Example Analysis using BiocMAP Output Objects

Nicholas J. Eagles[*1] ***and Leonardo Collado-Torres***[†1,2]

[1]Lieber Institute for Brain Development, Johns Hopkins Medical Campus
[2]Center for Computational Biology, Johns Hopkins University

[*]nickeagles77@gmail.com [†]lcolladotor@gmail.com

16 February 2022

Contents

1 Add Experiment Metadata to BiocMAP Outputs 2

2 Exploratory Plots 4

2.1 Bisulfite-Conversion Efficiency by Cell Population 4

2.2 Relationship between Methylation Fractions across Cytosine Context, by Cell Population 5

2.3 Fraction of Covered Cytosines by Cell Population 7

2.4 Distribution of Methylation Fractions across Cytosines by Cell Population 8

3 Bibliography 14

1 Add Experiment Metadata to BiocMAP Outputs

The `bsseq` output objects from BiocMAP contain methylation and coverage info for our samples in the dataset. However, we're interested in exploring how this information relates back to sample metadata and phenotype information, present in an external file. Our first step will therefore be to load the BiocMAP output objects into memory, and manually attach the additional sample metadata to each object.

```
#  Load required R packages
library("bsseq")
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
## Loading required package: GenomicRanges
## Loading required package: stats4
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
## Loading required package: IRanges
## Loading required package: GenomeInfoDb
## Loading required package: SummarizedExperiment
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
##
## Attaching package: 'MatrixGenerics'
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
```

Example Analysis using BiocMAP Output Objects

```
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
## Loading required package: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Attaching package: 'Biobase'
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
library("HDF5Array")
## Loading required package: DelayedArray
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:S4Vectors':
##
##     expand
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum, scale, sweep
## Loading required package: rhdf5
##
## Attaching package: 'HDF5Array'
## The following object is masked from 'package:rhdf5':
##
##     h5ls
library("ggplot2")

#  Path to the sample metadata and BiocMAP outputs. The outputs are too large
#  to host in this repository, so we reference local paths here
meta_file <- file.path(
    "/dcl02/lieber/ajaffe/FlowRNA_RNAseq/WGBS",
    "FlowRNA_WGBS_Sample_Information_with_Pheno_Info.csv"
)
out_dir <- file.path(
    "/dcs04/lieber/lcolladotor/flowRNA_LIBD001/flowRNA_WGBS/processed-data",
    "03_BiocMAP/BiocMAP_output"
)
```

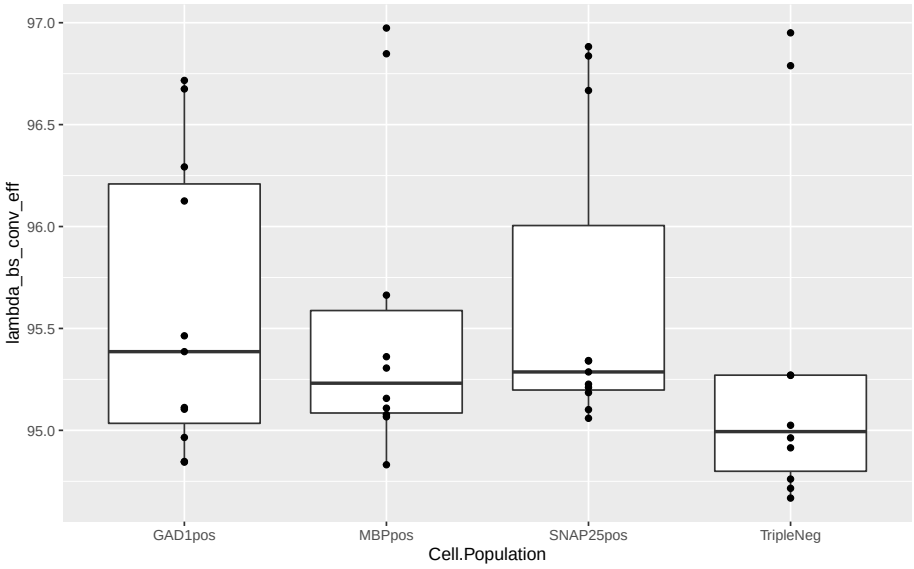
```
#  Load the 'CpG'-context object
bs_cpg <- loadHDF5SummarizedExperiment(
    file.path(out_dir, "BSobjects", "objects", "combined"),
    prefix = "CpG"
)

#  Load the 'CpH'-context object. Note: this requires quite a bit of memory
#  (~23GB) even though the assays are disk-backed!
bs_cph <- loadHDF5SummarizedExperiment(
    file.path(out_dir, "BSobjects", "objects", "combined"),
    prefix = "CpH"
)

#  Read in experiment-specific metadata and ensure sample ID orders match
meta <- read.csv(meta_file)
meta <- meta[match(colnames(bs_cpg), meta$LIBD.), ]

#  Add this metadata to the Bioconductor objects
colData(bs_cpg) <- cbind(colData(bs_cpg), meta)
colData(bs_cph) <- cbind(colData(bs_cph), meta)

#  Keep a copy of the metadata as a data frame, for easy plotting
meta_df <- data.frame(colData(bs_cpg))
```

2 Exploratory Plots

2.1 Bisulfite-Conversion Efficiency by Cell Population

This experiment used spike-ins of the lambda bacteriophage genome, which were quantified via BiocMAP to infer bisulfite-conversion rate. Successful bisulfite conversion is a pre-requisite for accurate methylation calls, so we'd like to see both that values are close to 1, and that values are not significantly different by sample (or by sample-related variables like cell population). We'll explore this visually below.

```
ggplot(meta_df, aes(x = Cell.Population, y = lambda_bs_conv_eff)) +
    geom_boxplot(outlier.shape = NA) +
    geom_point()
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
## Warning: Removed 1 rows containing missing values (geom_point).
```



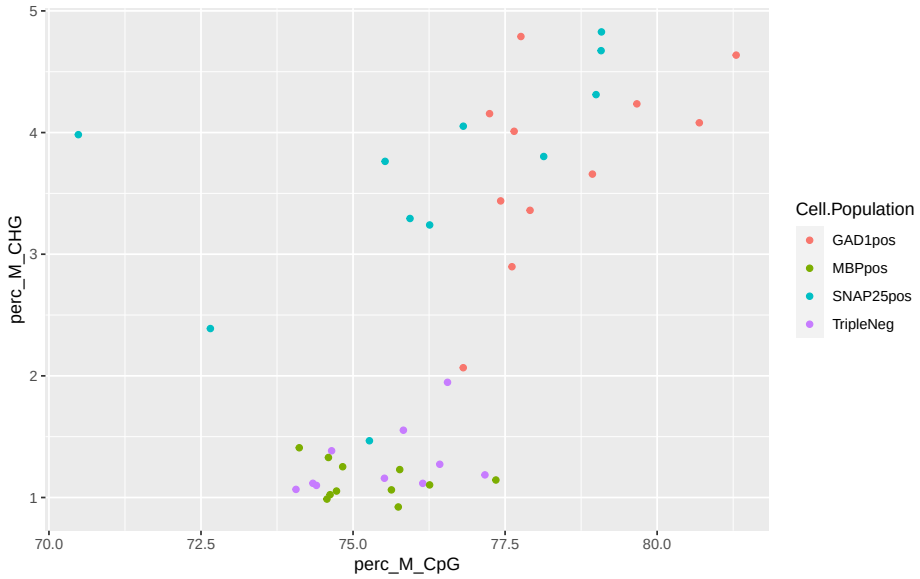
2.2 Relationship between Methylation Fractions across Cytosine Context, by Cell Population

Next, we'll explore if average methylation rate for each cytosine context correlates with that of other contexts across sample. For example, is a sample with highly methylated CpGs likely to have highly methylated CHGs (the first plot)?

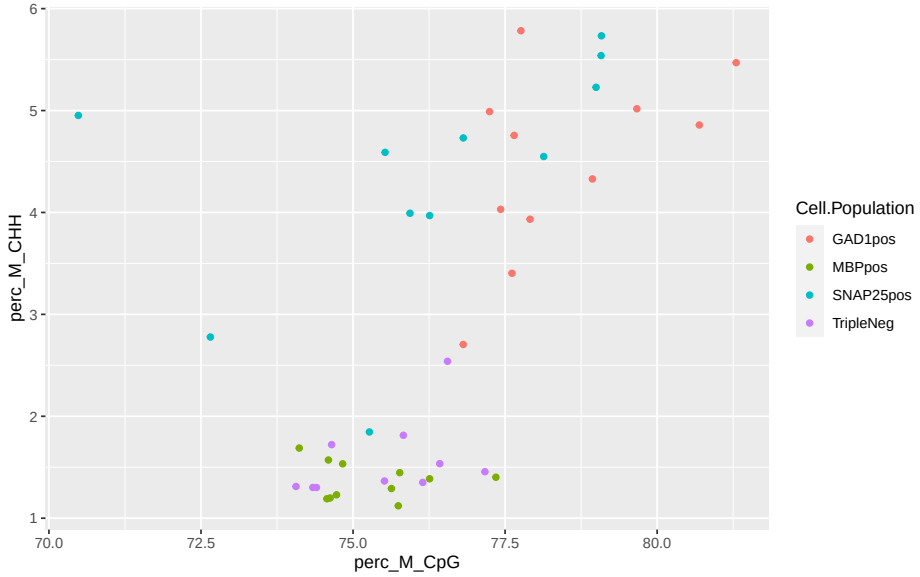
We observe a few interesting facts; first, there is a visibly obvious correlation between average methylation rates of different cytosine contexts by sample. This is highly pronounced between CpH contexts (CHG vs. CHH). In each case, the relation appears roughly linear, though this is questionable for CpG vs. CHH context comparison. Another observation is that samples tend to cluster fairly well by cell population. Finally, for comparisons of CpG vs. CpH context, the strength of correlation between methylation rates varies significantly by cell type, with `MBPpos` and `TripleNeg` showing only weak correlation at best.

```
#  Clean this up: code should be written around the idea we are plotting each
#  combination of methylation fractions (CpG, CHG, CHH), and plots should be
#  placed in a single grid

ggplot(meta_df, aes(x = perc_M_CpG, y = perc_M_CHG, color = Cell.Population)) +
    geom_point()
```

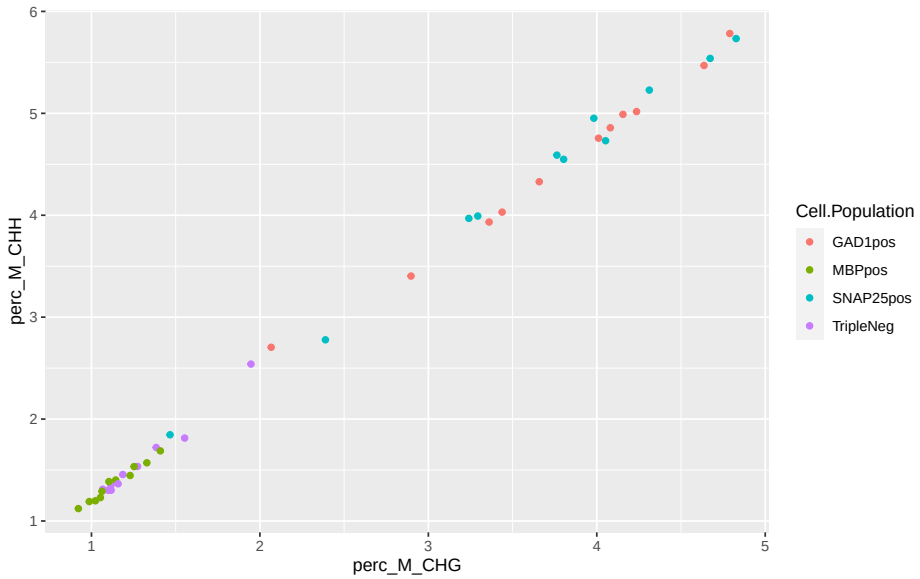
Example Analysis using BiocMAP Output Objects




```
ggplot(meta_df, aes(x = perc_M_CpG, y = perc_M_CHH, color = Cell.Population)) +
    geom_point()
```



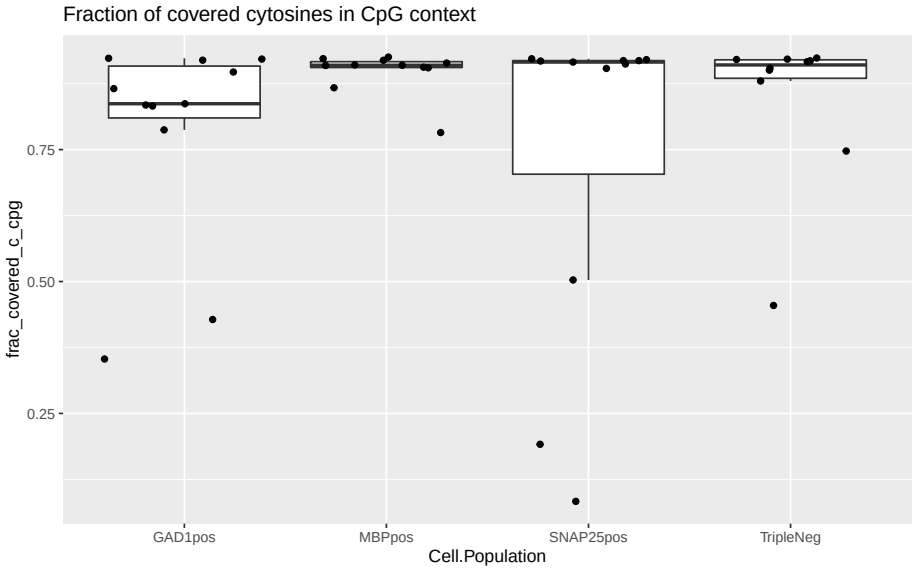

```
ggplot(meta_df, aes(x = perc_M_CHG, y = perc_M_CHH, color = Cell.Population)) +
    geom_point()
```



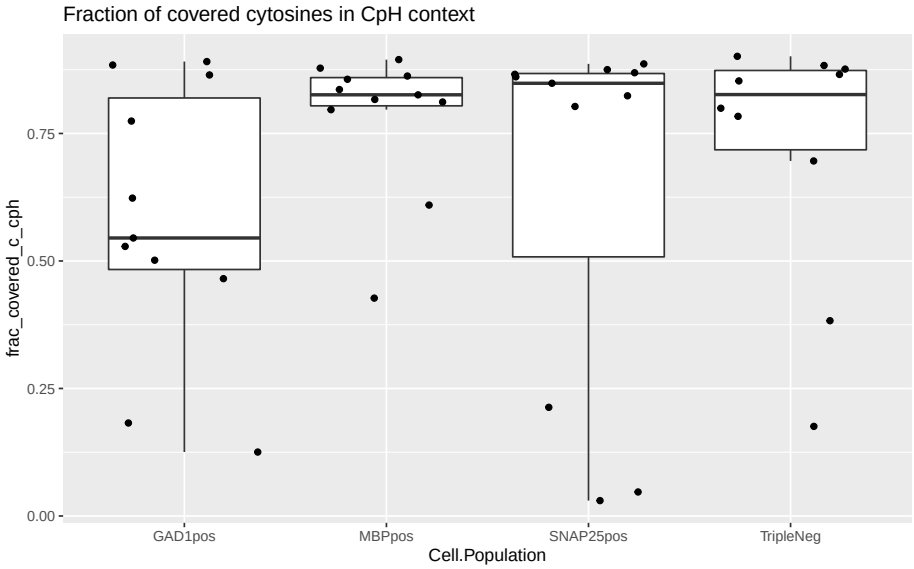
2.3 Fraction of Covered Cytosines by Cell Population

Another useful piece of information is how well-covered the genome is with methylation information. Does coverage of cytosines vary by a sample's cell type?

```
#  The matrices in 'assays(bs_cpg)' and 'assays(bs_cph)' are stored on disk. To
#  speed up some below computations, we raise the per-block memory size
setAutoBlockSize(1e9)
## automatic block size set to 1e+09 bytes (was 1e+08)

#  Get the proportion of cytosines in each object (context) that have non-zero
#  coverage in at least one sample
meta_df$frac_covered_c_cpg <- DelayedArray::colMeans(assays(bs_cpg)$Cov > 3)
meta_df$frac_covered_c_cph <- DelayedArray::colMeans(assays(bs_cph)$Cov > 3)

#   Plot fraction of covered CpG-context cytosines by cell population
ggplot(meta_df, aes(x = Cell.Population, y = frac_covered_c_cpg)) +
    geom_boxplot(outlier.shape = NA) +
    geom_point(position = "jitter") +
    labs(title = "Fraction of covered cytosines in CpG context")
```




```
#   Plot fraction of covered CpH-context cytosines by cell population
ggplot(meta_df, aes(x = Cell.Population, y = frac_covered_c_cph)) +
    geom_boxplot(outlier.shape = NA) +
    geom_point(position = "jitter") +
    labs(title = "Fraction of covered cytosines in CpH context")
```



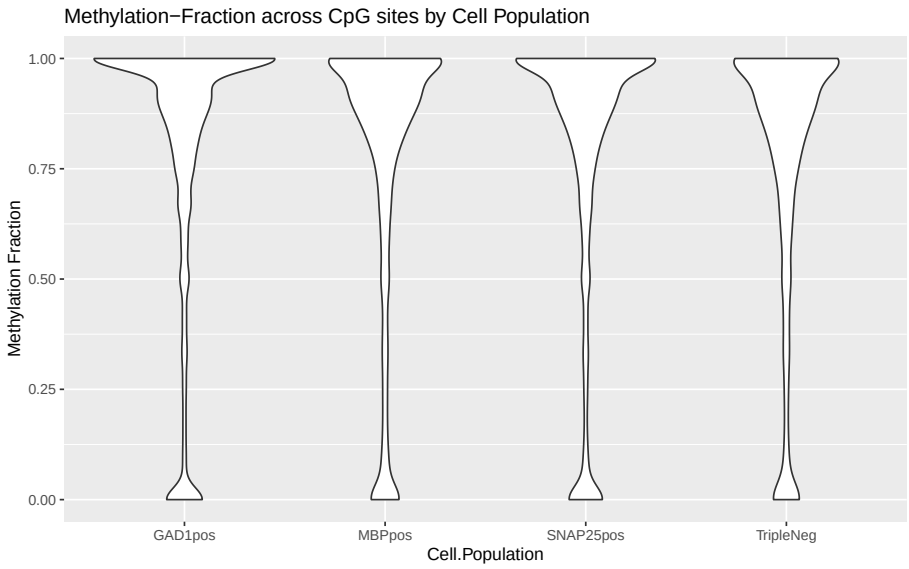
2.4 Distribution of Methylation Fractions across Cytosines by Cell Population

Grouping together all samples of a particular cell type, we'll explore the methylation-fraction distribution across cytosines, separately for both CpG and CpH contexts.

For both cytosine contexts, we observe a bimodal distribution with peaks at fractions of 0 and 1. This suggests that within a particular sample, a cytosine site is disproportionately likely to have consistent methylation pattern. For example, many sites are such that all observations of the cytosine are methylated for a particular sample. Similarly, we don't see many sites where around half of the observed site are methylated for a given sample. It's also worth noting that the apparent bimodal form is likely not an artifact of low coverage– i.e., only a small fraction of sites are covered just once or twice, a circumstance that would cause over-representation of the fractions of 0 or 1.

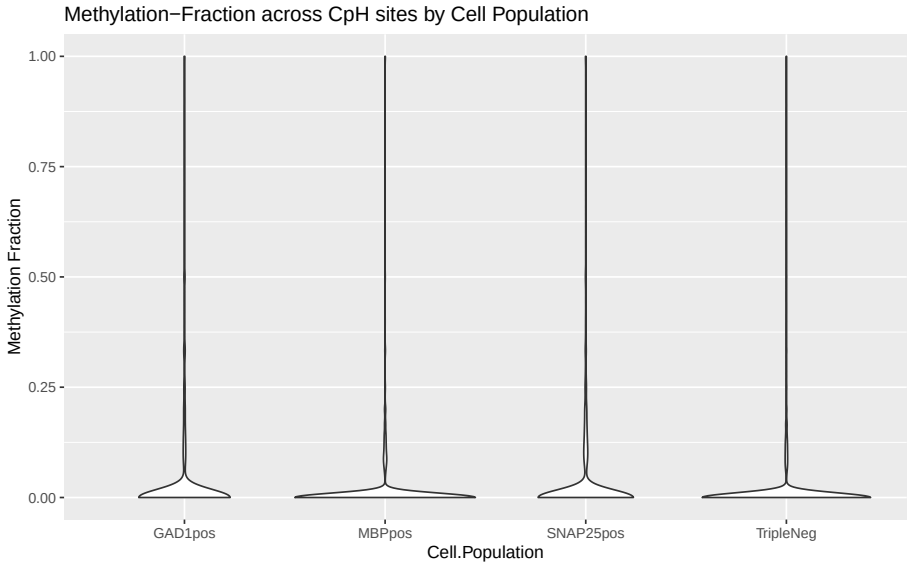
```
#   Randomly subset to a particular number of cytosines, to both control memory
#   and speed up plotting
max_sites <- 1000

#   Look at CpG sites first
indices <- sample(nrow(bs_cpg), max_sites)
m_frac <- assays(bs_cpg)$M[indices, ] / assays(bs_cpg)$Cov[indices, ]

#   It's worth looking at the distribution of coverage by site, since in theory
#   this could be cause for the bimodality observed in the plot below
table(assays(bs_cpg)$Cov[indices, ])
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 3549 1461 1347 1395 1480 1611 1631 1667 1755 1830 1895 1998 1928 2030 2022 1930
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
## 1930 1706 1621 1451 1264 1050  904  714  613  501  384  294  238  192  139   88
##   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47
##   68   50   24   24   11   17   10    5   10    5    5    5    2    7    3    3
##   48   49   50   51   52   53   54   55   56   57   58   59   60   62   63   64
##    4    4    3    3    3    2    2    3    3    2    3    2    2    1    2    2
##   66   67   68   69   70   71   72   74   75   77   78   79   84   86   87   89
##    1    2    2    2    5    1    5    1    2    2    1    2    1    2    2    1
##   91   94  105  117  121  126  134  135  138  141  142  143  152  154  157  161
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  162  164  165  170  172  174  175  178  180  182  188  192  193  196  198  200
##    1    1    1    1    1    1    1    1    2    2    1    1    1    1    1    1
##  204  206  208  214  220  221  229  231  236  246  249  251  254  264  273  277
##    1    1    1    1    1    1    1    1    1    1    1    2    1    1    1    1
##  281  283  286  290  313  316  326  348  356
##    1    1    1    1    1    1    1    1    1

#   Form a data frame for easy plotting: we'll collapse methylation data for all
#   samples into a single column, 'm_frac'. Here 'LIBD.' denotes sample ID
meth_df <- data.frame(
    "m_frac" = as.numeric(m_frac),
    "LIBD." = rep(colnames(m_frac), each = max_sites)
)

#   Label each observation (methylation fraction for a particular cytosine) with
#   the cell population of the associated sample
meth_df$Cell.Population <- meta_df$Cell.Population[
    match(meth_df$"LIBD.", meta_df$"LIBD.")
]
```

```
ggplot(meth_df, aes(x = Cell.Population, y = m_frac)) +
    geom_violin() +
    labs(
        title = "Methylation-Fraction across CpG sites by Cell Population",
        y = "Methylation Fraction"
    )
## Warning: Removed 3549 rows containing non-finite values (stat_ydensity).
```




```
#   Now look at CpH sites
indices <- sample(nrow(bs_cph), max_sites)
m_frac <- assays(bs_cph)$M[indices, ] / assays(bs_cph)$Cov[indices, ]

#   Again, we'll look at the distribution of coverage by site, since in theory
#   this could be cause for the bimodality observed in the plots below
table(assays(bs_cph)$Cov[indices, ])
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 5049 2565 2845 3128 3292 3573 3615 3609 3211 2888 2386 1963 1479 1090  754  552
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   32
##  360  212  133   94   42   33   28   16   11    8    6    4    1    3    1    2
##   35   37  100  167  216  223  234  260  261  263  286  287  299  303  312  315
##    3    1    1    1    1    1    1    1    2    1    1    1    1    1    1    1
##  320  345  352  359  362  364  366  367  369  370  372  379  381  384  390  411
##    1    1    1    1    2    1    2    1    1    2    1    1    1    1    1    1
##  412  417  422  424  432  433  485  491  535
##    1    1    1    1    1    1    1    1    1

#   Form a data frame for easy plotting: we'll collapse methylation data for all
#   samples into a single column, 'm_frac'. Here 'LIBD.' denotes sample ID
meth_df <- data.frame(
    "m_frac" = as.numeric(m_frac),
```

```
    "LIBD." = rep(colnames(m_frac), each = max_sites)
)

#   Label each observation (methylation fraction for a particular cytosine) with
#   the cell population of the associated sample
meth_df$Cell.Population <- meta_df$Cell.Population[
    match(meth_df$"LIBD.", meta_df$"LIBD.")
]

ggplot(meth_df, aes(x = Cell.Population, y = m_frac)) +
    geom_violin() +
    labs(
        title = "Methylation-Fraction across CpH sites by Cell Population",
        y = "Methylation Fraction"
    )
## Warning: Removed 5049 rows containing non-finite values (stat_ydensity).
```



Date the vignette was generated:

```
## [1] "2022-02-16 17:12:33 EST"
```

Wallclock time spent generating the vignette:

```
## Time difference of 2.307 hours
```

`R` session information:

```
## - Session info ---------------------------------------------------------------------------------------------------
##  setting  value
##  version  R version 4.1.2 Patched (2021-11-04 r81138)
##  os       CentOS Linux 7 (Core)
##  system   x86_64, linux-gnu
##  ui       X11
##  language (EN)
```

```
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       US/Eastern
##  date     2022-02-16
##  pandoc   2.13 @ /jhpce/shared/jhpce/core/conda/miniconda3-4.6.14/envs/svnR-4.1.x/bin/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------------------------------------------
##  package              * version  date (UTC) lib source
##  assertthat             0.2.1    2019-03-21 [2] CRAN (R 4.1.0)
##  Biobase              * 2.54.0   2021-10-26 [2] Bioconductor
##  BiocGenerics         * 0.40.0   2021-10-26 [2] Bioconductor
##  BiocIO                 1.4.0    2021-10-26 [2] Bioconductor
##  BiocManager            1.30.16  2021-06-15 [2] CRAN (R 4.1.2)
##  BiocParallel           1.28.3   2021-12-09 [2] Bioconductor
##  BiocStyle            * 2.22.0   2021-10-26 [1] Bioconductor
##  Biostrings             2.62.0   2021-10-26 [2] Bioconductor
##  bitops                 1.0-7    2021-04-24 [2] CRAN (R 4.1.0)
##  bookdown               0.24     2021-09-02 [1] CRAN (R 4.1.2)
##  BSgenome               1.62.0   2021-10-26 [2] Bioconductor
##  bsseq                * 1.30.0   2021-10-26 [2] Bioconductor
##  cli                    3.2.0    2022-02-14 [2] CRAN (R 4.1.2)
##  colorspace             2.0-2    2021-06-24 [2] CRAN (R 4.1.0)
##  crayon                 1.5.0    2022-02-14 [2] CRAN (R 4.1.2)
##  data.table             1.14.2   2021-09-27 [2] CRAN (R 4.1.2)
##  DBI                    1.1.2    2021-12-20 [2] CRAN (R 4.1.2)
##  DelayedArray         * 0.20.0   2021-10-26 [2] Bioconductor
##  DelayedMatrixStats     1.16.0   2021-10-26 [2] Bioconductor
##  digest                 0.6.29   2021-12-01 [2] CRAN (R 4.1.2)
##  dplyr                  1.0.8    2022-02-08 [2] CRAN (R 4.1.2)
##  ellipsis               0.3.2    2021-04-29 [2] CRAN (R 4.1.0)
##  evaluate               0.14     2019-05-28 [2] CRAN (R 4.1.0)
##  fansi                  1.0.2    2022-01-14 [2] CRAN (R 4.1.2)
##  farver                 2.1.0    2021-02-28 [2] CRAN (R 4.1.0)
##  fastmap                1.1.0    2021-01-25 [2] CRAN (R 4.1.0)
##  generics               0.1.2    2022-01-31 [2] CRAN (R 4.1.2)
##  GenomeInfoDb         * 1.30.1   2022-01-30 [2] Bioconductor
##  GenomeInfoDbData       1.2.7    2021-11-01 [2] Bioconductor
##  GenomicAlignments      1.30.0   2021-10-26 [2] Bioconductor
##  GenomicRanges        * 1.46.1   2021-11-18 [2] Bioconductor
##  ggplot2              * 3.3.5    2021-06-25 [2] CRAN (R 4.1.0)
##  glue                   1.6.1    2022-01-22 [2] CRAN (R 4.1.2)
##  gtable                 0.3.0    2019-03-25 [2] CRAN (R 4.1.0)
##  gtools                 3.9.2    2021-06-06 [2] CRAN (R 4.1.0)
##  HDF5Array            * 1.22.1   2021-11-14 [2] Bioconductor
##  htmltools              0.5.2    2021-08-25 [2] CRAN (R 4.1.2)
##  httr                   1.4.2    2020-07-20 [2] CRAN (R 4.1.0)
##  IRanges              * 2.28.0   2021-10-26 [2] Bioconductor
##  jsonlite               1.7.3    2022-01-17 [2] CRAN (R 4.1.2)
##  knitr                  1.37     2021-12-16 [2] CRAN (R 4.1.2)
##  labeling               0.4.2    2020-10-20 [2] CRAN (R 4.1.0)
##  lattice                0.20-45  2021-09-22 [3] CRAN (R 4.1.2)
```

```
##  lifecycle             1.0.1    2021-09-24 [2] CRAN (R 4.1.2)
##  limma                 3.50.0   2021-10-26 [2] Bioconductor
##  locfit                1.5-9.4  2020-03-25 [2] CRAN (R 4.1.0)
##  lubridate             1.8.0    2021-10-07 [2] CRAN (R 4.1.2)
##  magrittr              2.0.2    2022-01-26 [2] CRAN (R 4.1.2)
##  Matrix              * 1.4-0    2021-12-08 [3] CRAN (R 4.1.2)
##  MatrixGenerics      * 1.6.0    2021-10-26 [2] Bioconductor
##  matrixStats         * 0.61.0   2021-09-17 [2] CRAN (R 4.1.2)
##  munsell               0.5.0    2018-06-12 [2] CRAN (R 4.1.0)
##  permute               0.9-7    2022-01-27 [2] CRAN (R 4.1.2)
##  pillar                1.7.0    2022-02-01 [2] CRAN (R 4.1.2)
##  pkgconfig             2.0.3    2019-09-22 [2] CRAN (R 4.1.0)
##  plyr                  1.8.6    2020-03-03 [2] CRAN (R 4.1.0)
##  purrr                 0.3.4    2020-04-17 [2] CRAN (R 4.1.0)
##  R.methodsS3           1.8.1    2020-08-26 [2] CRAN (R 4.1.0)
##  R.oo                  1.24.0   2020-08-26 [2] CRAN (R 4.1.0)
##  R.utils               2.11.0   2021-09-26 [2] CRAN (R 4.1.2)
##  R6                    2.5.1    2021-08-19 [2] CRAN (R 4.1.2)
##  Rcpp                  1.0.8    2022-01-13 [2] CRAN (R 4.1.2)
##  RCurl                 1.98-1.6 2022-02-08 [2] CRAN (R 4.1.2)
##  RefManageR          * 1.3.0    2020-11-13 [1] CRAN (R 4.1.2)
##  restfulr              0.0.13   2017-08-06 [2] CRAN (R 4.1.0)
##  rhdf5               * 2.38.0   2021-10-26 [2] Bioconductor
##  rhdf5filters          1.6.0    2021-10-26 [2] Bioconductor
##  Rhdf5lib              1.16.0   2021-10-26 [2] Bioconductor
##  rjson                 0.2.21   2022-01-09 [2] CRAN (R 4.1.2)
##  rlang                 1.0.1    2022-02-03 [2] CRAN (R 4.1.2)
##  rmarkdown             2.11     2021-09-14 [2] CRAN (R 4.1.2)
##  Rsamtools             2.10.0   2021-10-26 [2] Bioconductor
##  rtracklayer           1.54.0   2021-10-26 [2] Bioconductor
##  S4Vectors           * 0.32.3   2021-11-21 [2] Bioconductor
##  scales                1.1.1    2020-05-11 [2] CRAN (R 4.1.0)
##  sessioninfo         * 1.2.2    2021-12-06 [2] CRAN (R 4.1.2)
##  sparseMatrixStats     1.6.0    2021-10-26 [2] Bioconductor
##  stringi               1.7.6    2021-11-29 [2] CRAN (R 4.1.2)
##  stringr               1.4.0    2019-02-10 [2] CRAN (R 4.1.0)
##  SummarizedExperiment * 1.24.0   2021-10-26 [2] Bioconductor
##  tibble                3.1.6    2021-11-07 [2] CRAN (R 4.1.2)
##  tidyselect            1.1.1    2021-04-30 [2] CRAN (R 4.1.0)
##  utf8                  1.2.2    2021-07-24 [2] CRAN (R 4.1.0)
##  vctrs                 0.3.8    2021-04-29 [2] CRAN (R 4.1.0)
##  withr                 2.4.3    2021-11-30 [2] CRAN (R 4.1.2)
##  xfun                  0.29     2021-12-14 [2] CRAN (R 4.1.2)
##  XML                   3.99-0.8 2021-09-17 [2] CRAN (R 4.1.2)
##  xml2                  1.3.3    2021-11-30 [2] CRAN (R 4.1.2)
##  XVector               0.34.0   2021-10-26 [2] Bioconductor
##  yaml                  2.2.2    2022-01-25 [2] CRAN (R 4.1.2)
##  zlibbioc              1.40.0   2021-10-26 [2] Bioconductor
##
##  [1] /users/neagles/R/4.1.x
##  [2] /jhpce/shared/jhpce/core/conda/miniconda3-4.6.14/envs/svnR-4.1.x/R/4.1.x/lib64/R/site-library
```

```
##  [3] /jhpce/shared/jhpce/core/conda/miniconda3-4.6.14/envs/svnR-4.1.x/R/4.1.x/lib64/R/library
##
## ------------------------------------------------------------------------------------------------------------------
```

3 Bibliography

This vignette was generated using *BiocStyle* (Oleś, 2021) with *knitr* (Xie, 2021) and *rmarkdown* (Allaire, Xie, McPherson, Luraschi, Ushey, Atkins, Wickham, Cheng, Chang, and Iannone, 2021) running behind the scenes.

Citations made with *RefManageR* (McLean, 2017).

[1] J. Allaire, Y. Xie, J. McPherson, et al. rmarkdown: Dynamic Documents for R. R package version 2.11. 2021. URL: https://github.com/rstudio/rmarkdown.

[2] M. W. McLean. "RefManageR: Import and Manage BibTeX and BibLaTeX References in R". In: The Journal of Open Source Software (2017). DOI: 10.21105/joss.00338.

[3] A. Oleś. BiocStyle: Standard styles for vignettes and other Bioconductor documents. R package version 2.22.0. 2021. URL: https://github.com/Bioconductor/BiocStyle.

[4] Y. Xie. knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.37. 2021. URL: https://yihui.org/knitr/.