Overview of the *DMRcaller* package

Nicolae Radu Zabet*

April 16, 2015

Contents

1 Introduction

DNA methylation is an epigenetic modification of the DNA where a methyl group is added to the cytosine nucleotides. This modification is heritable, able to control the gene regulation and, in general, is associated with transcriptional gene silencing. While in mammals the DNA is predominantly methylated in CG context, in plants non-CG methylation (CHG and CHH, where H can be any of the A, C or T nucleotides) is also present and is important for the epigenetic regulation of transcription. Sequencing of bisulfite converted DNA has become the method of choice to determine genome wide methylation distribution. The *DMRcaller* package computes the set of Differentially Methylated Regions (DMRs) between two samples. *DMRcaller* will compute the differentially methylated regions from Whole Genome Bisulfite Sequencing (WGBS) or Reduced Representation Bisulfite Sequencing (RRBS) data. There are several tools able to call DMRs, but most work has been done in mammalian systems and, thus, they were designed to primarily call CG methylation.

2 Methods

The package computes the DMRs using the CX report files generated by Bismark (Krueger and Andrews, 2011), which contain the number of methylated and unmmethylated reads for each cytosine in the genome. The coverage at each position on the genome is not homogeneous and this makes it difficult to compute the differentially methylated cytosines. Here, we implemented three methods:

- **noise_filter** where we use a kernel (Hebestreit et al., 2013) to smooth the number of methylated reads and the total number of reads (the *DMRcaller* package provides four kernels: `"uniform"`, `"triangular"`, `"gaussian"` and `"epanechnicov"`)

*e-mail: `n.r.zabet@gen.cam.ac.uk`, The Sainsbury Laboratory, University of Cambridge, UK

- **neighbourhood** where individual cytosines in a specific context are considered in the analysis without any smoothing
- **bins** where the genome is split into equal bins where all the reads are pooled together

The DMRs are then computed by performing a statistical test between the number of methylated reads and the total number of reads in the two conditions for each position, cytosine or bin. In particular, we implemented two statistical tests: (*i*) Fisher's exact test and (*ii*) the Score test. The former (Fisher's exact test) uses the `fisher.test` in the *stats* package.

The Score test is a statistical test of a simple null hypothesis that a parameter of interest is equal to some particular value. In our case, we are interested if the methylation levels in the two samples are equal or different. Given that m_1 is the number of methylated reads in condition 1, m_2 is the number of methylated reads in condition 2, n_1 is the total number of reads in condition 1 and n_2 is the total number of reads in condition 2, the Z-score of the Score test is

$$Z = \frac{(p_1 - p_2)\,\nu}{\sqrt{p(1-p)}} \tag{1}$$

where $p_1 = m_1/n_1$, $p_2 = m_2/n_2$,

$$p = \frac{m_1 + m_2}{n_1 + n_2} \quad \text{and} \quad \nu = \sqrt{\frac{n_1 n_2}{n_1 + n_2}} \tag{2}$$

We then convert the Z-score to the p-value assuming a normal distribution and a two sided test.

```
pValue <- 2*pnorm(-abs(zScore))
```

Finally, for both statistical tests (Fisher's exact test and Score test), we adjust the p-values for multiple testing using Benjamini and Hochberg's method (Benjamini and Hochberg, 1995) to control the false discovery

```
pValue <- p.adjust(pValue, method="fdr")
```

The algorithm performs the statistical test for each position, cytosine or bin and then marks as DMRs all positions/cytosines/bins that satisfy the following three conditions:

- the difference in methylation levels between the two conditions is statistically significant according to the statistical test;
- the difference in methylation proportion between the two conditions is higher than a threshold value;
- the mean number of reads per cytosine is higher than a threshold.

To group adjacent DMRs, we run an iterative process, where neighbouring DMRs (within a certain distance of each other) are joined only if these three conditions are still met after joining the DMRs.

Finally, we filter the DMRs as follow

- Remove DMRs whose lengths are less than a minimum size.
- Remove DMRs with fewer cytosines than a threshold value.

For a set of potential DMRs (e.g. genes, transposable elements or CpG islands) the user can call the function `filterDMRs` where all reads in a set of provided regions are pooled together and then the algorithm performs the statistical test for each region.

3 Description

3.1 Data

Bismark (Krueger and Andrews, 2011) is a popular tool for methylation call on WGBS or RRBS data. *DMRcaller* takes as inputs the CX report files generated by Bismark and stores this data in a `GRanges` object. In the package, we included two CX report files that contain the methylation calls of WT and *met1-3 Arabidopsis thaliana* (Stroud et al., 2013). *MET1* gene encodes for the main DNA methyltransferase in *Arabidopsis thaliana* and the *met1-3* mutation results in a genome-wide loss of DNA methylation (mainly in CG context). Due to running time, we restricted the data and analysis to the first 1 *Mb* of the third chromosome of *A. thaliana*.

```
library(DMRcaller)

#load presaved data
data(methylationDataList)
```

To load a different dataset, one can use `readBismark` function, which takes as input the filename of the CX report file to be loaded.

```
# specify the Bismark CX report files
saveBismark(methylationDataList[["WT"]],
            "chr3test_a_thaliana_wt.CX_report")
saveBismark(methylationDataList[["met1-3"]],
            "chr3test_a_thaliana_met13.CX_report")

# load the data
methylationDataWT <- readBismark("chr3test_a_thaliana_wt.CX_report")
methylationDataMet13 <- readBismark("chr3test_a_thaliana_met13.CX_report")
methylationDataList <- GRangesList("WT" = methylationDataWT,
                                   "met1-3" = methylationDataMet13)
```

`methylationDataList` is a `GRangesList` object , where the `GRanges` elements contain four metadata columns

- **context** - the context of the Cytosine (CG, CHG or CHH)
- **readsM** - the number of methylated reads
- **readsN** - the total number of reads
- **trinucleotide_context** - the specific context of the cytosine (H is replaced by the actual nucleotide)

If the data consists of two or more replicates, these can be pooled together using the function `poolMethylationDatasets` or `poolTwoMethylationDatasets` (in the case of pooling only two datasets). The latter function (`poolTwoMethylationDatasets`) is useful when the datasets are large and creating a `GRangesList` object is not possible (e.g. the `GRanges` objects are two large).

```
# load the data
methylationDataAll <- poolMethylationDatasets(methylationDataList)

# In the case of 2 elements, this is equivalent to
methylationDataAll <- poolTwoMethylationDatasets(methylationDataList[[1]],
                                                 methylationDataList[[2]])
```

Alternatively, one can use `readBismarkPool` to directly read a list of CX report files and pool them together.

```
# load the data
methylationDataAll <- readBismarkPool(c(file_wt, file_met13))
```

3.2 Low resolution profiles

The *DMRcaller* package also offers the possibility to visualise context specific global changes in the methylation profile. To achieve this, the user can call `plotMethylationProfileFromData` function, which computes the mean methylation proportion in tiling bins of fixed size; see Figure 1.

Alternatively, for a finer control, the user can use `computeMethylationProfile` function to compute the methylation profile at certain locations on the genome. This function returns a `GRanges` object with four metadata columns

- **sumReadsM** - the number of methylated reads
- **sumReadsN** - the total number of reads

```
par(mar=c(4, 4, 3, 1)+0.1)
plotMethylationProfileFromData(methylationDataList[["WT"]],
                               methylationDataList[["met1-3"]],
                               conditionsNames = c("WT","met1-3"),
                               windowSize = 10000,
                               autoscale = FALSE,
                               context = c("CG"))

## Recompute regions...
## Computing low resolution profiles...
## Calculating methylation profile for  Chr3:101..999999  using a window of  10000  bp
## Calculating methylation profile for  Chr3:101..999999  using a window of  10000  bp
```

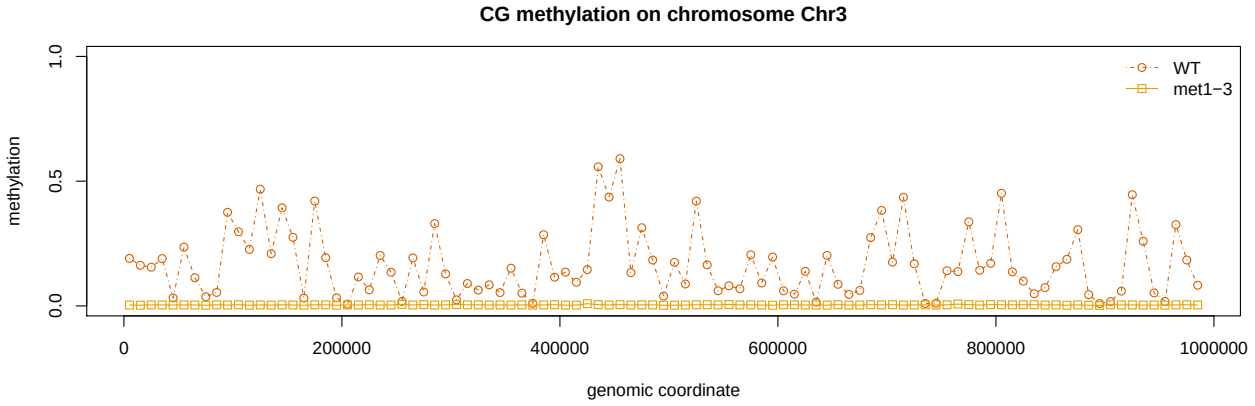

Figure 1: *Low resolution profile in CG context for WT and met1-3.*

- **Proportion** - the proportion of methylated reads
- **context** - the context

One or more of these `GRanges` objects can be put in a `GRangesList` object which is then passed as a parameter to the `plotMethylationProfile` function.

```
regions <- GRanges(seqnames = Rle("Chr3"), ranges = IRanges(1,1E6))

# compute low resolution profile in 10 Kb windows
profileCGWT <- computeMethylationProfile(methylationDataList[["WT"]],
                                         regions,
                                         windowSize = 10000,
                                         context = "CG")

profileCGMet13 <- computeMethylationProfile(methylationDataList[["met1-3"]],
                                            regions,
                                            windowSize = 10000,
                                            context = "CG")

profilesCG <- GRangesList("WT" = profileCGWT, "met1-3" = profileCGMet13)

#plot the low resolution profile
par(mar=c(4, 4, 3, 1)+0.1)
par(mfrow=c(1,1))
plotMethylationProfile(profilesCG,
                       autoscale = FALSE,
                       labels = NULL,
                       title="CG methylation on Chromosome 3",
                       col=c("#D55E00","#E69F00"),
                       pch = c(1,0),
                       lty = c(4,1))
```

3.3 Coverage of the bisulfite sequencing data

The number of reads from the bisulfite sequencing can differ significantly between different locations on the genome in the sense that cytosines in the same context (including neighbouring cytosines) can display large variability in the coverage. To plot the coverage of the bisulfite sequencing datasets, one can use `plotMethylationDataCoverage` function which takes as input one or two datasets and the vector with the thresholds used to compute the proportion of cytosines with at least that many reads; see Figure 2.

Alternatively, the *DMRcaller* also provides the `computeMethylationDataCoverage` function which returns a numeric vector with the number or proportion of cytosines in a specific context that have at least a certain number of reads specified by the input vector `breaks`.

```
# compute the coverage in the two contexts
coverageCGWT <- computeMethylationDataCoverage(methylationDataList[["WT"]],
                                               context="CG",
                                               breaks = c(1,5,10,15))
```

3.4 Calling DMRs

DMRcaller package provides `computeDMRs` function to call DMRs. The output of this function is a `GRanges` with 11 metadata columns.

- **direction** - a numeric value indicating whether the methylation was lost in the second condition compared to the first one (-1) or gained ($+1$)

```r
# plot the coverage in the two contexts
par(mar=c(4, 4, 3, 1)+0.1)
plotMethylationDataCoverage(methylationDataList[["WT"]],
                            methylationDataList[["met1-3"]],
                            breaks = c(1,5,10,15),
                            regions = NULL,
                            conditionsNames=c("WT","met1-3"),
                            context = c("CHH"),
                            proportion = TRUE,
                            labels=LETTERS,
                            contextPerRow = FALSE)
```

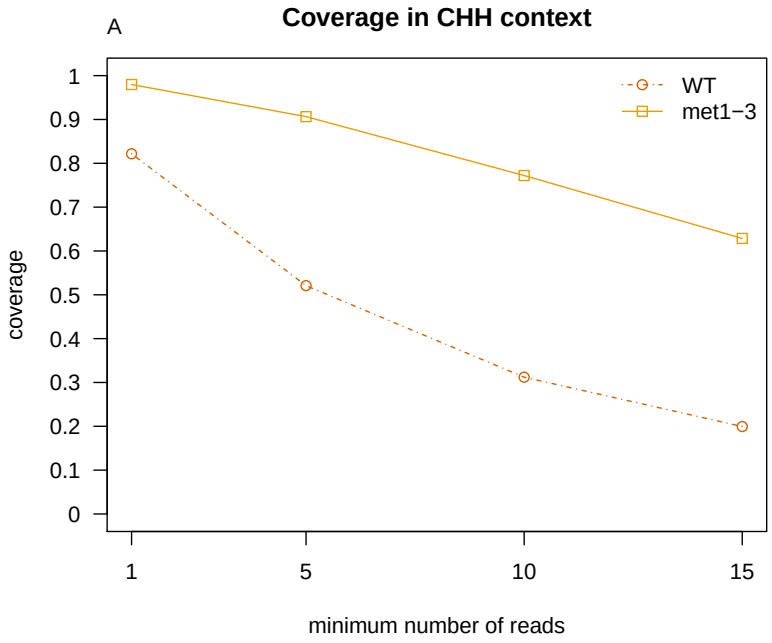

Figure 2: *Coverage*. For example, this figure shows that in WT only 30% of the cytosines in CHH context have at least 10 reads.

- **context** - the context of the cytosine (CG, CHG or CHH)
- **sumReadsM1** - the number of methylated reads in the DMR in condition 1
- **sumReadsN1** - the total number of reads in the DMR in condition 1
- **proportion1** - the proportion of methylated reads in the DMR in condition 1
- **sumReadsM2** - the number of methylated reads in the DMR in condition 2
- **sumReadsN2** - the total number of reads in the DMR in condition 2
- **proportion2** - the proportion of methylated reads in the DMR in condition 2
- **cytosinesCount** - the number of cytosines in the DMR
- **pValue** - the adjusted p-value of the statistical test
- **regionType** - a character string indicating whether the methylation was lost in the second condition compared to the first one (`"loss"`) or gained (`"gain"`)

For predifined regions (e.g. genes, transposons or CpG islands) the user can call `filterDMRs` function to extract the list of regions that are differentially methylated. The output of this function is again a GRanges with the same 11 metadata columns.

Below we present examples of calling both functions.

```
chr_local <- GRanges(seqnames = Rle("Chr3"), ranges = IRanges(5E5,6E5))

# compute the DMRs in CG context with noise_filter method
DMRsNoiseFilterCG <- computeDMRs(methylationDataList[["WT"]],
                      methylationDataList[["met1-3"]],
                      regions = chr_local,
                      context = "CG",
                      method = "noise_filter",
                      windowSize = 100,
                      kernelFunction = "triangular",
                      test = "score",
                      pValueThreshold = 0.01,
                      minCytosinesCount = 4,
                      minProportionDifference = 0.4,
                      minGap = 0,
                      minSize = 50,
                      minReadsPerCytosine = 4,
                      cores = 1)

## Parameters checking ...
## Extract methylation in the corresponding context
## Computing DMRs at  Chr3:500000..600000
## Calculating interpolations...
## Identifying DMRs...
## Analysed reads inside DMRs
## Merge DMRs iteratively
## Filter DMRs

print(DMRsNoiseFilterCG)

## GRanges object with 60 ranges and 11 metadata columns:
##        seqnames           ranges strand   | direction     context
##           <Rle>        <IRanges>  <Rle>   | <numeric> <character>
##    [1]     Chr3 [503043, 503148]      *   |        -1          CG
##    [2]     Chr3 [503390, 503542]      *   |        -1          CG
```

```
##    [3]     Chr3 [503612, 503901]      *   |        -1          CG
##    [4]     Chr3 [504042, 504093]      *   |        -1          CG
##    [5]     Chr3 [504255, 504348]      *   |        -1          CG
##    ...      ...               ...    ... ...       ...         ...
##   [56]     Chr3 [593906, 594076]      *   |        -1          CG
##   [57]     Chr3 [594128, 594214]      *   |        -1          CG
##   [58]     Chr3 [594285, 594385]      *   |        -1          CG
##   [59]     Chr3 [599027, 599107]      *   |        -1          CG
##   [60]     Chr3 [599509, 599634]      *   |        -1          CG
##        sumReadsM1 sumReadsN1 proportion1 sumReadsM2 sumReadsN2 proportion2
##         <numeric>  <numeric>   <numeric>  <numeric>  <numeric>   <numeric>
##    [1]        299        365   0.8191781          0        419 0.000000000
##    [2]        158        198   0.7979798          0        414 0.000000000
##    [3]        342        442   0.7737557          3        807 0.003717472
##    [4]         59         86   0.6860465          0        249 0.000000000
##    [5]        265        351   0.7549858          0        412 0.000000000
##    ...        ...        ...         ...        ...        ...         ...
##   [56]        216        253   0.8537549          1        648  0.00154321
##   [57]         27         45   0.6000000          2        107  0.01869159
##   [58]        128        149   0.8590604          0        258  0.00000000
##   [59]         57        111   0.5135135          3        154  0.01948052
##   [60]        168        219   0.7671233          0        201  0.00000000
##        cytosinesCount        pValue  regionType
##             <numeric>     <numeric> <character>
##    [1]             10 4.182506e-122        loss
##    [2]             12  1.866732e-98        loss
##    [3]             25 4.840050e-185        loss
##    [4]              6  7.283256e-47        loss
##    [5]             12 3.753532e-105        loss
##    ...            ...           ...         ...
##   [56]             16 2.230811e-158        loss
##   [57]              4  8.309914e-17        loss
##   [58]              4  5.301091e-72        loss
##   [59]              4  2.608676e-21        loss
##   [60]              8  1.198206e-57        loss
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

```
# compute the DMRs in CG context with neighbourhood method
DMRsNeighbourhoodCG <- computeDMRs(methylationDataList[["WT"]],
                                   methylationDataList[["met1-3"]],
                                   regions = chr_local,
                                   context = "CG",
                                   method = "neighbourhood",
                                   test = "score",
                                   pValueThreshold = 0.01,
                                   minCytosinesCount = 4,
                                   minProportionDifference = 0.4,
                                   minGap = 200,
                                   minSize = 1,
                                   minReadsPerCytosine = 4,
                                   cores = 1)

## Parameters checking ...
## Extract methylation in the corresponding context
## Computing DMRs
```

```
## Merge DMRs iteratively
## Filter DMRs

print(DMRsNeighbourhoodCG)

## GRanges object with 34 ranges and 16 metadata columns:
##        seqnames           ranges strand   |  context trinucleotide_context
##           <Rle>        <IRanges>  <Rle>   | <factor>              <factor>
##    [1]     Chr3 [503058, 503853]      *   |       CG                   CGG
##    [2]     Chr3 [504058, 504069]      *   |       CG                   CGG
##    [3]     Chr3 [504292, 504490]      *   |       CG                   CGA
##    [4]     Chr3 [506440, 506776]      *   |       CG                   CGT
##    [5]     Chr3 [507119, 507480]      *   |       CG                   CGA
##    ...      ...               ...    ... ...      ...                   ...
##   [30]     Chr3 [588591, 588633]      *   |       CG                   CGC
##   [31]     Chr3 [591681, 591790]      *   |       CG                   CGT
##   [32]     Chr3 [593736, 594337]      *   |       CG                   CGA
##   [33]     Chr3 [598934, 599219]      *   |       CG                   CGT
##   [34]     Chr3 [599556, 599586]      *   |       CG                   CGA
##          readsM1   readsN1   readsM2   readsN2        pValue sumReadsM1
##        <integer> <integer> <integer> <integer>     <numeric>  <numeric>
##    [1]        96       139         0       117  0.000000e+00        806
##    [2]        22        25         0        48  1.552219e-42         56
##    [3]        35        39         0        42 6.078309e-180        389
##    [4]        28        42         0        59 2.599691e-108        228
##    [5]         6         9         0        21 1.107359e-102        225
##    ...       ...       ...       ...       ...           ...        ...
##   [30]        11        12         0        38 1.396833e-145        353
##   [31]        25        31         2        59 1.274770e-143        268
##   [32]        65        75         1        56  0.000000e+00        659
##   [33]         6         6         0        15  2.419682e-39         83
##   [34]        46        55         0        63  4.166635e-57        166
##        sumReadsN1 proportion1 sumReadsM2 sumReadsN2 proportion2
##         <numeric>   <numeric>  <numeric>  <numeric>   <numeric>
##    [1]       1217   0.6622843          4       2205 0.001814059
##    [2]         78   0.7179487          0        210 0.000000000
##    [3]        584   0.6660959          0        911 0.000000000
##    [4]        370   0.6162162          3        629 0.004769475
##    [5]        415   0.5421687          1        687 0.001455604
##    ...        ...         ...        ...        ...         ...
##   [30]        426   0.8286385          4        509 0.007858546
##   [31]        296   0.9054054          4        486 0.008230453
##   [32]       1068   0.6170412          6       1817 0.003302146
##   [33]        178   0.4662921          3        331 0.009063444
##   [34]        217   0.7649770          0        200 0.000000000
##        cytosinesCount direction  regionType
##             <numeric> <numeric> <character>
##    [1]             65        -1        loss
##    [2]              5        -1        loss
##    [3]             24        -1        loss
##    [4]             20        -1        loss
##    [5]             22        -1        loss
##    ...            ...       ...         ...
##   [30]              8        -1        loss
##   [31]             14        -1        loss
##   [32]             41        -1        loss
##   [33]             13        -1        loss
```

```
##   [34]             7        -1        loss
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

```
# compute the DMRs in CG context with bins method
DMRsBinsCG <- computeDMRs(methylationDataList[["WT"]],
                          methylationDataList[["met1-3"]],
                          regions = chr_local,
                          context = "CG",
                          method = "bins",
                          binSize = 100,
                          test = "score",
                          pValueThreshold = 0.01,
                          minCytosinesCount = 4,
                          minProportionDifference = 0.4,
                          minGap = 200,
                          minSize = 50,
                          minReadsPerCytosine = 4,
                          cores = 1)

## Parameters checking ...
## Extract methylation in the corresponding context
## Computing DMRs at  Chr3:500000..600000
## Count inside each bin...
## Filter the bins...
## Identifying DMRs...
## Merge adjacent DMRs
## Merge DMRs iteratively
## Filter DMRs

print(DMRsBinsCG)

## GRanges object with 34 ranges and 11 metadata columns:
##        seqnames           ranges strand   | sumReadsM1 sumReadsN1
##           <Rle>        <IRanges>  <Rle>   |  <numeric>  <numeric>
##    [1]     Chr3 [503000, 503199]      *   |        299        731
##    [2]     Chr3 [503400, 504499]      *   |        959       1674
##    [3]     Chr3 [506400, 506699]      *   |        182        321
##    [4]     Chr3 [507300, 507499]      *   |        182        318
##    [5]     Chr3 [514800, 514899]      *   |        560        760
##    ...      ...              ...    ... ...        ...        ...
##   [30]     Chr3 [588500, 588699]      *   |        355        458
##   [31]     Chr3 [591600, 591799]      *   |        270        368
##   [32]     Chr3 [593700, 594399]      *   |        660       1151
##   [33]     Chr3 [599000, 599299]      *   |         77        184
##   [34]     Chr3 [599500, 599599]      *   |        168        219
##        proportion1 sumReadsM2 sumReadsN2  proportion2 cytosinesCount
##          <numeric>  <numeric>  <numeric>    <numeric>      <numeric>
##    [1]   0.4090287          1        776 0.0012886598             17
##    [2]   0.5728793          3       3183 0.0009425071             90
##    [3]   0.5669782          3        546 0.0054945055             18
##    [4]   0.5723270          1        464 0.0021551724             14
##    [5]   0.7368421          1        680 0.0014705882             10
##    ...         ...        ...        ...          ...            ...
##   [30]   0.7751092          5        605  0.008264463             10
##   [31]   0.7336957          4        652  0.006134969             18
```

```
##   [32]   0.5734144          7       1907  0.003670687             44
##   [33]   0.4184783          3        356  0.008426966             14
##   [34]   0.7671233          0        201  0.000000000              8
##           context direction        pValue  regionType
##       <character> <numeric>     <numeric> <character>
##    [1]          CG        -1  4.247517e-87        loss
##    [2]          CG        -1  0.000000e+00        loss
##    [3]          CG        -1  2.347520e-84        loss
##    [4]          CG        -1  3.122909e-76        loss
##    [5]          CG        -1 5.252114e-179        loss
##    ...         ...       ...           ...         ...
##   [30]          CG        -1 1.790777e-150        loss
##   [31]          CG        -1 1.698863e-139        loss
##   [32]          CG        -1 2.247435e-298        loss
##   [33]          CG        -1  5.498315e-37        loss
##   [34]          CG        -1  1.103348e-57        loss
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

```
# load the gene annotation data
data(GEs)

#select the genes
genes <- GEs[which(GEs$type == "gene")]

# compute the DMRs in CG context over genes
DMRsGenesCG <- filterDMRs(methylationDataList[["WT"]],
                          methylationDataList[["met1-3"]],
                          potentialDMRs = genes[overlapsAny(genes, chr_local)],
                          context = "CG",
                          test = "score",
                          pValueThreshold = 0.01,
                          minCytosinesCount = 4,
                          minProportionDifference = 0.4,
                          minReadsPerCytosine = 3,
                          cores = 1)

## Parameters checking ...
## Extract methylation in the corresponding context
## Computing DMRs at  Chr3:101..999999
## Selecting data...
## Identifying DMRs...

print(DMRsGenesCG)

## GRanges object with 3 ranges and 21 metadata columns:
##       seqnames           ranges strand |   source     type     score
##          <Rle>        <IRanges>  <Rle> | <factor> <factor> <numeric>
##   [1]     Chr3 [576378, 579559]      + |   TAIR10     gene      <NA>
##   [2]     Chr3 [528574, 532582]      - |   TAIR10     gene      <NA>
##   [3]     Chr3 [570134, 572345]      - |   TAIR10     gene      <NA>
##           phase          ID        Name                Note
##       <integer> <character> <character>     <CharacterList>
##   [1]      <NA>   AT3G02680   AT3G02680 protein_coding_gene
##   [2]      <NA>   AT3G02530   AT3G02530 protein_coding_gene
##   [3]      <NA>   AT3G02660   AT3G02660 protein_coding_gene
```

```
##               Parent       Index Derives_from           Alias sumReadsM1
##       <CharacterList> <character>  <character> <CharacterList>  <numeric>
##   [1]                        <NA>         <NA>                       1106
##   [2]                        <NA>         <NA>                       1150
##   [3]                        <NA>         <NA>                        874
##       sumReadsN1 proportion1 sumReadsM2 sumReadsN2 proportion2
##        <numeric>   <numeric>  <numeric>  <numeric>   <numeric>
##   [1]       2379   0.4649012         14       4546 0.003079630
##   [2]       2756   0.4172714         30       6207 0.004833253
##   [3]       1848   0.4729437         10       3487 0.002867795
##       cytosinesCount    pValue  regionType direction
##            <numeric> <numeric> <character> <numeric>
##   [1]            142         0        loss        -1
##   [2]            171         0        loss        -1
##   [3]            104         0        loss        -1
##   -------
##   seqinfo: 7 sequences from an unspecified genome; no seqlengths
```

3.5 Merge DMRs

Finally, for merging adjacent DMRs, *DMRcaller* provides the function `mergeDMRsIteratively` which can be used as follows:

```
DMRsNoiseFilterCGMerged <- mergeDMRsIteratively(DMRsNoiseFilterCG,
                                                minGap = 200,
                                                respectSigns = TRUE,
                                                methylationDataList[["WT"]],
                                                methylationDataList[["met1-3"]],
                                                context = "CG",
                                                minProportionDifference = 0.4,
                                                minReadsPerCytosine = 4,
                                                pValueThreshold = 0.01,
                                                test="score")

## Parameters checking ...
## Merge DMRs iteratively ...

print(DMRsNoiseFilterCGMerged)

## GRanges object with 37 ranges and 11 metadata columns:
##        seqnames           ranges strand   | direction     context
##           <Rle>        <IRanges>  <Rle>   | <numeric> <character>
##    [1]     Chr3 [503043, 503148]      *   |        -1          CG
##    [2]     Chr3 [503390, 504509]      *   |        -1          CG
##    [3]     Chr3 [506392, 506723]      *   |        -1          CG
##    [4]     Chr3 [507286, 507422]      *   |        -1          CG
##    [5]     Chr3 [514791, 514891]      *   |        -1          CG
##    ...      ...              ...    ... ...       ...         ...
##   [33]     Chr3 [588556, 588681]      *   |        -1          CG
##   [34]     Chr3 [591657, 591828]      *   |        -1          CG
##   [35]     Chr3 [593709, 594385]      *   |        -1          CG
##   [36]     Chr3 [599027, 599107]      *   |        -1          CG
##   [37]     Chr3 [599509, 599634]      *   |        -1          CG
##        sumReadsM1 sumReadsN1 proportion1 sumReadsM2 sumReadsN2
##         <numeric>  <numeric>   <numeric>  <numeric>  <numeric>
##    [1]        299        365   0.8191781          0        419
```

```
##    [2]        959       1674   0.5728793          3       3183
##    [3]        182        321   0.5669782          3        546
##    [4]        153        217   0.7050691          0        322
##    [5]        560        760   0.7368421          1        680
##    ...        ...        ...         ...        ...        ...
##   [33]        355        458   0.7751092          5        605
##   [34]        268        321   0.8348910          4        540
##   [35]        659       1068   0.6170412          6       1827
##   [36]         57        111   0.5135135          3        154
##   [37]        168        219   0.7671233          0        201
##         proportion2 cytosinesCount        pValue  regionType
##           <numeric>      <numeric>     <numeric> <character>
##    [1] 0.0000000000             10 4.182506e-122        loss
##    [2] 0.0009425071             90  0.000000e+00        loss
##    [3] 0.0054945055             18  1.449939e-84        loss
##    [4] 0.0000000000              8  1.209606e-70        loss
##    [5] 0.0014705882             10 2.039056e-178        loss
##    ...          ...            ...           ...         ...
##   [33]  0.008264463             10 4.022065e-150        loss
##   [34]  0.007407407             16 4.839238e-140        loss
##   [35]  0.003284072             42  0.000000e+00        loss
##   [36]  0.019480519              4  2.608676e-21        loss
##   [37]  0.000000000              8  1.198206e-57        loss
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Note that two neighbouring DMRs will be merged if all the conditions below are met

- they are within a distance from each other smaller than minGap
- the difference in methylation levels between the two conditions is statistically significant according to the statistical test when the two DMRs are joined
- the difference in methylation proportion between the two conditions is higher than a threshold value when the two DMRs are joined
- the number of reads per cytosine is higher than a threshold when the two DMRs are joined

3.6 Extract methylation data in regions

`analyseReadsInsideRegionsForCondition` function can extract additional information in a set of genomic regions (including DMRs) from any `methylationData` object. For example, to establish a link between the CG and CHH methylation, one might want to extract the number of methylated reads and the total number of reads in CHH context inside DMRs called in CG context.

```
#retrive the number of reads in CHH context in WT in CG DMRs
DMRsNoiseFilterCGreadsCHH <- analyseReadsInsideRegionsForCondition(
                             DMRsNoiseFilterCGMerged,
                             methylationDataList[["WT"]], context = "CHH",
                             label = "WT")

## Parameters checking ...
## Extract methylation levels in corresponding context ...
## Compute reads inside each region ...

print(DMRsNoiseFilterCGreadsCHH)
```

```
## GRanges object with 37 ranges and 15 metadata columns:
##        seqnames           ranges strand   | direction     context
##           <Rle>        <IRanges>  <Rle>   | <numeric> <character>
##    [1]     Chr3 [503043, 503148]      *   |        -1          CG
##    [2]     Chr3 [503390, 504509]      *   |        -1          CG
##    [3]     Chr3 [506392, 506723]      *   |        -1          CG
##    [4]     Chr3 [507286, 507422]      *   |        -1          CG
##    [5]     Chr3 [514791, 514891]      *   |        -1          CG
##    ...      ...               ...    ... ...       ...         ...
##   [33]     Chr3 [588556, 588681]      *   |        -1          CG
##   [34]     Chr3 [591657, 591828]      *   |        -1          CG
##   [35]     Chr3 [593709, 594385]      *   |        -1          CG
##   [36]     Chr3 [599027, 599107]      *   |        -1          CG
##   [37]     Chr3 [599509, 599634]      *   |        -1          CG
##        sumReadsM1 sumReadsN1 proportion1 sumReadsM2 sumReadsN2
##         <numeric>  <numeric>   <numeric>  <numeric>  <numeric>
##    [1]        299        365   0.8191781          0        419
##    [2]        959       1674   0.5728793          3       3183
##    [3]        182        321   0.5669782          3        546
##    [4]        153        217   0.7050691          0        322
##    [5]        560        760   0.7368421          1        680
##    ...        ...        ...         ...        ...        ...
##   [33]        355        458   0.7751092          5        605
##   [34]        268        321   0.8348910          4        540
##   [35]        659       1068   0.6170412          6       1827
##   [36]         57        111   0.5135135          3        154
##   [37]        168        219   0.7671233          0        201
##         proportion2 cytosinesCount        pValue  regionType
##           <numeric>      <numeric>     <numeric> <character>
##    [1] 0.0000000000             10 4.182506e-122        loss
##    [2] 0.0009425071             90  0.000000e+00        loss
##    [3] 0.0054945055             18  1.449939e-84        loss
##    [4] 0.0000000000              8  1.209606e-70        loss
##    [5] 0.0014705882             10 2.039056e-178        loss
##    ...          ...            ...           ...         ...
##   [33]  0.008264463             10 4.022065e-150        loss
##   [34]  0.007407407             16 4.839238e-140        loss
##   [35]  0.003284072             42  0.000000e+00        loss
##   [36]  0.019480519              4  2.608676e-21        loss
##   [37]  0.000000000              8  1.198206e-57        loss
##        sumReadsMWTCHH sumReadsNWTCHH proportionWTCHH cytosinesCountCHH
##             <numeric>      <numeric>       <numeric>         <numeric>
##    [1]              0            303     0.000000000                27
##    [2]             99           3323     0.029792356               309
##    [3]             10           1047     0.009551098                90
##    [4]              0            571     0.000000000                33
##    [5]              1            665     0.001503759                32
##    ...            ...            ...             ...               ...
##   [33]             12            672     0.017857143                32
##   [34]              6            792     0.007575758                52
##   [35]             29           2560     0.011328125               196
##   [36]              0            193     0.000000000                23
##   [37]              1            206     0.004854369                36
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

3.7 Plotting the distribution of DMRs

Sometimes, it is useful to obtain the distribution of the DMRs over the chromosomes. The *DMRcaller* provides the `computeOverlapProfile` function, which computes this distribution. The `GRanges` object generated by this function can then be added to a `GRangesList` object, which can be plotted using `plotOverlapProfile` function; see Figure 3. Additionally, the `plotOverlapProfile` function allows the user to specify two `GRangesList`, thus, allowing the plotting of distributions of hypo or hyper methylated DMRs separately.

```
# compute the distribution of DMRs
hotspots <- computeOverlapProfile(DMRsNoiseFilterCGMerged, chr_local,
                                  windowSize=5000, binary=TRUE)

## Calculating overlaps for  Chr3:500000..600000  using a window of  5000  bp

# plot the distribution of DMRs
plotOverlapProfile(GRangesList("Chr3"=hotspots))
```

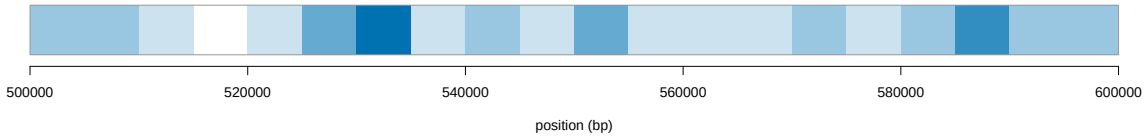

Figure 3: *Distribution of DMRs.* Darker colour indicates higher density, while lighter colour lower density.

3.8 Plotting profiles with DMRs

Finally, *DMRcaller* package also provides a function to plot methylation profiles at a specific location on the genome. To plot the methylation profile the user needs to call the `plotLocalMethylationProfile` function; see Figure 4.

4 Parallel computation

Computing the DMRs can be computationally intensive. For example, in the case of *A. thaliana* (with a genome of ≈ 130 *Mb*), it can take several hours to compute the DMRs depending on the method used and on the number of DMRs. To speed up computations, *DMRcaller* supports parallel computing of DMRs using the package *parallel*, but parallel computation is currently not supported on Windows.

The five functions used for computing and filtering the DMRs (`computeDMRs`, `filterDMRs`, `mergeDMRsIteratively` and `analyseReadsInsideRegionsForCondition`) accept the parameter `cores`, which specifies the number of cores that can be used when performing the corresponding computations. When using 10 cores, it can take between 10 and 30 minutes to compute the DMRs in *A. thaliana* depending on the selected parameters.

5 Session information

```
sessionInfo()

## R version 3.2.0 RC (2015-04-08 r68161)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows Server 2008 R2 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=C
## [2] LC_CTYPE=English_United States.1252
```

```
# select a 20 Kb location on the Chr3
chr3Reg <- GRanges(seqnames = Rle("Chr3"), ranges = IRanges(510000,530000))

# create a list with all DMRs
DMRsCGList <- list("noise filter" = DMRsNoiseFilterCGMerged,
                   "neighbourhood" = DMRsNeighbourhoodCG,
                   "bins" = DMRsBinsCG,
                   "genes" = DMRsGenesCG)
# plot the local profile
par(cex=0.9)
par(mar=c(4, 4, 3, 1)+0.1)
plotLocalMethylationProfile(methylationDataList[["WT"]],
                            methylationDataList[["met1-3"]],
                            chr3Reg,
                            DMRsCGList,
                            conditionsNames = c("WT", "met1-3"),
                            GEs,
                            windowSize = 300,
                            main="CG methylation")
```

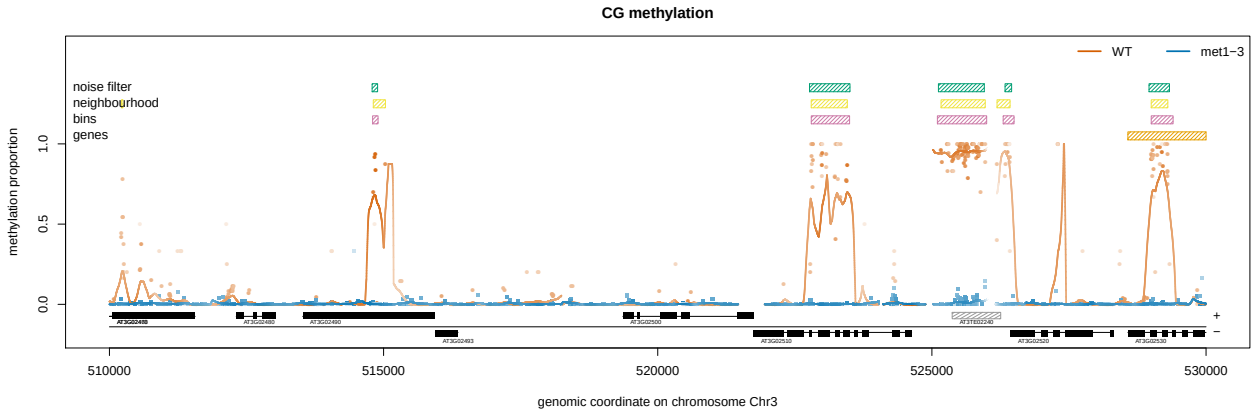

Figure 4: *Local methylation profile.* The points on the graph represent methylation proportion of individual cytosines, their colour (red or blue) which sample they belong to and the intensity of the the colour how many reads that particular cytosine had. This means that darker colours indicate stronger evidence that the corresponding cytosine has the corresponding methylation proportion, while lighter colours indicate a weaker evidence. The solid lines represent the smoothed profiles and the intensity of the colour the coverage at the corresponding position (darker colours indicate more reads while lighter ones less reads). The boxes on top represent the DMRs, where a filled box will represent a DMR which gained methylation while a box with a pattern represent a DMR that lost methylation. The DMRs need to have a metadata column `regionType` which can be either `"gain"` (where there is more methylation in condition 2 compared to condition 1) or `"loss"` (where there is less methylation in condition 2 compared to condition 1). In case this metadata column is missing all DMRs are drawn using filled boxes. Finally, we also allow annotation of the DNA sequence. We represent by black boxes all the exons, which are joined by a horizontal black line, thus, marking the full body of the gene. With grey boxes we mark the transposable elements. Both for genes and transposable elements we plot them over a mid line if they are on the positive strand and under the mid line if they are on the negative strand.

```
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
## [1] DMRcaller_1.0.0      GenomicRanges_1.20.0 GenomeInfoDb_1.4.0
## [4] IRanges_2.2.0        S4Vectors_0.6.0      BiocGenerics_0.14.0
##
## loaded via a namespace (and not attached):
## [1] formatR_1.1    XVector_0.8.0  tools_3.2.0    Rcpp_0.11.5
## [5] highr_0.4.1    knitr_1.9      RcppRoll_0.2.2 stringr_0.6.2
## [9] evaluate_0.6
```

References

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300.

Hebestreit, K., Dugas, M., and Klein, H.-U. (2013). Detection of significantly differentially methylated regions in targeted bisulfite sequencing data. *Bioinformatics*, 29(13):1647–1653.

Krueger, F. and Andrews, S. R. (2011). Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics*, 27(11):1571–1572.

Stroud, H., Greenberg, M. V., Feng, S., Bernatavichute, Y., and Jacobsen, S. E. (2013). Comprehensive analysis of silencing mutants reveals complex regulation of the *Arabidopsis* methylome. *Cell*, 152(1-2):352–364.