The DMRcate package user's guide

Peters TJ, Buckley MJ, Statham A, Pidsley R, Clark SJ, Molloy PL

March 30, 2016

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.

```
source("http://bioconductor.org/biocLite.R")
biocLite("DMRcate")
```

Load DMRcate into the workspace:

```
library(DMRcate)
```

Illumina®Array Workflow

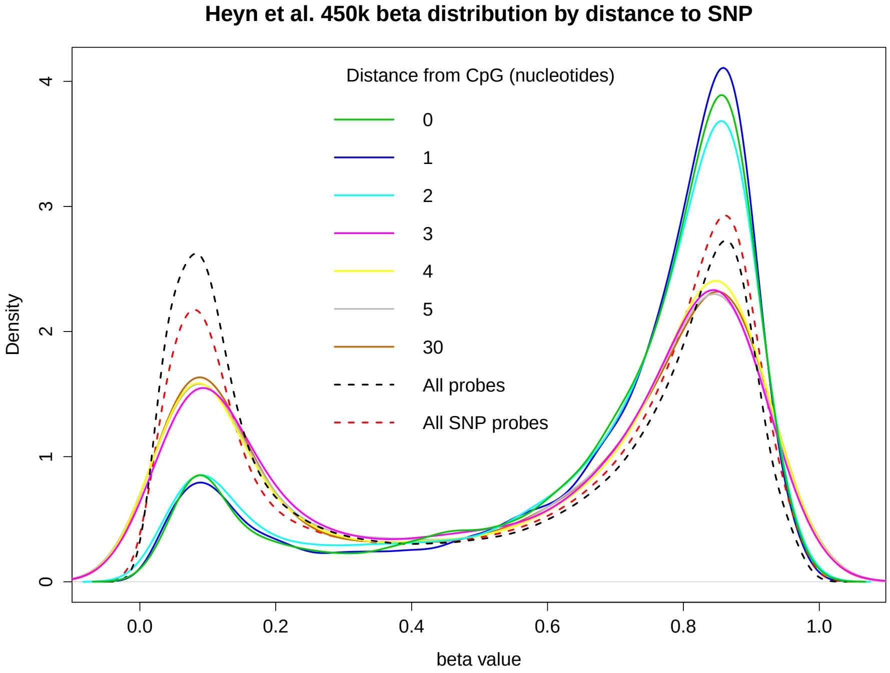
We now can load in the test data set of beta values. We assume at this point that normalisation and filtering out bad-quality probes via their detection p-values have already been done. Many packages are available for these purposes, including minfi, wateRmelon and methylumi. M-values (logit-transform of beta) are preferable to beta values for significance testing via limma because of increased sensitivity, but we will retain the beta matrix for visualisation purposes later on.

The TCGA (Cancer Genome Atlas - colorectal cancer) data in myBetas only comes from chromosome 20, but DMRcate will have no problem taking in the approximately half million probes as input for this pipeline either.

```
data(dmrcatedata)
myMs <- logit2(myBetas)
```

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]. In particular, probes that are 0, 1, or 2 nucleotides from the methylcytosine of

Figure 1: Beta distribution of 450K probes from publically available data from blood samples of healthy individuals [2] 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.



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 30,000 [1] cross-reactive probes, so as to reduce confounding. Here we use Illumina®'s database of approximately 150,000 potentially SNP-confounded probes, and an internally-loaded dataset of the probes from [1], to filter these probes out. About 600 are removed from our M-matrix of approximately 10,000:

```
nrow(illuminaSNPs)

## [1] 153113

nrow(myMs)

## [1] 10042

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

## [1] 9403
```

Next we want to annotate our matrix of M-values with relevant information. The default is the `ilmn12.hg19` annotation, but this can be substituted for any argument compatible with the interface provided by the `minfi` package. We also use the backbone of the `limma` pipeline for differential array analysis to get t-statistics changes and, optionally, filter probes by their fdr-corrected p-value. Here we have 38 patients with 2 tissue samples each taken from them. We want to compare within patients across tissue samples, so we set up our variables for a standard limma pipeline, and set `coef=39` in `cpg.annotate` since this corresponds to the phenotype comparison in `design`.

```
patient <- factor(sub("-.*", "", colnames(myMs)))
type <- factor(sub(".*-", "", colnames(myMs)))
design <- model.matrix(~patient + type)
myannotation <- cpg.annotate("array", myMs.noSNPs, analysis.type="differential",
    design=design, coef=39)

## Your contrast returned 6101 individually significant probes.  We
recommend the default setting of pcutoff in dmrcate().
## Loading required package:  IlluminaHumanMethylation450kanno.ilmn12.hg19
```

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

For each chromosome, two smoothed estimates are computed: one weighted with `myannotation$stat` and one not, for a null comparison. The two estimates are compared via a Satterthwaite approximation[3], 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 significant probes where the distance to the next consecutive probe is less than `lambda` nucleotides.

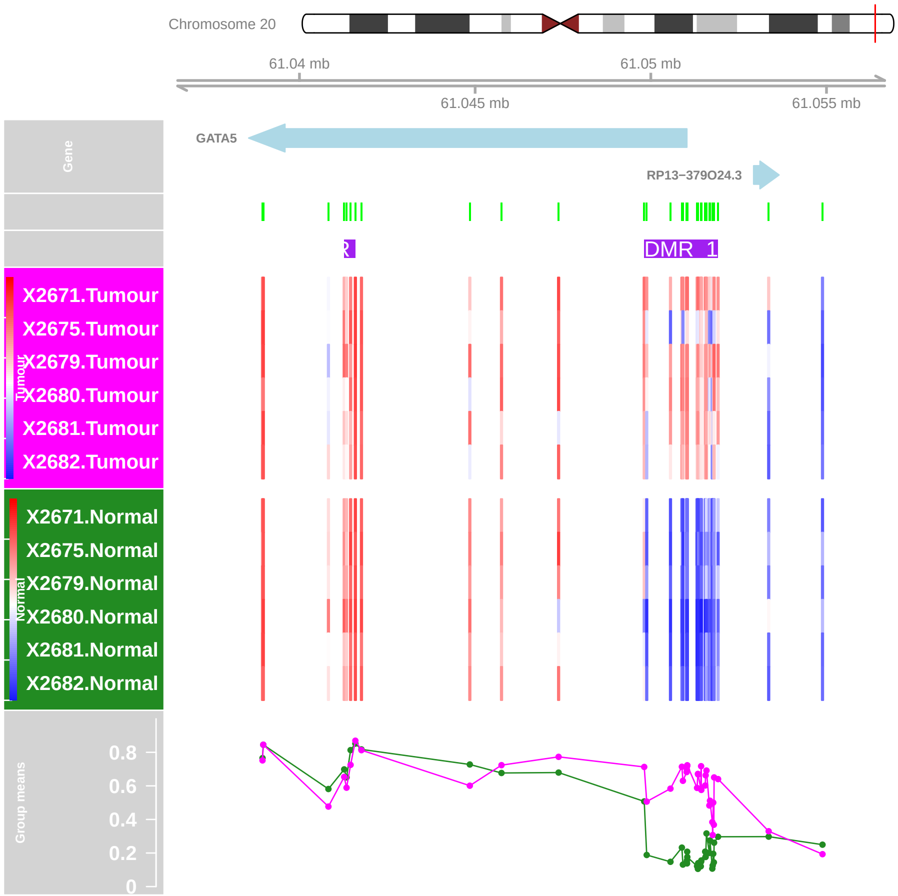
```
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)

## Fitting chr20...
## Demarcating regions...
## Done!
```

We can convert our DMR list to a GRanges object, which uses the `genome` argument to annotate overlapping promoter regions (+/- 2000 bp from TSS). and pass it to DMR.plot, which uses the `Gviz` package as a backend for contextualising each DMR. We'll choose one associated with the GATA5 locus.

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

## GRanges object with 739 ranges and 6 metadata columns:
##                            seqnames               ranges strand   |
##                               <Rle>            <IRanges>  <Rle>   |
##   chr20:61049813-61051915    chr20 [61049813, 61051915]      *   |
##   chr20:57424521-57431303    chr20 [57424521, 57431303]      *   |
##   chr20:24448859-24452131    chr20 [24448859, 24452131]      *   |
##   chr20:21491781-21498921    chr20 [21491781, 21498921]      *   |
##   chr20:61806628-61810795    chr20 [61806628, 61810795]      *   |
##                       ...      ...                  ...    ... ...
##   chr20:40321552-40321839    chr20 [40321552, 40321839]      *   |
##     chr20:3451292-3451627    chr20 [ 3451292,  3451627]      *   |
##   chr20:43729808-43730241    chr20 [43729808, 43730241]      *   |
##   chr20:44541804-44542136    chr20 [44541804, 44542136]      *   |
##     chr20:3214756-3214926    chr20 [ 3214756,  3214926]      *   |
##                              no.cpgs       minfdr      Stouffer
##                            <integer>    <numeric>     <numeric>
##   chr20:61049813-61051915         27  0.000000e+00  0.000000e+00
##   chr20:57424521-57431303         77  0.000000e+00 2.554946e-268
##   chr20:24448859-24452131         21  0.000000e+00 1.343441e-236
##   chr20:21491781-21498921         26 3.322748e-208 4.239591e-231
##   chr20:61806628-61810795         23  0.000000e+00 7.907402e-215
##                       ...        ...           ...           ...
##   chr20:40321552-40321839          3  1.607334e-10   0.002292294
##     chr20:3451292-3451627          8  1.408881e-12   0.003156977
##   chr20:43729808-43730241          9  1.815089e-16   0.019406577
##   chr20:44541804-44542136          2  3.002164e-12   0.019544574
##     chr20:3214756-3214926          4  1.272372e-12   0.148286525
##                              maxbetafc meanbetafc
##                              <numeric>  <numeric>
##   chr20:61049813-61051915   0.4770680 0.35455081
##   chr20:57424521-57431303  -0.2084268 0.07728631
##   chr20:24448859-24452131   0.4263522 0.28291317
##   chr20:21491781-21498921   0.4385002 0.25698803
##   chr20:61806628-61810795   0.4182034 0.24114120
##                       ...         ...        ...
##   chr20:40321552-40321839  0.10782550 0.05362530
##     chr20:3451292-3451627  0.12304730 0.02957047
##   chr20:43729808-43730241 -0.11708998 0.01597976
##   chr20:44541804-44542136  0.04326832 0.01645320
##     chr20:3214756-3214926  0.05158419 0.01150333
##
##
##   chr20:61049813-61051915
##   chr20:57424521-57431303 GNAS-050, GNAS-037, GNAS-058, GNAS-001, GNAS-009, GNAS-049, GNA
##   chr20:24448859-24452131
##   chr20:21491781-21498921
##   chr20:61806628-61810795
##                       ...
##   chr20:40321552-40321839
##     chr20:3451292-3451627
##   chr20:43729808-43730241
##   chr20:44541804-44542136
##     chr20:3214756-3214926
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Now we can plot a significant DMR. We use functionality from the `Gviz` package as a backend for this purpose. We will plot a DMR associated with the GATA5 locus for the first 6 tumour/normal matched pairs.

```
groups <- c(Tumour="magenta", Normal="forestgreen")
cols <- groups[as.character(type)]
samps <- c(1:6, 38+(1:6))
DMR.plot(ranges=results.ranges, dmr=1, CpGs=myBetas, phen.col=cols, genome="hg19",
         samps=samps)
```



WGBS Workflow

WGBS is a little different. Because the data is represented binomially (that is, by the number of methylated reads followed by the total coverage for that particular CpG site) rather than the continuous distribution afforded by array intensities, we must model the differential methylation signal in a way that respects this. A popular way of doing this is via the beta-binomial distribution. We currently recommend using the method implemented in the DSS package[4], because it uses dispersion shrinkage via a Bayesian framework - similar to `edgeR` for RNA-Seq count data.

The `CpGs` GRanges object contains simulated data for 3 Treatment vs. 3 Control samples for 10^5 CpG sites, generated by WGBSSuite[5].

```
CpGs

## GRanges object with 100000 ranges and 12 metadata columns:
```

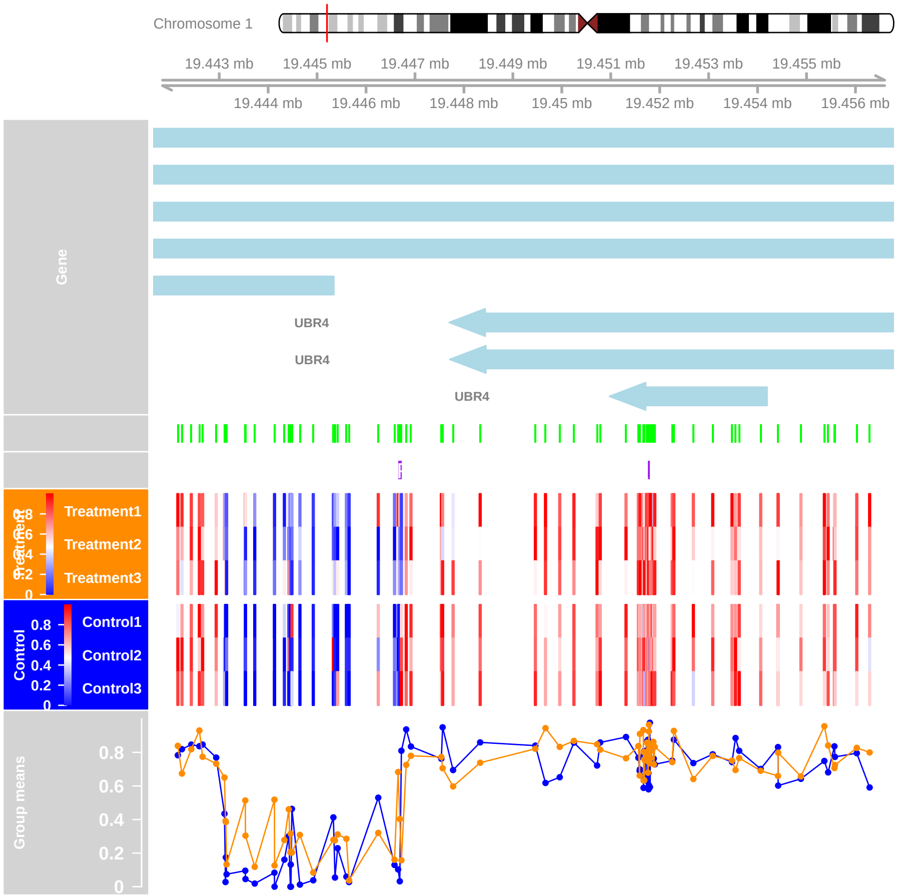
```
##            seqnames               ranges strand   | Treatment1.C
##               <Rle>            <IRanges>  <Rle>   |    <integer>
##        [1]     chr1           [  1,   1]      *   |           11
##        [2]     chr1           [ 54,  54]      *   |            9
##        [3]     chr1           [ 58,  58]      *   |           14
##        [4]     chr1           [320, 320]      *   |           12
##        [5]     chr1           [325, 325]      *   |           10
##        ...      ...                  ...    ... ...          ...
##    [99996]     chr1 [19705499, 19705499]      *   |           13
##    [99997]     chr1 [19705511, 19705511]      *   |           11
##    [99998]     chr1 [19705521, 19705521]      *   |           15
##    [99999]     chr1 [19705567, 19705567]      *   |           19
##   [100000]     chr1 [19705760, 19705760]      *   |           11
##            Treatment1.cov Treatment2.C Treatment2.cov Treatment3.C
##                 <integer>    <integer>      <integer>    <integer>
##        [1]             13            9             14           16
##        [2]             15           16             26           18
##        [3]             20           19             20           19
##        [4]             15           14             14           17
##        [5]             19           13             18           14
##        ...            ...          ...            ...          ...
##    [99996]             15           13             13           12
##    [99997]             13           16             19           16
##    [99998]             15           13             13           15
##    [99999]             20           11             17           18
##   [100000]             21           14             14           21
##            Treatment3.cov Control1.C Control1.cov Control2.C Control2.cov
##                 <integer>  <integer>    <integer>  <integer>    <integer>
##        [1]             19         11           15         16           23
##        [2]             20         17           18         10           17
##        [3]             27         16           16         12           14
##        [4]             20         13           25         15           21
##        [5]             22          5           14         16           23
##        ...            ...        ...          ...        ...          ...
##    [99996]             20         13           32         12           20
##    [99997]             19         12           27         14           22
##    [99998]             17         16           17          8           16
##    [99999]             20         18           24         18           20
##   [100000]             28         17           21         12           17
##            Control3.C Control3.cov
##             <integer>    <integer>
##        [1]         11           14
##        [2]         19           21
##        [3]         15           19
##        [4]         18           22
##          [5]          20           26
##          ...         ...          ...
##      [99996]          15           15
##      [99997]          11           19
##      [99998]          22           22
##      [99999]          16           17
##     [100000]          12           25
##     -------
##     seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Note the structure of the metadata columns for this object: samples come in column pairs, with the number of methylated reads followed by the total coverage for that CpG site. Naturally, ⟨sample⟩.cov must always be ≥ ⟨sample⟩.C. This structure must be in place in order for downstream tasks such as `DMR.plot()` to be run. Using this structure, we can now extract the methylation and coverage counts, and prepare a `bsseq` object as we would for `DSS`, and call differentially methylated CpG sites.

```
meth <- as.data.frame(CpGs)[,c(1:2, grep(".C$", colnames(as.data.frame(CpGs))))]
coverage <- as.data.frame(CpGs)[,c(1:2, grep(".cov$", colnames(as.data.frame(CpGs))))]

treat1 <- data.frame(chr=coverage$seqnames, pos=coverage$start,
                     N=coverage$Treatment1.cov, X=meth$Treatment1.C)

treat2 <- data.frame(chr=coverage$seqnames, pos=coverage$start,
                     N=coverage$Treatment2.cov, X=meth$Treatment2.C)

treat3 <- data.frame(chr=coverage$seqnames, pos=coverage$start,
                     N=coverage$Treatment3.cov, X=meth$Treatment3.C)

ctrl1 <- data.frame(chr=coverage$seqnames, pos=coverage$start,
                     N=coverage$Control1.cov, X=meth$Control1.C)

ctrl2 <- data.frame(chr=coverage$seqnames, pos=coverage$start,
                     N=coverage$Control2.cov, X=meth$Control2.C)

ctrl3 <- data.frame(chr=coverage$seqnames, pos=coverage$start,
                     N=coverage$Control3.cov, X=meth$Control3.C)

samples <- list(treat1, treat2, treat3, ctrl1, ctrl2, ctrl3)
sampnames <- sub("\\..*", "", colnames(meth))[-c(1:2)]

obj_bsseq <- makeBSseqData(samples, sampnames)
DSSres <- DMLtest(obj_bsseq, group1=sampnames[1:3], group2=sampnames[4:6], smoothing=FALSE)

## Estimating dispersion for each CpG site, this will take a while ...
```

We can now enter `DSSres` into the `DMRcate` workflow. Because CpGs are much closer together than they are when represented by Illumina arrays, we will shrink the kernel size by increasing `C`. We will also run this in serial (`mc.cores=1`). If you want to run `dmrcate()` in parallel (1 chromosome per core), please check your processor specifications by running `detectCores()`.

```
wgbsannot <- cpg.annotate("sequencing", DSSres)
wgbs.DMRs <- dmrcate(wgbsannot, lambda = 1000, C = 50, pcutoff = 0.05, mc.cores = 1)

## Fitting chr1...
## Demarcating regions...
## Done!

wgbs.ranges <- extractRanges(wgbs.DMRs, genome = "hg19")
groups <- c(Treatment="darkorange", Control="blue")
cols <- groups[sub("[0-9]", "", sampnames)]
DMR.plot(ranges=wgbs.ranges, dmr=1, CpGs=CpGs, phen.col=cols, genome="hg19")
```



```
sessionInfo()

## R version 3.2.4 Revised (2016-03-16 r70336)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## 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] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.2.1
##  [2] DMRcate_1.6.53
##  [3] DMRcatedata_1.6.1
##  [4] DSS_2.10.0
##  [5] bsseq_1.6.0
##  [6] minfi_1.16.1
##  [7] bumphunter_1.10.0
##  [8] locfit_1.5-9.1
##  [9] iterators_1.0.8
## [10] foreach_1.4.3
## [11] Biostrings_2.38.4
## [12] XVector_0.10.0
## [13] SummarizedExperiment_1.0.2
## [14] GenomicRanges_1.22.4
## [15] GenomeInfoDb_1.6.3
## [16] IRanges_2.4.8
## [17] S4Vectors_0.8.11
## [18] lattice_0.20-33
## [19] Biobase_2.30.0
## [20] BiocGenerics_0.16.1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-126             bitops_1.0-6
##  [3] matrixStats_0.50.1       RColorBrewer_1.1-2
##  [5] tools_3.2.4              doRNG_1.6
##  [7] nor1mix_1.2-1            rpart_4.1-10
##  [9] Hmisc_3.17-2             DBI_0.3.1
## [11] Gviz_1.14.6              colorspace_1.2-6
## [13] nnet_7.3-12              gridExtra_2.2.1
## [15] base64_1.1               preprocessCore_1.32.0
## [17] chron_2.3-47             formatR_1.3
## [19] pkgmaker_0.22            rtracklayer_1.30.4
## [21] scales_0.4.0             genefilter_1.52.1
## [23] quadprog_1.5-5           stringr_1.0.0
## [25] digest_0.6.9             Rsamtools_1.22.0
## [27] foreign_0.8-66           illuminaio_0.12.0
## [29] siggenes_1.44.0          R.utils_2.2.0
## [31] GEOquery_2.36.0          dichromat_2.0-0
## [33] BSgenome_1.38.0          limma_3.26.9
## [35] highr_0.5.1              RSQLite_1.0.0
## [37] mclust_5.1               BiocParallel_1.4.3
## [39] gtools_3.5.0             acepack_1.3-3.3
## [41] R.oo_1.20.0              VariantAnnotation_1.16.4
## [43] RCurl_1.95-4.8           magrittr_1.5
## [45] Formula_1.2-1            futile.logger_1.4.1
## [47] Rcpp_0.12.4              munsell_0.4.3
## [49] R.methodsS3_1.7.1        stringi_1.0-1
## [51] MASS_7.3-45              zlibbioc_1.16.0
## [53] plyr_1.8.3               grid_3.2.4
## [55] multtest_2.26.0          GenomicFeatures_1.22.13
## [57] annotate_1.48.0          knitr_1.12.3
## [59] beanplot_1.2             igraph_1.0.1
## [61] rngtools_1.2.4           corpcor_1.6.8
## [63] codetools_0.2-14         biomaRt_2.26.1
## [65] mixOmics_5.2.0           futile.options_1.0.0
## [67] XML_3.98-1.4             evaluate_0.8.3
## [69] biovizBase_1.18.0        latticeExtra_0.6-28
## [71] lambda.r_1.1.7           data.table_1.9.6
## [73] gtable_0.2.0             reshape_0.8.5
## [75] ggplot2_2.1.0            xtable_1.8-2
## [77] survival_2.38-3          GenomicAlignments_1.6.3
## [79] AnnotationDbi_1.32.3     registry_0.3
## [81] ellipse_0.3-8            cluster_2.0.3
## [83] rgl_0.95.1441
```

References

[1] 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).

[2] 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.

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

[4] 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.

[5] Rackham, OJL, Dellaportas P, Petretto E, Bottolo, L. WGBSSuite: Simulating Whole Genome Bisulphite Sequencing data and benchmarking differential DNA methylation analysis tools. *Bioinformatics* 2015. (Oxford, England), (March).