jDMRgrid: a heuristic DMR caller for WGBS data using grid approach

Robert S. Piecyk, Rashmi R. Hazarika, Yadi Shahryary, Frank Johannes

2023-10-24

Contents

1 Input files

For generation of region-level calls, jDMRgrid 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

```
load(system.file("data", "listFiles1.RData", package = "jDMRgrid"))
listFiles1$file <- system.file("extdata", listFiles1$file, package = "jDMRgrid")
listFiles1
```

```
                                                                                          file     sample repl
  1: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p1.txt methylomeA
  2: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p2.txt methylomeB
  3: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p3.txt methylomeC
  4: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p4.txt methylomeD
```

For pairwise control-treatment data-sets with replicates,additional columns “replicate” and “group” should be provided. See structure below.

> **file**: full PATH of file **sample**: a sample name **replicate**: label for replicates **group**: label for control and treatment groups

```
load(system.file("data", "listFiles2.RData", package = "jDMRgrid"))
listFiles2$file <- system.file("extdata", listFiles2$file, package = "jDMRgrid")
listFiles2
```

```
                                                                                          file  sample replica
  1: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p1.txt      WT      rep
  2: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p2.txt      WT      rep
  3: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p3.txt mutant1      rep
  4: /home/robert/R/x86_64-pc-linux-gnu-library/4.3/jDMRgrid/extdata/toyData/methimpute_p4.txt mutant1      rep
          group
  1:    control
  2:    control
  3: treatment1
  4: treatment1
```

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 jDMRgrid on a binned genome

This function uses a grid approach to bin the genome into equal sized bins. User specifies the window and step size as numeric values.

out.dir: PATH to output directory.

window: NUMERIC VALUE specifying bin size.

step: NUMERIC VALUE specifying step size. If bin and step size are equal, we are utilizing non-sliding window approach.

samplelist: DATAFRAME OBJECT containing information about file, sample and replicate. For control/treatment data an additional column specifying the replicates is required.

contexts: VECTOR or CHARACTER presenting 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: NUMERIC VALUE specifying percentile threshold based on empirical distribution of the cytosines across bins.

mincov: NUMERIC VALUE specifying minimum read coverage over cytosines. By default this option is set as 0.

include.intermediate: LOGICAL specifying whether or not the intermediate component should be included in the HMM model. By default this option is set as FALSE.

runName: CHARACTER as the name of the operation. By default this option is set to 'GridGenome'.

parallelApply: LOGICAL specifying if future.apply package should be used to use parallel operation. By default this option is set to FALSE.

numCores: NUMERIC VALUE specifying number of cores to perform parallel operation using foreach loop. By default this option is set to NULL.

```
library(jDMRgrid)
runjDMRgrid(out.dir = "~/folder_population/grid", window = 200, step = 50, samplelist = listFiles1,
    contexts = c("CG", "CHG", "CHH"), min.C = 10, mincov = 0, include.intermediate = TRUE,
    runName = "Arabidopsis")
runjDMRgrid(out.dir = "~/folder_replicate/grid", window = 200, step = 50, samplelist = listFiles2,
    contexts = c("CG", "CHG", "CHH"), min.C = 10, mincov = 0, include.intermediate = TRUE,
    runName = "Arabidopsis")
```

2.1.1 Output files of jDMR Grid approach

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

```
jDMR.out <- fread("~/folder_replicate/grid/methimpute_p1_CG.txt")
```

```
   seqnames start end context posteriorMax status rc.meth.lvl
1:        1     1 200      CG      0.97962      U     0.04022
2:        1    51 250      CG      0.79295      U     0.12897
3:        1   101 300      CG      0.79295      U     0.12897
4:        1   151 350      CG      0.64649      M     0.80224
5:        1   201 400      CG      0.64649      M     0.80224
6:        1   251 450      CG      0.64649      M     0.80224
```

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"

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

samplelist: DATAFRAME OBJECT containing information about file, sample and replicate. For control/treatment data an additional column specifying the replicates is required.

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".

postMax.out: By default this option is set as FALSE. You can set it to TRUE if you want to output the DMR matrix containing posterior probabilities for the status call of each region.

```
makeDMRmatrix(contexts = c("CG", "CHG", "CHH"), postMax.out = TRUE, samplelist = listFiles1,
    input.dir = "~/folder_population/grid", out.dir = "~/folder_population/matrix",
    include.intermediate = FALSE)

makeDMRmatrix(contexts = c("CG", "CHG", "CHH"), postMax.out = TRUE, samplelist = listFiles2,
    input.dir = "~/folder_replicate/grid", out.dir = "~/folder_replicate/matrix",
    include.intermediate = FALSE)
```

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("~/folder_replicate/matrix/CG_StateCalls.txt", header = TRUE)
```

```
     seqnames start end WT_rep1 WT_rep2 mutant1_rep1 mutant1_rep2
  1:        1     1 200       0       1            1            1
  2:        1    51 250       0       1            1            1
  3:        1   101 300       0       1            1            1
  4:        1   151 350       1       1            0            0
  5:        1   201 400       1       0            0            0
  6:        1   251 450       1       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("~/folder_replicate/matrix/CG_rcMethlvl.txt", header = TRUE)
```

```
     seqnames start end WT_rep1 WT_rep2 mutant1_rep1 mutant1_rep2
  1:        1     1 200 0.04022 0.91114      0.48309      0.90956
  2:        1    51 250 0.12897 0.85744      0.54839      0.79152
  3:        1   101 300 0.12897 0.85744      0.54839      0.79152
  4:        1   151 350 0.80224 0.85744      0.47546      0.49508
  5:        1   201 400 0.80224 0.47885      0.47546      0.49508
  6:        1   251 450 0.80224 0.47885      0.47546      0.49508
```

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

```
postMax <- fread("~/folder_replicate/matrix/CG_postMax.txt", header = TRUE)
```

```
     seqnames start end WT_rep1 WT_rep2 mutant1_rep1 mutant1_rep2
  1:        1     1 200 0.97962 0.99733      0.98386      0.88337
  2:        1    51 250 0.79295 0.87356      0.82874      0.64923
  3:        1   101 300 0.79295 0.87356      0.82874      0.64923
  4:        1   151 350 0.64649 0.87356      0.33333      0.33333
  5:        1   201 400 0.64649 0.33333      0.33333      0.33333
  6:        1   251 450 0.64649 0.33333      0.33333      0.33333
```

3.3 Split DMR matrix into pairwise groups

Ignore this step if you are running jDMR on population data without replicates

samplelist: DATAFRAME OBJECT containing information about file, sample and replicate. For control/treatment data an additional column specifying the replicates is required.

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".

postMax.out: by default this option is set to FALSE. If you want to output the matrix containing posterior probabilities set it to TRUE.

```r
splitGroups(samplelist = listFiles2, input.dir = "~/folder_replicate/matrix", out.dir = "~/folder_replicate/ma
```

4 Filter DMR matrix

4.1 Filter the DMR matrix

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

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.

samplelist: DATAFRAME OBJECT containing information about file, sample and replicate. For control/treatment data an additional column specifying the replicates is required.

if.mergingBins : Logical argument if merging consecutive bins having the same stateCalls should be performed. By default set to TRUE. (logical)

4.2 Filtered Output

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

```
statecallsFiltered <- fread("~/folder_population/matrix/CG_StateCalls-filtered.txt",
    header = TRUE)
```

```
     seqnames start end methylomeA_rep1 methylomeB_rep1 methylomeC_rep1 methylomeD_rep1
  1:        1     1 200               0               1               1               1
  2:        1    51 250               0               1               1               1
  3:        1   101 300               0               1               1               1
  4:        1   151 350               1               1               0               0
  5:        1   201 400               1               0               0               0
  6:        1   251 450               1               0               0               0
```

If "rc.methlvl.out" option is set to TRUE a filtered matrix with averaged methylation levels in generated.

```
rcmethlvlFiltered <- fread("~/folder_population/matrix/CG_rcMethlvl-filtered.txt",
    header = TRUE)
```

```
     seqnames start end methylomeA_rep1 methylomeB_rep1 methylomeC_rep1 methylomeD_rep1
  1:        1     1 200         0.04022         0.91114         0.48309         0.90956
  2:        1    51 250         0.12897         0.85744         0.54839         0.79152
  3:        1   101 300         0.12897         0.85744         0.54839         0.79152
  4:        1   151 350         0.80224         0.85744         0.47546         0.49508
  5:        1   201 400         0.80224         0.47885         0.47546         0.49508
  6:        1   251 450         0.80224         0.47885         0.47546         0.49508
```

5 Search for context-specific and annotate DMRs

5.1 Output context specific DMRs

Output DMRs specific for contexts i.e CG-only, CHG-only, CHH-only, non-CG and multi-context DMRs using the *StateCalls-filtered.txt files (if variable ifFiltered equals to TRUE) or* StateCalls.txt files (if variable ifFiltered equals to FALSE, as default).

```
context.specific.DMRs(samplelist = listFiles1, output.dir = "~/folder_population/context_DMRs",
    input.dir = "~/folder_population/matrix", if.filtered = FALSE)
context.specific.DMRs(samplelist = listFiles2, output.dir = "~/folder_replicate/context_DMRs",
    input.dir = "~/folder_replicate/matrix", if.filtered = FALSE)
```

5.2 Annotate DMRs

This function annotates the lists of DMRs. Any file(.txt) containing 3 columns (chr, start, stop) can be annotated using the annotateDMRs function. Please move all files to be annotated to a separate folder and set the full PATH to the "input.dir" 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.

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

out.dir: path to output folder

In the following example, I will annotate the files generated in section 4.3

```
gff.file_promoters <- system.file("extdata/toyData", "TAIR10_promoters.gff3", package = "jDMRgrid")
gff.file_TE <- system.file("extdata/toyData", "TAIR10_TE.gff3", package = "jDMRgrid")

annotateDMRs(gff.files = c(gff.file_promoters, gff.file_TE), annotation = c("promoters",
    "TE"), input.dir = "~/folder_population/context_DMRs", if.gff3 = FALSE, out.dir = "~/folder_population/ann

annotateDMRs(gff.files = c(gff.file_promoters, gff.file_TE), annotation = c("promoters",
    "TE"), input.dir = "~/folder_replicate/context_DMRs", if.gff3 = FALSE, out.dir = "~/folder_replicate/annot
```

5.3 Output files after annotation

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

```
DMRcounts <- fread("~/folder_replicate/annotate_DMRs/DMR-counts.txt", header = TRUE)
```

```
                              sample total.DMRs promoters TE multiple.overlaps
  1: WT_mutant1_multi-context-DMRs          2         1  0                 0
```

6 R session info

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 22.04.3 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               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    LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: Etc/UTC
tzcode source: system (glibc)

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

other attached packages:
[1] data.table_1.14.8 jDMRgrid_0.2.3

loaded via a namespace (and not attached):
 [1] tidyselect_1.2.0            dplyr_1.1.3                 R.utils_2.12.2              Biostrings_2.68.1
 [5] bitops_1.0-7                fastmap_1.1.1               RCurl_1.98-1.12             GenomicAlignments_1
 [9] XML_3.99-0.14               digest_0.6.33               lifecycle_1.0.3             magrittr_2.0.3
[13] compiler_4.3.1              rlang_1.1.1                 tools_4.3.1                 utf8_1.2.3
[17] yaml_2.3.7                  rtracklayer_1.60.1          knitr_1.44                  S4Arrays_1.0.6
[21] DelayedArray_0.26.7         plyr_1.8.9                  abind_1.4-5                 BiocParallel_1.34.2
[25] purrr_1.0.2                 R.oo_1.25.0                 BiocGenerics_0.46.0         grid_4.3.1
[29] stats4_4.3.1                fansi_1.0.5                 colorspace_2.1-0            future_1.33.0
[33] ggplot2_3.4.4               globals_0.16.2              scales_1.2.1                iterators_1.0.14
[37] SummarizedExperiment_1.30.2 cli_3.6.1                   rmarkdown_2.25              crayon_1.5.2
[41] generics_0.1.3              rstudioapi_0.15.0           future.apply_1.11.0         reshape2_1.4.4
[45] rjson_0.2.21                RUnit_0.4.32                ape_5.7-1                   stringr_1.5.0
[49] zlibbioc_1.46.0             parallel_4.3.1              formatR_1.14                BiocManager_1.30.22
[53] XVector_0.40.0              restfulr_0.0.15             matrixStats_1.0.0           vctrs_0.6.4
[57] Matrix_1.6-1.1              minpack.lm_1.2-4            IRanges_2.34.1              S4Vectors_0.38.2
[61] RBGL_1.76.0                 listenv_0.9.0               foreach_1.5.2               tidyr_1.3.0
[65] glue_1.6.2                  parallelly_1.36.0           codetools_0.2-19            stringi_1.7.12
[69] gtable_0.3.4                GenomeInfoDb_1.36.4         GenomicRanges_1.52.1        BiocIO_1.10.0
[73] munsell_0.5.0               tibble_3.2.1                pillar_1.9.0                htmltools_0.5.6.1
[77] graph_1.78.0                methimpute_1.22.0           GenomeInfoDbData_1.2.10     R6_2.5.1
[81] doParallel_1.0.17           lattice_0.21-9              evaluate_0.22               Biobase_2.60.0
[85] R.methodsS3_1.8.2           Rsamtools_2.16.0            Rcpp_1.0.11                 nlme_3.1-163
[89] xfun_0.40                   MatrixGenerics_1.12.3       biocViews_1.68.2            pkgconfig_2.0.3
```