**DMRcate** for bisulfite sequencing

Peters TJ

May 5, 2024

**Summary**

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

```r
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 = 100000000)
## 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:

```
```
bis_1072 <- renameSeqlevels(bis_1072, mapSeqlevels(seqlevels(bis_1072), "UCSC"))

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

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 = 10000000)
## 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.

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

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

# 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
```r
## Lymph-N-Tcon-R1 1 0 1 0 0
## Lymph-N-Tcon-R2 1 0 1 0 0
## Lymph-N-Tcon-R3 1 0 1 0 0
## Lymph-N-Treg-R1 1 0 0 1 0
## Lymph-N-Treg-R2 1 0 0 1 0
## Lymph-N-Treg-R3 1 0 0 1 0
## attr("assign")
## [1] 0 1 1 1 1
## attr("contrasts")
## attr("contrasts")$tissue
## [1] "contr.treatment"

# Methylation matrix design
methdesign <- edgeR::modelMatrixMeth(design)
methdesign
```

```r
## 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 1 0
## 16 0 0 0 0 0 0 0 1 0
## 17 0 0 0 0 0 0 0 0 1
## 18 0 0 0 0 0 0 0 0 0
## 19 0 0 0 0 0 0 0 0 0
## 20 0 0 0 0 0 0 0 0 0
## 21 0 0 0 0 0 0 0 0 0
## 22 0 0 0 0 0 0 0 0 0
## 23 0 0 0 0 0 0 0 0 0
## 24 0 0 0 0 0 0 0 0 0
## 25 0 0 0 0 0 0 0 0 0
## 26 0 0 0 0 0 0 0 0 0
## 27 0 0 0 0 0 0 0 0 0
## 28 0 0 0 0 0 0 0 0 0
## 29 0 0 0 0 0 0 0 0 0
```

<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>1</td>
<td>1</td>
<td></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></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></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></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></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></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></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></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></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></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>1</td>
<td>0</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></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></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>1</td>
<td>0</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></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></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>1</td>
<td>0</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></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>0</td>
<td></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)
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)
```

```
## Filtering out all CpGs where at least one sample has zero coverage...
## Processing BSseq object...
## Transforming counts...
## Fitting model...
## Your contrast returned 157 individually significant CpGs. We recommend
## the default setting of pcutoff in dmrcate().
seq_annot
```

```
# CpGannotated object describing 506908 CpG sites, with independent
# CpG threshold indexed at fdr=0.05 and 157 significant CpG sites.
```

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

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

```
## Fitting chr19...
## Demarcating regions...
## Done!
dmrcate.res
```

```
# DMResults object with 9 DMRs.
# Use extractRanges() to produce a GRanges object of these.
treg_vs_tcon.ranges <- extractRanges(dmrcate.res, genome="mm10")
```

```
# see ?DMRcatedata and browseVignettes('DMRcatedata') for documentation
# loading from cache
treg_vs_tcon.ranges
```
## GRanges object with 9 ranges and 8 metadata columns:
##
## seqnames ranges strand | no.cpgs min_smoothed_fdr Stouffer
## <Rle> <IRanges> <Rle> | <integer> <numeric> <numeric>
## [1] chr19 29270611-29272005 * | 16 4.32351e-94 1.000000
## [4] chr19 29374953-29375393 * | 12 1.48257e-54 1.000000
## [5] chr19 36378257-36379597 * | 27 1.53747e-76 1.000000
## [7] chr19 57092365-57092646 * | 10 3.80468e-36 0.139494
## [8] chr19 40808208-40809554 * | 26 3.43873e-63 1.000000
##
## HMFD Radeon Fisher maxdiff meandiff overlapping.genes
## <numeric> <numeric> <numeric> <numeric> <character>
## [1] 0.0151786 2.14645e-08 -6.40482 -4.22353 Jak2
## [2] 0.0078774 1.28163e-04 -6.40328 -3.53692 Smarca2
## [3] 0.0446759 1.50767e-04 5.81470 3.93201 Sgms1
## [4] 0.0282265 2.41191e-03 -6.09023 -3.02083 Cd274, Gm36043
## [5] 0.0482585 7.25026e-03 5.18388 2.93152 Wbp1l
## [6] 0.0512002 4.52566e-02 5.18388 2.93152 Wbp1l
## [7] 0.0711193 6.39021e-02 -4.67645 -3.36472 Ablim1
## [8] 0.1582571 3.05279e-01 -4.83855 -3.07494 Cc2d2b
## [9] 0.1858534 6.90217e-01 4.57011 2.56520 Rrp12
##
## -----
##
## seqinfo: 1 sequence from an unspecified genome; no seqlengths

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.

```
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:

```
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 *Gcnt1* promoter is hypomethylated not only relative to fat, but to the other tissues as well.
## [3] LC_TIME=en_GB LC_COLLATE=C
## [5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8 LC_NAME=C
## [9] LC_ADDRESS=C LC_TELEPHONE=C
##
## 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] IlluminaHumanMethylationEPICanno.ilm10b4.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] Biostrings_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_1.40.0
## [20] IRanges_2.38.0
## [21] S4Vectors_0.42.0
## [22] ExperimentHub_2.12.0
## [23] AnnotationHub_3.12.0
## [24] Bioconductor_2.12.0
## [25] dbplyr_2.5.0
## [26] BiocGenerics_0.50.0
## [27] DMRcate_3.0.0
##
## 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_1.4.1
## [20] magrittr_2.0.3
## [21] limma_3.60.0
## [22] Hmisc_5.1-2
## [23] rmarkdown_2.26
## [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_0.30.0
```r
## [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_1.4.1
## [66] xfun_0.43
## [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_1.1.1
## [73] latticeExtra_0.6-30
## [74] rhdf5filters_1.16.0
## [75] fansi_1.0.6
## [76] openssl_2.1.2
## [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_2.3.6
## [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
```
## 

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>httr_1.4.7</td>
<td></td>
</tr>
<tr>
<td>htmlwidgets_1.6.4</td>
<td></td>
</tr>
<tr>
<td>S4Arrays_1.4.0</td>
<td></td>
</tr>
<tr>
<td>pkgconfig_2.0.3</td>
<td></td>
</tr>
<tr>
<td>gtable_0.3.5</td>
<td></td>
</tr>
<tr>
<td>blob_1.2.4</td>
<td></td>
</tr>
<tr>
<td>siggenes_1.78.0</td>
<td></td>
</tr>
<tr>
<td>htmltools_0.5.8.1</td>
<td></td>
</tr>
<tr>
<td>ProtGenerics_1.36.0</td>
<td></td>
</tr>
<tr>
<td>scales_1.3.0</td>
<td></td>
</tr>
<tr>
<td>png_0.1-8</td>
<td></td>
</tr>
<tr>
<td>knitr_1.46</td>
<td></td>
</tr>
<tr>
<td>rstudioapi_0.16.0</td>
<td></td>
</tr>
<tr>
<td>tzdb_0.4.0</td>
<td></td>
</tr>
<tr>
<td>rjson_0.2.21</td>
<td></td>
</tr>
<tr>
<td>nlm_3.1-164</td>
<td></td>
</tr>
<tr>
<td>checkmate_2.3.1</td>
<td></td>
</tr>
<tr>
<td>curl_5.2.1</td>
<td></td>
</tr>
<tr>
<td>org.Hs.eg.db_3.19.1</td>
<td></td>
</tr>
<tr>
<td>cachem_1.0.8</td>
<td></td>
</tr>
<tr>
<td>rhdf5_2.48.0</td>
<td></td>
</tr>
<tr>
<td>stringr_1.5.1</td>
<td></td>
</tr>
<tr>
<td>BiocVersion_3.19.1</td>
<td></td>
</tr>
<tr>
<td>foreign_0.8-86</td>
<td></td>
</tr>
<tr>
<td>AnnotationDbi_1.66.0</td>
<td></td>
</tr>
<tr>
<td>restfulr_0.0.15</td>
<td></td>
</tr>
<tr>
<td>GEOquery_2.72.0</td>
<td></td>
</tr>
<tr>
<td>pillar_1.9.0</td>
<td></td>
</tr>
<tr>
<td>grid_4.4.0</td>
<td></td>
</tr>
<tr>
<td>reshape_0.8.9</td>
<td></td>
</tr>
<tr>
<td>vctrs_0.6.5</td>
<td></td>
</tr>
<tr>
<td>xtable_1.8-4</td>
<td></td>
</tr>
<tr>
<td>cluster_2.1.6</td>
<td></td>
</tr>
<tr>
<td>htmlTable_2.4.2</td>
<td></td>
</tr>
<tr>
<td>evaluate_0.23</td>
<td></td>
</tr>
<tr>
<td>readr_2.1.5</td>
<td></td>
</tr>
<tr>
<td>GenomicFeatures_1.56.0</td>
<td></td>
</tr>
<tr>
<td>cli_3.6.2</td>
<td></td>
</tr>
<tr>
<td>compiler_4.4.0</td>
<td></td>
</tr>
<tr>
<td>Rsamtools_2.20.0</td>
<td></td>
</tr>
<tr>
<td>rngtools_1.5.2</td>
<td></td>
</tr>
<tr>
<td>rlang_1.1.3</td>
<td></td>
</tr>
<tr>
<td>crayon_1.5.2</td>
<td></td>
</tr>
<tr>
<td>nor1mix_1.3-3</td>
<td></td>
</tr>
<tr>
<td>mclust_6.1.1</td>
<td></td>
</tr>
</tbody>
</table>
## References
