DMRcate for EPICv2

Peters TJ

July 16, 2024

Summary

Worked example to find DMRs from EPICv2 arrays between B cell ALL and T cell ALL samples.

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

The Illumina Infinium HumanMethylationEPIC v2.0 BeadChip (EPICv2) extends genomic coverage to more than 920,000 CpG sites. One of the new characteristics of this array compared to previous versions is that it contains instances of multiple probes (or "replicates") that map to the same CpG site. This is potentially useful for performance testing of probes, but for DMRcate we need a 1-to-1 mapping of probe to CpG site, otherwise the kernel will be biased towards those sites with more replicates. This vignette will show how to pare down an EPICv2 dataset to a suitable format for DMR calling. We use the AnnotationHub package EPICv2manifest[1] as a backend for this purpose.

```
library(DMRcate)
```

We use a subset of samples kindly taken from Noguera-Castells *et al.* (2023)[2], for finding DMRs between T cell acute lymphoblastic leukaemia (TALL) and B cell acute lymphoblastic leukaemia (BALL). We load the beta matrix from ExperimentHub:

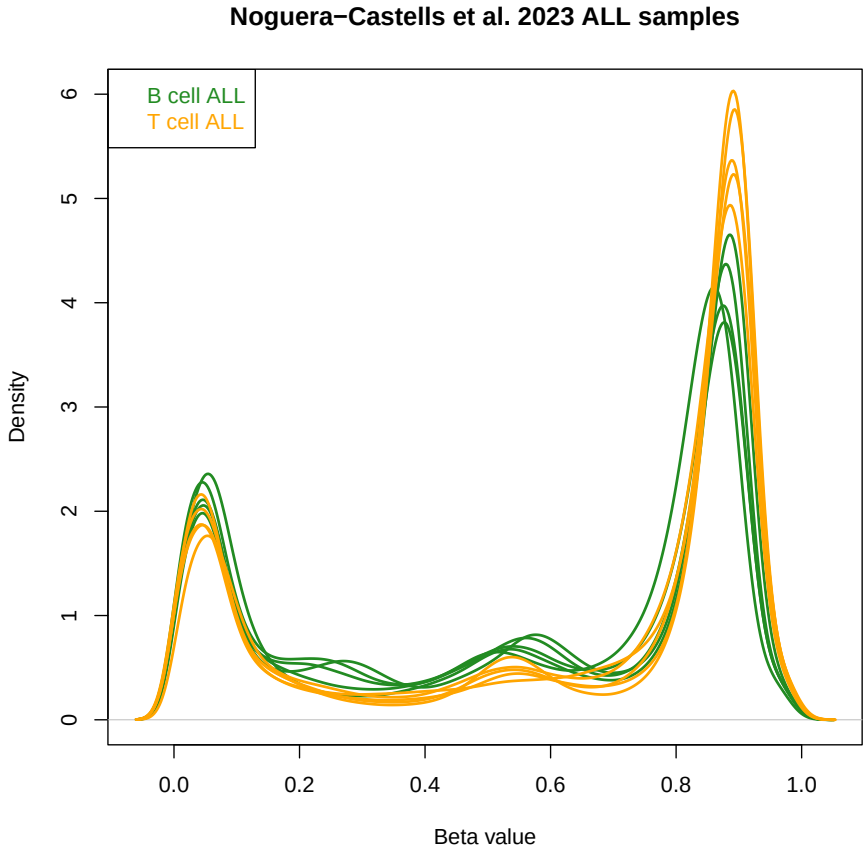
```
library(ExperimentHub)
eh <- ExperimentHub()
ALLbetas <- eh[["EH9451"]]
head(ALLbetas)
```

```
##                    BALL_01   BALL_02   BALL_03   BALL_04   BALL_05   TALL_01
## cg00000029_TC21 0.51221200 0.8123723 0.4522275 0.7993704 0.2033696 0.7856451
## cg00000109_TC21 0.87596630 0.4896535 0.8292074 0.2289017 0.6004454 0.9133542
## cg00000155_BC21 0.88927350 0.8942796 0.9044124 0.9417848 0.9060851 0.9204409
```

```
## cg00000158_BC21 0.91496990 0.9127220 0.9164246 0.9158492 0.9305084 0.9199422
## cg00000165_TC21 0.08777337 0.2067737 0.1017819 0.1056683 0.4797166 0.0651125
## cg00000221_BC21 0.84573060 0.8298593 0.7151204 0.8186862 0.8489285 0.8596191
##                    TALL_02   TALL_03    TALL_04   TALL_05
## cg00000029_TC21 0.88279680 0.8663721 0.88071230 0.8594255
## cg00000109_TC21 0.93277520 0.9023123 0.89303100 0.8841929
## cg00000155_BC21 0.91436200 0.9156632 0.91978220 0.9256892
## cg00000158_BC21 0.92858330 0.9306610 0.93371590 0.9376163
## cg00000165_TC21 0.07799849 0.3776332 0.06911442 0.0788177
## cg00000221_BC21 0.85424070 0.8713877 0.88269830 0.8905514
```

There are 5 TALL samples and 5 BALL samples, which, at a glance, may have distinct methylation profiles.

```
plot(density(ALLbetas[,1]), col="forestgreen", xlab="Beta value", ylim=c(0, 6),
     main="Noguera-Castells et al. 2023 ALL samples", lwd=2)
invisible(sapply(2:5, function (x) lines(density(ALLbetas[,x]), col="forestgreen", lwd=2)))
invisible(sapply(6:10, function (x) lines(density(ALLbetas[,x]), col="orange", lwd=2)))
legend("topleft", c("B cell ALL", "T cell ALL"), text.col=c("forestgreen", "orange"))
```



Now let's logit2 transform the betas into M-values, which approximate normality for linear modelling.

```
ALLMs <- minfi::logit2(ALLbetas)
```

Just like with the previous EPIC array, there are subsets of probes whose target is near a SNP, and also subsets for which there is evidence of cross-reactivity with other parts of the genome[1], both which can be filtered out with with `rmSNPandCH()`. We will filter out the SNP-proximal probes, but for this vignette, we will retain the EPICv2 probes for which there is evidence for preferential binding to off-targets, using `rmcrosshyb=FALSE`, since these off-targets have also been mapped[1].

```
nrow(ALLMs)

## [1] 894902
```

```
ALLMs.noSNPs <- rmSNPandCH(ALLMs, rmcrosshyb = FALSE)

## Assuming EPICv2 data.  Proceeding...
## see ?DMRcatedata and browseVignettes('DMRcatedata') for documentation
## loading from cache

nrow(ALLMs.noSNPs)

## [1] 893321
```

EPICv2 data contains groups of probes we call "replicates" that map to the same CpG site. Since DMRcate requires a 1-to-1 relationship between rows of methylation measurements and CpG loci, lest the kernel become biased towards sites with multiple mappings, we need to attenuate our M-value matrix to maintain this relationship. The function `rmPosReps()` does this, with user options specified by the argument `filter.strategy`. The default is to take the `mean` of each replicate group (with row name represented by the first probe in each group that appears in the manifest):

```
ALLMs.noSNPs.repmean <- rmPosReps(ALLMs.noSNPs, filter.strategy="mean")

## Loading EPICv2 manifest...
## loading from cache
## Averaging probes that map to the same CpG site...
```

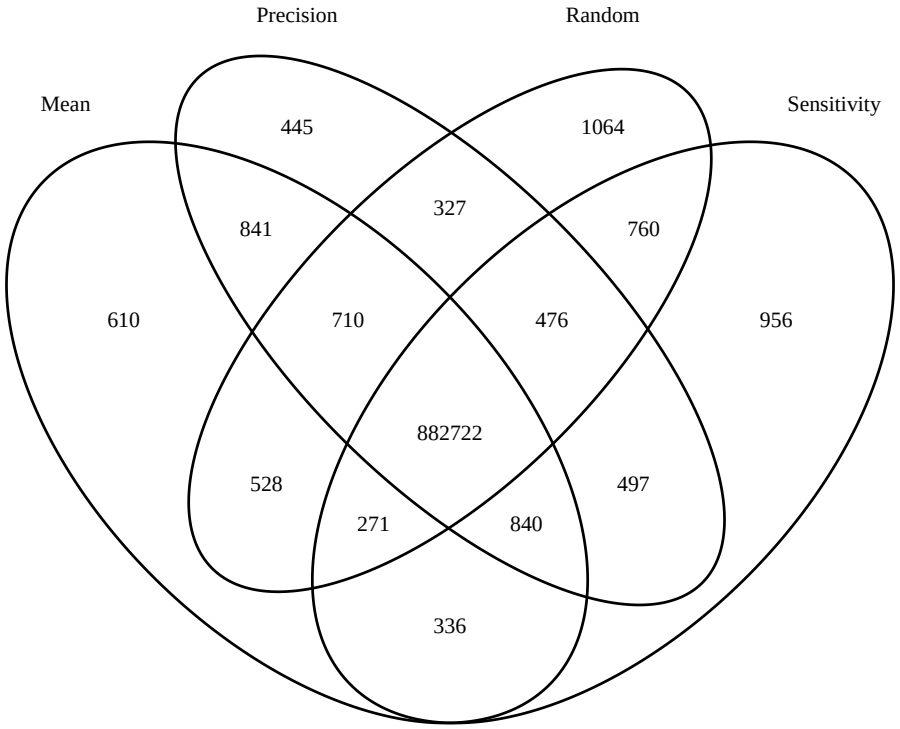
However, we can also select one probe from each replicate group based on its performance from cross-platform comparisons with EPICv1 and WGBS data (see Peters *et al.* 2024)[1]. Probe robustness is measured in terms of either:

- its sensitivity to methylation change and measurement precision, relative to a platform consensus obtained from matched EPICv1 and whole genome bisulphite sequencing (WGBS) values[3].
- minimum root mean squared error (RMSE) with WGBS (mainly for probes that don't appear on EPICv1)

In some cases, a replicated probe will prove superior in both sensitivity and precision, however in many cases superior sensitivity and precision is split between two probes in a replicate group. Furthermore, an individual probe may have superior sensitivity in a group, but superior precision is incurred by the group mean. Hence we give the user the option of choosing `sensitivity` or `precision`, based on their appetite for potential Type I error. In cases where there is insufficient evidence for superiority, a replicate probe is taken at random from the group. This strategy is generalised to all replicate groups for the `random` option. Figure 1 shows the differences in probe selection based on the `filter.strategy` argument. In practice, the number of differentially methylated CpGs (and hence DMRs) is unlikely to vary greatly depending on

the `filter.strategy` argument, but we provide this optionality to make use of all available probes on the array, and to potentially update the manifest with more rigorous probe benchmarking as usage proliferates.

Figure 1: Variation in probe selection based on the `filter.strategy` argument from `rmPosReps()`. Values in the "Random" group may differ when reproduced.



The main event, however, is calling DMRs. Let's take our replicate-averaged M-value matrix and specify our hypothesis to call DMRs between T cell and B cell ALL samples:

```
type <- gsub("_.*", "", colnames(ALLMs.noSNPs.repmean))
type

##  [1] "BALL" "BALL" "BALL" "BALL" "BALL" "TALL" "TALL" "TALL" "TALL" "TALL"

design <- model.matrix(~type)
```

```
design

##    (Intercept) typeTALL
## 1            1        0
## 2            1        0
## 3            1        0
## 4            1        0
## 5            1        0
## 6            1        1
## 7            1        1
## 8            1        1
## 9            1        1
## 10           1        1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$type
## [1] "contr.treatment"
```

Now for `cpg.annotate()`. In DMRcate 3.0 we have added an extra logical argument `epicv2Remap` giving the user the option of reassigning probes to an off-target CpG site where there is evidence (based on both *in silico* alignment and concordance with WGBS data) that the probe is preferentially hybridising to that site. As stated previously, this will only apply when you have specified `rmcrosshyb=FALSE` in `rmSNPandCH()`.

```
myannotation <- cpg.annotate("array", object=ALLMs.noSNPs.repmean, what = "M",
                             arraytype = "EPICv2", epicv2Remap = TRUE,
                             analysis.type="differential", design=design, coef=2,
                             fdr=0.001)

## EPICv2 specified.  Loading manifest...
## loading from cache
## Remapping 4214 cross-hybridising probes to their more likely offtarget...
## Your contrast returned 29047 individually significant probes.  We
recommend the default setting of pcutoff in dmrcate().
```

And then, as usual, call DMRs and annotate them. Unlike previous arrays, the EPICv2 manifest is in hg38:

```
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!
```

```
results.ranges <- extractRanges(dmrcoutput, genome = "hg38")
```

```
## see ?DMRcatedata and browseVignettes('DMRcatedata') for documentation
## loading from cache
```

```
results.ranges
```

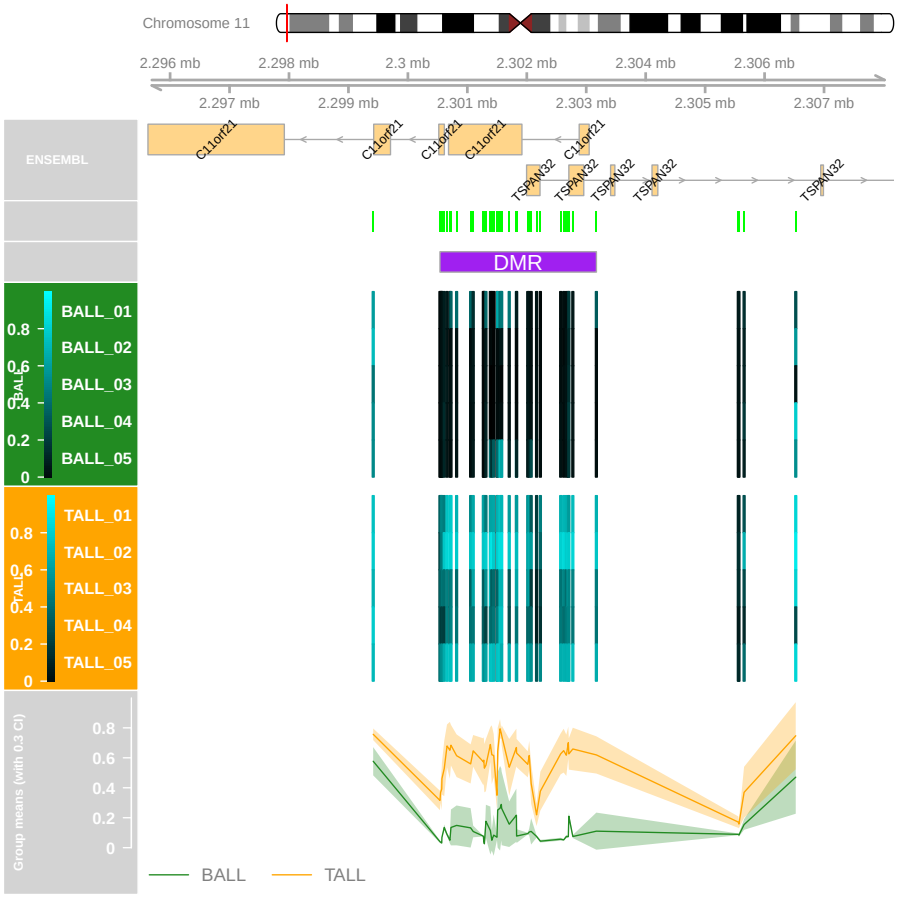
```
## GRanges object with 5417 ranges and 8 metadata columns:
##          seqnames              ranges strand |   no.cpgs min_smoothed_fdr
##             <Rle>           <IRanges>  <Rle> | <integer>        <numeric>
##      [1]     chr1 153627003-153629180      * |        23                0
##      [2]    chr11     2300540-2303171      * |        38                0
##      [3]    chr16   88975385-88978474      * |        18                0
##      [4]     chr4 186723466-186727204      * |        24                0
##      [5]     chr6   30687420-30690899      * |        38                0
##      ...      ...                 ...    ... .       ...              ...
##   [5413]    chr20   36645878-36646871      * |        12      8.10764e-77
##   [5414]    chr21   33401858-33402739      * |         8      7.43757e-62
##   [5415]    chr19   41363946-41365269      * |        15     3.39818e-301
##   [5416]    chr12   65882508-65882583      * |         2      4.45357e-41
##   [5417]     chr1 211326296-211326306      * |         2      1.09849e-41
##             Stouffer       HMFDR      Fisher   maxdiff   meandiff
##            <numeric>   <numeric>   <numeric> <numeric>  <numeric>
##      [1] 1.05022e-80 5.88555e-06 3.51116e-74  0.770396   0.531299
```

```
##      [2] 1.67834e-78 7.96194e-05 1.63728e-70  0.627270   0.465320
##      [3] 8.03527e-68 2.60133e-06 9.82218e-62  0.768491   0.563418
##      [4] 6.15941e-59 5.20802e-06 4.80956e-56 -0.692987  -0.457100
##      [5] 1.44171e-43 2.19887e-05 5.47427e-56  0.698081   0.333864
##      ...         ...         ...         ...       ...        ...
##   [5413]    0.308926 3.04865e-05   0.0279987  0.410876  0.0282449
##   [5414]    0.710193 3.97004e-05   0.0296453  0.628688  0.1061681
##   [5415]    0.936890 7.48404e-06   0.0754426  0.782168  0.0564337
##   [5416]    0.204983 1.89738e-01   0.2274645  0.266638  0.1776157
##   [5417]    0.707108 5.75957e-01   0.7315879 -0.122462 -0.0420904
##               overlapping.genes
##                     <character>
##      [1] S100A1, S100A13, AL1..
##      [2]      TSPAN32, C11orf21
##      [3]                CBFA2T3
##      [4]                   FAT1
##      [5]           PPP1R18, NRM
##      ...                    ...
##   [5413]                   SLA2
##   [5414]                   <NA>
##   [5415] AC011462.1, TMEM91, ..
##   [5416]                  HMGA2
##   [5417]                   <NA>
##   -------
##   seqinfo: 23 sequences from an unspecified genome; no seqlengths
```

And finally, plotting one of the top DMRs.

```
groups <- c(BALL="forestgreen", TALL="orange")
cols <- groups[as.character(type)]
DMR.plot(ranges=results.ranges, dmr=2, CpGs=ALLbetas,
         what = "Beta", arraytype = "EPICv2", phen.col=cols, genome = "hg38")

## Loading EPICv2 manifest...
## loading from cache
## Ensembl site unresponsive, trying useast mirror
```




```
sessionInfo()

## R version 4.4.1 (2024-06-14 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows Server 2022 x64 (build 20348)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=C
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
```

```
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] DMRcatedata_2.23.0
##  [2] IlluminaHumanMethylationEPICanno.ilm10b4.hg19_0.6.0
##  [3] IlluminaHumanMethylationEPICmanifest_0.3.0
##  [4] FlowSorted.Blood.EPIC_2.9.0
##  [5] minfi_1.51.0
##  [6] bumphunter_1.47.0
##  [7] locfit_1.5-9.10
##  [8] iterators_1.0.14
##  [9] foreach_1.5.2
## [10] Biostrings_2.73.1
## [11] XVector_0.45.0
## [12] SummarizedExperiment_1.35.1
## [13] Biobase_2.65.0
## [14] MatrixGenerics_1.17.0
## [15] matrixStats_1.3.0
## [16] GenomicRanges_1.57.1
## [17] GenomeInfoDb_1.41.1
## [18] IRanges_2.39.1
## [19] S4Vectors_0.43.1
## [20] ExperimentHub_2.13.0
## [21] AnnotationHub_3.13.0
## [22] BiocFileCache_2.13.0
## [23] dbplyr_2.5.0
## [24] BiocGenerics_0.51.0
## [25] DMRcate_3.1.5
##
## loaded via a namespace (and not attached):
##   [1] splines_4.4.1
##   [2] BiocIO_1.15.0
##   [3] bitops_1.0-7
##   [4] filelock_1.0.3
##   [5] cellranger_1.1.0
##   [6] tibble_3.2.1
##   [7] R.oo_1.26.0
##   [8] preprocessCore_1.67.0
##   [9] XML_3.99-0.17
```

```
##  [10] rpart_4.1.23
##  [11] lifecycle_1.0.4
##  [12] httr2_1.0.2
##  [13] edgeR_4.3.5
##  [14] base64_2.0.1
##  [15] MASS_7.3-61
##  [16] lattice_0.22-6
##  [17] ensembldb_2.29.0
##  [18] scrime_1.3.5
##  [19] backports_1.5.0
##  [20] magrittr_2.0.3
##  [21] limma_3.61.4
##  [22] Hmisc_5.1-3
##  [23] rmarkdown_2.27
##  [24] yaml_2.3.9
##  [25] doRNG_1.8.6
##  [26] askpass_1.2.0
##  [27] Gviz_1.49.0
##  [28] DBI_1.2.3
##  [29] RColorBrewer_1.1-3
##  [30] abind_1.4-5
##  [31] zlibbioc_1.51.1
##  [32] quadprog_1.5-8
##  [33] purrr_1.0.2
##  [34] R.utils_2.12.3
##  [35] AnnotationFilter_1.29.0
##  [36] biovizBase_1.53.0
##  [37] RCurl_1.98-1.16
##  [38] nnet_7.3-19
##  [39] VariantAnnotation_1.51.0
##  [40] rappdirs_0.3.3
##  [41] GenomeInfoDbData_1.2.12
##  [42] genefilter_1.87.0
##  [43] annotate_1.83.0
##  [44] permute_0.9-7
##  [45] DelayedMatrixStats_1.27.2
##  [46] codetools_0.2-20
##  [47] DelayedArray_0.31.8
##  [48] xml2_1.3.6
##  [49] tidyselect_1.2.1
##  [50] UCSC.utils_1.1.0
##  [51] beanplot_1.3.1
##  [52] base64enc_0.1-3
##  [53] illuminaio_0.47.0
##  [54] GenomicAlignments_1.41.0
```

```
##  [55] jsonlite_1.8.8
##  [56] multtest_2.61.0
##  [57] Formula_1.2-5
##  [58] survival_3.7-0
##  [59] missMethyl_1.39.13
##  [60] tools_4.4.1
##  [61] progress_1.2.3
##  [62] Rcpp_1.0.12
##  [63] glue_1.7.0
##  [64] gridExtra_2.3
##  [65] SparseArray_1.5.21
##  [66] xfun_0.45
##  [67] dplyr_1.1.4
##  [68] HDF5Array_1.33.3
##  [69] withr_3.0.0
##  [70] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.6.1
##  [71] BiocManager_1.30.23
##  [72] fastmap_1.2.0
##  [73] latticeExtra_0.6-30
##  [74] rhdf5filters_1.17.0
##  [75] fansi_1.0.6
##  [76] openssl_2.2.0
##  [77] digest_0.6.36
##  [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_2.61.2
##  [85] RSQLite_2.3.7
##  [86] R.methodsS3_1.8.2
##  [87] tidyr_1.3.1
##  [88] utf8_1.2.4
##  [89] generics_0.1.3
##  [90] data.table_1.15.4
##  [91] rtracklayer_1.65.0
##  [92] prettyunits_1.2.0
##  [93] httr_1.4.7
##  [94] htmlwidgets_1.6.4
##  [95] S4Arrays_1.5.4
##  [96] pkgconfig_2.0.3
##  [97] gtable_0.3.5
##  [98] blob_1.2.4
##  [99] siggenes_1.79.0
```

```
## [100] htmltools_0.5.8.1
## [101] ProtGenerics_1.37.0
## [102] scales_1.3.0
## [103] png_0.1-8
## [104] knitr_1.48
## [105] rstudioapi_0.16.0
## [106] tzdb_0.4.0
## [107] rjson_0.2.21
## [108] nlme_3.1-165
## [109] checkmate_2.3.1
## [110] curl_5.2.1
## [111] org.Hs.eg.db_3.19.1
## [112] cachem_1.1.0
## [113] rhdf5_2.49.0
## [114] stringr_1.5.1
## [115] BiocVersion_3.20.0
## [116] foreign_0.8-87
## [117] AnnotationDbi_1.67.0
## [118] restfulr_0.0.15
## [119] GEOquery_2.73.0
## [120] pillar_1.9.0
## [121] grid_4.4.1
## [122] reshape_0.8.9
## [123] vctrs_0.6.5
## [124] xtable_1.8-4
## [125] cluster_2.1.6
## [126] htmlTable_2.4.2
## [127] evaluate_0.24.0
## [128] bsseq_1.41.0
## [129] readr_2.1.5
## [130] GenomicFeatures_1.57.0
## [131] cli_3.6.3
## [132] compiler_4.4.1
## [133] Rsamtools_2.21.0
## [134] rngtools_1.5.2
## [135] rlang_1.1.4
## [136] crayon_1.5.3
## [137] nor1mix_1.3-3
## [138] mclust_6.1.1
## [139] interp_1.1-6
## [140] plyr_1.8.9
## [141] stringi_1.8.4
## [142] deldir_2.0-4
## [143] BiocParallel_1.39.0
## [144] munsell_0.5.1
```

```
## [145] lazyeval_0.2.2
## [146] Matrix_1.7-0
## [147] BSgenome_1.73.0
## [148] hms_1.1.3
## [149] sparseMatrixStats_1.17.2
## [150] bit64_4.0.5
## [151] ggplot2_3.5.1
## [152] Rhdf5lib_1.27.0
## [153] KEGGREST_1.45.1
## [154] statmod_1.5.0
## [155] highr_0.11
## [156] memoise_2.0.1
## [157] bit_4.0.5
## [158] readxl_1.4.3
```

References

[1] Peters, T.J., Meyer, B., Ryan, L., Achinger-Kawecka, J., Song, J., Campbell, E.M., Qu, W., Nair, S., Loi-Luu, P., Stricker, P., Lim, E., Stirzaker, C., Clark, S.J. and Pidsley, (2024). Characterisation and reproducibility of the HumanMethylationEPIC v2.0 BeadChip for DNA methylation profiling. *BMC Genomics* 25, 251. doi:10.1186/s12864-024-10027-5.

[2] Noguera-Castells A, Garcia-Prieto CA, Alvarez-Errico D, Esteller, M. (2023). Validation of the new EPIC DNA methylation microarray (900K EPIC v2) for high-throughput profiling of the human DNA methylome. *Epigenetics* 18(1), 2185742.

[3] Peters TJ, French HJ, Bradford ST, Pidsley R, Stirzaker C, Varinli H, Nair, S, Qu W, Song J, Giles KA, Statham AL, Speirs H, Speed TP, Clark, SJ. (2019). Evaluation of cross-platform and interlaboratory concordance via consensus modelling of genomic measurements. *Bioinformatics*, 2019 35(4), 560-570.