The **DMRcate** package user’s guide

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

March 14, 2017

**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.

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

Load **DMRcate** into the workspace:

```r
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) 450K 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.

```r
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, 2]. 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 [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.

Heyn et al. 450k beta distribution by distance to SNP

Distance from CpG (nucleotides)

- 0
- 1
- 2
- 3
- 4
- 5
- 30
- All probes
- All SNP probes

Density

beta value
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 63,000 [1, 2] cross-reactive probes on either 450K and/or EPIC,
so as to reduce confounding. Here we use a combination of in silico analysis
from [1, 2] and Illumina®’s database of approximately 150,000 potentially SNP-
confounded probes, to filter these probes out. About 600 are removed from our
M-matrix of approximately 10,000:

```r
nrow(snpsall)
## [1] 208568
nrow(myMs)
## [1] 10042
myMs.noSNPs <- rmSNPandCH(myMs, dist=2, mafcut=0.05)
nrow(myMs.noSNPs)
## [1] 9382
```

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
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.

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

```r
patient <- factor(sub("\-.*", "", colnames(myMs)))
type <- factor(sub(".*-\", "", colnames(myMs)))
design <- model.matrix(~patient + type)
myannotation <- cpg.annotate("array", myMs.noSNPs, what="M", arraytype = "450K",
  analysis.type="differential", design=design, coef=39)
```

```text
# Loading required package: IlluminaHumanMethylation450kanno.ilmn12.hg19
# Your contrast returned 6091 individually significant probes. We
# recommend the default setting of pcutoff in dmr cate().

# Or, alternatively
grset <- makeGenomicRatioSetFromMatrix(myMs.noSNPs, array = "IlluminaHumanMethylation450k",
  annotation = "ilmn12.hg19", mergeManifest = TRUE,
  what = "M")
```
myannotation <- cpg.annotate("array", grset, analysis.type="differential", design=design, coef=39)

## Your contrast returned 6091 individually significant probes. We recommend the default setting of pcut in dmrcate().

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[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 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 743 ranges and 6 metadata columns:
## seqnames ranges strand | no.cpgs
## chr20:61049813-61051915 chr20 [61049813, 61051915] * | 27
## chr20:57424521-57431303 chr20 [57424521, 57431303] * | 77
## chr20:24448859-24452131 chr20 [24448859, 24452131] * | 21
## chr20:21491781-21498921 chr20 [21491781, 21498921] * | 26
## chr20:61806628-61810795 chr20 [61806628, 61810795] * | 23
## ... ... ... ... ... ... ...
## chr20:3026614-3027467 chr20 [3026614, 3027467] * | 5
## chr20:3451292-3451627 chr20 [3451292, 3451627] * | 8
## chr20:3214756-3214926 chr20 [3214756, 3214926] * | 3
## chr20:43729808-43730241 chr20 [43729808, 43730241] * | 9
## chr20:44541804-44542136 chr20 [44541804, 44542136] * | 2
## minfdr Stouffer maxbetafc
## <numeric> <numeric> <numeric>  
## chr20:61049813-61051915 0.000000e+00 0.000000e+00 0.4770680
## chr20:57424521-57431303 0.000000e+00 2.725637e-268 -0.2084268

4
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.

```r
groups <- c(Tumour="magenta", Normal="forestgreen")
cols <- groups[as.character(type)]
samps <- c(1:6, 38+(1:6))
```
**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[5], 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[6].
CpGs

## GRanges object with 10000 ranges and 12 metadata columns:
##
## seqnames ranges strand | Treatment1.C
## <Rle> <IRanges> <Rle> | <integer>
## [1] chr1 [ 1, 1] * | 11
## [3] chr1 [ 58, 58] * | 14
## ...
## [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.cov Treatment2.cov Treatment3.C
## <integer> <integer> <integer> <integer>
## [1] 13 9 14 16
## [2] 15 16 26 18
## [3] 20 19 20 19
## [4] 19 13 18 14
## [5] ...
## [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
## 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, \langle sample\rangle.cov must always be \geq \langle sample\rangle.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.

```r
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")
Chromosome 1

12.814 mb
12.815 mb
12.816 mb
12.817 mb
12.818 mb
12.819 mb
12.82 mb
12.821 mb
12.822 mb
12.823 mb

Gene

CpGs

DMRs

Treatment

Control

Group means

sessionInfo()

## R version 3.3.3 (2017-03-06)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.2 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
##
## attached base packages:
## [1] splines stats4 parallel stats graphics grDevices utils
## [8] datasets methods base
##
## other attached packages:
## [1] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.6.0
## [2] DMRcate_1.10.8
## [3] DMRcatedata_1.10.1
## [4] DSS_2.14.0
## [5] bsseq_1.10.0
## [7] minfi_1.20.2
## [8] bumphunter_1.14.0
## [9] locfit_1.5-9.1
## [10] iterators_1.0.8
## [11] foreach_1.4.3
## [12] Biostrings_2.42.1
## [13] XVector_0.14.0
## [14] SummarizedExperiment_1.4.0
## [16] GenomeInfoDb_1.10.3
## [17] IRanges_2.8.1
## [18] S4Vectors_0.12.1
## [19] Biobase_2.34.0
## [20] BiocGenerics_0.20.0
##
## loaded via a namespace (and not attached):
## [1] colorspace_1.3-2
## [2] siggenes_1.48.0
## [3] mclust_5.2.3
## [4] biovizBase_1.22.0
## [5] htmlTable_1.9
## [6] base64enc_0.1-3
## [7] dichromat_2.0-0
## [8] base64_2.0
## [9] IlluminaHumanMethylationEPICanno.ilm10b2.hg19_0.6.0
## [10] interactiveDisplayBase_1.12.0
## [11] AnnotationDbi_1.36.2
## [12] codetools_0.2-15
## [13] R.methodsS3_1.7.1
## [14] methylumi_2.20.0
## [15] knitr_1.15.1
## [16] Formula_1.2-1
## [17] Rsamtools_1.26.1
## [18] annotate_1.52.1
## [19] GO.db_3.4.0
## [20] cluster_2.0.5
## [21] R.oo_1.21.0
## [22] shiny_1.0.0
## [23] httr_1.2.1
## [24] backports_1.0.5
## [25] assertthat_0.1
## [26] Matrix_1.2-8
## [27] lazyeval_0.2.0
## [28] acepack_1.4.1
## [29] htmltools_0.3.5
## [30] tools_3.3.3
## [31] gtable_0.2.0
## [32] doRNG_1.6
## [33] Rcpp_0.12.9
## [34] multtest_2.30.0
## [35] preprocessorCore_1.36.0
## [36] nlme_3.1-131
## [37] rtracklayer_1.34.2
## [38] stringr_1.2.0
## [39] mime_0.5
## [40] ensemblDB_1.6.2
## [41] rngtools_1.2.4
## [42] gtools_3.5.0
## [43] statmod_1.4.29
## [44] XML_3.98-1.5
## [45] beanplot_1.2
## [46] org.Hs.eg.db_3.4.0
## [47] AnnotationHub_2.6.4
## [48] zlibbioc_1.20.0
## [49] MASS_7.3-45
## [50] scales_0.4.1
## [51] BSgenome_1.42.0
## [52] VariantAnnotation_1.20.2
## [53] BiocInstaller_1.24.0
## [54] GEOquery_2.40.0
## [55] RColorBrewer_1.1-2
## [56] missMethyl_1.8.0
## [57] yaml_2.1.14
## [58] memoise_1.0.0
## [59] gridExtra_2.2.1
## [60] ggplot2_2.2.1
## [61] pkgmaker_0.22
## [62] biomaRt_2.30.0
## [63] rpart_4.1-10
## [64] reshape_0.8.6
## [65] latticeExtra_0.6-28
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>stringi_1.1.2</td>
<td></td>
</tr>
<tr>
<td>RSQLite_1.1-2</td>
<td></td>
</tr>
<tr>
<td>highr_0.6</td>
<td></td>
</tr>
<tr>
<td>genefilter_1.56.0</td>
<td></td>
</tr>
<tr>
<td>checkmate_1.8.2</td>
<td></td>
</tr>
<tr>
<td>permute_0.9-4</td>
<td></td>
</tr>
<tr>
<td>GenomicFeatures_1.26.3</td>
<td></td>
</tr>
<tr>
<td>BiocParallel_1.8.1</td>
<td></td>
</tr>
<tr>
<td>matrixStats_0.51.0</td>
<td></td>
</tr>
<tr>
<td>bitops_1.0-6</td>
<td></td>
</tr>
<tr>
<td>nor1mix_1.2-2</td>
<td></td>
</tr>
<tr>
<td>evaluate_0.10</td>
<td></td>
</tr>
<tr>
<td>lattice_0.20-34</td>
<td></td>
</tr>
<tr>
<td>ruv_0.9.6</td>
<td></td>
</tr>
<tr>
<td>GenomicAlignments_1.10.0</td>
<td></td>
</tr>
<tr>
<td>htmlwidgets_0.8</td>
<td></td>
</tr>
<tr>
<td>plyr_1.8.4</td>
<td></td>
</tr>
<tr>
<td>magrittr_1.5</td>
<td></td>
</tr>
<tr>
<td>R6_2.2.0</td>
<td></td>
</tr>
<tr>
<td>Hmisc_4.0-2</td>
<td></td>
</tr>
<tr>
<td>IlluminaHumanMethylation450kmanifest_0.4.0</td>
<td></td>
</tr>
<tr>
<td>DBI_0.6</td>
<td></td>
</tr>
<tr>
<td>foreign_0.8-67</td>
<td></td>
</tr>
<tr>
<td>survival_2.40-1</td>
<td></td>
</tr>
<tr>
<td>RCurl_1.95-4.8</td>
<td></td>
</tr>
<tr>
<td>nnet_7.3-12</td>
<td></td>
</tr>
<tr>
<td>tibble_1.2</td>
<td></td>
</tr>
<tr>
<td>grid_3.3.3</td>
<td></td>
</tr>
<tr>
<td>data.table_1.10.4</td>
<td></td>
</tr>
<tr>
<td>digest_0.6.12</td>
<td></td>
</tr>
<tr>
<td>xtable_1.8-2</td>
<td></td>
</tr>
<tr>
<td>httpuv_1.3.3</td>
<td></td>
</tr>
<tr>
<td>R.utils_2.5.0</td>
<td></td>
</tr>
<tr>
<td>illuminaio_0.16.0</td>
<td></td>
</tr>
<tr>
<td>openssl_0.9.6</td>
<td></td>
</tr>
<tr>
<td>munsell_0.4.3</td>
<td></td>
</tr>
<tr>
<td>registry_0.3</td>
<td></td>
</tr>
<tr>
<td>BiasedUrn_1.07</td>
<td></td>
</tr>
<tr>
<td>Gviz_1.18.2</td>
<td></td>
</tr>
<tr>
<td>quadprog_1.5-5</td>
<td></td>
</tr>
</tbody>
</table>
References


