**DMRcate** for bisulfite sequencing

Peters TJ

May 29, 2024

**Summary**

Worked example to find DMRs from whole genome bisulfite sequencing data.

```
if (!require("BiocManager"))
  install.packages("BiocManager")
BiocManager::install("DMRcate")
```

Load **DMRcate** into the workspace:

```r
library(DMRcate)
```

Bisulfite sequencing assays are fundamentally different to arrays, because methylation is represented as a pair of methylated and unmethylated reads per sample, instead of a single beta value. Although we could simply take the logit-transformed fraction of methylated reads per CpG, this removes the effect of varying read depth across the genome. For example, a sampling depth of 30 methylated reads and 10 unmethylated reads is a much more precise estimate of the methylation level of a given CpG site than 3 methylated and 1 unmethylated. Hence, we take advantage of the fact that the overall effect can be expressed as an interaction between the coefficient of interest and a two-level factor representing methylated and unmethylated reads [1].

The example shown here will be performed on a BSseq object containing bisulfite sequencing of regulatory T cells from various tissues as part of the **tissueTreg** package[2], imported using ExperimentHub. First, we will import the data:

```r
library(ExperimentHub)
eh <- ExperimentHub()
bis_1072 <- eh[["EH1072"]]
bis_1072
```

## An object of type 'BSseq' with
## 21867550 methylation loci
## 15 samples
## has been smoothed with
## BSmooth (ns = 70, h = 1000, maxGap = 10000000)
## All assays are in-memory

colnames(bis_1072)

# [1] "Fat-Treg-R1"  "Fat-Treg-R2"  "Fat-Treg-R3"  "Liver-Treg-R1"
# [5] "Liver-Treg-R2"  "Liver-Treg-R3"  "Skin-Treg-R1"  "Skin-Treg-R2"
# [9] "Skin-Treg-R3"  "Lymph-N-Tcon-R1"  "Lymph-N-Tcon-R2"  "Lymph-N-Tcon-R3"
# [13] "Lymph-N-Treg-R1"  "Lymph-N-Treg-R2"  "Lymph-N-Treg-R3"

The data contains 15 samples: 3 (unmatched) replicates of mouse Tregs from fat, liver, skin and lymph node, plus a group of 3 CD4+ conventional lymph node T cells (Tcon). We will annotate the BSseq object to reflect this phenotypic information:

bsseq::pData(bis_1072) <- data.frame(replicate=gsub(".*-", "", colnames(bis_1072)),
  tissue=substr(colnames(bis_1072), 1, nchar(colnames(bis_1072)) - 3),
  row.names=colnames(bis_1072))
colData(bis_1072)$tissue <- gsub("-", "_", colData(bis_1072)$tissue)
as.data.frame(colData(bis_1072))

# replicate tissue
# Fat-Treg-R1 R1 Fat_Treg
# Fat-Treg-R2 R2 Fat_Treg
# Fat-Treg-R3 R3 Fat_Treg
# Liver-Treg-R1 R1 Liver_Treg
# Liver-Treg-R2 R2 Liver_Treg
# Liver-Treg-R3 R3 Liver_Treg
# Skin-Treg-R1 R1 Skin_Treg
# Skin-Treg-R2 R2 Skin_Treg
# Skin-Treg-R3 R3 Skin_Treg
# Lymph-N-Tcon-R1 R1 Lymph_N_Tcon
# Lymph-N-Tcon-R2 R2 Lymph_N_Tcon
# Lymph-N-Tcon-R3 R3 Lymph_N_Tcon
# Lymph-N-Treg-R1 R1 Lymph_N_Treg
# Lymph-N-Treg-R2 R2 Lymph_N_Treg
# Lymph-N-Treg-R3 R3 Lymph_N_Treg

For standardisation purposes (and for DMR.plot to recognise the genome) we will change the chromosome naming convention to UCSC:

2
For demonstration purposes, we will retain CpGs on chromosome 19 only:

```r
bis_1072 <- renameSeqlevels(bis_1072, mapSeqlevels(seqlevels(bis_1072), "UCSC"))
bis_1072 <- bis_1072[seqnames(bis_1072)="chr19",]
```

An object of type 'BSseq' with
558056 methylation loci
15 samples
has been smoothed with
BSmooth (ns = 70, h = 1000, maxGap = 100000000)
All assays are in-memory

Now we can prepare the model to be fit for `sequencing.annotate()`. The arguments are equivalent to `cpg.annotate()` but for a couple of exceptions:

- There is an extra argument `all.cov` giving an option whether to retain only CpGs where all samples have non-zero coverage, or whether to retain CpGs with only partial sample representation.
- The design matrix should be constructed to reflect the 2-factor structure of methylated and unmethylated reads. Fortunately, `edgeR::modelMatrixMeth()` can take a regular design matrix and transform it into the appropriate structure ready for model fitting.

```r
tissue <- factor(pData(bis_1072)$tissue)
tissue <- relevel(tissue, "Liver_Treg")

design <- model.matrix(~tissue)
colnames(design) <- gsub("tissue", ", colnames(design))
colnames(design)[1] <- "Intercept"
rownames(design) <- colnames(bis_1072)
```

```
# Intercept Fat_Treg Lymph_N_Tcon Lymph_N_Treg Skin_Treg
# Fat-Treg-R1  1  1  0  0  0
# Fat-Treg-R2  1  1  0  0  0
# Fat-Treg-R3  1  1  0  0  0
# Liver-Treg-R1  1  0  0  0  0
# Liver-Treg-R2  1  0  0  0  0
# Liver-Treg-R3  1  0  0  0  0
# Skin-Treg-R1  1  0  0  0  1
# Skin-Treg-R2  1  0  0  0  1
# Skin-Treg-R3  1  0  0  0  1
```
# Methylation matrix design

methdesign <- edgeR::modelMatrixMeth(design)

methdesign

```
## Sample1 Sample2 Sample3 Sample4 Sample5 Sample6 Sample7 Sample8 Sample9
## 1 1 0 0 0 0 0 0 0 0
## 2 1 0 0 0 0 0 0 0 0
## 3 0 1 0 0 0 0 0 0 0
## 4 0 1 0 0 0 0 0 0 0
## 5 0 0 1 0 0 0 0 0 0
## 6 0 0 1 0 0 0 0 0 0
## 7 0 0 0 1 0 0 0 0 0
## 8 0 0 0 1 0 0 0 0 0
## 9 0 0 0 0 1 0 0 0 0
## 10 0 0 0 0 1 0 0 0 0
## 11 0 0 0 0 0 1 0 0 0
## 12 0 0 0 0 0 0 1 0 0
## 13 0 0 0 0 0 0 0 1 0
## 14 0 0 0 0 0 0 0 1 0
## 15 0 0 0 0 0 0 0 0 1
```
<table>
<thead>
<tr>
<th></th>
<th>Sample10</th>
<th>Sample11</th>
<th>Sample12</th>
<th>Sample13</th>
<th>Sample14</th>
<th>Sample15</th>
<th>Intercept</th>
<th>Fat_Treg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Lymph_N_Tcon</th>
<th>Lymph_N_Treg</th>
<th>Skin_Treg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Just like for `cpg.annotate()`, we can specify a contrast matrix to find our comparisons of interest.

```r
cont.mat <- limma::makeContrasts(treg_vs_tcon=Lymph_N_Treg-Lymph_N_Tcon, 
                                  fat_vs_ln=Fat_Treg-Lymph_N_Treg, 
                                  skin_vs_ln=Skin_Treg-Lymph_N_Treg, 
                                  fat_vs_skin=Fat_Treg-Skin_Treg, 
                                  levels=methdesign)
```

```r
cont.mat
## Contrasts
## Levels treg_vs_tcon fat_vs_ln skin_vs_ln fat_vs_skin
## Sample1 0 0 0 0
## Sample2 0 0 0 0
## Sample3 0 0 0 0
## Sample4 0 0 0 0
## Sample5 0 0 0 0
## Sample6 0 0 0 0
## Sample7 0 0 0 0
## Sample8 0 0 0 0
## Sample9 0 0 0 0
## Sample10 0 0 0 0
## Sample11 0 0 0 0
## Sample12 0 0 0 0
## Sample13 0 0 0 0
## Sample14 0 0 0 0
## Sample15 0 0 0 0
```
Say we want to find DMRs between the regulatory and conventional T cells from the lymph node. First we would fit the model, where `sequencing.annotate()` transforms counts into log2CPMs (via `limma::voom()`) and uses `limma` under the hood to generate per-CpG $t$-statistics, indexing the FDR at 0.05:

```r
seq_annot <- sequencing.annotate(bis_1072, methdesign, all.cov = TRUE,
                                 contrasts = TRUE, cont.matrix = cont.mat,
                                 coef = "treg_vs_tcon", fdr=0.05)
```

And then, just like before, we can call DMRs with `dmrcate()`:

```r
dmrcate.res <- dmrcate(seq_annot, C=2, min.cpgs = 5)
```

```r
treg_vs_tcon.ranges <- extractRanges(dmrcate.res, genome="mm10")
```
## GRanges object with 9 ranges and 8 metadata columns:

<table>
<thead>
<tr>
<th>seqnames</th>
<th>ranges</th>
<th>strand</th>
<th>no.cpgs</th>
<th>min_smoothed_fdr</th>
<th>Stouffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr19</td>
<td>29270611-29272005</td>
<td>*</td>
<td>16</td>
<td>4.32351e-94</td>
<td>1.000000</td>
</tr>
<tr>
<td>chr19</td>
<td>26683453-26684174</td>
<td>*</td>
<td>12</td>
<td>1.77927e-57</td>
<td>1.000000</td>
</tr>
<tr>
<td>chr19</td>
<td>32276886-32278089</td>
<td>*</td>
<td>13</td>
<td>1.74620e-56</td>
<td>1.000000</td>
</tr>
<tr>
<td>chr19</td>
<td>29374953-29375393</td>
<td>*</td>
<td>12</td>
<td>1.48257e-54</td>
<td>1.000000</td>
</tr>
<tr>
<td>chr19</td>
<td>36378257-36379597</td>
<td>*</td>
<td>27</td>
<td>1.53747e-76</td>
<td>1.000000</td>
</tr>
<tr>
<td>chr19</td>
<td>46653280-46654180</td>
<td>*</td>
<td>19</td>
<td>3.94008e-59</td>
<td>1.000000</td>
</tr>
<tr>
<td>chr19</td>
<td>57092365-57092646</td>
<td>*</td>
<td>10</td>
<td>3.80468e-36</td>
<td>0.139494</td>
</tr>
<tr>
<td>chr19</td>
<td>40808208-40809554</td>
<td>*</td>
<td>26</td>
<td>3.43873e-63</td>
<td>1.000000</td>
</tr>
<tr>
<td>chr19</td>
<td>41874401-41874895</td>
<td>*</td>
<td>22</td>
<td>2.75829e-39</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

**HMFDR Fisher maxdiff meandiff overlapping.genes**

| seqinfo | :
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 sequence from an unspecified genome; no seqlengths</td>
</tr>
</tbody>
</table>

Looks like the top DMR is associated with the **Jak2** locus and hypomethylated in the Treg cells (since `meandiff < 0`). We can plot it like so:

```r
cols <- as.character(plyr::mapvalues(tissue, unique(tissue),
                                    c("darkorange", "maroon", "blue",
                                    "black", "magenta")))
names(cols) <- tissue

DMR.plot(treg_vs_tcon.ranges, dmr = 1,
         CpGs=bis_1072[tissue %in% c("Lymph_N_Tcon", "Lymph_N_Treg")],
         phen.col = cols[tissue %in% c("Lymph_N_Tcon", "Lymph_N_Treg")],
         genome="mm10")
```
Now, let’s find DMRs between fat and skin Tregs.

```r
seq_annot <- sequencing.annotate(bis_1072, methdesign, all.cov = TRUE,
                               contrasts = TRUE, cont.matrix = cont.mat,
                               coef = "fat_vs_skin", fdr=0.05)
```

## Filtering out all CpGs where at least one sample has zero coverage...
## Processing BSseq object...
## Transforming counts...
## Fitting model...
## Your contrast returned 5 individually significant CpGs; a small
## but real effect. Consider increasing the 'fdr' parameter using changeFDR(),
## but be warned there is an increased risk of Type I errors.

Because this comparison is a bit more subtle, there are very few significantly
differential CpGs at this threshold. So we can use `changeFDR()` to relax the
FDR to 0.25, taking into account that there is an increased risk of false positives.

```r
seq_annot <- changeFDR(seq_annot, 0.25)
## Threshold is now set at FDR=0.25, resulting in 63 significantly differential CpGs.

dmrcate.res <- dmrcate(seq_annot, C=2, min.cpgs = 5)
## Fitting chr19...
## Demarcating regions...
## Done!
fat_vs_skin.ranges <- extractRanges(dmrcate.res, genome="mm10")
## see ?DMRcatedata and browseVignettes(‘DMRcatedata’) for documentation
## loading from cache

Now let’s plot the top DMR with not only fat and skin, but with all samples:

```r
cols
## Fat_Treg Fat_Treg Fat_Treg Liver_Treg Liver_Treg Liver_Treg
## "darkorange" "darkorange" "darkorange" "maroon" "maroon" "maroon"
## Skin_Treg Skin_Treg Skin_Treg Lymph_N_Tcon Lymph_N_Tcon Lymph_N_Tcon
## "blue" "blue" "blue" "black" "black" "black"
## Lymph_N_Treg Lymph_N_Treg Lymph_N_Treg
## "magenta" "magenta" "magenta"

DMR.plot(fat_vs_skin.ranges, dmr = 1, CPGs=bis_1072, phen.col = cols, genome="mm10")
```
Here we can see the methylation of skin cells over this region near the \textit{Gcnt1} promoter is hypomethylated not only relative to fat, but to the other tissues as well.

```r
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 22.04.4 LTS
##
## Matrix products: default
## BLAS: /home/biocbuild/bbs-3.19-bioc/R/lib/libRblas.so
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8   LC_NUMERIC=C
```
## time zone: America/New_York
## tzcode source: system (glibc)
##
## attached base packages:
##  [1] parallel  stats4   stats    graphics  grDevices  utils  datasets
##  [8] methods   base
##
## other attached packages:
##  [1] bsseq_1.40.0
##  [2] tissueTreg_1.24.0
##  [3] DMRcatedata_2.22.0
##  [4] IlluminaHumanMethylationEPICannoilm10b4.hg19_0.6.0
##  [5] IlluminaHumanMethylationEPICmanifest_0.3.0
##  [7] minfi_1.50.0
##  [8] bumphunter_1.46.0
##  [9] locfit_1.5-9.9
## [10] iterators_1.0.14
## [11] foreach_1.5.2
## [12] Biostings_2.72.0
## [13] XVector_0.44.0
## [14] SummarizedExperiment_1.34.0
## [15] Biobase_2.64.0
## [16] MatrixGenerics_1.16.0
## [17] matrixStats_1.3.0
## [18] GenomicRanges_1.56.0
## [19] GenomeInfoDb_1.40.1
## [20] IRanges_2.38.0
## [21] S4Vectors_0.42.0
## [22] ExperimentHub_2.12.0
## [23] AnnotationHub_3.12.0
## [24] BiocFileCache_2.12.0
## [25] dbplyr_2.5.0
## [26] BiocGenerics_0.50.0
## [27] DMRcate_3.0.1
##
## loaded via a namespace (and not attached):
##  [1] splines_4.4.0
## [3] bitops_1.0-7
## [4] filelock_1.0.3
## [5] cellranger_1.1.0
## [6] tibble_3.2.1
## [8] preprocessCore_1.66.0
## [9] XML_3.99-0.16.1
## [10] rpart_4.1.23
## [11] lifecycle_1.0.4
## [12] httr2_1.0.1
## [13] edgeR_4.2.0
## [14] base64_2.0.1
## [15] MASS_7.3-60.2
## [16] lattice_0.22-6
## [17] ensembldb_2.28.0
## [18] scrime_1.3.5
## [19] backports_1.5.0
## [20] magrittr_2.0.3
## [21] limma_3.60.2
## [22] Hmisc_5.1-3
## [23] rmarkdown_2.27
## [24] yaml_2.3.8
## [25] doRNG_1.8.6
## [26] askpass_1.2.0
## [27] Gviz_1.48.0
## [28] DBI_1.2.2
## [29] RColorBrewer_1.1-3
## [30] abind_1.4-5
## [31] zlibbioc_1.50.0
## [32] quadprog_1.5-8
## [33] purrr_1.0.2
## [34] R.utils_2.12.3
## [35] AnnotationFilter_1.28.0
## [36] biovizBase_1.52.0
## [37] RCurl_1.98-1.14
## [38] nnet_7.3-19
## [39] VariantAnnotation_1.50.0
## [40] rappdirs_0.3.3
## [41] GenomeInfoDbData_1.2.12
## [42] genefilter_1.86.0
## [43] annotate_1.82.0
## [44] permute_0.9-7
## [45] DelayedMatrixStats_1.26.0
## [46] codetools_0.2-20
## [47] DelayedArray_0.30.1
## [48] xml2_1.3.6  
## [49] tidyselect_1.2.1  
## [50] UCSC.utils_1.0.0  
## [51] beanplot_1.3.1  
## [52] base64enc_0.1-3  
## [53] illuminaio_0.46.0  
## [54] GenomicAlignments_1.40.0  
## [55] jsonlite_1.8.8  
## [56] multtest_2.60.0  
## [57] Formula_1.2-5  
## [58] survival_3.6-4  
## [59] missMethyl_1.38.0  
## [60] tools_4.4.0  
## [61] progress_1.2.3  
## [62] Rcpp_1.0.12  
## [63] glue_1.7.0  
## [64] gridExtra_2.3  
## [65] SparseArray_1.4.8  
## [66] xfun_0.44  
## [67] dplyr_1.1.4  
## [68] HDF5Array_1.32.0  
## [69] withr_3.0.0  
## [70] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.6.1  
## [71] BiocManager_1.30.23  
## [72] fastmap_1.2.0  
## [73] latticeExtra_0.6-30  
## [74] rhdf5filters_1.16.0  
## [75] fansi_1.0.6  
## [76] openssl_2.2.0  
## [77] digest_0.6.35  
## [78] mime_0.12  
## [79] R6_2.5.1  
## [80] colorspace_2.1-0  
## [81] gtools_3.9.5  
## [82] jpeg_0.1-10  
## [83] dichromat_2.0-0.1  
## [84] biomaRt_2.60.0  
## [85] RSQLite_2.3.7  
## [86] R.methodsS3_1.8.2  
## [87] tidyr_1.3.1  
## [88] utf8_1.2.4  
## [89] generics_0.1.3  
## [90] data.table_1.15.4  
## [91] rtracklayer_1.64.0  
## [92] prettyunits_1.2.0
## [93] httr_1.4.7
## [94] htmlwidgets_1.6.4
## [95] S4Arrays_1.4.1
## [96] pkgconfig_2.0.3
## [97] gtable_0.3.5
## [98] blob_1.2.4
## [99] siggenes_1.78.0
## [100] htmltools_0.5.8.1
## [101] ProtGenerics_1.36.0
## [102] scales_1.3.0
## [103] png_0.1-8
## [104] knitr_1.47
## [105] rstudioapi_0.16.0
## [106] tzdb_0.4.0
## [107] rjson_0.2.21
## [108] nlme_3.1-164
## [109] checkmate_2.3.1
## [110] curl_5.2.1
## [111] org.Hs.eg.db_3.19.1
## [112] cachem_1.1.0
## [113] rhdf5_2.48.0
## [114] stringr_1.5.1
## [115] BiocVersion_3.19.1
## [116] foreign_0.8-86
## [117] AnnotationDbi_1.66.0
## [118] restfulr_0.0.15
## [119] GEOquery_2.72.0
## [120] pillar_1.9.0
## [121] grid_4.4.0
## [122] reshape_0.8.9
## [123] vctrs_0.6.5
## [124] xtable_1.8-4
## [125] cluster_2.1.6
## [126] htmlTable_2.4.2
## [127] evaluate_0.23
## [128] readr_2.1.5
## [129] GenomicFeatures_1.56.0
## [130] cli_3.6.2
## [131] compiler_4.4.0
## [132] Rsamtools_2.20.0
## [133] rngtools_1.5.2
## [134] rlang_1.1.3
## [135] crayon_1.5.2
## [136] nor1mix_1.3-3
## [137] mclust_6.1.1
## References
