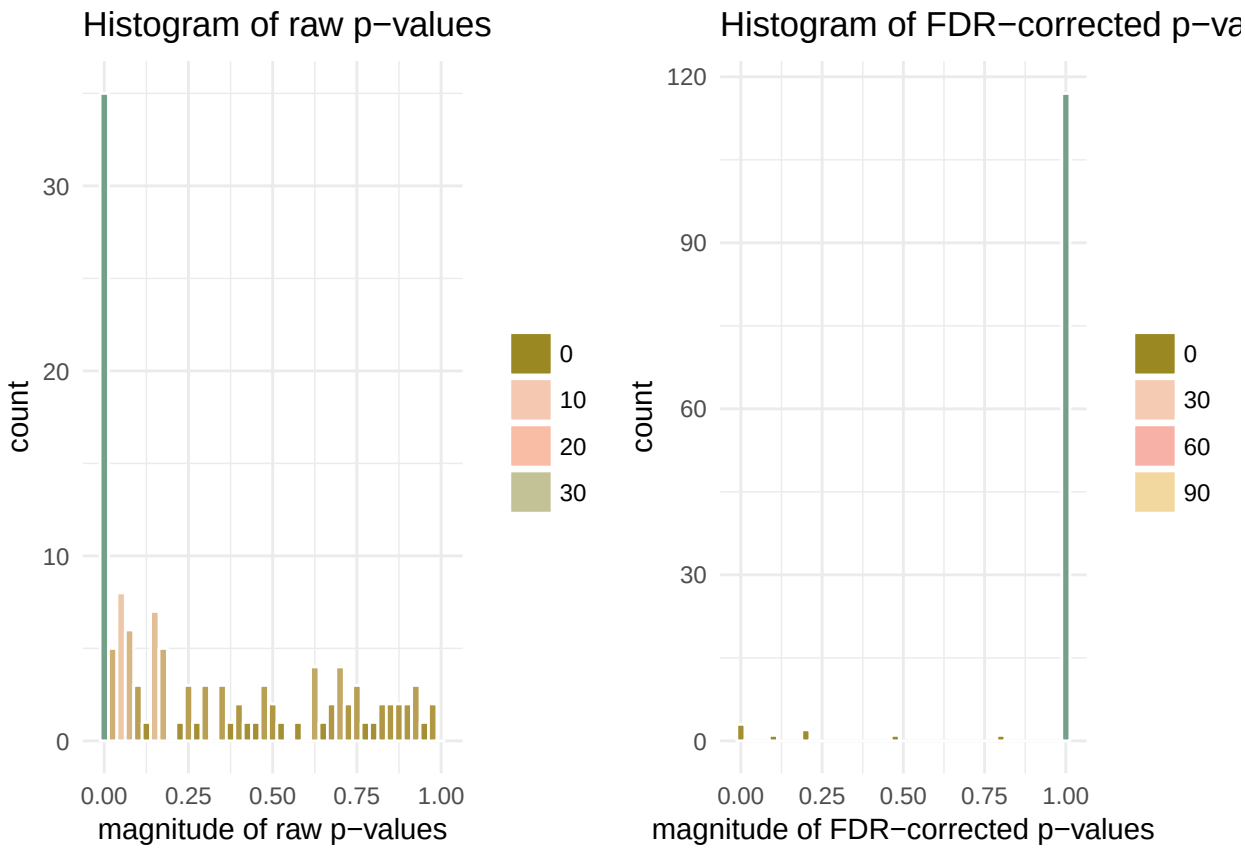
Having explored the results of our analysis numerically, we now proceed to use the visualization tools provided with the `methyvim` R package to further enhance our understanding of the results.

0.6.1 Visualization of Results

While making allowance for users to explore the full set of results produced by the estimation procedure (by way of exposing these directly to the user), the `methyvim` package also provides *three* (3) visualization utilities that produce plots commonly used in examining the results of differential methylation analyses.

A simple call to `plot` produces side-by-side histograms of the raw p-values computed as part of the estimation process and the corrected p-values obtained from using the FDR-MSA procedure.

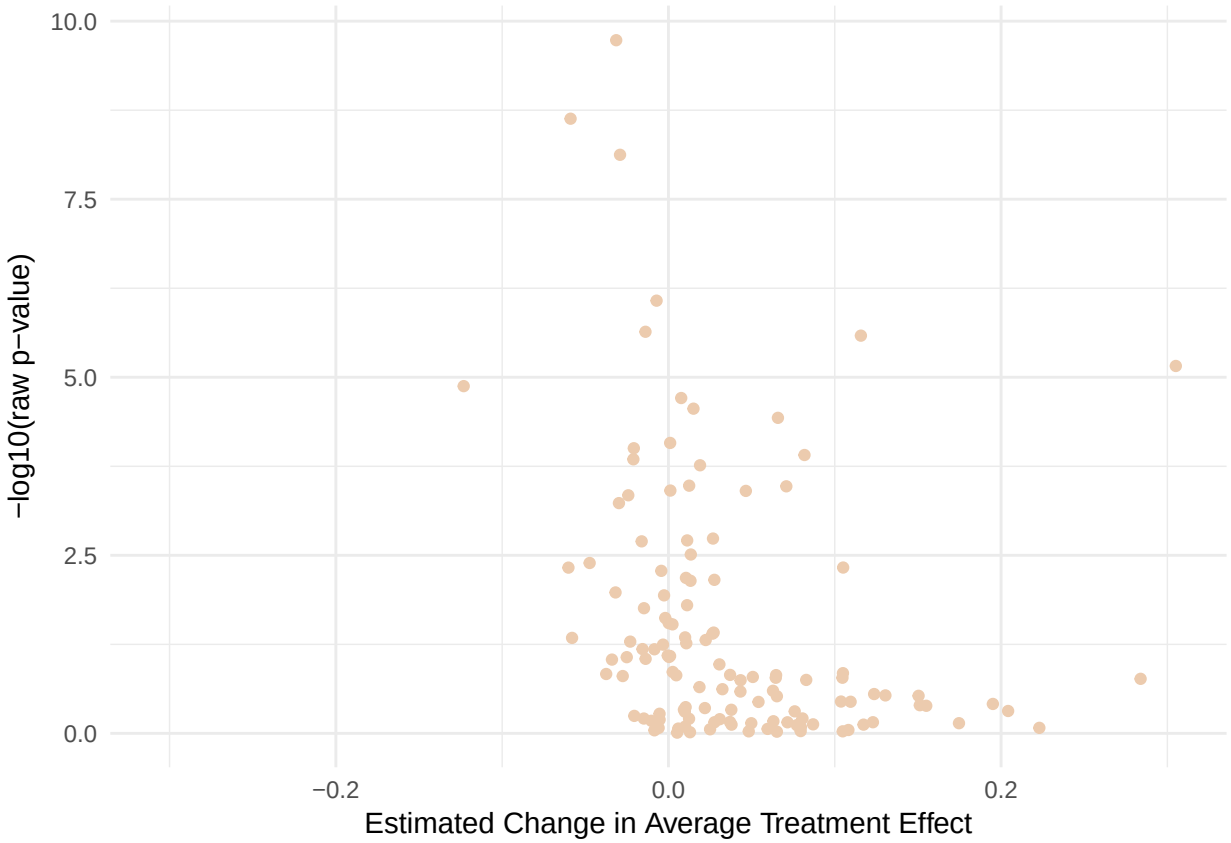
```
plot(methyvim_cancer_ate)
```



Remark: The plots displayed above may also be generated separately by explicitly setting the argument "type" to `plot.methytmle`. For a plot of the raw p-values, specify `type = "raw_pvals"`, and for a plot of the FDR-corrected p-values, specify `type = "fdr_pvals"`.

While histograms of the p-values may be generally useful in inspecting the results of the estimation procedure, a more common plot used in examining the results of differential methylation procedures is the volcano plot, which plots the parameter estimate along the x-axis and $-\log_{10}(\text{p-value})$ along the y-axis. We implement such a plot in the `methyvolc` function:

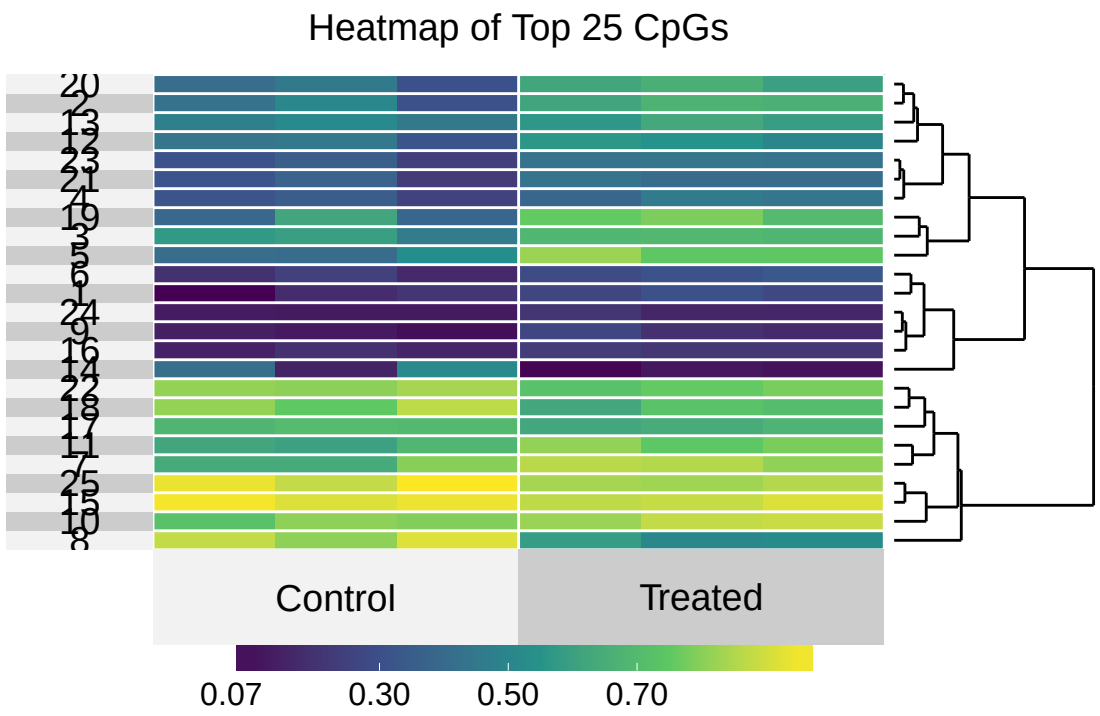
```
methyvolc(methyvim_cancer_ate)
```



The purpose of such a plot is to ensure that very low (possibly statistically significant) p-values do not arise from cases of low variance. This appears to be the case in the plot above (notice that most parameter estimates are *near zero*, even in cases where the raw p-values are quite low).

Yet another popular plot for visualizing effects in such settings is the heatmap, which plots estimates of the raw methylation effects (as measured by the assay) across subjects using a heat gradient. We implement this in the `methyheat` function:

```
methyheat(methyvim_cancer_ate)
```



Invoking `methyheat` in this manner produces a plot of the top sites (25, by default) based on the raw p-value, using the raw methylation measures in the plot. This uses the exceptional `superheat` R package (Barter and Yu 2017).

0.7 Session Information

```
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-apple-darwin16.7.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A
## LAPACK: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
##  [1] splines   stats4    parallel  methods   stats     graphics  grDevices
##  [8] utils     datasets  base
##
## other attached packages:
##  [1] bindrcpp_0.2
##  [2] minfiData_0.22.0
##  [3] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.6.0
```

```
##  [4] IlluminaHumanMethylation450kmanifest_0.4.0
##  [5] arm_1.9-3
##  [6] lme4_1.1-13
##  [7] Matrix_1.2-11
##  [8] MASS_7.3-47
##  [9] earth_4.5.1
## [10] plotmo_3.3.4
## [11] TeachingDemos_2.10
## [12] plotrix_3.6-6
## [13] nnet_7.3-12
## [14] gam_1.14-4
## [15] methyvimData_0.99.4
## [16] methyvim_0.99.2
## [17] minfi_1.22.1
## [18] bumphunter_1.16.0
## [19] locfit_1.5-9.1
## [20] iterators_1.0.8
## [21] foreach_1.4.3
## [22] Biostrings_2.44.2
## [23] XVector_0.16.0
## [24] SummarizedExperiment_1.6.4
## [25] DelayedArray_0.2.7
## [26] matrixStats_0.52.2
## [27] Biobase_2.36.2
## [28] GenomicRanges_1.28.5
## [29] GenomeInfoDb_1.12.2
## [30] IRanges_2.10.3
## [31] S4Vectors_0.14.4
## [32] BiocGenerics_0.22.0
## [33] tmle_1.2.0-5
## [34] SuperLearner_2.0-23-9000
## [35] nnls_1.4
## [36] BiocStyle_2.4.1
## [37] dplyr_0.7.3
## [38] purrr_0.2.3
## [39] readr_1.1.1
## [40] tidyr_0.7.1
## [41] tibble_1.3.4
## [42] ggplot2_2.2.1
## [43] tidyverse_1.1.1
## [44] devtools_1.13.3
## [45] nima_0.4.5
## [46] fcuk_0.1.21
## [47] prettycode_1.0.0
##
## loaded via a namespace (and not attached):
##   [1] readxl_1.0.0             backports_1.1.0
##   [3] plyr_1.8.4               lazyeval_0.2.0
##   [5] listenv_0.6.0            BiocParallel_1.10.1
##   [7] digest_0.6.12            htmltools_0.3.6
##   [9] wesanderson_0.3.2        magrittr_1.5
##  [11] memoise_1.1.0            cluster_2.0.6
##  [13] limma_3.32.7             globals_0.10.2
```

```
##  [15] annotate_1.54.0          modelr_0.1.1
##  [17] doFuture_0.5.1           siggenes_1.50.0
##  [19] colorspace_1.3-2         blob_1.1.0
##  [21] rvest_0.3.2              haven_1.1.0
##  [23] crayon_1.3.2             RCurl_1.95-4.8
##  [25] jsonlite_1.5             genefilter_1.58.1
##  [27] bindr_0.1                GEOquery_2.42.0
##  [29] survival_2.41-3          glue_1.1.1
##  [31] registry_0.3             gtable_0.2.0
##  [33] zlibbioc_1.22.0          superheat_0.1.0
##  [35] ProjectTemplate_0.8      abind_1.4-5
##  [37] scales_0.5.0             DBI_0.7
##  [39] rngtools_1.2.4           ggthemes_3.4.0
##  [41] Rcpp_0.12.12             xtable_1.8-2
##  [43] foreign_0.8-69           bit_1.1-12
##  [45] mclust_5.3               preprocessCore_1.38.1
##  [47] httr_1.3.1               RColorBrewer_1.1-2
##  [49] pkgconfig_2.0.1          reshape_0.8.7
##  [51] XML_3.98-1.9             labeling_0.3
##  [53] rlang_0.1.2.9000         reshape2_1.4.2
##  [55] AnnotationDbi_1.38.2     munsell_0.4.3
##  [57] cellranger_1.1.0         tools_3.4.1
##  [59] RSQLite_2.0              broom_0.4.2
##  [61] ggdendro_0.1-20          evaluate_0.10.1
##  [63] stringr_1.2.0.9000       yaml_2.1.14
##  [65] knitr_1.17               bit64_0.9-7
##  [67] beanplot_1.2             future_1.6.1
##  [69] nlme_3.1-131             doRNG_1.6.6
##  [71] nor1mix_1.2-3            xml2_1.1.1
##  [73] biomaRt_2.32.1           compiler_3.4.1
##  [75] stringi_1.1.5            GenomicFeatures_1.28.4
##  [77] forcats_0.2.0            lattice_0.20-35
##  [79] nloptr_1.0.4             psych_1.7.8
##  [81] multtest_2.32.0          stringdist_0.9.4.6
##  [83] data.table_1.10.4        bitops_1.0-6
##  [85] rtracklayer_1.36.4       R6_2.2.2
##  [87] gridExtra_2.3            codetools_0.2-15
##  [89] gtools_3.5.0             assertthat_0.2.0
##  [91] openssl_0.9.7            pkgmaker_0.22
##  [93] rprojroot_1.2            withr_2.0.0
##  [95] GenomicAlignments_1.12.2 Rsamtools_1.28.0
##  [97] mnormt_1.5-5             GenomeInfoDbData_0.99.0
##  [99] hms_0.3                  quadprog_1.5-5
##  [ reached getOption("max.print") -- omitted 7 entries ]
```

References

Aryee, Martin J, Andrew E Jaffe, Hector Corrada-Bravo, Christine Ladd-Acosta, Andrew P Feinberg, Kasper D Hansen, and Rafael A Irizarry. 2014. "Minfi: A Flexible and Comprehensive Bioconductor Package for the Analysis of Infinium DNA Methylation Microarrays." *Bioinformatics* 30 (10). Oxford University Press (OUP): 1363–9.

doi:10.1093/bioinformatics/btu049.

Barter, Rebecca L, and Bin Yu. 2017. "Superheat: An R Package for Creating Beautiful and Extendable Heatmaps for Visualizing Complex Data."

Benjamini, Yoav, and Yosef Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society. Series B (Methodological)*. JSTOR, 289–300.

Breiman, Leo. 1996. "Stacked Regressions." *Machine Learning* 24 (1). Springer: 49–64.

Chambaz, Antoine, Pierre Neuvial, and Mark J van der Laan. 2012. "Estimation of a Non-Parametric Variable Importance Measure of a Continuous Exposure." *Electronic Journal of Statistics* 6. NIH Public Access: 1059.

Dedeurwaerder, Sarah, Matthieu Defrance, Martin Bizet, Emilie Calonne, Gianluca Bontempi, and François Fuks. 2013. "A Comprehensive Overview of Infinium Humanmethylation450 Data Processing." *Briefings in Bioinformatics*. Oxford Univ Press, bbt054.

Fortin, Jean-Philippe, Aurelie Labbe, Mathieu Lemire, Brent W Zanke, Thomas J Hudson, Elana J Fertig, Celia MT Greenwood, and Kasper D Hansen. 2014. "Functional Normalization of 450k Methylation Array Data Improves Replication in Large Cancer Studies." *bioRxiv*. Cold Spring Harbor Labs Journals.

Hernan, Miguel A, and James M Robins. 2018, forthcoming. *Causal Inference*. Chapman & Hall/Crc Texts in Statistical Science. Taylor & Francis.

Pearl, Judea. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Tuglus, Catherine, and Mark J. van der Laan. 2009. "Modified FDR Controlling Procedure for Multi-Stage Analyses." *Statistical Applications in Genetics and Molecular Biology* 8 (1). Walter de Gruyter: 1–15. doi:10.2202/1544-6115.1397.

van der Laan, Mark J, and Sherri Rose. 2011. *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer Science & Business Media.

———. 2017. *Targeted Learning in Data Science: Causal Inference for Complex Longitudinal Studies*. Springer Science & Business Media.

van der Laan, Mark J, Eric C Polley, and Alan E Hubbard. 2007. "Super Learner." *Statistical Applications in Genetics and Molecular Biology* 6 (1).