An Introduction to the methylumi package

Sean Davis and Sven Bilke

November 1, 2022

1 Introduction

Gene expression patterns are very important in understanding any biologic system. The regulation of gene expression is a complex, poorly understood process. However, DNA methylation is one biological process that is known to be important in gene regulation. In short, DNA methylation is a chemical modification of DNA CpG sites in which a methyl group is added number 5 carbon of the cytosine pyrimidine ring. In humans, the DNA methyltransferases (DNMT1, DNMT3a, and DNMT3b) are the enzymes responsible for carrying out the methylation.

The Illumina GoldenGate methylation profiling technology specifically targets more than 1500 CpG sites throughout the genome, specifically targeting approximately 700 "cancer genes". Samples are run in 96-well format, making the technology very high-throughput. After a two-color hybridization, a laser captures the intensities associated with the methylated state and the accompanying unmethylated state. The Illumina BeadStudio software is then used for quality control and basic visualization tasks. A newer platform, the Illumina Infinium Methylation platform provides a more "whole-genome" view of DNA methylation. Utilizing the Infinium profiling technology on bisulfite-treated DNA, the methylation status of more than 25,000 individual CpG sites is assayed simultaneously. This package can handle both types of data. Note that normalization functions here are really specific to the GoldenGate platform and are probably not optimal for Infinium data.

The methylumi package provides convenient mechanisms for loading the results of the Illumina methylation platform into R/Bioconductor. Classes based on common Bioconductor classes for encapsulating the data and facilitate data manipulation are at the core of the package, with methods for quality control, normalization (for GoldenGate, in particular), and plotting.

2 Loading Data

After exporting the data from BeadStudio, methylumi can read them in with a single command. To include rich sample annotation, it is possible to supply a `data.frame` including that sample annotation. This can be read from disk or constructed on-the-fly for flexibility.

If used, a *SampleID* column must be present and match the sample IDs used in the BeadStudio export file. Also, if a column called *SampleLabel* is present in the data frame and it includes unique names, the values from that column will be used for the sampleNames of the resulting *MethyLumiSet*.

Two different formats can be read by methylumi. The "Final Report" format includes all the data in a single file. The package will look "[Header]" in the file to determine when that file format is to be used. The data block "[Sample Methylation Profile]" needs to be present in the "Final Report" format. If the data block "[Control Probe Profile]" is present, these data will be included in the *QCdata* of the resulting *MethyLumiSet* object. The second format can be a simple tab-delimited text file with headers from BeadStudio. If this format is used, the sample data and the QC data can be in separate files. The QC data need not be present for either format, but it can be helpful for quality control of data. For the examples in this vignette, a small sample set run on the Illumina GoldenGate platform will be used and the file format is the tab-delimited format.

```
suppressPackageStartupMessages(library(methylumi,quietly=TRUE))
samps <- read.table(system.file("extdata/samples.txt",
                                package = "methylumi"),sep="\t",header=TRUE)
mldat <- methylumiR(system.file('extdata/exampledata.samples.txt',package='methylumi'),
                    qcfile=system.file('extdata/exampledata.controls.txt',package="methy
                    sampleDescriptions=samps)
```

Only a subset of an entire plate is included here for illustration purposes. The `mldat` object now contains the data (in an *eSet*-like object) and quality control information (available as `QCdata(mldat)`, also an *eSet*-like object) from a set of experiments. The details of what was loaded can be viewed:

```
mldat
```

```
##
## Object Information:
## MethyLumiSet (storageMode: lockedEnvironment)
## assayData: 1536 features, 10 samples
##   element names: Avg_NBEADS, BEAD_STDERR, betas, methylated, pvals, unmethylated
## protocolData: none
## phenoData
##   sampleNames: M_1 M_2 ... F_10 (10 total)
##   varLabels: sampleID SampleLabel Sample
##     Gender
##   varMetadata: labelDescription
## featureData
##   featureNames: AATK_E63_R AATK_P519_R ...
##     ZP3_P220_F (1536 total)
```

```
##   fvarLabels: TargetID ProbeID ... PRODUCT
##     (17 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation:
## Major Operation History:
##             submitted            finished
## 1 2022-11-01 18:10:01 2022-11-01 18:10:02
##
## 1 methylumiR(filename = system.file("extdata/exampledata.samples.txt",     package =
```

Accessors for various slots of the resulting object are outlined in the online help. It is worth noting, though, that the *QCdata* will return another *eSet*-like object of class *MethyLumiQC*; the data contained here can be useful if an array has failed.

Note that the assayData names have been changed from the original column identifiers in the data file from Illumina. The mappings are available via the function `getAssayDataNameSubstitutions`.

```
getAssayDataNameSubstitutions()
```

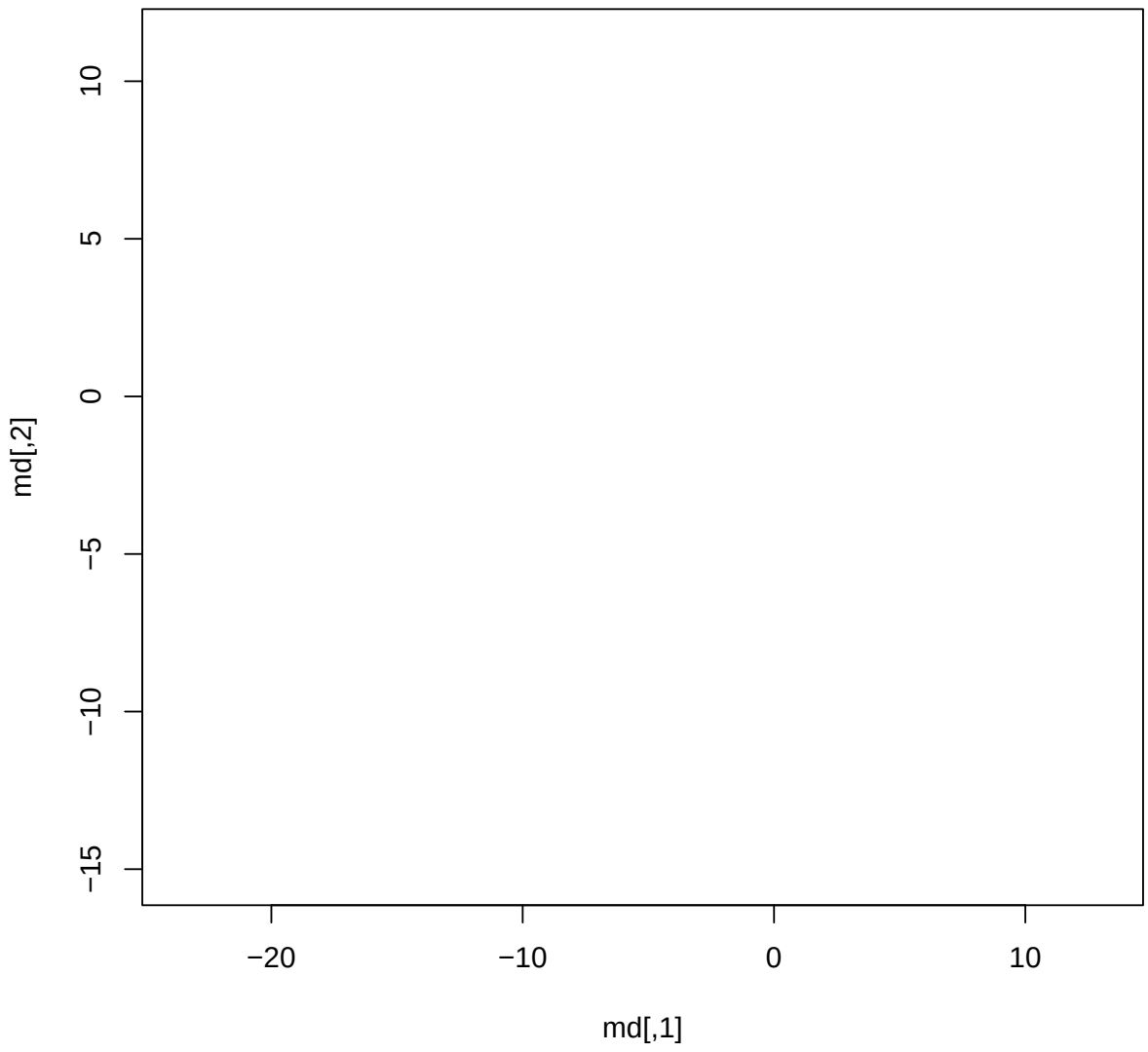
```
##            regex       result
## 1       AVG_Beta        betas
## 2 Detection Pval        pvals
## 3     Signal CY3 unmethylated
## 4     Signal CY5   methylated
## 5     Signal_Red unmethylated
## 6     Signal_Grn   methylated
## 7       Signal_A unmethylated
## 8       Signal_B   methylated
```

3 Quality Control

The data that are included with the methylumi package are all normal samples from the same tissue. The samples are labeled with the presumed gender. The data are meant to be illustrative of some typical quality-control and sample-labeling problems. In order to get a quick overview of the samples, it is useful to look at an MDS plot of the samples, using only probes on the X chromosome. Since females undergo X-inactivation, they should show something approximating hemi-methylation on that chromosome while males should show very little methylation on the X chromosome.

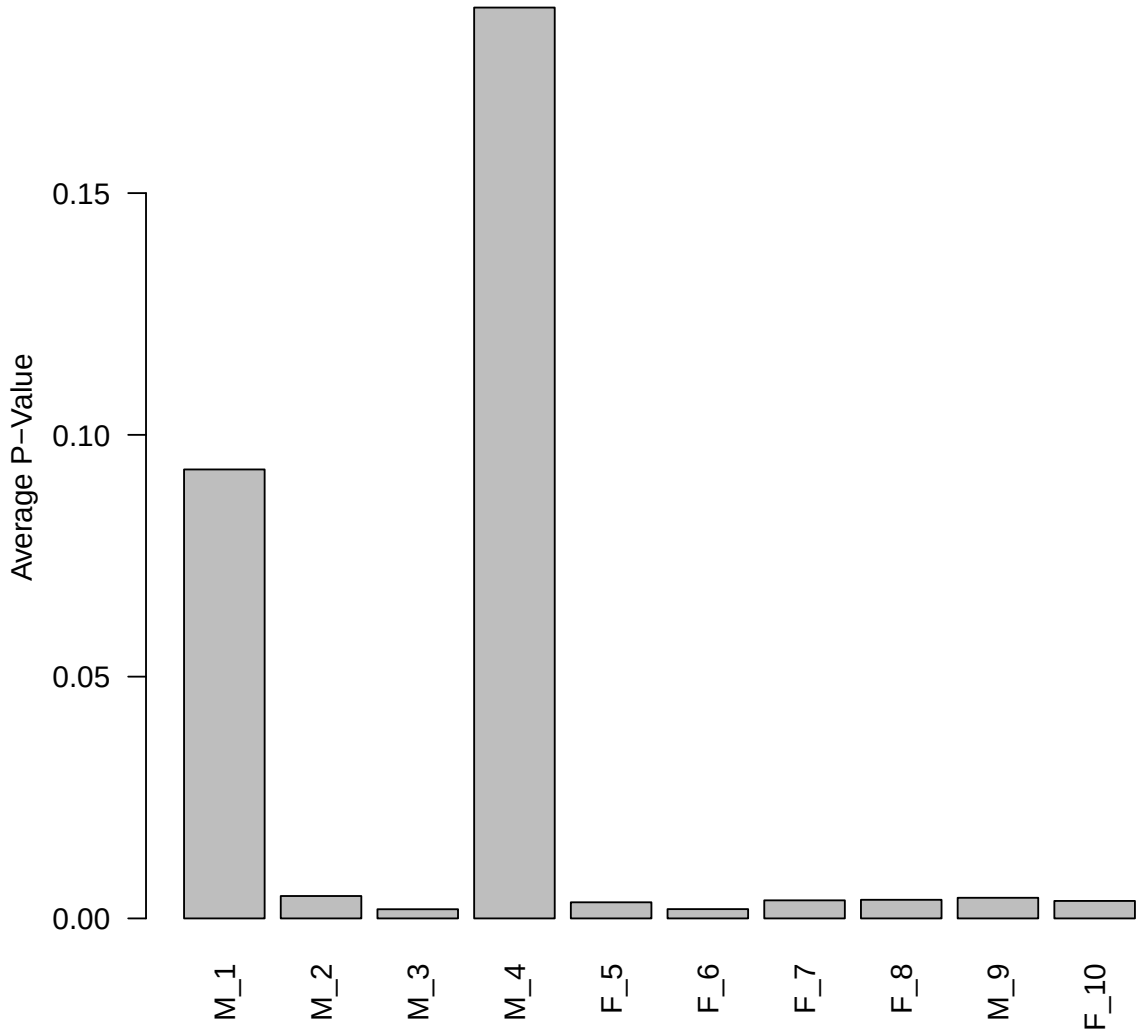
```
md <- cmdscale(dist(t(exprs(mldat)[fData(mldat)$CHROMOSOME=='X',])),2)
```

```
plot(md,pch=c('F','M')[pData(mldat)$Gender],col=c('red','blue')[pData(mldat)$Gender])
```



The MDS plot shows that the males and females are quite distinct except for a single male that groups with the females. Upon consultation with the laboratory investigator, the sample was found to be mislabeled. Also, it is worth noting that the males do not cluster nearly as tightly as the females. A quick evaluation of the p-values for detection for the samples will show what the problem is:

```
avgPval <- colMeans(pvals(mldat))
par(las=2)
barplot(avgPval,ylab="Average P-Value")
```



So, it is quite obvious that there are two arrays that fail QC with a large percentage of the reporters showing lack of measurement.

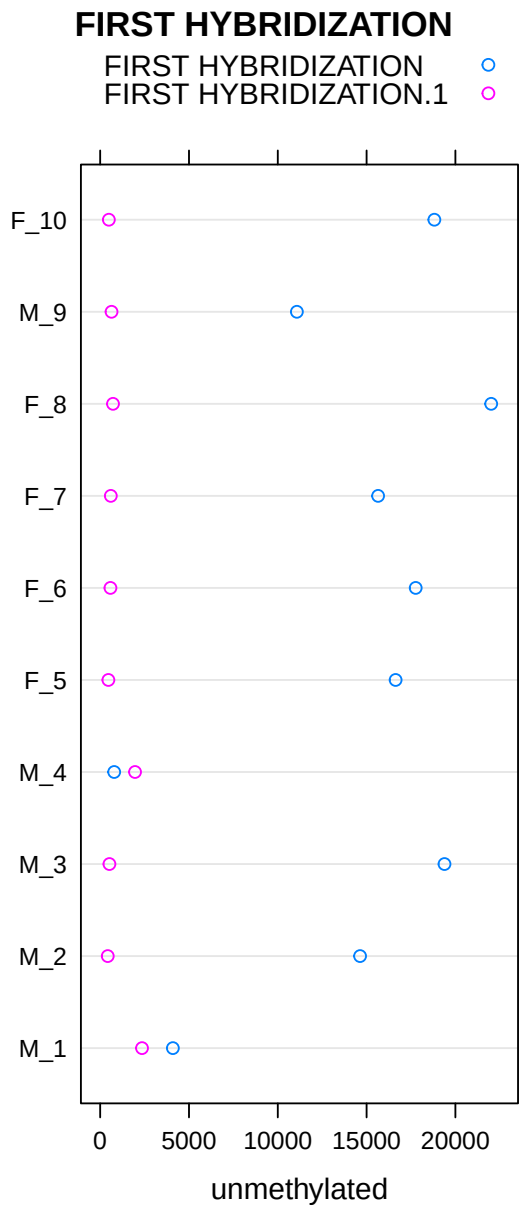
It is also possible to use the *qcplot* method in combination with *controlTypes* to examine the QC data in more detail. The control types for the GoldenGate platform are:

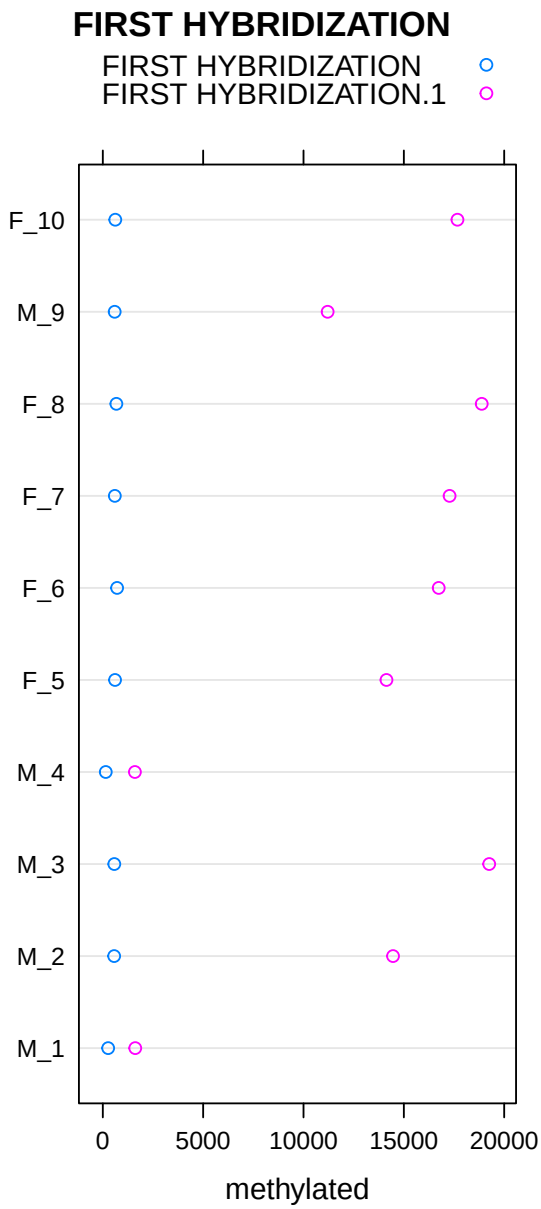
```
controlTypes(mldat)
```

```
## [1] "ALLELE SPECIFIC EXTENSION"
## [2] "EXTENSION GAP"
## [3] "FIRST HYBRIDIZATION"
## [4] "GENDER"
## [5] "NEGATIVE"
## [6] "PCR CONTAMINATION"
## [7] "SECOND HYBRIDIZATION"
```

Looking more closely at the hybridization controls ("FIRST HYBRIDIZATION") is telling here:

```
qcplot(mldat,"FIRST HYBRIDIZATION")
```





So, it appears that the hybridization controls (at least) failed for samples "M_1" and "M_4", which might help explain why the samples failed.

4 Normalization

The Illumina platform shows a significant dye bias in the two channels which will lead to bias in the estimates of Beta on the GoldenGate platform. Therefore, some normalization is required. The function, `normalizeMethyLumiSet` does this normalization. Basically, it

looks at the median intensities in the methylated and unmethylated channels (each measured in one color on the GoldenGate platform) at very low and very high beta values and sets these medians equal. Using the transformed unmethylated and methylated values, new beta values are calculated using one of two "map" functions. The `ratio` function is the default and is the same as used by Illumina in the BeadStudio software, but values using the `atan` selection should be similar.

First, a bit of cleanup is needed. The two samples with significantly poorer quality are removed. The gender of the mis-labeled sample is also corrected.

```
toKeep <- (avgPval<0.05)
pData(mldat)$Gender[9] <- "F"
mldat.norm <- normalizeMethyLumiSet(mldat[,toKeep])
```

5 Example Analysis

As a simple example of an analysis, we can look for the differences between males and females. We already know there is a strong difference based on simple unsupervized methods (the MDS plot). However, methylation is particularly informative for sex differences because females undergo X-inactivation and are, therefore, expected to have one copy of the X chromosome largely methylated. **Note that, while limma is used here to illustrate a point, an appropriate statistical framework for finding differential methylation targets based on the Illumina methylation platforms with data that is not normally distributed under the null is a current research topic for a number of groups.**

```
library(limma)

##
## Attaching package:  'limma'
## The following object is masked from 'package:BiocGenerics':
##
##     plotMA

dm <- model.matrix(~1+Gender,data=pData(mldat.norm))
colnames(dm)

## [1] "(Intercept)" "GenderM"

fit1 <- lmFit(exprs(mldat.norm),dm)
fit2 <- eBayes(fit1)
tt <- topTable(fit2,coef=2,genelist=fData(mldat.norm)[,c('SYMBOL','CHROMOSOME')],number=
x <- aggregate(tt$adj.P.Val,by=list(tt$CHROMOSOME),median)
colnames(x) <- c('Chromosome','Median adjusted P-value')
```

```
library(xtable)
xt <- xtable(x,label="tab:chromosomepvals",caption="The median adjusted p-value for each
digits(xt) <- 6
print(xt,include.rownames=FALSE,align="cr")
```

Chromosome	Median adjusted P-value
1	0.998947
10	0.998947
11	0.998947
12	0.998947
13	0.998947
14	0.998947
15	0.998947
16	0.998947
17	0.998947
18	0.998947
19	0.998947
2	0.998947
20	0.998947
21	0.998947
22	0.998947
3	0.998947
4	0.998947
5	0.998947
6	0.998947
7	0.998947
8	0.998947
9	0.998947
X	0.000114

Table 1: The median adjusted p-value for each chromosome showing that the X-chromosome is highly significantly different between males and females

Looking at the median adjusted p-values for the resulting differences (calculated using limma), one can quickly see that the X chromosome is, indeed, quite significantly different, on the whole, between males and females. The actual p-values are plotted to show the distribution.

6 sessionInfo

```
toLatex(sessionInfo())
```

- R version 4.2.1 Patched (2022-07-09 r82577), `x86_64-apple-darwin17.0`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_GB/en_US.UTF-8`
- Running under: `macOS Big Sur ... 10.16`
- Matrix products: default
- BLAS: `/Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib`
- LAPACK: `/Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.60.0, Biobase 2.58.0, BiocGenerics 0.44.0, Biostrings 2.66.0, FDb.InfiniumMethylation.hg19 2.2.0, GenomeInfoDb 1.34.0, GenomicFeatures 1.50.0, GenomicRanges 1.50.0, IRanges 2.32.0, MatrixGenerics 1.10.0, S4Vectors 0.36.0, SummarizedExperiment 1.28.0, TxDb.Hsapiens.UCSC.hg19.knownGene 3.2.2, XVector 0.38.0, bumphunter 1.40.0, foreach 1.5.2, ggplot2 3.3.6, iterators 1.0.14, knitr 1.40, lattice 0.20-45, limma 3.54.0, locfit 1.5-9.6, matrixStats 0.62.0, methylumi 2.44.0, minfi 1.44.0, org.Hs.eg.db 3.16.0, reshape2 1.4.4, scales 1.2.1, xtable 1.8-4
- Loaded via a namespace (and not attached): BiocFileCache 2.6.0, BiocIO 1.8.0, BiocParallel 1.32.0, DBI 1.1.3, DelayedArray 0.24.0, DelayedMatrixStats 1.20.0, GEOquery 2.66.0, GenomeInfoDbData 1.2.9, GenomicAlignments 1.34.0, HDF5Array 1.26.0, KEGGREST 1.38.0, MASS 7.3-58.1, Matrix 1.5-1, R6 2.5.1, RColorBrewer 1.1-3, RCurl 1.98-1.9, RSQLite 2.2.18, Rcpp 1.0.9, Rhdf5lib 1.20.0, Rsamtools 2.14.0, XML 3.99-0.12, annotate 1.76.0, askpass 1.1, assertthat 0.2.1, base64 2.0.1, beanplot 1.3.1, biomaRt 2.54.0, bit 4.0.4, bit64 4.0.5, bitops 1.0-7, blob 1.2.3, cachem 1.0.6, cli 3.4.1, codetools 0.2-18, colorspace 2.0-3, compiler 4.2.1, crayon 1.5.2, curl 4.3.3, data.table 1.14.4, dbplyr 2.2.1, digest 0.6.30, doRNG 1.8.2, dplyr 1.0.10, ellipsis 0.3.2, evaluate 0.17, fansi 1.0.3, fastmap 1.1.0, filelock 1.0.2, genefilter 1.80.0, generics 0.1.3, glue 1.6.2, grid 4.2.1, gtable 0.3.1, highr 0.9, hms 1.1.2, httr 1.4.4, illuminaio 0.40.0, lifecycle 1.0.3, magrittr 2.0.3, mclust 6.0.0, memoise 2.0.1, multtest 2.54.0, munsell 0.5.0, nlme 3.1-160, nor1mix 1.3-0, openssl 2.0.4, pillar 1.8.1, pkgconfig 2.0.3, plyr 1.8.7, png 0.1-7, preprocessCore 1.60.0, prettyunits 1.1.1, progress 1.2.2, purrr 0.3.5, quadprog 1.5-8, rappdirs 0.3.3, readr 2.1.3, reshape 0.8.9, restfulr 0.0.15, rhdf5 2.42.0, rhdf5filters 1.10.0, rjson 0.2.21, rlang 1.0.6, rngtools 1.5.2, rtracklayer 1.58.0, scrime 1.3.5, siggenes 1.72.0,

sparseMatrixStats 1.10.0, splines 4.2.1, stringi 1.7.8, stringr 1.4.1, survival 3.4-0, tibble 3.1.8, tidyr 1.2.1, tidyselect 1.2.0, tools 4.2.1, tzdb 0.3.0, utf8 1.2.2, vctrs 0.5.0, withr 2.5.0, xfun 0.34, xml2 1.3.3, yaml 2.3.6, zlibbioc 1.44.0

```
print(xyplot(-log10(adj.P.Val)~CHROMOSOME,
       tt,ylab="-log10(Adjusted P-value)",
       main="P-values for probes\ndistinguising males from females"))
      ## Warning in order(as.numeric(x)):  NAs introduced by coercion
    ## Warning in diff(as.numeric(x[ord])):  NAs introduced by coercion
     ## Warning in (function (x, y, type = "p", groups = NULL, pch = if
     (is.null(groups)) plot.symbol$pch else superpose.symbol$pch, :  NAs
                            introduced by coercion
```

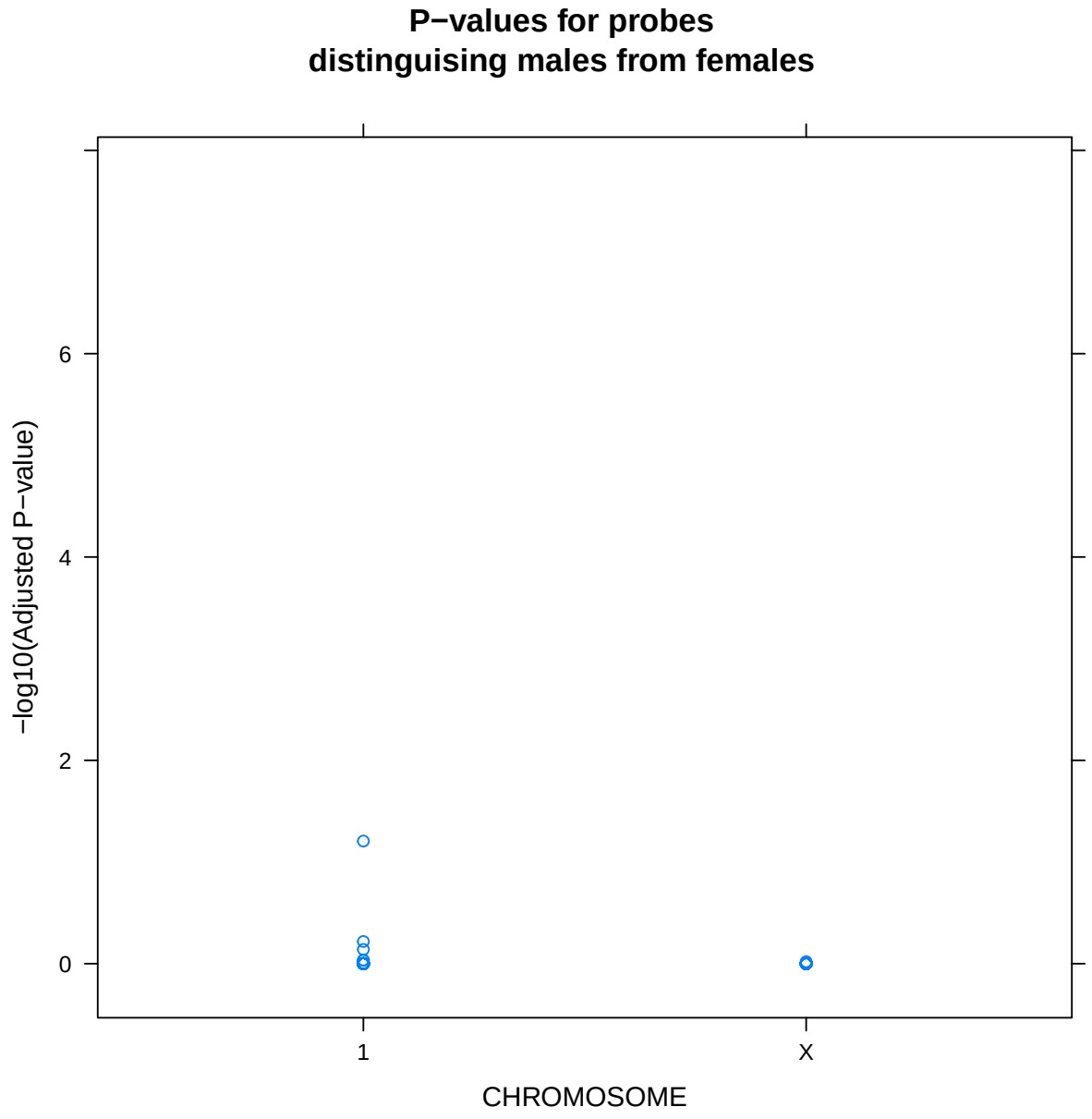


Figure 1: Probes differentially methylated plotted by chromosome. Note that the p-values plotted here are based on a linear model. Since the underlying data are not normally distributed, the p-values representing the outcomes of the linear models are not exact.