jDMR: a heuristic DMR caller for WGBS data

Rashmi Hazarika, Y.Shahryary & Frank Johannes

2022-02-17

Contents

1 Input files

For generation of region-level calls, jDMR requires the following inputs.

1.1 Methimpute files:

Base-level methylome outputs (generated using the R package "Methimpute")

1.2 A metadata file containing description about samples

For population data-sets without replicates, listfiles.fn should have the structure below.

> **file**: full PATH of file.
>
> **sample**: a sample name

```
samplefile1 <- system.file("extdata", "listFiles1.fn", package = "jDMR")
fread(samplefile1, header = TRUE)
```

```
                                   file     sample
  1: methimpute-out/methylome_A_All.txt methylomeA
  2: methimpute-out/methylome_B_All.txt methylomeB
  3: methimpute-out/methylome_C_All.txt methylomeC
  4: methimpute-out/methylome_D_All.txt methylomeD
  5: methimpute-out/methylome_E_All.txt methylomeE
  6: methimpute-out/methylome_F_All.txt methylomeF
```

For pairwise control-treatment data-sets with replicates, an additional column "replicate" should be provided. See structure below.

> **file**: full PATH of file
>
> **sample**: a sample name
>
> **replicate**: label for replicates

```
samplefile2 <- system.file("extdata", "listFiles2.fn", package = "jDMR")
fread(samplefile2, header = TRUE)
```

```
                                   file  sample replicate
  1: methimpute-out/methylome_A_All.txt    Col0      rep1
  2: methimpute-out/methylome_B_All.txt    Col0      rep2
  3: methimpute-out/methylome_C_All.txt mutant1      rep1
  4: methimpute-out/methylome_D_All.txt mutant1      rep2
  5: methimpute-out/methylome_E_All.txt mutant2      rep1
  6: methimpute-out/methylome_F_All.txt mutant2      rep2
```

2 Generate cytosine region calls from genome

jDMR detects DMRs using two approaches a) finding cytosine clusters in the genome (section 2.1) b) using a binning approach (section 2.2). You can use either of the methods to obtain the region calls. The remaining steps, makeDMRmatrix, filterDMRmatrix annotateDMRs are the same for both methods.

2.1 Run jDMR on cytosine clusters extracted from genome

fasta.file: PATH to FASTA file

samplefiles: PATH to file containing description about the samples

genome: a string containing name of the genome

out.dir: PATH to output directory

contexts: sequence contexts of the cytosine. By default this option is set to c("CG", "CHG", "CHH"). If you want to run for a single context such as CG, set it as "CG".

```
library(jDMR)
```

```
out.dir <- "/myfolder/DMR-results"
myfasta <- system.file("extdata", "TAIR10_chr_all.fa", package = "jDMR")
samplefile <- system.file("extdata", "listFiles1.fn", package = "jDMR")

runjDMRregions(fasta.file = myfasta, samplefiles = samplefile, genome = "Arabidopsis",
    out.dir = out.dir)
```

2.1.1 Output files of jDMR Regions approach

Rdata files containing coordinates of Cytosine clusters will be generated for each chromosome and cytosine context.

Output file "Arabidopsis_regions_chr1_CG.Rdata" contains coordinates of cytosine clusters

```
regionfile <- dget(system.file("extdata", "min.C_5/fp0.01/Arabidopsis_regions_chr1_CG.Rdata",
    package = "jDMR"))
```

```
head(regionfile$reg.obs)
```

```
  chr start   end cluster.length region
1   1  3696  3856            160   reg1
2   1 12100 12155             55   reg2
3   1 20991 21026             35   reg3
4   1 21257 21293             36   reg4
5   1 29966 30008             42   reg5
6   1 46099 46141             42   reg6
```

Region files containing state calls and methylation levels will be generated for each sample and each cytosine context.

```
   seqnames start   end context posteriorMax status rc.meth.lvl
1:        1  3696  3856      CG      0.99999      U     0.01732
2:        1 12100 12155      CG      0.99999      U     0.01732
3:        1 20991 21026      CG      0.99999      U     0.01732
4:        1 21257 21293      CG      0.99999      U     0.01732
5:        1 29966 30008      CG      0.99999      M     0.87383
6:        1 46099 46141      CG      0.99999      U     0.01732
```

seqnames, start and end: Chromosome coordinates

context: Sequence context of cytosine i.e CG,CHG,CHH

posteriorMax: Posterior value of the methylation state call

status : Methylation status

rc.meth.lvl: Recalibrated methylation level calculated from the posteriors and fitted parameters

2.2 Run jDMR on a binned genome

A non-sliding window approach will be used to bin the genome. The bin and step size will be determined automatically using the min.C parameter.

fasta.file: PATH to FASTA file

samplefiles: PATH to file containing description about the samples

genome: a string containing name of the genome

out.dir: PATH to output directory

contexts: sequence contexts of the cytosine. By default this option is set to c("CG", "CHG", "CHH"). If you want to run for a single context such as CG, set it as "CG".

min.C: minimum number of cytosines in a bin. bins lower than specified theshold will be dropped.

```
out.dir <- "/myfolder/DMR-results"
myfasta <- system.file("extdata", "TAIR10_chr_all.fa", package = "jDMR")
samplefile <- system.file("extdata", "listFiles1.fn", package = "jDMR")

runjDMRgrid(out.dir = out.dir, fasta.file = myfasta, samplefiles = samplefile, min.C = 10,
    genome = "Arabidopsis")
```

2.2.1 Output files of jDMR Grid approach

Region files containing state calls and methylation levels will be generated for each sample and for each context.

```
   seqnames start  end context posteriorMax status rc.meth.lvl
1:        1     1  500      CG      0.99999      M     0.74660
2:        1   501 1000      CG      0.99999      M     0.74660
3:        1  1001 1500      CG      0.99999      U     0.03057
4:        1  1501 2000      CG      0.99999      U     0.03057
5:        1  2001 2500      CG      0.99999      U     0.03057
6:        1  2501 3000      CG      0.99999      U     0.03057
```

seqnames, start and end: Chromosome coordinates

context: Sequence context of cytosine i.e CG,CHG,CHH

posteriorMax: Posterior value of the methylation state call

status : Methylation status

rc.meth.lvl: Recalibrated methylation level calculated from the posteriors and fitted parameters

3 Generate DMR matrix

3.1 Run "makeDMRmatrix"

"makeDMRmatrix" function generates matrix of state calls, rc.meth.lvls and posterior probabilities for all samples in one dataframe.

samplefiles: PATH to file containing description about the samples

input.dir: PATH to directory containing region files.

out.dir: PATH to output directory.

contexts: sequence contexts of the cytosine. By default this option is set to c("CG", "CHG", "CHH"). If you want to run for a single context such as CG, set it as "CG".

```
input.dir <- "/myfolder/DMR-results"
out.dir <- "/myfolder/DMRmatrix-results"
samplefile <- system.file("extdata", "listFiles1.fn", package = "jDMR")

makeDMRmatrix(samplefiles = samplefile, input.dir = myinput, out.dir = out.dir)
```

3.2 Output files of DMRmatrix function

"CG_StateCalls.txt" has the following structure. "0" in the output matrix denotes "Unmethylated" and "1" stands for "Methylated".

```
statecalls <- fread(paste0(out.dir, "CG_StateCalls.txt", sep = ""), header = TRUE)
head(statecalls)
```

```
   seqnames start  end WT_rep1 mutantA_rep1 mutantB_rep1
1:        1     1  500       1            1            1
2:        1   501 1000       1            1            1
3:        1  1001 1500       0            0            0
4:        1  1501 2000       0            0            0
5:        1  2001 2500       0            0            0
6:        1  2501 3000       0            0            0
```

"CG_rcMethlvl.txt" has the following structure. The output matrix contains recalibrated methylation levels for each sample and for the specific region.

```
rcmethlvls <- fread(paste0(out.dir, "CG_rcMethlvl.txt", sep = ""), header = TRUE)
head(rcmethlvls)
```

```
   seqnames start  end WT_rep1 mutantA_rep1 mutantB_rep1
1:        1     1  500 0.74660      0.67491      0.71563
2:        1   501 1000 0.74660      0.67491      0.71563
3:        1  1001 1500 0.03057      0.01907      0.02159
4:        1  1501 2000 0.03057      0.01907      0.02159
5:        1  2001 2500 0.03057      0.01907      0.02159
6:        1  2501 3000 0.03057      0.01907      0.02159
```

"CG_postMax.txt" has the following structure. The output matrix contains posterior probabilities for each sample and for the specific region.

```
postMax <- fread(paste0(out.dir, "CG_postMax.txt", sep = ""), header = TRUE)
head(postMax)
```

```
   seqnames start  end WT_rep1 mutantA_rep1 mutantB_rep1
1:        1     1  500 0.99999            1      0.99999
2:        1   501 1000 0.99999            1      0.99999
3:        1  1001 1500 0.99999            1      0.99999
4:        1  1501 2000 0.99999            1      0.99999
5:        1  2001 2500 0.99999            1      0.99999
6:        1  2501 3000 0.99999            1      0.99999
```

4 Filter DMR matrix

4.1 Filter the DMR matrix with the following options

This function filters the DMR matrix for non-polymorphic patterns.

gridDMR: set this option to TRUE if Grid approach was used otherwise set to FALSE. The output will contain merged regions.

data.dir: PATH to folder containing DMR matrix

epiMAF.cutoff: Applicable for calling calling population DMRs. This option can be used to filter for Minor Epi-Allele frequency as specified by user. By default, this option is set to NULL.

replicate.consensus : Applicable for control-treatment data-sets with replicates. Users can specify the percentage of concordance in methylation states in samples with multiple replicates. For datasets with just 2 replicates, *replicate.consensus* should be set as 1 (means 100% concordance). By default, this option is set to NULL.

rc.methlvl.out: Output filtered matrix containing recalibrtated methylation levels. By default, this option is set to FALSE.

context.specific.DMRs: Output context specific DMRs i.e CG-only, CHG-only, CHH-only, non-CG and multi-context DMRs. By default, this option is set to TRUE.

```
out.dir <- "/myfolder/DMRmatrix-results"
filterDMRmatrix(gridDMR=TRUE, #if region DMRs set to FALSE
                data.dir=out.dir)
```

4.2 Filtered Output

"CG_StateCalls-filtered.txt" has the following structure.

```
statecallsFiltered <- fread(paste0(out.dir, "CG_StateCalls-filtered.txt", sep = ""),
    header = TRUE)
head(statecallsFiltered)
```

```
     seqnames  start    end width WT_rep1 mutantA_rep1 mutantB_rep1
  1:        1  32001  32500   500       0            1            0
  2:        1  35501  36000   500       1            0            0
  3:        1  44501  45000   500       1            0            0
  4:        1  93001  93500   500       0            1            1
  5:        1  95001  95500   500       0            1            1
  6:        1 133001 133500   500       1            0            1
```

Additionally, the following files viz, CG-only-DMRs.txt, CHG-only-DMRs.txt, CHH-only-DMRs.txt, nonCG-DMRs.txt and multi-context-DMRs.txt will be generated.

5 Annotate DMRs

Multiple gff3 annotation files can be supplied as a vector with the *gff* option. Single/multiple files containing filtered DMR matrix should be provided with the *file.list* option.

gff.files: Multiple gff3 annotation files can be supplied as a vector

annotation: specify annotation categories

input.dir: path to folder containing only files to be annotated. Any file containing 3 columns (chr, start, stop) can be annotated using the annotateDMRs function.

gff3.out: whether to output annotated files in gff3 format

out.dir: path to output folder

```
# annotation files
gff.AT <- "/Annotations/Arabidopsis_thaliana.TAIR10.47.gff3"
gff.TE <- "/Annotations/TAIR10_TE.gff3"
gff.pr <- "/Annotations/TAIR10_promoters.gff3"
```

```r
mydir <- "/myfolder/annotate_DMRs/"

annotateDMRs(gff.files = c(gff.AT, gff.TE, gff.pr), annotation = c("gene", "promoters",
    "TE"), input.dir = mydir, gff3.out = TRUE, out.dir = mydir)
```

5.1 Output files after annotation

Mapped files are output in .txt and/or .gff3 format. Addiitonally, a DMR count table is generated.

```r
DMRcounts <- fread(paste0(out.dir, "annotate_DMRs/DMR-counts.txt", sep = ""), header = TRUE)
DMRcounts
```

```
              sample total.DMRs gene promoters    TE multiple.overlaps
1:       CG-only-DMRs       7238 4551       118   262              2122
2:      CHG-only-DMRs        729   42        47   303               208
3:      CHH-only-DMRs      41292 9809      3082 13082              9260
4: multi-context-DMRs       1235   59       145   305               608
5:          nonCG-DMRs      10389  193       481  2138              2583
```

6 R session info

```r
sessionInfo()
```

```
R version 4.0.1 (2020-06-06)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS  10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] parallel  stats4    stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] jDMR_0.1.0           R.utils_2.11.0       R.oo_1.24.0          R.methodsS3_1.8.1    GenomicRanges_1.42.
 [6] GenomeInfoDb_1.26.7  IRanges_2.24.1       S4Vectors_0.28.1     BiocGenerics_0.36.1  data.table_1.14.2

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.8                  lattice_0.20-45             tidyr_1.1.4                 Rsamtools_2.6.0
 [5] Biostrings_2.58.0           assertthat_0.2.1            digest_0.6.29               utf8_1.2.2
 [9] R6_2.5.1                    plyr_1.8.6                  evaluate_0.14               ggplot2_3.3.5
[13] pillar_1.6.5                zlibbioc_1.36.0             rlang_1.0.0                 rstudioapi_0.13
[17] Matrix_1.4-0                rmarkdown_2.11              BiocParallel_1.24.1         stringr_1.4.0
[21] RCurl_1.98-1.5              munsell_0.5.0               DelayedArray_0.16.3         compiler_4.0.1
[25] rtracklayer_1.50.0          xfun_0.29                   pkgconfig_2.0.3             htmltools_0.5.2
[29] tidyselect_1.1.1            SummarizedExperiment_1.20.0 tibble_3.1.6                GenomeInfoDbData_1.
[33] matrixStats_0.61.0          XML_3.99-0.8                fansi_1.0.2                 crayon_1.4.2
[37] dplyr_1.0.7                 GenomicAlignments_1.26.0    bitops_1.0-7                grid_4.0.1
[41] gtable_0.3.0                lifecycle_1.0.1             DBI_1.1.2                   magrittr_2.0.2
[45] formatR_1.11                scales_1.1.1                cli_3.1.1                   stringi_1.7.6
[49] XVector_0.30.0              reshape2_1.4.4              ellipsis_0.3.2              generics_0.1.1
```

```
[53] vctrs_0.3.8                methimpute_1.12.0        tools_4.0.1              Biobase_2.50.0
[57] glue_1.6.1                 purrr_0.3.4              MatrixGenerics_1.2.1     fastmap_1.1.0
[61] yaml_2.2.2                 colorspace_2.0-2         minpack.lm_1.2-1         knitr_1.37
```