


Example of Methylome Analysis with MethylIT using Cancer Datasets

Robersy Sanchez¹ 
rus547@psu.edu

Sally Mackenzie^{1,2} 
sam795@psu.edu

¹Department of Biology, Pennsylvania State University, University Park, PA 16802

²Department of Plant Science, Pennsylvania State University, University Park, PA 16802

23 January 2020

An example of methylation analysis with Methyl-IT, a novel methylome analysis procedure based on a signal-detection and machine-learning approach, is provided here. Methylation analysis involves a signal detection problem, and the method was designed to discriminate methylation regulatory signal from background noise induced by thermal fluctuations. Methyl-IT enhances the resolution of genome methylation behavior to reveal network-associated responses, offering resolution of gene pathway influences not attainable with previous methods. Herein, an example of MethylIT application to the analysis of breast cancer methylomes is presented.

Table of Contents

Methyl-IT	2
Available datasets and reading	3
<i>Reading datasets</i>	3
The reference individual	4
Hellinger divergence estimation	6
<i>Descriptive statistical analysis</i>	8
<i>Histogram and boxplots of divergences estimated in each sample</i>	8
<i>Checking cytosine read counts</i>	11
Nonlinear fit of probability distribution models	13
Signal detection	15
<i>Potential methylation signal</i>	15
<i>Histogram and boxplots of methylation potential signals</i>	18
Cutpoint estimation	20

<i>Cutpoint based on Youden index</i>	21
<i>Cutpoint estimated using a model classifier</i>	23
DMPs	24
The descriptive statistic of the read counts at DMPs	24
<i>Venn Diagram of DMPs</i>	25
Monte Carlo Evaluation of the Model Classification Performance.....	27
Differentially informative methylated genomic regions (DIMRs)	28
<i>Differentially methylated genes (DMGs)</i>	28
Concluding remarks.....	33
Acknowledgments	33
S2. Session Information	33
References.....	35

Methyl-IT

MethylIT is an R package for methylome analysis based on a signal-detection machine learning approach (SD-ML). This approach is postulated to provide greater sensitivity for resolving true signal from the methylation background within the methylome (Sanchez and Mackenzie 2016; Sanchez et al. 2019). Because the biological signal created within the dynamic methylome environment characteristic of plants is not free from background noise, the approach, designated Methyl-IT, includes the application of signal detection theory (Greiner, Pfeiffer, and Smith 2000; Carter et al. 2016; Harpaz et al. 2013; Kruspe et al. 2017). A basic requirement for the application of signal detection is the knowledge of the probability distribution of the background noise, which is used as null hypothesis in the detection of the methylation signal. This probability distribution is, in general, a member of generalized gamma distribution family, and it can be deduced on a statistical mechanical/thermodynamics basis from DNA methylation induced by thermal fluctuations (Sanchez and Mackenzie 2016).

Herein, we provide an example of Methyl-IT application to the analysis of breast cancer methylomes. Due to the size of human methylome the current example only covers the analysis of chromosome 13. A full description of Methyl-IT application of methylome analysis in plants is given in the manuscript (Sanchez et al. 2019).

The R packages required in this example are:

```
library(MethylIT)
library(ggplot2) # graphic
library(reshape2) # To reshape the data frame
library(grid) # For multiple plots
library(gridExtra) # For multiple plots
```

```
library(VennDiagram)
library(rtracklayer) # To import gene annotation
```

Available datasets and reading

Methylome datasets of whole-genome bisulfite sequencing (WGBS) are available at Gene Expression Omnibus (GEO DataSets). For the current example, datasets from breast tissues (normal and cancer) and embryonic stem cells will be downloaded from GEO. The data set are downloaded providing the GEO accession numbers for each data set to the function 'getGEOSuppFiles' (see the function [getGEOSuppFiles](#) help). Some times the ftp site is not available and the information must be manually downloaded from GEO database.

```
# Embryonic stem cells datasets
esc.files = getGEOSuppFiles(GEO = c("GSM2041690", "GSM2041691", "GSM2041692"),
                             verbose = FALSE)
# Breast tissues (normal, cancer, metastasis)
cancer.files = getGEOSuppFiles(GEO = c("GSM1279517", "GSM1279514",
                                         "GSM1279513"), verbose = FALSE)
```

The file path and name of each downloaded dataset is found in the output variables 'esc.files' and 'cancer.files'.

Reading datasets

Datasets for our example can be read with function [readCounts2GRangesList](#). To specify the reading of only chromosome 13, we can specify the parameter 'chromosomes = "Chr13"'. The symbol chromosome 13, in this case "Chr13", must be consistent with the annotation provided in the given GEO dataset. Each file is wholly read with the setting 'chromosomes = "Chr13"' and then the GRanges are built only with chromosome 13, which could be time consuming. However, users working on Linux OS can specify the reading of specific lines from each file by using regular expressions.

For example, if only chromosomes 1 and 3 are required, then we can set chromosomes = NULL (default) and 'chromosome.pattern = "^Chr[1,3]"'. This will read all the lines in the downloaded files starting with the words "Chr1" or "Chr3". If we are interested in chromosomes 1 and 2, then we can set 'chromosome.pattern = "^Chr[1-2]"'. If all the chromosomes are required, then set chromosomes = NULL and chromosome.pattern = NULL (default). Obviously, before read the files, user must check them to see which annotation was used by the experimenters to denote the chromosomes, e.g., "chr13" or just "13", etc.

```
# Embryonic stem cells datasets
ref = readCounts2GRangesList(filenamees = esc.files,
                             sample.id = c("ESC1", "ESC2", "ESC3"),
                             columns = c(seqnames = 1, start = 2,
                                           mC = 4, uC = 5), pattern = "^chr13",
                             remove = TRUE, verbose = FALSE)
# Breast tissues (normal, cancer, metastasis)
LR = readCounts2GRangesList(filenamees = cancer.files,
                             sample.id = c("Breast_normal", "Breast_cancer",
```

```

        "Breast_metastasis"),
columns = c(seqnames = 1, start = 2,
            mC = 3, uC = 4),
remove = TRUE, pattern = "^13",
chromosome.names = "chr13", verbose = FALSE)

```

In real data analysis, a bigger sample size must be used. For the purpose of this example, to illustrate the application of methylation analysis with Methyl-IT, we will just add an artificial normal sample.

```

LR$Breast_normal11 <- poolFromGRlist(c(ref[1], LR$Breast_normal,
                                       LR$Breast_normal, LR$Breast_normal,
                                       LR$Breast_normal, LR$Breast_normal),
stat = "mean", num.cores = 6L, verbose = FALSE)

```

In the metacolumn of the last GRanges object, mC and uC stand for the methylated and unmethylated read counts, respectively. Notice that option 'remove = TRUE' remove the decompressed files (default: FALSE, see ?readCounts2GRangesList for more details about this function).

The reference individual

Any two objects can be compared based on some measured variable if, and only if, the measurements were taken in the same metric space, the same coordinate system, in respect to the same origin of coordinates. Usually, in our daily 3D experience, our brain automatically sets up the origin of coordinates equal to zero. The differences found in the comparison depend on the reference used to perform the measurements and from the metric system. The space where the objects are located (or the set of objects) together with the metric is called metric space.

Unfortunately, this is not the case for the methylation process. Each individual from the same population and even each chromosome follows an independent stochastic methylation process. This is not a surprise, since cytosine methylation affects the mechanical properties of DNA molecule with strong dependence of the DNA sequence context (until, at least, 6 bases at both sides of each cytosine site) (Severin et al. 2011; Ngo et al. 2016).

To evaluate the methylation differences between individuals from control and treatment we introduce a metric in the bidimensional space of methylation levels $P_i = (p_i, 1 - p_i)$. Vectors P_i provide a measurement of the uncertainty of methylation levels (Sanchez et al. 2019). However, to perform the comparison between the uncertainty of methylation levels from each group of individuals, control (c) and treatment (t), we should estimate the uncertainty variation with respect to the same individual reference on the mentioned metric space. The reason to measure the uncertainty variation with respect to the same reference resides in that even sibling individuals follow an independent ontogenetic development and, consequently, their developments follow independent stochastic processes. This is a consequence of the "omnipresent" action of the second law of

thermodynamics in living organisms. In the current example, we will create the reference individual by pooling the methylation counts from the embryonic stem cells.

It should be noticed that the results are sensitive to the reference used. The statistics mean, median, or sum of the read counts at each cytosine site of some control samples can be used to create a virtual reference sample. A virtual reference individual can be built with function `poolFromGRlist`. It is up to the user whether to apply the 'row sum', 'row mean' or 'row median' of methylated and unmethylated read counts at each cytosine site across individuals. Notice that when "mean" is selected the virtual reference individual is the group centroid, which play a fundamental role in multivariate statistic.

```
Ref = poolFromGRlist(ref, stat = "mean", num.cores = 6L, verbose = FALSE)
```

Ref

```
## GRanges object with 1560637 ranges and 2 metadata columns:
##           seqnames      ranges strand |           mC           uC
##           <Rle> <IRanges>  <Rle> | <numeric> <numeric>
##      [1]   chr13  19020631      * |           1           1
##      [2]   chr13  19020633      * |           2           2
##      [3]   chr13  19020642      * |           1           1
##      [4]   chr13  19020643      * |           2           2
##      [5]   chr13  19020679      * |           1           1
##      ...     ...      ...      ... |     ...     ...
## [1560633]   chr13 115108993      * |           1           3
## [1560634]   chr13 115109022      * |           1           1
## [1560635]   chr13 115109023      * |           3           4
## [1560636]   chr13 115109523      * |           2           2
## [1560637]   chr13 115109524      * |           1           1
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Only direct lab experiments can reveal whether differences detected with distinct references outside the experimental conditions for control and treatment groups are real. The best reference would be estimated using a subset of individuals from control group. Such a reference will contribute to remove the intragroup variation, in control and in treatment groups, induced by environmental changes external to or not controlled by the experimental conditions.

Methylation analysis for each cytosine position is frequently performed in the bidimensional space of (*methylated*, *unmethylated*) read counts. Frequently, Fisher's exact test is applied to a single cytosine position, under the null hypothesis that the proportions $p_{ct} = \text{methylated}_{ct} / (\text{methylated}_{ct} + \text{unmethylated}_{ct})$ and $p_{tt} = \text{methylated}_{tt} / (\text{methylated}_{tt} + \text{unmethylated}_{tt})$ are the same for control and treatment, respectively. In this case, the implicit reference point for the counts at every cytosine positions is (*methylated* = 0, *unmethylated* = 0), which corresponds to the point $P_i = (0,1)$.

In our case, the Hellinger divergence (one of the metric used, here) of each individual in respect to the reference is the variable to test in place of (*methylated*, *unmethylated*) read counts or the methylation levels $P_i = (p_i, 1 - p_i)$ (Sanchez et al. 2019).

The use of references is restricted by the thermodynamics basis of the theory. The current information-thermodynamics based approach is supported on the following postulate:

“High changes of Hellinger divergences are less frequent than low changes, provided that the divergence is proportional to the amount of energy required to process the change of information in the methylation system”.

The last postulate acknowledges the action of the second law of thermodynamics on the biomolecular methylation system. For the methylation system, it implies that the frequencies of the information divergences between methylation levels must be proportional to a Boltzmann factor (see supplementary information from reference (Sanchez and Mackenzie 2016)). In other words, the frequencies of information divergences values should follow a trend proportional to an exponential decay. If we do not observe such a behavior, then either the reference is too far from experimental condition or we are dealing with an extreme situation where the methylation machinery in the cell is dysfunctional. The last situation is found, for example, in the silencing mutation at the gene of cytosine-DNA-methyltransferase in *Arabidopsis thaliana*. Methylation of 5-methylcytosine at CpG dinucleotides is maintained by MET1 in plants.

Hellinger divergence estimation

To perform the comparison between the uncertainty of methylation levels from each group of individuals, control (c) and treatment (t), the divergence between the methylation levels of each individual is estimated with respect to the same reference on the metric space formed by the vector set $P_i = (p_i, 1 - p_i)$ and the Hellinger divergence H . Basically, the information divergence between the methylation levels of an individual j and reference sample r is estimated according to the Hellinger divergence given by the formula:

$$H(\hat{p}_{ij}, \hat{p}_{ir}) = w_i [(\sqrt{\hat{p}_{ij}} - \sqrt{\hat{p}_{ir}})^2 + (\sqrt{1 - \hat{p}_{ij}} - \sqrt{1 - \hat{p}_{ir}})^2]$$

where $w_i = 2 \frac{m_{ij}m_{ir}}{m_{ij}+m_{ir}}$, $m_{ij} = n_i^{mCj} + n_i^{uCj} + 1$, $m_{ir} = n_i^{mCr} + n_i^{uCr} + 1$ and $j \in \{c, t\}$. This equation for Hellinger divergence is given in reference (Basu, Mandal, and Pardo 2010), but other information theoretical divergences can be used as well. Next, the information divergence for control (Breast_normal) and treatment (Breast_cancer and Breast_metastasis) samples are estimated with respect to the reference virtual individual. A Bayesian correction of counts can be selected or not. In a Bayesian framework, methylated read counts are modeled by a beta-binomial distribution, which accounts for both the biological and sampling variations (Hebestreit, Dugas, and Klein 2013; Robinson et al. 2014; Dolzhenko and Smith 2014). In our case we adopted the Bayesian approach suggested in reference (Baldi and Brunak 2001) (Chapter 3). In a Bayesian framework with uniform priors, the methylation level can be defined as: $p = (mC + 1)/(mC + uC + 2)$.

However, the most natural statistical model for replicated BS-seq DNA methylation measurements is beta-binomial (the beta distribution is a prior conjugate of binomial distribution). We consider the parameter p (methylation level) in the binomial distribution

as randomly drawn from a beta distribution. The hyper-parameters α and β from the beta-binomial distribution are interpreted as pseudo-counts (Sanchez et al. 2019). The information divergence is estimated here using the function `estimateDivergence`:

```
HD = estimateDivergence(ref = Ref, indiv = LR, Bayesian = TRUE,
                        min.coverage = 5, high.coverage = 300,
                        percentile = 0.999, num.cores = 6L, tasks = 0L,
                        verbose = FALSE)

HD$Breast_cancer

## GRanges object with 583235 ranges and 9 metadata columns:
##           seqnames      ranges strand |           c1           t1           c2           t2
##           <Rle> <IRanges> <Rle> | <numeric> <numeric> <numeric> <numeric>
##      [1] chr13  19020631      * |           1           1          14          24
##      [2] chr13  19020633      * |           2           2          14          25
##      [3] chr13  19020643      * |           2           2           7          38
##      [4] chr13  19020782      * |           1           1           6          56
##      [5] chr13  19020786      * |           1           1          10          53
##      ...      ...      ...      ... |      ...      ...      ...      ...
## [583231] chr13 115108776      * |           1           1          52          20
## [583232] chr13 115108789      * |           2           2          27          43
## [583233] chr13 115108993      * |           1           3          72           5
## [583234] chr13 115109023      * |           3           4          56          36
## [583235] chr13 115109524      * |           1           1          31           9
##
##           p1           p2           TV
##           <numeric> <numeric> <numeric>
##      [1] 0.454113362273475 0.390954889464515 -0.131578947368421
##      [2] 0.454348995976309 0.381553592948877 -0.141025641025641
##      [3] 0.454348995976309 0.186339122607807 -0.344444444444444
##      [4] 0.454113362273475 0.121807860928068 -0.403225806451613
##      [5] 0.454113362273475 0.180939975343107 -0.341269841269841
##      ...      ...      ...
## [583231] 0.454113362273475 0.722220749983958 0.222222222222222
## [583232] 0.454348995976309 0.397697909666978 -0.114285714285714
## [583233] 0.451781432579768 0.928149668925976 0.685064935064935
## [583234] 0.453423974969116 0.611797407851488 0.180124223602485
## [583235] 0.454113362273475 0.771793453785385 0.275
##
##           bay.TV           hdiv
##           <numeric> <numeric>
##      [1] -0.06315847280896 0.0227965968799609
##      [2] -0.0727954030274322 0.0484806896715985
##      [3] -0.268009873368502 0.770572964508297
##      [4] -0.332305501345407 0.829619642501239
##      [5] -0.273173386930369 0.512220461858664
##      ...      ...
## [583231] 0.268107387710483 0.436946710684087
## [583232] -0.0566510863093313 0.0306804911627062
## [583233] 0.476368236346207 2.89411608418132
## [583234] 0.158373432882373 0.373514034748116
## [583235] 0.31768009151191 0.615372705059761
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```


Function 'estimateDivergence' returns a list of GRanges objects with the four columns of counts, the information divergence, and additional columns:

1. The original matrix of methylated (c_i) and unmethylated (t_i) read counts from control ($i = 1$) and treatment ($i = 2$) samples.
2. "p1" and "p2": methylation levels for control and treatment, respectively.
3. "bay.TV": total variation $TV = p2 - p1$.
4. "TV": total variation based on simple counts: $TV = c1/(c1 + t1) - c2/(c2 + t2)$.
5. "hdiv": Hellinger divergence.

If Bayesian = TRUE, results are based on the posterior estimations of methylation levels $p1$ and $p2$. Filtering by coverage is provided at this step which would be used unless previous filtering by coverage had been applied. This is a pairwise filtering. Cytosine sites with 'coverage' > 'min.coverage' and 'coverage' < 'percentile' (e.g., 99.9 coverage percentile) in at least one of the samples are preserved. The coverage percentile used is the maximum estimated from both samples: reference and individual.

For some GEO datasets only the methylation levels for each cytosine site are provided. In this case, Hellinger divergence can be estimated as given in reference (Sanchez and Mackenzie 2016):

$$H(\hat{p}_{ij}, \hat{p}_{ir}) = 2[(\sqrt{\hat{p}_{ij}} - \sqrt{\hat{p}_{ir}})^2 + (\sqrt{1 - \hat{p}_{ij}} - \sqrt{1 - \hat{p}_{ir}})^2]$$

Descriptive statistical analysis

A descriptive analysis on the distribution of Hellinger divergences of methylation levels is recommended. Like in any other statistical analysis, descriptive statistical analysis help us to detect potential issues in the raw data. It is the user responsibility to perform quality check of his/her dataset before start to apply Methyl-IT. Nevertheless, the quality checking of the raw data is not perfect. So, it is healthy to sees for potential issues.

Histogram and boxplots of divergences estimated in each sample

First, the data of interest (Hellinger divergences, "hdiv") are selected from the GRanges objects:

```
normal = HD$Breast_normal[, "hdiv"]
normal1 = HD$Breast_normal1[, "hdiv"]
cancer = HD$Breast_cancer[, "hdiv"]
metastasis = HD$Breast_metastasis[, "hdiv"]
```

Next, a single GRanges object is built from the above set of GRanges objects using the function `uniqueGRanges`. Notice that the number of cores to use for parallel computation can be specified.

```
hd = uniqueGRanges(list(normal, normal1, cancer, metastasis), missing = NA,
                    verbose = FALSE, num.cores = 6L)
```



```
colnames(mcols(hd)) <- c("normal", "normal1", "cancer", "metastasis")
hd

## GRanges object with 793181 ranges and 4 metadata columns:
##           seqnames      ranges strand |           normal           normal1
##           <Rle> <IRanges> <Rle> |           <numeric>           <numeric>
##      [1]   chr13   19020631      * |      0.288114717525655  0.287759722800159
##      [2]   chr13   19020633      * |      0.869257795833625  0.714264129461205
##      [3]   chr13   19020643      * |      0.555916920998297  0.560786257097847
##      [4]   chr13   19020680      * |      0.0237432667649364 0.0282048959784729
##      [5]   chr13   19020687      * | 7.57195235075071e-05 0.00202752610421691
##      ...     ...     ...     ... |      ...           ...
## [793177]   chr13 115108776      * |      0.575468339697292 0.571095407369092
## [793178]   chr13 115108789      * |      0.519827721839907 0.491452444591154
## [793179]   chr13 115108993      * |      2.52391385084963  2.25112539931814
## [793180]   chr13 115109023      * |      3.0321878626068  2.95933795531802
## [793181]   chr13 115109524      * |      0.946473945048512 0.92093811839244
##           cancer           metastasis
##           <numeric>           <numeric>
##      [1] 0.0227965968799609 2.07568760927939
##      [2] 0.0484806896715985 3.18068868077232
##      [3] 0.770572964508297 0.000723026944202173
##      [4]                <NA>                <NA>
##      [5]                <NA>                <NA>
##      ...     ...     ...
## [793177] 0.436946710684087                <NA>
## [793178] 0.0306804911627062                <NA>
## [793179] 2.89411608418132 3.28511983457169
## [793180] 0.373514034748116 0.671851570028068
## [793181] 0.615372705059761                <NA>
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Now, the Hellinger divergences estimated for each sample are in a single matrix on the metacolumn of the GRanges object and we can proceed to build the histogram and boxplot graphics for these data.

Finally, the data included in the graphic are:

```
# Define an auxiliar function
fun_length <- function(x){
  return(data.frame(y = median(x) + 1, label = paste0("n = ", length(x))))
}

data <- data.frame(normal = hd$normal, normal1 = hd$normal1,
                  cancer = hd$cancer, metastasis = hd$metastasis)
data = suppressMessages(melt(data))
colnames(data) <- c("Breast.tissue", "HD")
data = data[data$HD > 0, ]
DataFrame(data)

## DataFrame with 3172724 rows and 2 columns
##           Breast.tissue           HD
##           <factor>           <numeric>
## 1           normal 0.288114717525655
```

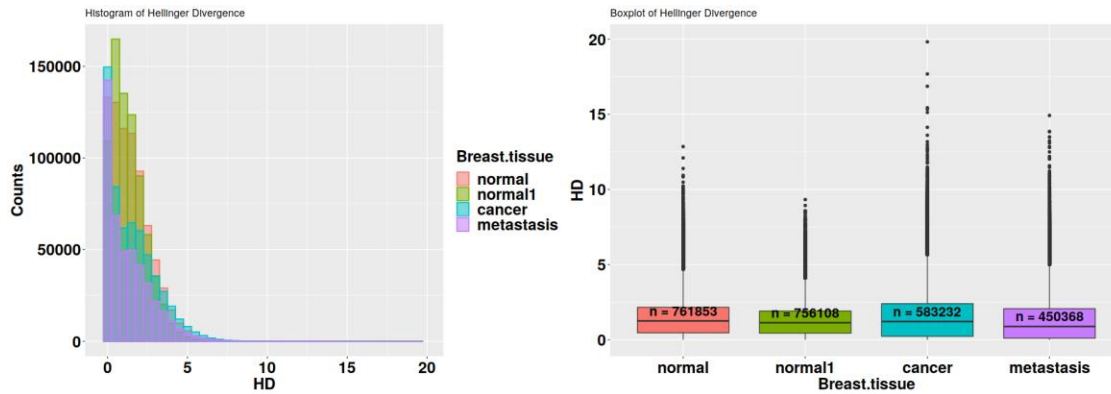
```
## 2          normal    0.869257795833625
## 3          normal    0.555916920998297
## 4          normal    0.0237432667649364
## 5          normal 7.57195235075071e-05
## ...          ...          ...
## NA.621156      NA          NA
## NA.621157      NA          NA
## 3172722      metastasis    3.28511983457169
## 3172723      metastasis    0.671851570028068
## NA.621158      NA          NA
```

```
p1 = ggplot(data, aes(x = HD, fill = Breast.tissue, colour = Breast.tissue)) +
  geom_histogram(alpha = 0.5, binwidth = 0.5, position = "identity", na.rm = TRUE,
    size = 0.7) + xlim(-0.4,20) +
  theme(axis.title.x = element_text(face = "bold", size = 20),
    axis.text.x = element_text(face = "bold", size = 20, color = "black",
      hjust = 0.5, vjust = 0.75),
    axis.text.y = element_text(face = "bold", size = 20, color = "black"),
    axis.title.y = element_text(face = "bold", size = 20,color = "black"),
    legend.text = element_text(size = 20, face = "bold"),
    legend.title = element_text(size = 20, face = "bold")
  ) +
  ylab("Counts" ) +
  ggtitle("Histogram of Hellinger Divergence")
```

For visualization purposes HD is limited to the interval 0 to 50

```
dt = data[ which(data$HD < 50), ]
p2 = ggplot(dt,aes(x = Breast.tissue, y = HD , fill = Breast.tissue)) +
  geom_boxplot(na.rm = TRUE) + ylim(-0.1, 20) +
  stat_summary(fun.data = fun_length, geom = "text",
    position = position_dodge(width = 0.9), vjust = 1,
    size = 6, fontface = "bold") +
  theme(axis.title.x = element_text(face = "bold", size = 20),
    axis.text.x = element_text(face = "bold", size = 20, color = "black",
      hjust = 0.5, vjust = 0.75),
    axis.text.y = element_text(face = "bold", size = 20, color = "black"),
    axis.title.y = element_text(face = "bold", size = 20,color = "black"),
    legend.position = "none"
  ) +
  ggtitle("Boxplot of Hellinger Divergence")
grid.arrange(p1, p2, ncol = 2)
```

```
## Warning: Removed 4 rows containing non-finite values (stat_summary).
```



Except for the tail, most of the methylation changes occurred under the area covered by the density curve corresponding to the normal breast tissue. This is theoretically expected. This area is explainable in statistical physical terms and, theoretically, it should fit a Weibull distribution. The tails regions cover the methylation changes that, with high probability, are not induced by thermal fluctuation and are not addressed to stabilize the DNA molecule. These changes are methylation signal. Professor David J. Miller (Department of Electrical Engineering, Penn State) proposed modeling the distribution as a mixed distributions to simultaneously describe the background methylation noise and the methylation signal (personal communication, January, 2018). This model approach seems to be supported by the above histogram, but it must be studied before being incorporated in a future version of Methyl-IT.

Checking cytosine read counts

According to the parameter values set for the function `estimateDivergence`, for each cytosine site, at least one individual must have a 10 reads or more, and the maximum coverage must be 500 or less.

```
# Maximum coverage at each cytosine site
covr <- lapply(HD, function(x) {
  cov1 <- x$c1 + x$t1
  cov2 <- x$c2 + x$t2
  cov <- apply(cbind(cov1, cov2), 1, max)
  return(cov)
})

# Quantiles
do.call(rbind, lapply(covr, function(x) {
  q60 <- quantile(x, 0.6)
  q9999 <- quantile(x, 0.9999)
  idx1 <- which(x >= q60)
  idx2 <- which(x <= 500)
  q95 <- quantile(x, 0.95)

  idx <- intersect(idx1, idx2)
  return(c(round(summary(x)), q60, quantile(x, c(0.95, 0.99, 0.999, 0.9999)),
    num.siteGreater_8 = sum(x >= 8), q60_to_500 = sum((x >= q60) & (x <=
500))),
```

```

        num.siteGreater_500 = sum(x > 500))})
)

##           Min. 1st Qu. Median Mean 3rd Qu. Max. 60% 95% 99% 99.9%
## Breast_normal      5      19      30      31      41      78      34      57      68      76
## Breast_cancer      5      17      31      37      50     128      37      87     109     125
## Breast_metastasis   5      14      27      31      43     105      32      69      88     102
## Breast_normal1      5      17      26      26      35      65      29      48      57      64
##           99.99% num.siteGreater_8 q60_to_500 num.siteGreater_500
## Breast_normal      78           745740      317962              0
## Breast_cancer     128           568171      241561              0
## Breast_metastasis  105           433990      186721              0
## Breast_normal1     65           737309      317308              0

```

The quantiles indicates acceptable values for the coverages. Descriptive statistics for methylated reads can be checked as well.

```

# Maximum coverage at each cytosine site
methc <- lapply(HD, function(x) {
  r <- apply(cbind(x$c1, x$c2), 1, max)
  return(r)
})

# Quantiles
do.call(rbind, lapply(methc, function(x) {
  q10 <- quantile(x, 0.1)
  q60 <- quantile(x, 0.6)
  q9999 <- quantile(x, 0.9999)
  idx1 <- which(x >= q60)
  idx2 <- which(x <= 500)
  q95 <- quantile(x, 0.95)

  idx <- intersect(idx1, idx2)
  return(c(round(summary(x)), q10, q60,
    quantile(x, c(0.95, 0.99, 0.999, 0.9999)),
    num.siteGreater_8 = sum(x >= 8),
    q60_to_500 = sum((x >= q60) & (x <= 500)),
    num.siteGreater_500 = sum(x > 500))}))
)

##           Min. 1st Qu. Median Mean 3rd Qu. Max. 10% 60% 95% 99% 99.9%
## Breast_normal      4      14      24      25      34      78      7      28      49      59      69
## Breast_cancer      4      10      21      27      38     127      5      27      73      96     115
## Breast_metastasis   4       7      16      21      30     103      4      21      55      74      91
## Breast_normal1      4      12      20      21      29      65      6      24      41      50      58
##           99.99% num.siteGreater_8 q60_to_500 num.siteGreater_500
## Breast_normal      74           676305      308116              0
## Breast_cancer     122           471387      235980              0
## Breast_metastasis   99           327545      183189              0
## Breast_normal1     62           663067      302809              0

```

The critical values from empirical cumulative probability distributions will be used in the downstream analysis.

```
critical.val <- do.call(rbind, lapply(HD, function(x) {
  hd.95 = quantile(x$hdiv, 0.95)
  tv.95 = quantile(abs(x$bay.TV), 0.95)
  return(c(tv = tv.95, hd = hd.95))
}))

critical.val

##              tv.95%   hd.95%
## Breast_normal      0.5044527 3.785860
## Breast_cancer      0.5186351 4.380013
## Breast_metastasis  0.5106455 4.027645
## Breast_normal1     0.4902885 3.276831
```

Nonlinear fit of probability distribution models

A basic requirement for the application of signal detection is the knowledge of the probability distribution of the background noise. Probability distribution, as a Weibull distribution model, can be deduced on a statistical mechanical/thermodynamics basis for DNA methylation induced by thermal fluctuations ([Sanchez and Mackenzie 2016](#)). Assuming that this background methylation variation is consistent with a Poisson process, it can be distinguished from variation associated with methylation regulatory machinery, which is non-independent for all genomic regions ([Sanchez and Mackenzie 2016](#)). An information-theoretic divergence to express the variation in methylation induced by background thermal fluctuations will follow a Weibull distribution model, provided that it is proportional to the minimum energy dissipated per bit of information associated with the methylation change. The nonlinear fit to a Weibull distribution model is performed by the function [nonlinearFitDist](#).

In general, an information divergence of methylation levels will follow a probability distribution from the family of generalized gamma distribution. Next, the best fitted model will be selected from one of the following distributions: two- and three-parameter Weibull distributions, gamma with two- and three-parameters (“Gamma2P” or “Gamma3P”), and generalized gamma with three-parameter (“GGamma3P”).

The best model is selected on the basis of Akaike’s information criterion and the correlation coefficient of cross-validations for the nonlinear regressions (R.Cross.val). These criteria evaluate different information inferred from the models. AIC deals with the trade-off between the goodness of fit (GOF) of the model and the complexity of the model, while R.Cross.val provides information on the prediction power/performance of the model when confronted with external dataset. Cross-validations for the nonlinear regressions (R.Cross.val) were performed as described in reference ([Stevens 2009](#)). In addition, Stein’s formula for adjusted R squared (ρ) was used as an estimator of the average cross-validation predictive power ([Stevens 2009](#)).

In general, the best fitted model yields the best DMP classification performance (see below). However, on special dataset, as the used in the current example. Function [gofReport](#) search for the best fitted model between the set of models requested by the user.

The best model including “GGamma3P” model:

```
d <- c("Weibull2P", "Weibull3P", "Gamma2P", "Gamma3P", "GGamma3P")
gof_1 <- gofReport(HD = HD, column = 9L, model = d, num.cores = 6L, output = "all",
  verbose = FALSE)

|=====| 100%
##
## *** Creating report ...

gof_1$bestModel

##      Breast_normal      Breast_cancer Breast_metastasis      Breast_normal1
##      "GGamma3P"         "GGamma3P"         "GGamma3P"         "GGamma3P"
```

The best model excluding “GGamma3P” model:

```
d <- c("Weibull2P", "Weibull3P", "Gamma2P", "Gamma3P")
gof_2 <- gofReport(HD = HD, column = 9L, model = d, num.cores = 6L, output = "all",
  verbose = FALSE)

|=====| 100%
##
## *** Creating report ...

gof_2$bestModel

##      Breast_normal      Breast_cancer Breast_metastasis      Breast_normal1
##      "Gamma3P"         "Gamma3P"         "Gamma3P"         "Gamma3P"
```

When “GGamma3P” model is included, the goodness-of-fit indicators used here suggest that it is best fitted model. However, a final decision must take into account the DMP classification performance as well (see below), which is accomplished after a signal detection step. The reader must keep in mind that we are dealing here with numerical algorithms, which are not perfect. Depending on dataset depending on dataset, the numerical computation will confront different challengers.

The difference between g3p_R.Cross.val and gg3p_R.Cross.val suggests that model predictions powers for “GGamma3P” and “Gamma3P” are close.

```
gof_1$stats

##      w2p_AIC w2p_R.Cross.val w3p_AIC w3p_R.Cross.val g2p_AIC
## Breast_normal -3078521      0.9950574      NA      NA -2994586
## Breast_cancer -1841229      0.9878700      NA      NA -1962005
## Breast_metastasis -1460630      0.9870808      NA      NA -1637906
## Breast_normal1 -3623417      0.9974568      NA      NA -3470904
##      g2p_R.Cross.val g3p_AIC g3p_R.Cross.val gg3p_AIC
## Breast_normal      0.9932819 -3856372      0.9978151 -4094992
## Breast_cancer      0.9882769 -2328337      0.9935563 -2589444
## Breast_metastasis  0.9906456 -1689487      0.9916644 -2143320
## Breast_normal1     0.9967108 -4117173      0.9985194 -5242885
##      gg3p_R.Cross.val bestModel
## Breast_normal      0.9983753      gg3p
## Breast_cancer      0.9958887      gg3p
```

## Breast_metastasis	0.9969515	gg3p
## Breast_normal1	0.9996704	gg3p

It is worthy to observe that although in the current example we successfully performed the nonlinear fit for GGamm3P model, this is not the general case when the number of cytosine sites goes close or above 10^6 , for which the computational cost is very high. The nonlinear fit of GGamm3P or GGamm4P is, in general, difficult since two different set of estimated parameters could produce very close distribution curves. Also, it is important to check the value of the scaling parameter from each distribution. The numerical algorithms could find the best fitted model with scaling parameter values close to zero, say e.g. 10^{-8} , and with the highest GOFs. Situations like this are biologically meaningless and the fitted models must be discarded ([Sanchez and Mackenzie 2016](#)).

Signal detection

The information thermodynamics-based approach is postulated to provide greater sensitivity for resolving true signal from the thermodynamic background within the methylome ([Sanchez and Mackenzie 2016](#)). Because the biological signal created within the dynamic methylome environment characteristic of plants is not free from background noise, the approach, designated Methyl-IT, includes the application of signal detection theory (Greiner, Pfeiffer, and Smith 2000; Carter et al. 2016; Harpaz et al. 2013; Kruspe et al. 2017). Signal detection is a critical step to increase sensitivity and resolution of methylation signal by reducing the signal-to-noise ratio and objectively controlling the false positive rate and prediction accuracy/risk.

Potential methylation signal

The first estimation in our signal detection step is the identification of the cytosine sites carrying potential methylation signal *PS*. The methylation regulatory signal does not hold the theoretical distribution and, consequently, for a given level of significance α (Type I error probability, e.g. $\alpha = 0.05$), cytosine positions k with information divergence $H_k \geq H_{\alpha=0.05}$ can be selected as sites carrying potential signals *PS*. The value of α can be specified. For example, potential signals with $H_k > H_{\alpha=0.01}$ can be selected. For each sample, cytosine sites are selected based on the corresponding fitted theoretical distribution model estimated in the previous step. Additionally, since cytosine with $|TV_{d_k}| < 0.1$ are the most abundant sites, depending on the sample (experiment), cytosine positions k with $H_k \geq H_{\alpha=0.05}$ and $|TV_{d_k}| < 0.1$ can be observed. To prevent the last situation we can select the *PS* with the additional constraint $|TV_{d_k}| > TV_0$, where TV_0 ('tv.cut') is a user specified value. The *PS* is detected with the function `getPotentialDIMP`:

```
PS1 = getPotentialDIMP(LR = HD, dist.name = gof_1$bestModel, nlms = gof_1$nlms,
                      div.col = 9, alpha = 0.05, tv.col = 8, tv.cut = 0.5)
PS2 = getPotentialDIMP(LR = HD, dist.name = gof_2$bestModel, nlms = gof_2$nlms,
                      div.col = 9, alpha = 0.05, tv.col = 8, tv.cut = 0.5)
PS1$Breast_normal
```



```
## GRanges object with 20263 ranges and 10 metadata columns:
##      seqnames      ranges strand |      c1      t1      c2      t2
##      <Rle> <IRanges> <Rle> | <numeric> <numeric> <numeric> <numeric>
## [1] chr13 19082356      * |      2      3      32      0
## [2] chr13 19177615      * |      4      4      23      0
## [3] chr13 19240007      * |      2      2      47      0
## [4] chr13 19240932      * |      3      3      32      0
## [5] chr13 19296278      * |      3      3      28      0
## ...      ...      ...      ... |      ...      ...      ...      ...
## [20259] chr13 115096223      * |      2      2      44      0
## [20260] chr13 115096666      * |      3      3      34      0
## [20261] chr13 115097021      * |      3      3      21      0
## [20262] chr13 115097976      * |      2      2      42      0
## [20263] chr13 115098695      * |      5      6      27      0
##      p1      p2      TV
##      <numeric> <numeric> <numeric>
## [1] 0.451291141301997 0.966907578309412 0.6
## [2] 0.453059717163291 0.956651735386458 0.5
## [3] 0.452540088615202 0.976266258433735 0.5
## [4] 0.452801333044291 0.966907578309412 0.5
## [5] 0.452801333044291 0.963018950689629 0.5
## ...      ...      ...
## [20259] 0.452540088615202 0.974843378117459 0.5
## [20260] 0.452801333044291 0.968560537905369 0.5
## [20261] 0.452801333044291 0.953445527269274 0.5
## [20262] 0.452540088615202 0.97379606280508 0.5
## [20263] 0.452084580416743 0.961899676528001 0.545454545454545
##      bay.TV      hdiv      wprob
##      <numeric> <numeric> <numeric>
## [1] 0.515616437007415 4.15647457049208 0.012312044885481
## [2] 0.503592018223167 4.91370739287271 0.000699892925358736
## [3] 0.523726169818533 4.00903023416115 0.0183352461007624
## [4] 0.51410624526512 4.7067660769561 0.00179872975159443
## [5] 0.510217617645338 4.45249390070021 0.00481084809952303
## ...      ...      ...
## [20259] 0.522303289502256 3.93208684512193 0.0222081071884813
## [20260] 0.515759204861077 4.82054071434893 0.00108912361728619
## [20261] 0.500644194224983 3.89433980895292 0.0243046790859968
## [20262] 0.521255974189877 3.87695537917288 0.0253151351459034
## [20263] 0.509815096111259 6.58816673565954 1.70622502417152e-10
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

PS2\$Breast_normal

```
## GRanges object with 11314 ranges and 10 metadata columns:
##      seqnames      ranges strand |      c1      t1      c2      t2
##      <Rle> <IRanges> <Rle> | <numeric> <numeric> <numeric> <numeric>
## [1] chr13 19177615      * |      4      4      23      0
## [2] chr13 19240932      * |      3      3      32      0
## [3] chr13 19296278      * |      3      3      28      0
## [4] chr13 19366559      * |      3      3      41      0
## [5] chr13 19377594      * |      3      5      40      1
## ...      ...      ...      ... |      ...      ...      ...      ...
## [11310] chr13 115090609      * |      4      4      60      1
```

```

## [11311] chr13 115090684 * | 4 4 76 2
## [11312] chr13 115094605 * | 3 3 38 0
## [11313] chr13 115096666 * | 3 3 34 0
## [11314] chr13 115098695 * | 5 6 27 0
## p1 p2 TV
## <numeric> <numeric> <numeric>
## [1] 0.453059717163291 0.956651735386458 0.5
## [2] 0.452801333044291 0.966907578309412 0.5
## [3] 0.452801333044291 0.963018950689629 0.5
## [4] 0.452801333044291 0.973239008761225 0.5
## [5] 0.450322519953442 0.951980598633333 0.600609756097561
## ...
## [11310] 0.453059717163291 0.966306151596363 0.483606557377049
## [11311] 0.453059717163291 0.961222815946982 0.474358974358974
## [11312] 0.452801333044291 0.971416061497986 0.5
## [11313] 0.452801333044291 0.968560537905369 0.5
## [11314] 0.452084580416743 0.961899676528001 0.545454545454545
## bay.TV hdiv wprob
## <numeric> <numeric> <numeric>
## [1] 0.503592018223167 4.91370739287271 0.029737586671658
## [2] 0.51410624526512 4.7067660769561 0.0351828105894626
## [3] 0.510217617645338 4.45249390070021 0.0432102524538469
## [4] 0.520437675716933 5.16357749160076 0.0242498537338447
## [5] 0.501658078679891 5.41899286255755 0.0196653983966234
## ...
## [11310] 0.513246434433072 6.36868111042499 0.00895184278510633
## [11311] 0.508163098783691 6.28426714972153 0.00960479847283617
## [11312] 0.518614728453694 5.02598611668935 0.0271362792477895
## [11313] 0.515759204861077 4.82054071434893 0.0320791517033604
## [11314] 0.509815096111259 6.58816673565954 0.00745163566165919
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths

```

Model “GGamma3P” reported 20263 - 11314 = 8949 more DMPs in the normal tissue than model “Gamma3P”.

Notice that the total variation distance $TV_d = |TV|$ is an information divergence as well and that it can be used in place of Hellinger divergence ([Sanchez and Mackenzie 2016](#)). The set of vectors $P_i = (p_i, 1 - p_i)$ and distance function TV_d integrate a metric space ([Sanchez et al. 2019](#)). In particular, after takes the Manhattan distance between healthy individuals and patients methylation levels we have:

$$M_d(p_{ij}, p_{ir}) = \frac{1}{2} (|\hat{p}_{ij} - \hat{p}_{ir}| + |(1 - \hat{p}_{ij}) - (1 - \hat{p}_{ir})|) = |\hat{p}_{ij} - \hat{p}_{ir}| = TV_d(p_{ij}, p_{ir})$$

.

That is, the absolute difference of methylation levels is a particular case of Manhattan distance, which is named total variation distance TV_d . The quantitative effect of the vector components $1 - \hat{p}_{ij}$ and $1 - \hat{p}_{ir}$ (in our case, the effect of unmethylated read counts) is not present in TV_d as in $H(\hat{p}_{ij}, \hat{p}_{ir})$.

Histogram and boxplots of methylation potential signals

As before, a single GRanges object is built from the above set GRanges objects using the function `uniqueGRanges`, and the Hellinger divergences of the cytosine sites carrying *PS* (for each sample) are located in a single matrix on the metacolumn of the GRanges object.

```
# Define an auxiliary function
fun_length <- function(x){
  return(data.frame(y = median(x) + 1.5, label = paste0("n = ", length(x))))
}

# The data to be used in the boxplot
ps1 = uniqueGRanges(PS1, missing = NA, columns = 9, verbose = FALSE, num.cores = 6L)
colnames(mcols(ps1)) <- c("normal", "cancer", "metastasis", "normal1")
dat1 = data.frame(normal = ps1$normal, normal1 = ps1$normal1, cancer = ps1$cancer,
  metastasis = ps1$metastasis)
dat1 = suppressMessages(melt(dat1))
colnames(dat1) <- c("Breast.tissue", "HD")
idx <- which(is.na(dat1$HD))
dat1 <- dat1[-idx, ] # To remove missing data

ps2 <- uniqueGRanges(PS2, missing = NA, columns = 9, verbose = FALSE, num.cores = 6L)
colnames(mcols(ps2)) <- c("normal", "cancer", "metastasis", "normal1")
dat2 = data.frame(normal = ps2$normal, normal1 = ps2$normal1, cancer = ps2$cancer,
  metastasis = ps2$metastasis)
dat2 = suppressMessages(melt(dat2))
colnames(dat2) <- c("Breast.tissue", "HD")
idx <- which(is.na(dat2$HD))
dat2 <- dat2[-idx, ] # To remove missing data

p1 = ggplot(dat1, aes(x = HD, fill = Breast.tissue, colour = Breast.tissue)) +
  geom_histogram(alpha = 0.5, bins = 50, position = "identity", na.rm = TRUE,
    size = 0.7) + ylab("Counts") + xlim(0,30) +
  theme(axis.title.x = element_text(face = "bold", size = 20),
    axis.text.x = element_text(face = "bold", size = 20, color = "black",
      hjust = 0.5, vjust = 0.75),
    axis.text.y = element_text(face = "bold", size = 20, color = "black"),
    axis.title.y = element_text(face = "bold", size = 20, color = "black"),
    legend.text = element_text(size = 20, face = "bold"),
    legend.title = element_text(size = 20, face = "bold")
  ) +
  ggtitle("Histogram for Potential methylation signal. GGamma3P")

p2 = ggplot(dat1, aes(x = Breast.tissue, y = HD, fill = Breast.tissue)) +
  geom_boxplot(na.rm = TRUE) + ylim(0, 15) +
  stat_summary(fun.data = fun_length, geom = "text", na.rm = TRUE,
    position = position_dodge(width = 0.9), vjust = 1,
    size = 6, fontface = "bold") +
  geom_hline(yintercept = 6, linetype="dashed", color = "red") +
  theme(axis.title.x = element_text(face = "bold", size = 20),
    axis.text.x = element_text(face = "bold", size = 20, color = "black",
      hjust = 0.5, vjust = 0.75),
    axis.text.y = element_text(face = "bold", size = 20, color = "black"),
    axis.title.y = element_text(face = "bold", size = 20, color = "black"),
```

```

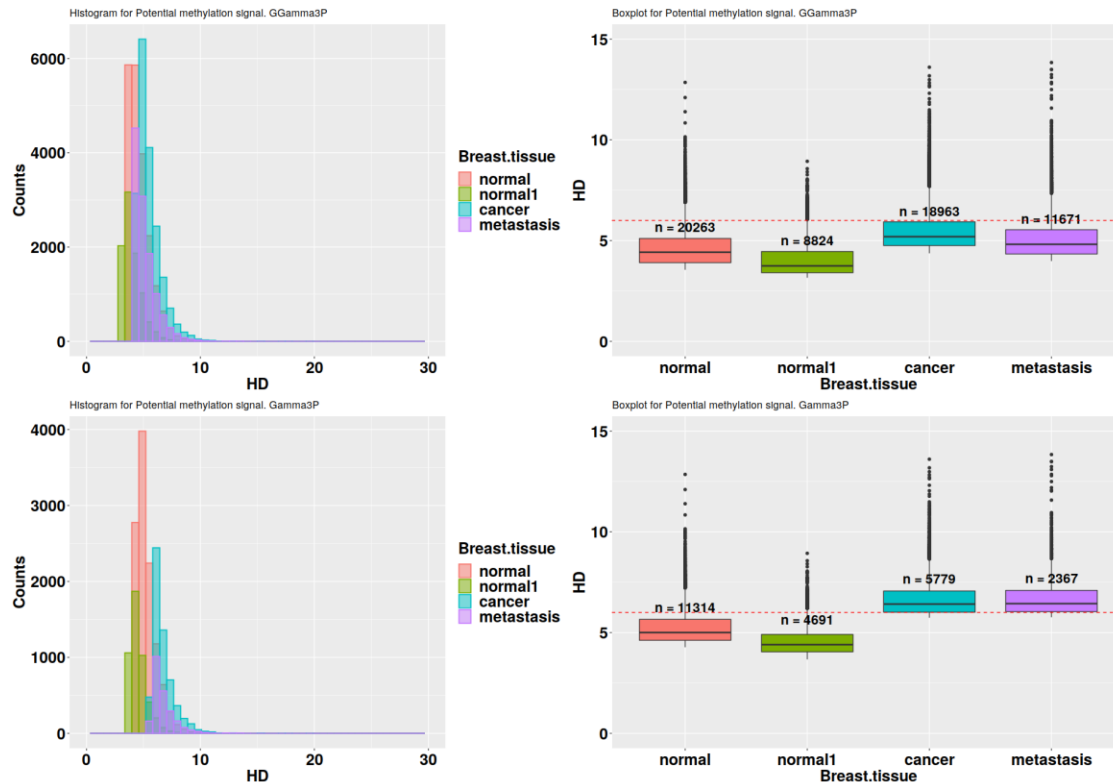
    legend.position = "none"
  ) +
  ggtitle("Boxplot for Potential methylation signal. GGamma3P")

p3 = ggplot(dat2, aes(x = HD, fill = Breast.tissue, colour = Breast.tissue)) +
  geom_histogram(alpha = 0.5, bins = 50, position = "identity", na.rm = TRUE,
    size = 0.7) + ylab("Counts") + xlim(0,30) +
  theme(axis.title.x = element_text(face = "bold", size = 20),
    axis.text.x = element_text(face = "bold", size = 20, color = "black",
      hjust = 0.5, vjust = 0.75),
    axis.text.y = element_text(face = "bold", size = 20, color = "black"),
    axis.title.y = element_text(face = "bold", size = 20, color = "black"),
    legend.text = element_text(size = 20, face = "bold"),
    legend.title = element_text(size = 20, face = "bold")
  ) +
  ggtitle("Histogram for Potential methylation signal. Gamma3P")

p4 = ggplot(dat2, aes(x = Breast.tissue, y = HD , fill = Breast.tissue)) +
  geom_boxplot(na.rm = TRUE) + ylim(0, 15) +
  stat_summary(fun.data = fun_length, geom = "text", na.rm = TRUE,
    position = position_dodge(width = 0.9), vjust = 1,
    size = 6, fontface = "bold") +
  geom_hline(yintercept = 6, linetype="dashed", color = "red") +
  theme(axis.title.x = element_text(face = "bold", size = 20),
    axis.text.x = element_text(face = "bold", size = 20, color = "black",
      hjust = 0.5, vjust = 0.75),
    axis.text.y = element_text(face = "bold", size = 20, color = "black"),
    axis.title.y = element_text(face = "bold", size = 20, color = "black"),
    legend.position = "none"
  ) +
  ggtitle("Boxplot for Potential methylation signal. Gamma3P")

grid.arrange(p1, p2, p3, p4, ncol = 2)

```



The graphics show that “Gamm3P” provides more information on the discrimination of treatment potential DMPs from the control.

Cutpoint estimation

Laws of statistical physics can account for background methylation, a response to thermal fluctuations that presumably functions in DNA stability (Sanchez and Mackenzie 2016; Sanchez et al. 2019). True signal is detected based on the optimal cutpoint (López-Ratón et al. 2014), which can be estimated from the area under the curve (AUC) of a receiver operating characteristic (ROC) curve built from a logistic regression performed with the potential signals from controls and treatments. The ROC AUC is equivalent to the probability that a randomly-chosen positive instance is ranked more highly than a randomly-chosen negative instance (Fawcett 2005). In the current context, the AUC is equivalent to the probability to distinguish a randomly-chosen methylation regulatory signal induced by the treatment from a randomly-chosen signal in the control.

The need for the application of (what is now known as) signal detection in cancer research was pointed out by Youden in the midst of the last century (Youden 1950). In the next example, the simple cutpoint estimation available in Methyl-IT is based on the application of Youden index (Youden 1950). Although cutpoints are estimated for a single variable, the classification performance can be evaluated for several variables and applying different model classifiers. A optimal cutpoint distinguishes disease stages from healthy individual. The performance of this classification is given in the output of function `estimateCutPoint`. A model classifier can be requested for further predictions and its classification performance is also provided. Below, the selected model classifier is a quadratic discriminant analysis

(QDA) (*classifier1* = "qda", *clas.perf* = TRUE). Four predictor variables are available: the Hellinger divergence of methylation levels (*hdiv*), total variation distance (*TV*, absolute difference of methylation levels), relative position of cytosine site in the chromosome (*pos*), and the logarithm base two of the probability to observe a Hellinger divergence value H greater than the critical value $H_{\alpha=0.05}$ (values given as probabilities in object PS, *wprob*).

Notice that the cutpoint can be estimated for any of the two currently available information divergences: *Hellinger divergence* (*div.col* = 9) or the *total variation distance* (with Bayesian correction, *div.col* = 8).

Cutpoint based on Youden index

Null distribution "GGamma3P":

```
cutpoint1 = estimateCutPoint(LR = PS1,
                             control.names = c("Breast_normal", "Breast_normal1"),
                             treatment.names = c("Breast_cancer",
"Breast_metastasis"),
                             simple = TRUE,
                             classifier1 = "pca.logistic",
                             column = c(hdiv = TRUE, bay.TV = TRUE,
                                         wprob = TRUE, pos = TRUE),
                             n.pc = 4, center = TRUE, scale = TRUE,
                             div.col = 9, clas.perf = TRUE)

cutpoint1$cutpoint
## [1] 4.383074

cutpoint1$testSetPerformance

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CT   TT
##           CT 1123 1084
##           TT 4007 9824
##
##           Accuracy : 0.6826
##           95% CI : (0.6753, 0.6898)
##           No Information Rate : 0.6801
##           P-Value [Acc > NIR] : 0.2575
##
##           Kappa : 0.1408
##
##           Mcnemar's Test P-Value : <2e-16
##
##           Sensitivity : 0.9006
##           Specificity : 0.2189
##           Pos Pred Value : 0.7103
##           Neg Pred Value : 0.5088
##           Prevalence : 0.6801
##           Detection Rate : 0.6125
##           Detection Prevalence : 0.8624
##           Balanced Accuracy : 0.5598
```

```
##
##      'Positive' Class : TT
##
```

```
cutpoint1$testSetModel.FDR
```

```
## [1] 0.2897115
```

Null distribution “Gamma3P”:

```
cutpoint2 = estimateCutPoint(LR = PS2,
                             control.names = c("Breast_normal", "Breast_normal1"),
                             treatment.names = c("Breast_cancer",
"Breast_metastasis"),
                             simple = TRUE,
                             classifier1 = "pca.logistic",
                             column = c(hdiv = TRUE, bay.TV = TRUE,
wprob = TRUE, pos = TRUE),
                             n.pc = 4, center = TRUE, scale = TRUE,
                             div.col = 9, clas.perf = TRUE)
```

```
cutpoint2$cutpoint
```

```
## [1] 5.736235
```

```
cutpoint2$testSetPerformance
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
## Prediction  CT   TT
##           CT 1157    0
##           TT    0 3262
```

```
##
```

```
##           Accuracy : 1
##           95% CI : (0.9992, 1)
## No Information Rate : 0.7382
## P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##           Kappa : 1
```

```
##
```

```
## McNemar's Test P-Value : NA
```

```
##
```

```
##           Sensitivity : 1.0000
##           Specificity : 1.0000
##           Pos Pred Value : 1.0000
##           Neg Pred Value : 1.0000
##           Prevalence : 0.7382
##           Detection Rate : 0.7382
##           Detection Prevalence : 0.7382
##           Balanced Accuracy : 1.0000
```

```
##
```

```
##      'Positive' Class : TT
```

```
##
```

```
cutpoint2$testSetModel.FDR
```



```
## [1] 0
```

The cutpoint estimated based “GGamma3P” model is lower than the value based on “Gamma3P” model. Often, this is a good detail, since it permits the identification of more sites with high probability to carry methylation signal. However, the evaluation of the classification performance has the last word on model selection. In the current case, the DMP classification performance is much better for “Gamma3P” model than for “GGamma3P”. In particular, the probabilities derived from each distribution model provide fundamental information to the model classifier. Hence, the probability that a potential DMP could carry a methylation signal estimated using “Gamma3P” model is more informative (for the model classifier) than the probability estimated using “GGamma3P” model.

Cutpoint estimated using a model classifier

In the current case, the DMP classification performance for the signal derived with null distribution “Gamma3P” can be improved searching for an optimal cutpoint with a model classifier and using the total variation distance (with Bayesian correction, bay.TV).

```
cutpoint2 = estimateCutPoint(LR = PS2,
                             control.names = c("Breast_normal", "Breast_normal1"),
                             treatment.names = c("Breast_cancer",
"Breast_metastasis"),
                             simple = FALSE,
                             classifier1 = "pca.logistic",
                             classifier2 = "pca.qda",
                             column = c(hdiv = TRUE, bay.TV = TRUE,
                                         wprob = TRUE, pos = TRUE),
                             n.pc = 4 , center = TRUE, scale = TRUE,
                             div.col = 8)

cutpoint2$cutpoint

## [1] 0.5000047

cutpoint2$testSetPerformance

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CT   TT
##           CT 6402    0
##           TT    0 3260
##
##           Accuracy : 1
##           95% CI : (0.9996, 1)
##           No Information Rate : 0.6626
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 1
##
##           McNemar's Test P-Value : NA
##
##           Sensitivity : 1.0000
```

```
##           Specificity : 1.0000
##           Pos Pred Value : 1.0000
##           Neg Pred Value : 1.0000
##           Prevalence : 0.3374
##           Detection Rate : 0.3374
##           Detection Prevalence : 0.3374
##           Balanced Accuracy : 1.0000
##
##           'Positive' Class : TT
##

cutpoint2$testSetModel.FDR

## [1] 0
```

The good DMP classification performance was kept at lower cutpoint value.

DMPs

Cytosine sites carrying a methylation signal are designated *differentially informative methylated positions* (DMPs). The probability that a DMP is not induced by the treatment is given by the probability of false alarm (P_{FA} , false positive, [Sanchez et al. 2019](#)). That is, the biological signal is naturally present in the control as well as in the treatment. According to the classical view in the epigenomic field, a DMP is a cytosine position carrying a significant methylation signal, which might be identified as differentially methylated cytosine site by Fisher's exact test (or other current tests). ***In Methyl-IT, a DMP is a DNA cytosine position with high probability to be differentially hyper-methylated or hypo-methylated in the treatment with respect to a given control (reference individual).*** Notice that Methyl-IT definition of DMP is not deterministic in an ordinary sense, but stochastic-deterministic in physico-mathematical terms. In other words, if a given cytosine position is identified as a hyper-methylated DMP in the treatment, then with high probability such a site will be found more frequently hyper-methylated than hypo-methylated in the ***treatment population***. Observe that it does not mean that, for all arbitrary subset of samples, the assumed cytosine position will be hypermethylated.

DMPs are selected with the function [selectDIMP](#)

```
DMPs = selectDIMP(PS2, div.col = 8, cutpoint = cutpoint2$cutpoint)
```

Error! Hyperlink reference not valid.The descriptive statistic of the read counts at DMPs

After the signal detection step the read counts at DMPs is notable greater than at the general methylated cytosine background:

```
# Maximum coverage at each cytosine site
methc <- lapply(DMPs, function(x) {
  r <- apply(cbind(x$c1, x$c2), 1, max)
  return(r)
})
```

```
# Quantiles
do.call(rbind, lapply(methc, function(x) {
  q10 <- quantile(x, 0.1)
  q60 <- quantile(x, 0.6)
  q9999 <- quantile(x, 0.9999)
  idx1 <- which(x >= q60)
  idx2 <- which(x <= 500)
  q95 <- quantile(x, 0.95)

  idx <- intersect(idx1, idx2)
  return(c(round(summary(x)), q10, q60,
    quantile(x, c(0.95, 0.99, 0.999, 0.9999)),
    num.siteGreater_8 = sum(x >= 8),
    q60_to_500 = sum((x >= q60) & (x <= 500)),
    num.siteGreater_500 = sum(x > 500))))))
})

##           Min. 1st Qu. Median Mean 3rd Qu. Max. 10% 60% 95% 99%
## Breast_normal      19      31      40      40      47      77      26      43.0      61      69
## Breast_cancer       14      38      51      56      73      126      29      58.0     101     116
## Breast_metastasis    15      35      45      48      58      103      27      49.2      80      91
## Breast_normal1      23      29      35      37      44      64      26      39.0      52      59
##           99.9% 99.99% num.siteGreater_8 q60_to_500
## Breast_normal    76.000  77.0000             11314      4648
## Breast_cancer    124.000 126.0000             5786      2388
## Breast_metastasis 101.638 102.7638             2363      945
## Breast_normal1    63.310  64.0000             4691      1886
##           num.siteGreater_500
## Breast_normal              0
## Breast_cancer              0
## Breast_metastasis          0
## Breast_normal1            0
```

The minimum amount of methylated read counts is observed “Breast_metastasis” (15 reads) and for any the samples the tenth percent of DMPs is 26 or more reads.

Venn Diagram of DMPs

The Venn diagram of DMPs reveals that the number cytosine site carrying methylation signal with a divergence level comparable to that observed in breast tissues with cancer and metastasis is relatively small (2797 DMPs). The number of DMPs decreased in the breast tissue with metastasis, but, as shown in the last boxplot, the intensity of the signal increased.

```
n12 = length(GenomicRanges::intersect(DMPs$Breast_normal,
                                       DMPs$Breast_cancer))
n13 = length(GenomicRanges::intersect(DMPs$Breast_normal,
                                       DMPs$Breast_metastasis))
n23 = length(GenomicRanges::intersect(DMPs$Breast_cancer,
                                       DMPs$Breast_metastasis))
n123 = length(Reduce(GenomicRanges::intersect,
                    list(DMPs$Breast_normal, DMPs$Breast_cancer,
                        DMPs$Breast_metastasis)))
```

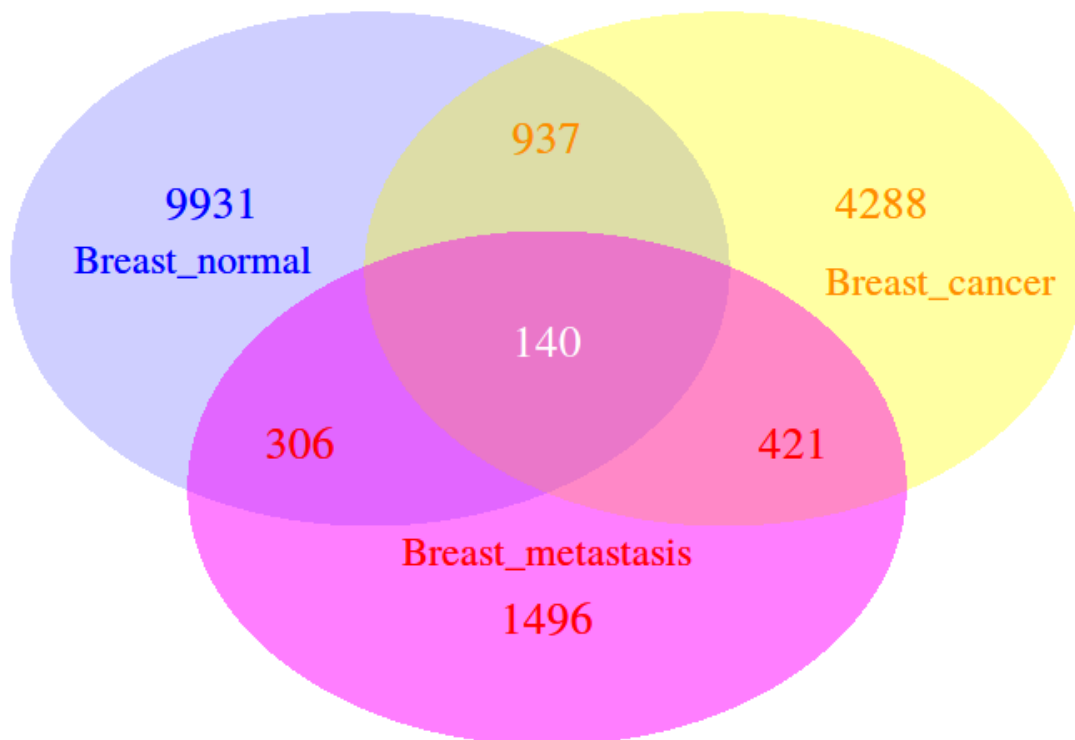
```

grid.newpage()
v = draw.triple.venn(area1 = length(DMPs$Breast_normal),
  area2 = length(DMPs$Breast_cancer),
  area3 = length(DMPs$Breast_metastasis),
  n12 = n12, n23 = n23, n13 = n13, n123 = n123,
  category = c("Breast_normal", "Breast_cancer",
    "Breast_metastasis"),
  lty = rep("blank", 3), fill = c("blue", "yellow",
    "magenta"),

  alpha = c(0.1, 0.2, 0.3),
  cat.pos = c(-80, 90, 0),
  cat.col = c("blue", "darkorange", "red"),
  cat.dist = c(-0.1, -0.08, -0.26),
  cex = rep(1.7, 7),
  cat.cex = c(1.5, 1.5, 1.5),
  label.col = c("blue", "darkorange", "darkorange",
    "red",
    "white", "red", "red"),

  scaled = TRUE)
grid.draw(v)

```



Notice that natural methylation regulatory signals (not induced by the treatment) are present in both groups, control and treatment. The signal detection step permits us to discriminate the “ordinary” signals observed in the control from those induced by the treatment (a disease in the current case). In addition, this diagram reflects a classification of DMPs only based on the cytosine positions. That is, this Venn diagram cannot tell us

whether DMPs at the same position can be distinguishable or not. For example, DMPs at the same positions in control and treatment can happen with different probabilities estimated from their corresponding fitted probability distributions (see below).

Monte Carlo Evaluation of the Model Classification Performance

The regulatory methylation signal is an output from a natural process that continuously takes place across the ontogenetic development of the organism. Therefore, we expect to see methylation signal in natural, ordinary conditions. Function 'evaluateDIMPclass' can be used to perform a classification of DMPs into two classes: DMPs from control and DMPs from treatment samples, as well as an evaluation of the classification performance (for more details see [evaluateDIMPclass](#)). As a matter of fact this function is called by function [estimateCutPoint](#). Additional feature not used before is the possibility to perform Monte Carlo evaluation of the model classifier performance. In above split of the sample into two subsets, training and test datasets, we would be just lucky getting the right proportion that yields the best classification performance. What about is the random split is repeat 300 times?

The performance of the "pca.qda" classifier is:

```
performance <- evaluateDIMPclass(LR = DMPs,
                                control.names = c("Breast_normal",
"Breast_normal1"),
                                treatment.names = c("Breast_cancer",
"Breast_metastasis"),
                                column = c(hdiv = TRUE, bay.TV = TRUE,
wprob = TRUE, pos = TRUE),
                                classifier = "pca.qda", prop = 0.6,
                                n.pc = 4, center = TRUE, scale = TRUE,
                                num.boot = 300, output = "mc.val",
                                num.cores = 6L, tasks = 2L)
```

```
performance
```

##	Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
##	Min. :1	Min. :1	Min. :0.9996	Min. :1	Min. :0.6626
##	1st Qu.:1	1st Qu.:1	1st Qu.:0.9996	1st Qu.:1	1st Qu.:0.6626
##	Median :1	Median :1	Median :0.9996	Median :1	Median :0.6626
##	Mean :1	Mean :1	Mean :0.9996	Mean :1	Mean :0.6626
##	3rd Qu.:1	3rd Qu.:1	3rd Qu.:0.9996	3rd Qu.:1	3rd Qu.:0.6626
##	Max. :1	Max. :1	Max. :0.9996	Max. :1	Max. :0.6626
##					
##	AccuracyPValue	McNemarPValue	Sensitivity	Specificity	Pos Pred Value
##	Min. :0	Min. : NA	Min. :1	Min. :1	Min. :1
##	1st Qu.:0	1st Qu.: NA	1st Qu.:1	1st Qu.:1	1st Qu.:1
##	Median :0	Median : NA	Median :1	Median :1	Median :1
##	Mean :0	Mean :NaN	Mean :1	Mean :1	Mean :1
##	3rd Qu.:0	3rd Qu.: NA	3rd Qu.:1	3rd Qu.:1	3rd Qu.:1
##	Max. :0	Max. : NA	Max. :1	Max. :1	Max. :1
##		NA's :300			
##	Neg Pred Value	Precision	Recall	F1	Prevalence
##	Min. :1	Min. :1	Min. :1	Min. :1	Min. :0.3374
##	1st Qu.:1	1st Qu.:1	1st Qu.:1	1st Qu.:1	1st Qu.:0.3374

```
## Median :1      Median :1      Median :1      Median :1      Median :0.3374
## Mean    :1      Mean    :1      Mean    :1      Mean    :1      Mean    :0.3374
## 3rd Qu.:1      3rd Qu.:1      3rd Qu.:1      3rd Qu.:1      3rd Qu.:0.3374
## Max.    :1      Max.    :1      Max.    :1      Max.    :1      Max.    :0.3374
##
## Detection Rate  Detection Prevalence  Balanced Accuracy
## Min.    :0.3374  Min.    :0.3374      Min.    :1
## 1st Qu.:0.3374  1st Qu.:0.3374      1st Qu.:1
## Median :0.3374  Median :0.3374      Median :1
## Mean    :0.3374  Mean    :0.3374      Mean    :1
## 3rd Qu.:0.3374  3rd Qu.:0.3374      3rd Qu.:1
## Max.    :0.3374  Max.    :0.3374      Max.    :1
##
```

Differentially methylated genomic regions (DMRs)

Our degree of confidence in whether DMP counts in both groups of samples, control and treatment, represent true biological signal was determined in the signal detection step. To estimate DIMRs, we followed similar steps to those proposed in Bioconductor R package DESeq2 (Love, Huber, and Anders 2014), but our GLM test looks for statistical difference between the groups based on gene-body DMP counts overlapping a given genomic region rather than read counts. The regression analysis of the generalized linear model (GLM) with logarithmic link was applied to test the difference between group counts. The fitting algorithmic approaches provided by 'glm' and 'glm.nb' functions from the R packages stat and MASS, respectively, were used for Poisson (PR), Quasi-Poisson (QPR) and Negative Binomial (NBR) linear regression analyses, respectively.

In Methyl-IT, the concept of DMR is generalized and it is not limited to any particular genomic region found with specific clustering algorithm. It can be applied to any naturally or algorithmically defined genomic region. For example, an exon region identified statistically to be differentially methylated by using generalized linear regression model (GLM) is a DMR. Differentially methylated genes (DMGs) are estimated from group comparisons for the number of DMPs on gene-body regions between control and treatment. This permits the extension of the concept of DMR by considering a differentially methylated gene as a particular case of DMRs.

Differentially methylated genes (DMGs)

We shall call DMGs those DIMRs restricted to gene-body regions. DMGs are detected using function `countTest2`.

Gene annotation is taken from [Ensembl](#)

```
# To Load human gene annotation
# AG = import(con = paste0("ftp://ftp.ensembl.org/pub/release-91/gff3/",
#                           "homo_sapiens/Homo_sapiens.GRCh38.91.gff3.gz"))
#
# genes = AG[ AG$type == "gene", c( "gene_id", "biotype", "Name" ) ]
# genes = genes[ genes$biotype == "protein_coding", "gene_id" ]
# seqlevels(genes, "coarse") <- "13" # To keep a consistent chromosome annotation
# seqlevels(genes) <- "chr13"
```

Here, for the sake of brevity, we just load the “RData” compressed file containing only the gene annotation:

```
con =
url("https://git.psu.edu/genomath/MethylIT_examples/raw/master/Homo_sapiens_GRCh38.91_genes.RData")
load(con)
genes = genes[ genes$biotype == "protein_coding", ]
seqlevels(genes, "coarse") <- "13" # To keep a consistent chromosome annotation
seqlevels(genes) <- "chr13"
genes

## GRanges object with 323 ranges and 3 metadata columns:
##      seqnames      ranges strand |      gene_id      biotype
##      <Rle>        <IRanges> <Rle> | <character> <character>
## [1] chr13 19173770-19181852 - | ENSG00000198033 protein_coding
## [2] chr13 19422877-19536762 - | ENSG00000132958 protein_coding
## [3] chr13 19633681-19673459 + | ENSG00000196199 protein_coding
## [4] chr13 19674752-19783019 - | ENSG00000121390 protein_coding
## [5] chr13 19823482-19863636 - | ENSG00000132950 protein_coding
## ...
## [319] chr13 113977783-114132611 - | ENSG00000185989 protein_coding
## [320] chr13 114179238-114223084 + | ENSG00000283361 protein_coding
## [321] chr13 114234887-114272723 + | ENSG00000130177 protein_coding
## [322] chr13 114281584-114305817 + | ENSG00000169062 protein_coding
## [323] chr13 114314513-114327328 + | ENSG00000198824 protein_coding
##      Name
##      <character>
## [1] TUBA3C
## [2] TPTE2
## [3] MPHOSPH8
## [4] PSPC1
## [5] ZMYM5
## ...
## [319] RASA3
## [320] AL160396.2
## [321] CDC16
## [322] UPF3A
## [323] CHAMP1
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Function [getDIMPatGenes](#) is used to count the number of DMPs at gene-body. Nevertheless, it can be used for any arbitrary specified genomic region as well. The operation of this function is based on the ‘*findOverlaps*’ function from the ‘*GenomicRanges*’ Bioconductor R package. The ‘*findOverlaps*’ function has several critical parameters like, for example, ‘*maxgap*’, ‘*minoverlap*’, and ‘*ignore.strand*’. In our function [getDIMPatGenes](#), except for setting `ignore.strand = TRUE` and `type = "within"`, we preserve the rest of default ‘*findOverlaps*’ parameters. In this case, these are important parameter settings because the local mechanical effect of methylation changes on a DNA region where a gene is located is independent of the strand where the gene is encoded. That is, methylation changes located in any of the two DNA strands inside the gene-body region will affect the flexibility of the DNA molecule (Choy et al. 2010; Severin et al. 2011).


```
DMPsBN = getDIMPatGenes(GR = DMPs$Breast_normal, GENES = genes)
DMPsBN1 = getDIMPatGenes(GR = DMPs$Breast_normal1, GENES = genes)
DMPsBC = getDIMPatGenes(GR = DMPs$Breast_cancer, GENES = genes)
DMPsBM = getDIMPatGenes(GR = DMPs$Breast_metastasis, GENES = genes)
```

The number of DMPs on the strand where a gene is encoded is obtained by setting `ignore.strand = FALSE`. However, for the current example results will be the same since the datasets downloaded from GEO do not have strand information. Next, the above GRanges objects carrying the DMP counts from each sample are grouped into a single GRanges object. Since we have only one control, to perform group comparison and to move forward with this example, we duplicated 'Breast_normal' sample. Obviously, the confidence on the results increases with the number of sample replications per group (in this case, it is only an illustrative example on how to perform the analysis, since a fair comparison requires for more than one replicate in the control group).

```
Genes.DMPs = uniqueGRanges( list(DMPsBN[, 2], DMPsBN1[, 2],
                                DMPsBC[, 2], DMPsBM[, 2]),
                             type = "equal", verbose = FALSE,
                             ignore.strand = TRUE )
colnames( mcols(Genes.DMPs)) <- c("Breast_normal", "Breast_normal1",
                                   "Breast_cancer", "Breast_metastasis")
```

Next, the set of mapped genes are annotated

```
GeneID = subsetByOverlaps(genes, Genes.DMPs, type = "equal",
                           ignore.strand = FALSE)
names( Genes.DMPs ) <- GeneID$gene_id
Genes.DMPs
```

```
## GRanges object with 279 ranges and 4 metadata columns:
##           seqnames           ranges strand | Breast_normal
##           <Rle>           <IRanges> <Rle> | <numeric>
##   ENSG00000198033   chr13  19173770-19181852   * |           1
##   ENSG00000132958   chr13  19422877-19536762   * |           2
##   ENSG00000196199   chr13  19633681-19673459   * |           2
##   ENSG00000121390   chr13  19674752-19783019   * |           6
##   ENSG00000132950   chr13  19823482-19863636   * |           1
##           ...           ...           ...   ... |           ...
##   ENSG00000283199   chr13  113953705-113973997   * |          11
##   ENSG00000185989   chr13  113977783-114132611   * |          75
##   ENSG00000283361   chr13  114179238-114223084   * |          31
##   ENSG00000130177   chr13  114234887-114272723   * |          26
##   ENSG00000169062   chr13  114281584-114305817   * |          20
##           Breast_normal1 Breast_cancer Breast_metastasis
##           <numeric>      <numeric>      <numeric>
##   ENSG00000198033           0           0           1
##   ENSG00000132958           2           0           0
##   ENSG00000196199           1           0           1
##   ENSG00000121390           2           0           3
##   ENSG00000132950           0           0           0
##           ...           ...           ...           ...
##   ENSG00000283199           5          10           5
##   ENSG00000185989          34         101          30
##   ENSG00000283361           7          42          26
```

```
## ENSG00000130177      8      32      10
## ENSG00000169062      7      19      6
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Now, we build a `glmDataSet` object.

```
## An experiment design is set.
colData <- data.frame(condition = factor(c("BN","BN","BC","BC"),
                                         levels = c("BN", "BC")),
                      c("Breast_normal","Breast_normal1",
                        "Breast_cancer","Breast_metastasis"),
                      row.names = 2)
## A RangedGlmDataSet is created
ds <- glmDataSet(GR = Genes.DMPs, colData = colData)
```

DMG analysis is performed with the function `countTest2`

```
DMGs = countTest2(DS = ds, num.cores = 4L, minCountPerIndv = 8,
                  countFilter = TRUE, maxGrpCV = c(1, 1),
                  Minlog2FC = 1, pvalCutOff = 0.05,
                  MVrate = .95, verbose = FALSE)
DMGs
```

```
## GRanges object with 37 ranges and 12 metadata columns:
##           seqnames      ranges strand | Breast_normal
##           <Rle>        <IRanges> <Rle> | <numeric>
## ENSG00000102683 chr13  23180952-23325165 * |      12
## ENSG00000027001 chr13  23730189-23889419 * |      13
## ENSG00000102699 chr13  24420926-24512810 * |      20
## ENSG00000139505 chr13  25246201-25288009 * |      10
## ENSG00000132964 chr13  26254104-26405238 * |      18
## ...      ...      ...      ... | ...
## ENSG00000134873 chr13  95433604-95579759 * |      21
## ENSG00000102572 chr13  98445185-98577940 * |      28
## ENSG00000204442 chr13 107163510-107866735 * |     106
## ENSG00000126217 chr13 112894378-113099739 * |      10
## ENSG00000185896 chr13 113297241-113323672 * |      15
##           Breast_normal1 Breast_cancer Breast_metastasis
##           <numeric>      <numeric>      <numeric>
## ENSG00000102683          5          0          0
## ENSG00000027001          4          0          0
## ENSG00000102699          6          6          2
## ENSG00000139505          6          0          1
## ENSG00000132964          4          0          0
## ...      ...      ...      ...
## ENSG00000134873          5          0          0
## ENSG00000102572          8          6          1
## ENSG00000204442         51         24          4
## ENSG00000126217          6          2          1
## ENSG00000185896          6          5          2
##           log2FC scaled.deviance      pvalue
##           <numeric>      <numeric>      <numeric>
## ENSG00000102683 -2.25129179860649 10.3380474313826 0.00482271130561193
## ENSG00000027001 -2.2512917986065  8.35180398450755 0.00927205013625302
```

```
## ENSG00000102699 -1.17865499634165 4.12049089254401 0.045424160854394
## ENSG00000139505 -1.79175946922806 6.68890141902843 0.00337278030755834
## ENSG00000132964 -2.484906649788 13.1724952257927 0.00170745868048471
## ...
## ENSG00000134873 -2.63905732961526 15.3578845622781 0.000810938754780871
## ENSG00000102572 -1.6376087894008 5.77349436895078 0.0166426011951339
## ENSG00000204442 -1.7240412951731 8.12159155176724 0.00309930302954073
## ENSG00000126217 -1.67397643357167 6.30622765547802 0.0156562836950803
## ENSG00000185896 -1.09861228866811 5.18205609520561 0.0290050118593663
## model adj.pval CT.SignalDensity
## <character> <numeric> <numeric>
## ENSG00000102683 Neg.Binomial.W 0.0114052809403684 5.89401861123053e-05
## ENSG00000027001 Neg.Binomial.W 0.0155764557196077 5.33815651474901e-05
## ENSG00000102699 Neg.Binomial 0.046735140959701 0.000141481199325244
## ENSG00000139505 Neg.Binomial.W 0.00959945164458911 0.0001913463608314
## ENSG00000132964 Neg.Binomial.W 0.00631759711779344 7.27826115724352e-05
## ...
## ENSG00000134873 Neg.Binomial.W 0.00428639056098461 8.89460576370453e-05
## ENSG00000102572 Neg.Binomial 0.0205258748073318 0.000135587092108831
## ENSG00000204442 Neg.Binomial 0.00959945164458911 0.000111628409643557
## ENSG00000126217 Neg.Binomial.W 0.0205258748073318 3.8955600354496e-05
## ENSG00000185896 Neg.Binomial 0.0315642776116634 0.000397245762711864
## TT.SignalDensity SignalDensityVariation
## <numeric> <numeric>
## ENSG00000102683 0 -5.89401861123053e-05
## ENSG00000027001 0 -5.33815651474901e-05
## ENSG00000102699 4.35326767154595e-05 -9.7948522609784e-05
## ENSG00000139505 1.19591475519625e-05 -0.000179387213279437
## ENSG00000132964 0 -7.27826115724352e-05
## ...
## ENSG00000134873 0 -8.89460576370453e-05
## ENSG00000102572 2.63641567989394e-05 -0.000109222935309892
## ENSG00000204442 1.99082514013987e-05 -9.17201582421583e-05
## ENSG00000126217 7.30417506646799e-06 -3.1651425288028e-05
## ENSG00000185896 0.000132415254237288 -0.000264830508474576
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Here, we add the gene alias:

```
hits = findOverlaps(DMGs, genes, type = "equal")
DMGs = DMGs[queryHits(hits)]
DMGs$alias <- genes$Name[subjectHits(hits)]
```

BRCA2, a breast cancer associated risk gene, is found between the DMGs

```
# DMGs
DMGs[ grep( "ENSG00000139618", names(DMGs) ) ]

## GRanges object with 1 range and 13 metadata columns:
##           seqnames      ranges strand | Breast_normal
##           <Rle>         <IRanges> <Rle> | <numeric>
## ENSG00000139618 chr13 32315474-32400266 * | 25
##           Breast_normal1 Breast_cancer Breast_metastasis
##           <numeric>      <numeric>      <numeric>
```

```
## ENSG00000139618      12      2      0
##      log2FC scaled.deviance      pvalue
##      <numeric>      <numeric>      <numeric>
## ENSG00000139618 -2.27726728500976 4.28814325502562 0.016083135361757
##      model      adj.pval      CT.SignalDensity
##      <character>      <numeric>      <numeric>
## ENSG00000139618 Neg.Binomial.W 0.0205258748073318 0.000218178387366882
##      TT.SignalDensity SignalDensityVariation      alias
##      <numeric>      <numeric> <character>
## ENSG00000139618 1.17934263441558e-05 -0.000206384961022726      BRCA2
## -----
## seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Concluding remarks

Results fundamentally depend on the use of the proper reference samples. If there is not enough control samples to create an independent reference virtual individual, then the centroid from the control group can be used instead, which can be built with function [poolFromGRlist](#).

The *reference* is used to put individuals from control and patients in the same *system of reference*. The information divergences of methylation levels with respect to the reference individual are estimated for each cytosine site for each individual, from control and treatment. As a result, the information on the natural spontaneous variation in the control and treatment populations is carried in the probability distributions of the information divergences, estimated for each individual. *It does not matter how statistically significant a DMP would be in the group of patients. What really matter is how big is the probability to observe the same methylation event in the control group in respect to the patient group.* Decisions are ultimately made on the basis of such classification probabilities at single cytosine positions. The signal detection based approach implemented in Methyl-IT is addressed to confront the mentioned issue and, in consequence, the level of resolution of Methyl-IT approach reach the single cytosine site.

Acknowledgments

We thank Professor David J Miller for valuable conversations and suggestions on our mathematical modeling. # Funding The work was supported by funding from NSF-SBIR (2015-33610-23428-UNL) and the Bill and Melinda Gates Foundation (OPP1088661).

S2. Session Information

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.3 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
```

```

##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      parallel  stats4     stats      graphics  grDevices  utils
## [8] datasets  methods    base
##
## other attached packages:
## [1] VennDiagram_1.6.20      futile.logger_1.4.3      gridExtra_2.3
## [4] reshape2_1.4.3          ggplot2_3.2.1            MethyKIT_0.3.2
## [7] rtracklayer_1.44.4      GenomicRanges_1.36.1     GenomeInfoDb_1.20.0
## [10] IRanges_2.18.3          S4Vectors_0.22.1         BiocGenerics_0.30.0
## [13] knitr_1.25
##
## loaded via a namespace (and not attached):
## [1] nlme_3.1-141             bitops_1.0-6
## [3] matrixStats_0.55.0       lubridate_1.7.4
## [5] bit64_0.9-7              tools_3.6.1
## [7] backports_1.1.4          R6_2.4.0
## [9] rpart_4.1-15             DBI_1.0.0
## [11] lazyeval_0.2.2           colorspace_1.4-1
## [13] nnet_7.3-12              withr_2.1.2
## [15] tidyselect_0.2.5         bit_1.1-14
## [17] compiler_3.6.1           Biobase_2.44.0
## [19] formatR_1.7              DelayedArray_0.10.0
## [21] labeling_0.3             scales_1.0.0
## [23] genefilter_1.66.0        stringr_1.4.0
## [25] digest_0.6.21            Rsamtools_2.0.1
## [27] rmarkdown_1.16           XVector_0.24.0
## [29] pkgconfig_2.0.3          htmltools_0.3.6
## [31] rlang_0.4.0              RSQLite_2.1.2
## [33] generics_0.0.2           BiocParallel_1.18.1
## [35] dplyr_0.8.3              ModelMetrics_1.2.2
## [37] RCurl_1.95-4.12          magrittr_1.5
## [39] nls2_0.2                 GenomeInfoDbData_1.2.1
## [41] Matrix_1.2-17            Rcpp_1.0.2
## [43] munsell_0.5.0            stringi_1.4.3
## [45] yaml_2.2.0               MASS_7.3-51.4
## [47] SummarizedExperiment_1.14.1 zlibbioc_1.30.0
## [49] plyr_1.8.4               recipes_0.1.7
## [51] blob_1.2.0               crayon_1.3.4
## [53] lattice_0.20-38          Biostrings_2.52.0
## [55] splines_3.6.1            annotate_1.56.1
## [57] zeallot_0.1.0           pillar_1.4.2
## [59] codetools_0.2-16        futile.options_1.0.1
## [61] XML_3.98-1.20           glue_1.3.1
## [63] evaluate_0.14            lambda.r_1.2.4
## [65] data.table_1.12.6        vctrs_0.2.0
## [67] foreach_1.4.7           gtable_0.3.0

```

```
## [69] purrr_0.3.2          assertthat_0.2.1
## [71] xfun_0.10            gower_0.2.1
## [73] prodlim_2018.04.18   xtable_1.8-4
## [75] e1071_1.7-2          ArgumentCheck_0.10.2
## [77] class_7.3-15         survival_2.44-1.1
## [79] timeDate_3043.102    minpack.lm_1.2-1
## [81] tibble_2.1.3         iterators_1.0.12
## [83] GenomicAlignments_1.20.1 AnnotationDbi_1.46.1
## [85] memoise_1.1.0        lava_1.6.6
## [87] caret_6.0-84         ipred_0.9-9
```

References.

Baldi, Pierre, and Soren Brunak. 2001. *Bioinformatics: the machine learning approach*. Second. Cambridge: MIT Press.

Basu, A., A. Mandal, and L. Pardo. 2010. "Hypothesis testing for two discrete populations based on the Hellinger distance." *Statistics & Probability Letters* 80 (3-4). Elsevier B.V.: 206–14. <https://doi.org/10.1016/j.spl.2009.10.008>.

Carter, Jane V., Jianmin Pan, Shesh N. Rai, and Susan Galandiuk. 2016. "ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves." *Surgery* 159 (6). Mosby: 1638–45. <https://doi.org/10.1016/j.surg.2015.12.029>.

Choy, John S., Sijie Wei, Ju Yeon Lee, Song Tan, Steven Chu, and Tae Hee Lee. 2010. "DNA methylation increases nucleosome compaction and rigidity." *Journal of the American Chemical Society* 132 (6). American Chemical Society: 1782–3. <https://doi.org/10.1021/ja910264z>.

Dolzhenko, Egor, and Andrew D Smith. 2014. "Using beta-binomial regression for high-precision differential methylation analysis in multifactor whole-genome bisulfite sequencing experiments." *BMC Bioinformatics* 15 (1). BioMed Central: 215. <https://doi.org/10.1186/1471-2105-15-215>.

Fawcett, Tom. 2005. "An introduction to ROC analysis." <https://doi.org/10.1016/j.patrec.2005.10.010>.

Greiner, M, D Pfeiffer, and R D Smith. 2000. "Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests." *Preventive Veterinary Medicine* 45 (1-2): 23–41. [https://doi.org/10.1016/S0167-5877\(00\)00115-X](https://doi.org/10.1016/S0167-5877(00)00115-X).

Harpaz, Rave, William DuMouchel, Paea LePendur, Anna Bauer-Mehren, Patrick Ryan, and Nigam H Shah. 2013. "Performance of Pharmacovigilance Signal Detection Algorithms for the FDA Adverse Event Reporting System." *Clin Pharmacol Ther* 93 (6): 1–19. <https://doi.org/10.1038/clpt.2013.24.Performance>.

Hebestreit, Katja, Martin Dugas, and Hans-Ulrich Klein. 2013. "Detection of significantly differentially methylated regions in targeted bisulfite sequencing data." *Bioinformatics (Oxford, England)* 29 (13): 1647–53. <https://doi.org/10.1093/bioinformatics/btt263>.

Kruspe, Sven, David D. Dickey, Kevin T. Urak, Giselle N. Blanco, Matthew J. Miller, Karen C. Clark, Elliot Burghardt, et al. 2017. "Rapid and Sensitive Detection of Breast Cancer Cells in Patient Blood with Nuclease-Activated Probe Technology." *Molecular Therapy - Nucleic Acids* 8 (September). Cell Press: 542–57. <https://doi.org/10.1016/j.omtn.2017.08.004>.

Love, M I, W Huber, and S Anders. 2014. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome Biology* 15 (12): 1–34. <https://doi.org/10.1186/S13059-014-0550-8>.

López-Ratón, Mónica, María Xosé Rodríguez-Álvarez, Carmen Cadarso-Suárez, Francisco Gude-Sampedro, and Others. 2014. "OptimalCutpoints: an R package for selecting optimal cutpoints in diagnostic tests." *Journal of Statistical Software* 61 (8). Foundation for Open Access Statistics: 1–36. <https://www.jstatsoft.org/article/view/v61i08>.

Ngo, Thuy T.M., Jehoong Yoo, Qing Dai, Qiucen Zhang, Chuan He, Aleksei Aksimentiev, and Taekjip Ha. 2016. "Effects of cytosine modifications on DNA flexibility and nucleosome mechanical stability." *Nature Communications* 7 (February). Nature Publishing Group: 10813. <https://doi.org/10.1038/ncomms10813>.

Robinson, Mark D., Abdullah Kahraman, Charity W. Law, Helen Lindsay, Malgorzata Nowicka, Lukas M. Weber, and Xiaobei Zhou. 2014. "Statistical methods for detecting differentially methylated loci and regions." *Frontiers in Genetics* 5 (SEP). Frontiers: 324. <https://doi.org/10.3389/fgene.2014.00324>.

Sanchez, Robersy, and Sally A. Mackenzie. 2016. "Information Thermodynamics of Cytosine DNA Methylation." Edited by Barbara Bardoni. *PLOS ONE* 11 (3). Public Library of Science: e0150427. <https://doi.org/10.1371/journal.pone.0150427>.

Sanchez, Robersy, Xiaodong Yang, Thomas Maher, and Sally Mackenzie. 2019. "Discrimination of DNA Methylation Signal from Background Variation for Clinical Diagnostics." *Int. J. Mol. Sci.* 20 (21): 5343. <https://doi.org/10.3390/ijms20215343>.

Severin, Philip M D, Xueqing Zou, Hermann E Gaub, and Klaus Schulten. 2011. "Cytosine methylation alters DNA mechanical properties." *Nucleic Acids Research* 39 (20): 8740–51. <https://doi.org/10.1093/nar/gkr578>.

Stevens, James P. 2009. *Applied Multivariate Statistics for the Social Sciences*. Fifth Edit. Routledge Academic.

Youden, W. J. 1950. "Index for rating diagnostic tests." *Cancer* 3 (1): 32–35. [https://doi.org/10.1002/1097-0142\(1950\)3:1<32::AID-CNCR2820030106>3.0.CO;2-3](https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3).