The DMRcate package user's guide

Peters TJ

January 10, 2020

Summary

DMRcate extracts the most differentially methylated regions (DMRs) and variably methylated regions (VMRs) from both Whole Genome Bisulphite Sequencing (WGBS) and Illumina®Infinium BeadChip Array samples via kernel smoothing.

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

Load DMRcate into the workspace:

```
library(DMRcate)
```

Illumina®Array Workflow

For this vignette, we will demonstrate DMRcate's array utility using data from ExperimentHub, namely Illumina HumanMethylationEPIC data from the data packages FlowSorted.Blood.EPIC. Specifically, we are interested in the methylation differences between CD4+ and CD8+ T cells.

```
library(ExperimentHub)
eh <- ExperimentHub()
FlowSorted.Blood.EPIC <- eh[["EH1136"]]
tcell <- FlowSorted.Blood.EPIC[,colData(FlowSorted.Blood.EPIC)$CD4T==100 |
                                colData(FlowSorted.Blood.EPIC)$CD8T==100]
```

Firstly we have to filter out any probes where any sample has a failed position. Then we will normalise using minfi::preprocessNoob. After this, we extract the M-values from the GenomicRatioSet.

```
detP <- detectionP(tcell)

## Loading required package:  IlluminaHumanMethylationEPICmanifest

remove <- apply(detP, 1, function (x) any(x > 0.01))
tcell <- tcell[!remove,]
tcell <- preprocessFunnorm(tcell)

## [preprocessFunnorm] Background and dye bias correction with noob
## [preprocessFunnorm] Mapping to genome
## [preprocessFunnorm] Quantile extraction
## [preprocessFunnorm] Normalization

tcellms <- getM(tcell)
```

M-values (logit-transform of beta) are preferable to beta values for significance testing via `limma` because of increased sensitivity, but we will transform this to a beta matrix for visualisation purposes later on.

Some of the methylation measurements on the array may be confounded by proximity to SNPs, and cross-hybridisation to other areas of the genome[1, 2]. In particular, probes that are 0, 1, or 2 nucleotides from the methylcytosine of interest show a markedly different distribution to those farther away, in healthy tissue (Figure 1).

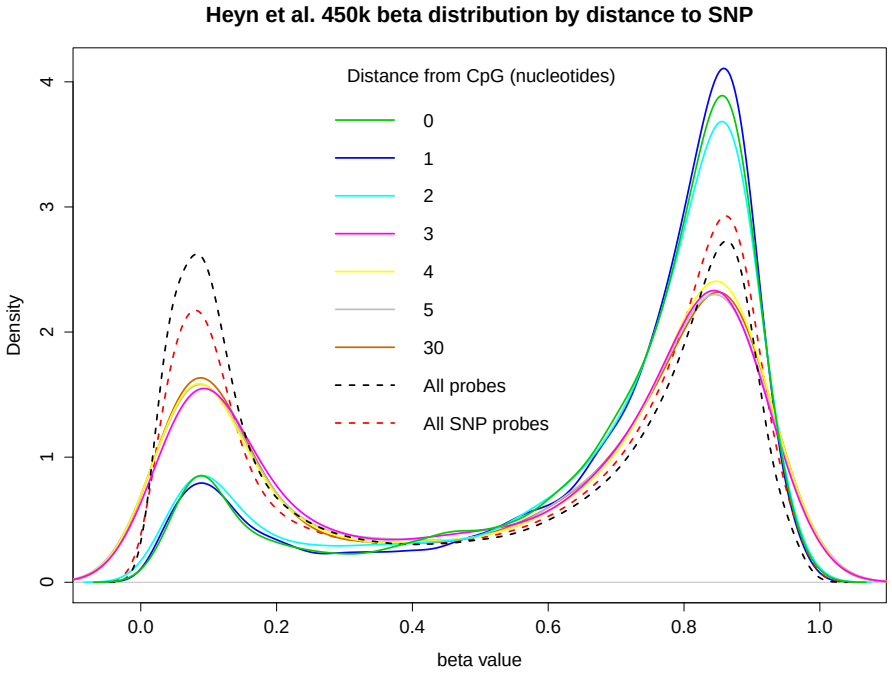
It is with this in mind that we filter out probes 2 nucleotides or closer to a SNP that have a minor allele frequency greater than 0.05, and the approximately 48,000 [1, 2] cross-reactive probes on either 450K and/or EPIC, so as to reduce confounding. Here we use a combination of *in silico* analyses from [1, 2]. About 60,000 are removed from our M-matrix of approximately 864,000:

```
nrow(tcellms)

## [1] 864039

tcellms.noSNPs <- rmSNPandCH(tcellms, dist=2, mafcut=0.05)
nrow(tcellms.noSNPs)

## [1] 803420
```

Here we have 6 CD8+ T cell assays, and 7 CD4+ T cell assays; we want to call DMRs between these groups. One of the CD4+ assays is a technical replicate, so we will average these two replicates like so:

```
tcell$Replicate

##  [1] ""          ""          ""          ""          ""          ""
##  [7] ""          ""          ""          "Th2535-1" "Th2535-1" ""
## [13] ""
```

Figure 1: Beta distribution of 450K probes from publically available data from blood samples of healthy individuals [3] by their proximity to a SNP. "All SNP probes" refers to the 153 113 probes listed by Illumina® whose values may potentially be confounded by a SNP.



```
tcell$Replicate[tcell$Replicate==""] <- tcell$Sample_Name[tcell$Replicate==""]
tcellms.noSNPs <- limma::avearrays(tcellms.noSNPs, tcell$Replicate)
tcell <- tcell[,!duplicated(tcell$Replicate)]
tcell <- tcell[rownames(tcellms.noSNPs),]
assays(tcell)[["M"]] <- tcellms.noSNPs
assays(tcell)[["Beta"]] <- ilogit2(tcellms.noSNPs)
```

Next we want to annotate our matrix of M-values with relevant information. We also use the backbone of the `limma` pipeline for differential array analysis. We want to compare within patients across tissue samples, so we set up our variables for a standard limma pipeline, and set `coef=2` in `cpg.annotate` since this corresponds to the phenotype comparison in `design`.

`cpg.annotate()` takes either a data matrix with Illumina probe IDs, or an already prepared GenomicRatioSet from `minfi`.

```
type <- factor(tcell$CellType)
design <- model.matrix(~type)
myannotation <- cpg.annotate("array", tcell, arraytype = "EPIC",
                             analysis.type="differential", design=design, coef=2)
```

```
myannotation

## CpGannotated object describing 803420 CpG sites, with independent
## CpG threshold indexed at fdr=0.05 and 31633 significant CpG sites.
```

Now we can find our most differentially methylated regions with `dmrcate()`.

For each chromosome, two smoothed estimates are computed: one weighted with per-CpG t-statistics and one not, for a null comparison. The two estimates are compared via a Satterthwaite approximation[4], and a significance test is calculated at all hg19 coordinates that an input probe maps to. After fdr-correction, regions are then agglomerated from groups of post-smoothed significant probes where the distance to the next consecutive probe is less than `lambda` nucleotides.

```
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)

## Fitting chr1...
## Fitting chr2...
## Fitting chr3...
## Fitting chr4...
## Fitting chr5...
## Fitting chr6...
## Fitting chr7...
## Fitting chr8...
## Fitting chr9...
```

```
## Fitting chr10...
## Fitting chr11...
## Fitting chr12...
## Fitting chr13...
## Fitting chr14...
## Fitting chr15...
## Fitting chr16...
## Fitting chr17...
## Fitting chr18...
## Fitting chr19...
## Fitting chr20...
## Fitting chr21...
## Fitting chr22...
## Fitting chrX...
## Fitting chrY...
## Demarcating regions...
## Done!

dmrcoutput

## DMResults object with 4949 DMRs.
## Use extractRanges() to produce a GRanges object of these.
```

We can convert our DMR list to a GRanges object, which uses the `genome` argument to annotate overlapping gene loci.

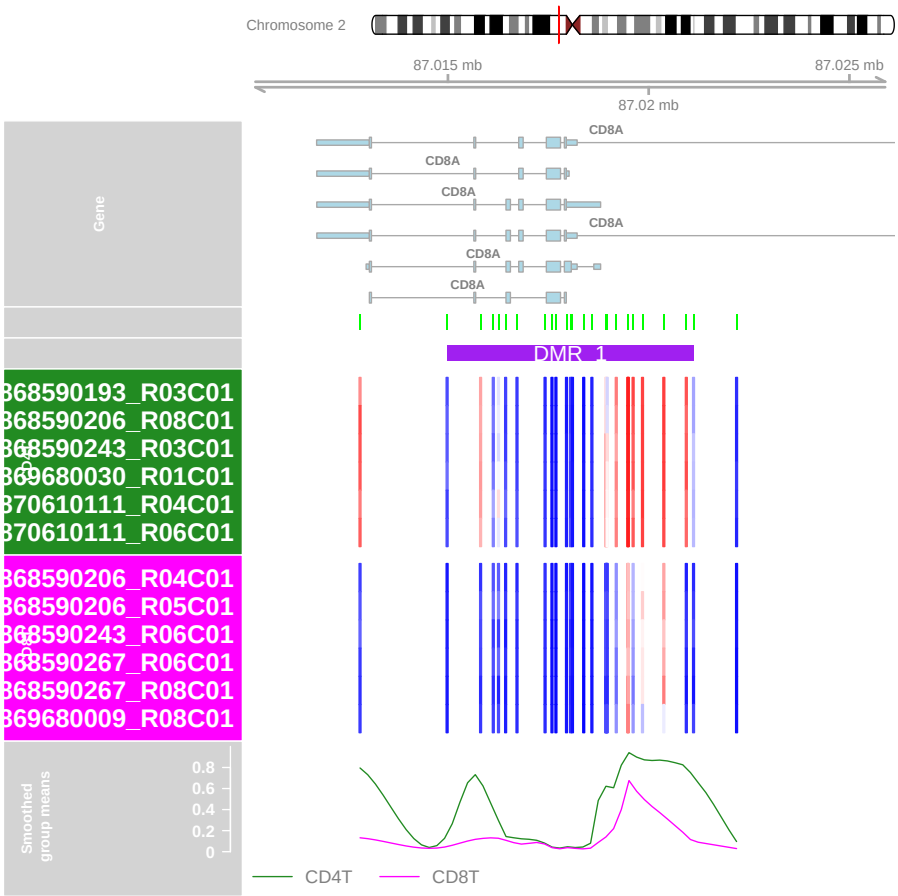
```
results.ranges <- extractRanges(dmrcoutput, genome = "hg19")
results.ranges

## GRanges object with 4949 ranges and 8 metadata columns:
##          seqnames              ranges strand |   no.cpgs      min_smoothed_fdr
##             <Rle>           <IRanges>  <Rle> | <integer>             <numeric>
##      [1]     chr2   87014979-87021117      * |        26                     0
##      [2]    chr17   47286445-47289036      * |        20                     0
##      [3]    chr12     6442329-6444675      * |        14                     0
##      [4]     chr1     2160666-2166155      * |        16 1.84211074257401e-171
##      [5]     chrX 135728914-135730413      * |        10                     0
##      ...      ...                 ...    ... .       ...                   ...
##   [4945]     chr6   31697652-31698899      * |        30  4.94727689667916e-23
##   [4946]     chr6   32120584-32121843      * |        39  1.61729581793529e-20
##   [4947]     chr6   42882703-42883389      * |         2  9.48999919269778e-12
##   [4948]    chr19   53662261-53662353      * |         2  5.81554772757448e-11
##   [4949]    chr10   64578469-64578476      * |         2  1.30283635369131e-10
##                     Stouffer                HMFDR              Fisher
##                    <numeric>            <numeric>           <numeric>
##      [1] 4.02760414580254e-53  6.8073238827719e-07 8.62111492091922e-64
```

```
##      [2] 2.13220507856045e-34 2.55286114637069e-06 4.79029222302714e-41
##      [3] 9.90230381917337e-34 8.04487333597537e-07 6.64603234719223e-41
##      [4] 5.84420550339003e-39 7.55488977783522e-06 2.19863812077882e-39
##      [5]  6.0114742001323e-42 6.94893061084617e-07 6.01039734771414e-39
##      ...                  ...                  ...                  ...
##   [4945]     0.99898176343113  0.00885228447939413    0.824190551257804
##   [4946]    0.999216743080738   0.0271688089869133    0.855903250102537
##   [4947]    0.860229971435707    0.762326002010376    0.899748955165383
##   [4948]    0.863172819930373    0.776670023474993    0.908883156746113
##   [4949]    0.951728160160997    0.865069070547641    0.966347558591277
##                     maxdiff             meandiff
##                   <numeric>            <numeric>
##      [1]  -0.733559196947324   -0.236923818756527
##      [2]  -0.635726177012658    -0.20297012977379
##      [3]   -0.63513162760869   -0.304662807536568
##      [4]   0.448376298467363    0.208764071796863
##      [5]   0.764572442533011    0.543704625452575
##      ...                  ...                  ...
##   [4945]  -0.385048848181761  -0.0320309885414187
##   [4946]  -0.146157983314745  0.00272269666532804
##   [4947]  0.0356160369671412   0.0205559012677759
##   [4948] 0.00326501166984835  0.00219677162536589
##   [4949] 0.00158381435573284 0.000546767339421035
##
##
##      [1] SNORA73, SNORA64, SNORA12, SNORA74, SNORA19, snR65, 5S_rRNA, SNORA4, SNORD11, SN
##      [2]
##      [3]
##      [4]
##      [5]
##      ...
##   [4945]
##   [4946]
##   [4947]
##   [4948]
##   [4949]
##   -------
##   seqinfo: 23 sequences from an unspecified genome; no seqlengths
```

DMRs are ranked by Fisher's multiple comparison statistic, but `Stouffer` scores and the harmonic mean of the individual component FDRs (`HMFDR`) are also given in this object as alternative options for ranking DMR significance.

We can then pass this GRanges object to `DMR.plot()`, which uses the `Gviz` package as a backend for contextualising each DMR.

```
groups <- c(CD8T="magenta", CD4T="forestgreen")
cols <- groups[as.character(type)]
cols

##           CD4T           CD8T           CD8T           CD4T           CD4T
## "forestgreen"      "magenta"      "magenta" "forestgreen" "forestgreen"
##           CD8T           CD8T           CD8T           CD8T           CD4T
##      "magenta"      "magenta"      "magenta"      "magenta" "forestgreen"
##           CD4T           CD4T
## "forestgreen" "forestgreen"

DMR.plot(ranges=results.ranges, dmr=1, CpGs=getBeta(tcell), what="Beta",
         arraytype = "EPIC", phen.col=cols, genome="hg19")
```



Consonant with the expected biology, our top DMR shows the CD8+ T cells hypomethylated across parts of the CD8A locus. The two distinct hypomethylated sections have been merged because they are less than 1000 bp apart -

specified by `lambda` in the call to `dmrcate()`. To call these as separate DMRs, make `lambda` smaller.

Lastly, we would like to do a gene ontology test on our DMRs; this is made possible by the `goregion()` function in the `missMethyl` package. We will take the top 100 DMRs for this enrichment test.

```
library(missMethyl)
enrichment_GO <- goregion(results.ranges[1:100], all.cpg = rownames(tcell),
                          collection = "GO", array.type = "EPIC")
enrichment_GO <- enrichment_GO[order(enrichment_GO$P.DE),]
head(enrichment_GO, 10)
```

```
##            ONTOLOGY                                     TERM    N DE         P.DE
## GO:0046649       BP                  lymphocyte activation  733 20 2.180442e-10
## GO:0001775       BP                       cell activation 1440 25 2.238168e-08
## GO:0098552       CC                    side of membrane  587 15 1.216611e-07
## GO:0002682       BP  regulation of immune system process 1659 25 1.417994e-07
## GO:0045321       BP                  leukocyte activation 1284 22 1.523709e-07
## GO:0042110       BP                     T cell activation  464 14 1.707874e-07
## GO:0030098       BP           lymphocyte differentiation  353 12 4.737771e-07
## GO:0009897       CC     external side of plasma membrane  393 11 6.017648e-07
## GO:0045785       BP positive regulation of cell adhesion  403 13 8.606610e-07
## GO:0042287       MF                   MHC protein binding   40  5 9.180044e-07
##                       FDR
## GO:0046649 4.990597e-06
## GO:0001775 2.561359e-04
## GO:0098552 6.514969e-04
## GO:0002682 6.514969e-04
## GO:0045321 6.514969e-04
## GO:0042110 6.514969e-04
## GO:0030098 1.549116e-03
## GO:0009897 1.721649e-03
## GO:0045785 2.101129e-03
## GO:0042287 2.101129e-03
```

From this enrichment test we can see the most enriched terms are germane to the contrast at hand, including lymphocyte activation and differentiation, T cell activation and MHC protein binding.

Bisulfite sequencing workflow

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-proportion 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 [5].

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[6], imported using ExperimentHub. First, we will import the data:

```
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:

```
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",]
bis_1072

## 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 is 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))
```

```r
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
## 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"
```

```r
#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       1       0       0       0
## 13       0       0       0       0       0       0       1       0       0
## 14       0       0       0       0       0       0       1       0       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       1
## 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
## 30       0       0       0       0       0       0       0       0       0
##    Sample10 Sample11 Sample12 Sample13 Sample14 Sample15 Intercept Fat_Treg
## 1         0        0        0        0        0        0         1        1
## 2         0        0        0        0        0        0         0        0
## 3         0        0        0        0        0        0         1        1
## 4         0        0        0        0        0        0         0        0
## 5         0        0        0        0        0        0         1        1
## 6         0        0        0        0        0        0         0        0
## 7         0        0        0        0        0        0         1        0
## 8         0        0        0        0        0        0         0        0
## 9         0        0        0        0        0        0         1        0
## 10        0        0        0        0        0        0         0        0
## 11        0        0        0        0        0        0         1        0
## 12        0        0        0        0        0        0         0        0
## 13        0        0        0        0        0        0         1        0
## 14        0        0        0        0        0        0         0        0
## 15        0        0        0        0        0        0         1        0
## 16        0        0        0        0        0        0         0        0
## 17        0        0        0        0        0        0         1        0
## 18        0        0        0        0        0        0         0        0
## 19        1        0        0        0        0        0         1        0
## 20        1        0        0        0        0        0         0        0
## 21        0        1        0        0        0        0         1        0
## 22        0        1        0        0        0        0         0        0
## 23        0        0        1        0        0        0         1        0
## 24        0        0        1        0        0        0         0        0
## 25        0        0        0        1        0        0         1        0
## 26        0        0        0        1        0        0         0        0
## 27        0        0        0        0        1        0         1        0
## 28        0        0        0        0        1        0         0        0
## 29        0        0        0        0        0        1         1        0
```

```
## 30        0        0        0        0        0        1        0        0
##    Lymph_N_Tcon Lymph_N_Treg Skin_Treg
## 1             0            0         0
## 2             0            0         0
## 3             0            0         0
## 4             0            0         0
## 5             0            0         0
## 6             0            0         0
## 7             0            0         0
## 8             0            0         0
## 9             0            0         0
## 10            0            0         0
## 11            0            0         0
## 12            0            0         0
## 13            0            0         1
## 14            0            0         0
## 15            0            0         1
## 16            0            0         0
## 17            0            0         1
## 18            0            0         0
## 19            1            0         0
## 20            0            0         0
## 21            1            0         0
## 22            0            0         0
## 23            1            0         0
## 24            0            0         0
## 25            0            1         0
## 26            0            0         0
## 27            0            1         0
## 28            0            0         0
## 29            0            1         0
## 30            0            0         0
```

Just like for `cpg.annotate()`, we can specify a contrast matrix to find our comparisons of interest.

```
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
##   Intercept              0         0         0         0
##   Fat_Treg               0         1         0         1
##   Lymph_N_Tcon          -1         0         0         0
##   Lymph_N_Treg           1        -1        -1         0
##   Skin_Treg              0         0         1        -1
```

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:

```
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()`:

```
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")
```

```
## snapshotDate():  2019-10-22
## see ?DMRcatedata and browseVignettes('DMRcatedata') for documentation
## loading from cache
```

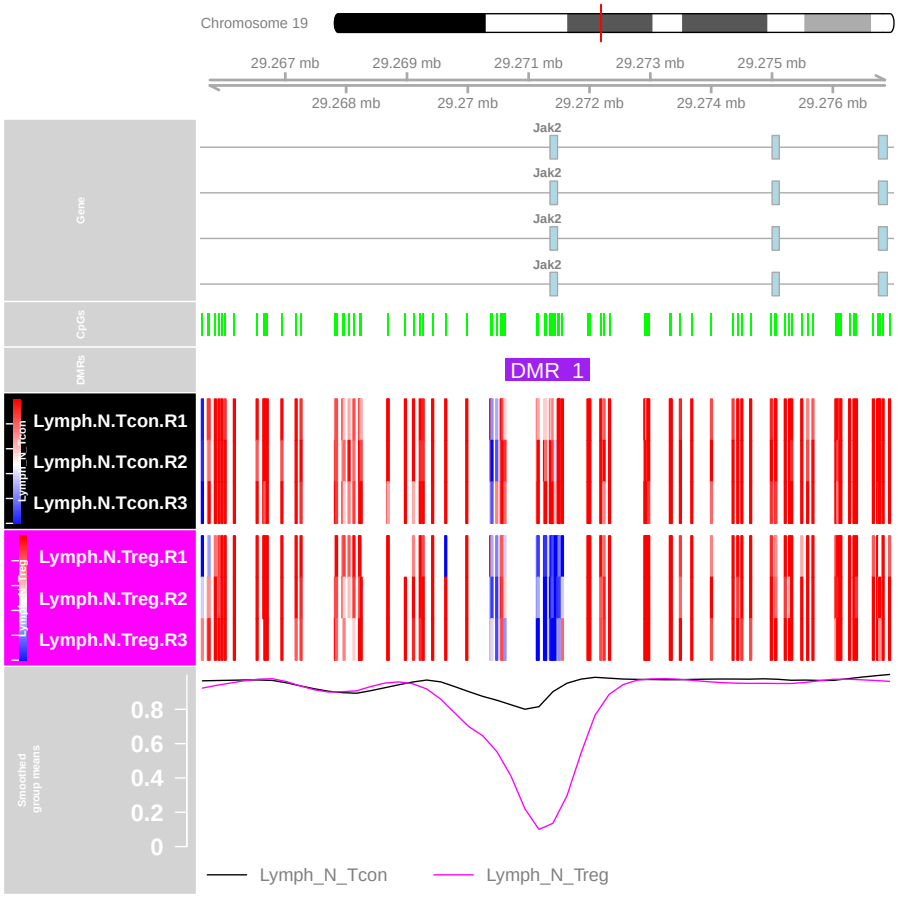
```
treg_vs_tcon.ranges
```

```
## GRanges object with 9 ranges and 8 metadata columns:
##       seqnames            ranges strand |   no.cpgs     min_smoothed_fdr
##          <Rle>         <IRanges>  <Rle> | <integer>           <numeric>
##   [1]    chr19 29270611-29272005      * |        16 4.32351382071251e-94
##   [2]    chr19 26683453-26684174      * |        12 1.77927194052734e-57
##   [3]    chr19 32276886-32278089      * |        13 1.74619734491989e-56
##   [4]    chr19 29374953-29375393      * |        12 1.48256678096532e-54
##   [5]    chr19 36378257-36379597      * |        27 1.53747431922626e-76
##   [6]    chr19 46653280-46654180      * |        19 3.94008431277526e-59
##   [7]    chr19 57092365-57092646      * |        10 3.80467545599821e-36
##   [8]    chr19 40808208-40809554      * |        26 3.43872692336671e-63
##   [9]    chr19 41874401-41874895      * |        22 2.75828631295851e-39
##                Stouffer               HMFDR               Fisher
##               <numeric>           <numeric>            <numeric>
##   [1]                 1  0.0151786468767818 2.14645276178178e-08
##   [2]                 1 0.00787739676169919 0.000128162777894713
##   [3]                 1  0.0446758599711491 0.000150766828389467
##   [4]                 1   0.028226547829304  0.00241190715810346
##   [5]                 1  0.0482585210991692  0.00725026464290328
##   [6]                 1  0.0512002475534357   0.0452566259528858
##   [7] 0.139493604737052  0.0711192891942912   0.0639020544512846
##   [8]                 1   0.180257052466182    0.305279374376655
##   [9]                 1   0.185853374084263    0.690216893652041
##                 maxdiff            meandiff overlapping.genes
##               <numeric>           <numeric>       <character>
##   [1] -6.40482000070317 -4.22352877428813              Jak2
##   [2]  -6.4032835829523 -3.53692271812483           Smarca2
##   [3]  5.81469634649427  3.93201028394053             Sgms1
##   [4] -6.10902321908482 -3.02082671194158 Cd274, AC119228.1
##   [5] -6.09624814631612  -3.0355027665284             Pcgf5
##   [6]   5.1838807268792  2.93151799486439             Wbp1l
```

```
##   [7] -4.67644961940502 -3.36472486679425            Ablim1
##   [8] -4.83855429451068  -3.0749438164254            Cc2d2b
##   [9]  4.57010787202641  2.56520038289291             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:

```
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.

```
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.
```

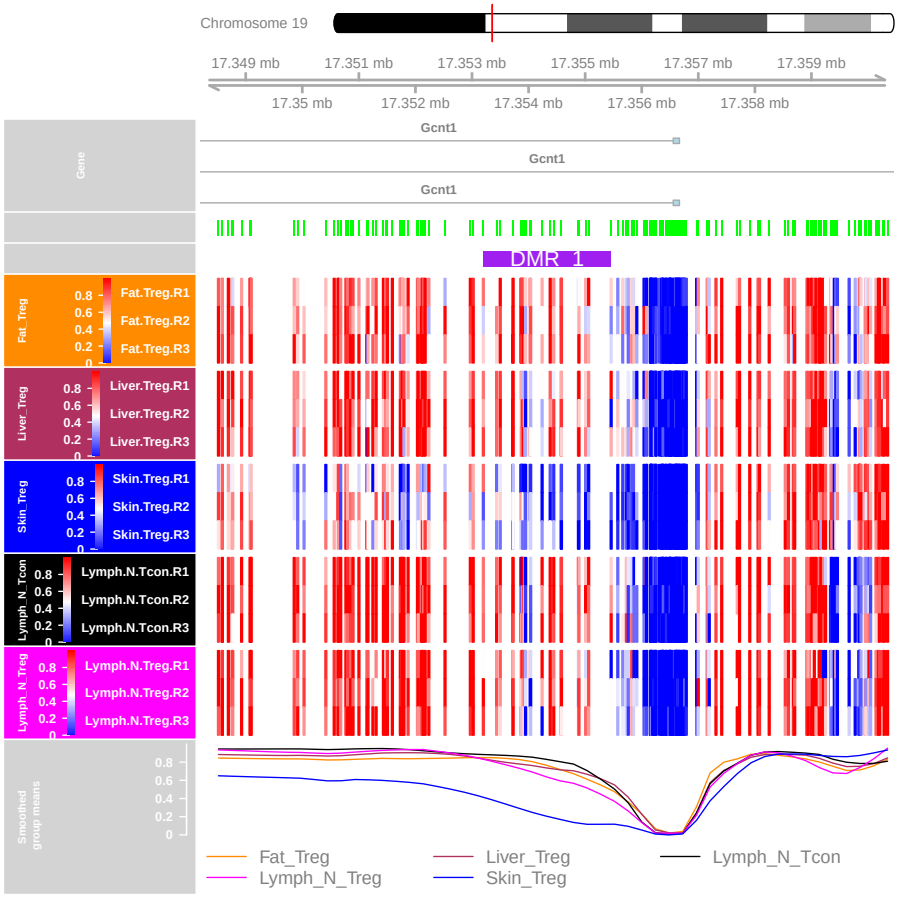
Because this comparison is a bit more subtle, there are very few significantly differential CpGs at this threshold. So we can use `changeFDR()` to relax the FDR to 0.25, taking into account that there is an increased risk of false positives.

```
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")

## snapshotDate():  2019-10-22
## 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 section of *Gcnt1* is hypomethylated not only relative to fat, but to the other tissues as well.

As an alternative to `limma`, there is also the option of taking CpG-level differential statistics using `DSS::DMLtest()` or `DSS::DMLtest.multiFactor()`. There is no need to pass arguments such as `design`, `coef`, etc. to `sequencing.annotate()` in this case since we do this outside of the function. `fdr`, however, must be specified. For example:

```
library(DSS)
DMLfit <- DMLfit.multiFactor(bis_1072, design=data.frame(tissue=tissue), formula=~tissue)

## Fitting DML model for CpG site: 100000 , 200000 , 300000 , 400000 , 500000 ,

DSS_treg.vs.tcon <- DMLtest.multiFactor(DMLfit, Contrast=matrix(c(0, 0, -1, 1, 0)))
#Make sure to filter out all sites where the test statistic is NA
DSS_treg.vs.tcon <- DSS_treg.vs.tcon[!is.na(DSS_treg.vs.tcon$stat),]
```

```
seq_annot <- sequencing.annotate(obj=DSS_treg.vs.tcon, fdr=0.05)
seq_annot

## CpGannotated object describing 544489 CpG sites, with independent
## CpG threshold indexed at fdr=0.05 and 450 significant CpG sites.

dmrcate.res <- dmrcate(seq_annot, C=2, min.cpgs = 5)
DSS.treg_vs_tcon.ranges <- extractRanges(dmrcate.res, genome="mm10")

findOverlaps(treg_vs_tcon.ranges, DSS.treg_vs_tcon.ranges)

## Hits object with 9 hits and 0 metadata columns:
##       queryHits subjectHits
##       <integer>   <integer>
##   [1]         1           1
##   [2]         2           3
##   [3]         3           5
##   [4]         4           9
##   [5]         5           2
##   [6]         6          15
##   [7]         7          26
##   [8]         8          18
##   [9]         9          24
##   -------
##   queryLength: 9 / subjectLength: 30
```

All of the 9 DMRs found using results from `limma` are also found using `DSS::DMLtest.multiFactor()`, with an extra 21 DMRs found by the latter at the same FDR. This suggests that DMLtest.multiFactor() is a little more permissive in calling differential methylation.

```
sessionInfo()

## R version 3.6.2 (2019-12-12)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.3 LTS
##
## Matrix products: default
## BLAS:   /home/biocbuild/bbs-3.10-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.10-bioc/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        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
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
##  [1] splines   stats4    parallel  stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] DSS_2.34.0
##  [2] bsseq_1.22.0
##  [3] tissueTreg_1.6.0
##  [4] missMethyl_1.20.2
##  [5] DMRcatedata_2.2.0
##  [6] IlluminaHumanMethylationEPICmanifest_0.3.0
##  [7] FlowSorted.Blood.EPIC_1.4.1
##  [8] IlluminaHumanMethylationEPICanno.ilm10b4.hg19_0.6.0
##  [9] nlme_3.1-143
## [10] quadprog_1.5-8
## [11] genefilter_1.68.0
## [12] ExperimentHub_1.12.0
## [13] AnnotationHub_2.18.0
## [14] BiocFileCache_1.10.2
## [15] dbplyr_1.4.2
## [16] DMRcate_2.0.7
## [17] minfi_1.32.0
## [18] bumphunter_1.28.0
## [19] locfit_1.5-9.1
## [20] iterators_1.0.12
## [21] foreach_1.4.7
## [22] Biostrings_2.54.0
## [23] XVector_0.26.0
## [24] SummarizedExperiment_1.16.1
## [25] DelayedArray_0.12.2
## [26] BiocParallel_1.20.1
## [27] matrixStats_0.55.0
## [28] Biobase_2.46.0
## [29] GenomicRanges_1.38.0
## [30] GenomeInfoDb_1.22.0
## [31] IRanges_2.20.1
## [32] S4Vectors_0.24.1
## [33] BiocGenerics_0.32.0
##
## loaded via a namespace (and not attached):
##   [1] R.utils_2.9.2
##   [2] tidyselect_0.2.5
```

```
##   [3] RSQLite_2.2.0
##   [4] AnnotationDbi_1.48.0
##   [5] htmlwidgets_1.5.1
##   [6] grid_3.6.2
##   [7] munsell_0.5.0
##   [8] codetools_0.2-16
##   [9] preprocessCore_1.48.0
##  [10] statmod_1.4.33
##  [11] withr_2.1.2
##  [12] colorspace_1.4-1
##  [13] highr_0.8
##  [14] knitr_1.26
##  [15] rstudioapi_0.10
##  [16] GenomeInfoDbData_1.2.2
##  [17] bit64_0.9-7
##  [18] rhdf5_2.30.1
##  [19] vctrs_0.2.1
##  [20] xfun_0.11
##  [21] biovizBase_1.34.1
##  [22] R6_2.4.1
##  [23] illuminaio_0.28.0
##  [24] AnnotationFilter_1.10.0
##  [25] bitops_1.0-6
##  [26] reshape_0.8.8
##  [27] assertthat_0.2.1
##  [28] promises_1.1.0
##  [29] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.6.0
##  [30] scales_1.1.0
##  [31] nnet_7.3-12
##  [32] gtable_0.3.0
##  [33] methylumi_2.32.0
##  [34] ensembldb_2.10.2
##  [35] rlang_0.4.2
##  [36] zeallot_0.1.0
##  [37] rtracklayer_1.46.0
##  [38] lazyeval_0.2.2
##  [39] acepack_1.4.1
##  [40] GEOquery_2.54.1
##  [41] dichromat_2.0-0
##  [42] checkmate_1.9.4
##  [43] yaml_2.2.0
##  [44] BiocManager_1.30.10
##  [45] GenomicFeatures_1.38.0
##  [46] backports_1.1.5
##  [47] httpuv_1.5.2
```

```
##  [48] Hmisc_4.3-0
##  [49] tools_3.6.2
##  [50] nor1mix_1.3-0
##  [51] ggplot2_3.2.1
##  [52] RColorBrewer_1.1-2
##  [53] siggenes_1.60.0
##  [54] Rcpp_1.0.3
##  [55] plyr_1.8.5
##  [56] base64enc_0.1-3
##  [57] progress_1.2.2
##  [58] zlibbioc_1.32.0
##  [59] purrr_0.3.3
##  [60] RCurl_1.95-4.12
##  [61] BiasedUrn_1.07
##  [62] prettyunits_1.1.0
##  [63] rpart_4.1-15
##  [64] openssl_1.4.1
##  [65] cluster_2.1.0
##  [66] magrittr_1.5
##  [67] data.table_1.12.8
##  [68] ProtGenerics_1.18.0
##  [69] mime_0.8
##  [70] hms_0.5.3
##  [71] evaluate_0.14
##  [72] xtable_1.8-4
##  [73] XML_3.98-1.20
##  [74] jpeg_0.1-8.1
##  [75] readxl_1.3.1
##  [76] mclust_5.4.5
##  [77] gridExtra_2.3
##  [78] compiler_3.6.2
##  [79] biomaRt_2.42.0
##  [80] tibble_2.1.3
##  [81] crayon_1.3.4
##  [82] R.oo_1.23.0
##  [83] htmltools_0.4.0
##  [84] later_1.0.0
##  [85] Formula_1.2-3
##  [86] tidyr_1.0.0
##  [87] DBI_1.1.0
##  [88] MASS_7.3-51.5
##  [89] rappdirs_0.3.1
##  [90] Matrix_1.2-18
##  [91] readr_1.3.1
##  [92] permute_0.9-5
```

```
##  [93] R.methodsS3_1.7.1
##  [94] Gviz_1.30.0
##  [95] pkgconfig_2.0.3
##  [96] GenomicAlignments_1.22.1
##  [97] registry_0.5-1
##  [98] IlluminaHumanMethylation450kmanifest_0.4.0
##  [99] foreign_0.8-74
## [100] xml2_1.2.2
## [101] annotate_1.64.0
## [102] rngtools_1.4
## [103] pkgmaker_0.27
## [104] multtest_2.42.0
## [105] beanplot_1.2
## [106] ruv_0.9.7.1
## [107] bibtex_0.4.2.2
## [108] doRNG_1.7.1
## [109] scrime_1.3.5
## [110] stringr_1.4.0
## [111] VariantAnnotation_1.32.0
## [112] digest_0.6.23
## [113] cellranger_1.1.0
## [114] base64_2.0
## [115] htmlTable_1.13.3
## [116] edgeR_3.28.0
## [117] DelayedMatrixStats_1.8.0
## [118] curl_4.3
## [119] shiny_1.4.0
## [120] Rsamtools_2.2.1
## [121] gtools_3.8.1
## [122] lifecycle_0.1.0
## [123] Rhdf5lib_1.8.0
## [124] askpass_1.1
## [125] limma_3.42.0
## [126] BSgenome_1.54.0
## [127] pillar_1.4.3
## [128] lattice_0.20-38
## [129] fastmap_1.0.1
## [130] httr_1.4.1
## [131] survival_3.1-8
## [132] GO.db_3.10.0
## [133] interactiveDisplayBase_1.24.0
## [134] glue_1.3.1
## [135] png_0.1-7
## [136] BiocVersion_3.10.1
## [137] bit_1.1-14
```

```
## [138] stringi_1.4.4
## [139] HDF5Array_1.14.1
## [140] blob_1.2.0
## [141] org.Hs.eg.db_3.10.0
## [142] latticeExtra_0.6-29
## [143] memoise_1.1.0
## [144] dplyr_0.8.3
```

References

[1] Pidsley R, Zotenko E, Peters TJ, Lawrence MG, Risbridger GP, Molloy P, Van Dijk S, Muhlhausler B, Stirzaker C, Clark SJ. Critical evaluation of the Illumina MethylationEPIC BeadChip microarray for whole-genome DNA methylation profiling. *Genome Biology*. 2016 17(1), 208.

[2] Chen YA, Lemire M, Choufani S, Butcher DT, Grafodatskaya D, Zanke BW, Gallinger S, Hudson TJ, Weksberg R. Discovery of cross-reactive probes and polymorphic CpGs in the Illumina Infinium HumanMethylation450 microarray. *Epigenetics*. 2013 Jan 11;8(2).

[3] Heyn H, Li N, Ferreira HJ, Moran S, Pisano DG, Gomez A, Esteller M. Distinct DNA methylomes of newborns and centenarians. *Proceedings of the National Academy of Sciences*. 2012 **109**(26), 10522-7.

[4] Satterthwaite FE. An Approximate Distribution of Estimates of Variance Components., *Biometrics Bulletin*. 1946 **2**: 110-114

[5] Chen Y, Pal B, Visvader JE, Smyth GK. Differential methylation analysis of reduced representation bisulfite sequencing experiments using edgeR. *F1000Research*. 2017 **6**, 2055.

[6] Delacher M, Imbusch CD, Weichenhan D, Breiling A, Hotz-Wagenblatt A, Trager U, ... Feuerer M. (2017). Genome-wide DNA-methylation landscape defines specialization of regulatory T cells in tissues. *Nature Immunology*. 2017 **18**(10), 1160-1172.

[7] Feng H, Conneely KN, Wu H. A Bayesian hierarchical model to detect differentially methylated loci from single nucleotide resolution sequencing data. *Nucleic Acids Research*. 2014 **42**(8), e69.

[8] Park Y, and Wu H. Differential methylation analysis for BS-seq data under general experimental design. *Bioinformatics*. 2016 **32**(10), 1446-1453.