Differential methylation analysis of reduced representation bisulfite sequencing experiments using edgeR

Yunshun Chen¹,², Bhupinder Pal¹,², Jane E. Visvader¹,², and Gordon K. Smyth¹,³

¹The Walter and Eliza Hall Institute of Medical Research, 1G Royal Parade, Parkville, Victoria 3052, Australia
²Department of Medical Biology, The University of Melbourne, Victoria 3010, Australia
³School of Mathematics and Statistics, The University of Melbourne, Victoria 3010, Australia

Abstract
Cytosine methylation is an important DNA epigenetic modification. In vertebrates, methylation occurs at CpG sites, which are dinucleotides where a cytosine (C) is immediately followed by a guanine (G) in the DNA sequence from 5’ to 3’. When located in a gene promoter, DNA methylation is often associated with transcriptional silencing and with down-regulation of the gene. Aberrant DNA methylation is associated with the development of diseases such as cancer. One of the most commonly used technologies of studying DNA methylation is bisulfite sequencing (BS-seq), which can be used to measure genomewide methylation levels on the single-nucleotide scale. Notably, BS-seq can also be combined with enrichment strategies, such as reduced representation bisulfite sequencing (RRBS), to target CpG-rich regions in order to save per-sample costs. A typical DNA methylation analysis involves identifying differentially methylated regions (DMRs) between different experimental conditions. Many statistical methods have been developed for finding DMRs in BS-seq data. In this workflow, we propose a novel approach of detecting DMRs using edgeR. By providing a complete analysis of RRBS profiles of epithelial populations in the mouse mammary gland, we will demonstrate that differential methylation analyses can be fit into the existing pipelines previously developed for RNA-seq differential expression studies.

The edgeR generalized linear model framework offers great flexibility for analyzing complex experimental designs, while still accounting for the biological variability. The analysis approach illustrated in this article can be applied to any BS-seq data that includes some replication, but it is especially appropriate for RRBS data with small numbers of biological replicates.
Introduction

Cytosine methylation is an important epigenetic DNA modification that is generally associated with transcriptional silencing [1]. In vertebrates, methylation occurs at CpG sites, which are dinucleotides where a cytosine (C) is immediately followed by a guanine (G) in the DNA sequence from 5’ to 3’. CpG dinucleotides are relatively uncommon on the human genome but occur more frequently in gene promoters and exons [2]. About 72% of human gene promoters are enriched for CpGs [2]. CpGs in gene promoters tend to cluster in CpG islands (CGIs), which are regions of a few hundred to a couple of thousand base pairs with very strong enrichment of CpGs [3].

The relationship of DNA methylation to transcription in vertebrates is complex [4]. Methylation of CGIs causes robust transcriptional repression and is required for long-term mono-allelic silencing including X inactivation and genomic imprinting [1]. Methylation of CpG-poor promoters is more weakly associated with gene expression, and the mechanisms by which this occurs are unclear [1]. Methylation of gene bodies on the other hand may be positively associated with gene expression [5, 4]. DNA methylation is relatively stable in that most CGIs do not change methylation state during normal cell development. Nevertheless DNA methylation is understood to play a regulatory role in differentiation and commitment in adult cell lineages [6, 7]. Aberrant methylation patterns are also associated with the development of diseases such as cancer [8, 9]. Bisulfite sequencing (BS-seq) is increasingly used to profile DNA methylation [10, 11]. Unmethylated cytosines (C) are converted to Uracils (U) by sodium bisulfite and then deaminated to thymines (T) during PCR amplification. Methylated Cs, on the other hand, remain intact after bisulfite treatment. This strategy produces whole genome bisulfite sequencing (WGBS) when combined with sequencing of the entire genome. WGBS is sometimes considered the “gold standard” for methylation profiling because it provides single-nucleotide resolution and whole-genome coverage [11]. However it requires large quantities of DNA and is expensive because of the amount of sequencing required.

The fact that CpG islands constitute only a small percentage of the genome makes the WGBS approach inefficient in terms of information content per sequenced read. To improve efficiency and reduce costs, enrichment strategies have been developed and combined with BS-seq to target a specific fraction of the genome. A common targeted approach is reduced representation bisulfite sequencing (RRBS) that targets CpG-rich regions [12, 11]. Under the RRBS strategy, small fragments that compose only 1% of the genome are generated using MspI digestion, which means fewer reads need to be sequenced in total to provide reasonable coverage of the targeted regions. The RRBS approach can capture approximately 70% of gene promoters and 85% of CpG islands, while requiring only small quantities of input sample [13]. RRBS has great advantages in cost and efficiency, especially when CGI or gene-orientated results are required. RRBS is also applicable for single cell studies [14].

The first step of analyzing BS-seq data is to align short sequence reads to a reference genome. The number of C-to-T conversions are then counted for all the mapped reads. A number of software tools have been developed to facilitate read mapping and methylation calling, including Bismark [15], MethylCoder [16], BRAT [17], BSeeker [18] and BSMAP [19]. Most of these tools rely on existing short read aligners, such as Bowtie [20, 21]. Typical downstream DNA methylation studies often involve finding differentially methylated regions (DMRs) between different experimental conditions. A number of statistical methods and software packages have been developed for detecting DMRs using the BS-seq technology. methylkit [22] and RnBeads [23] implement Fisher’s Exact Test, which is a popular choice for two-group comparisons with no replicates. In the case of complex experimental designs, regression methods are widely used to model methylation levels or read counts. RnBeads offers a linear regression approach based on the moderated t-test and empirical Bayes method implemented in limma [24]. BSmooth [25] is another analysis pipeline that uses linear regression and empirical Bayes together with a local likelihood smoother. methylkit also has an option to apply logistic regression with overdispersion correction [22]. Some other methods have been developed based on beta-binomial distribution to achieve better variance modelling. For example, DSS fits a Bayesian hierarchical beta-binomial model to BS-seq data and uses Wald tests to detect DMRs [26]. Other software using beta-binomial model include BisSeq [27], MAOBIS [28] and RADmeth [29].

In this workflow, we demonstrate an edgeR approach of differential methylation analysis. edgeR is one of the most popular Bioconductor packages for assessing differential expression in RNA-seq data [30, 31]. It is based on the negative binomial (NB) distribution and it models the variation between biological replicates through the NB dispersion parameter. Unlike other approaches to methylation sequencing data, the analysis explained in this workflow keeps the counts for methylated and unmethylated reads as separate observations. edgeR linear models are used to fit the total read count (methylated plus unmethylated) at each genomic locus, in such a way that the proportion of methylated reads at each locus is modelled indirectly as an over-dispersed binomial-like distribution. This approach has a number of advantages. First, it allows the differential methylation analysis to be undertaken using existing edgeR pipelines developed originally for RNA-seq differential expression analyses. The edgeR generalized linear model (GLM) framework offers great flexibility for analysing complex experimental designs while still accounting for the biological variability [32]. Second, keeping methylated and unmethylated read count as separate data observations allows the inherent variability of the data to be modeled more directly and perhaps more realistically. Differential methylation is assessed by likelihood ratio tests so we do not need to assume that the log-fold-changes or other coefficient estimators are normally distributed.
This article presents an analysis of an RRBS data set generated by the authors containing replicated RRBS profiles of basal and luminal cell populations from the mouse mammary epithelium. As with other articles in the Bioconductor Gateway series, our aim is to provide an example analysis with complete start to finish code. As with other Bioconductor workflow articles, we illustrate one analysis strategy in detail rather than comparing different pipelines. The analysis approach illustrated in this article can be applied to any BS-seq data that includes some replication, but is especially appropriate for RRBS data with small numbers of biological replicates. The results shown in this article were generated using Bioconductor Release 3.6.

The next section gives an expository introduction to the edgeR approach to methylation data. The analysis of the mammary epithelial data starts afterwards.

**Introducing the NB linear modeling approach to BS-seq data**

**A very small example**

To introduce the edgeR linear modeling approach to BS-seq data, consider a genomic locus that has $m_A$ methylated and $u_A$ unmethylated reads in condition A and $m_B$ methylated and $u_B$ unmethylated reads in condition B. Our approach is to model all four counts as NB distributed with the same dispersion but different means. Suppose the data is as given in Table 1. If this were a complete data set, then it could be analyzed in edgeR as follows.

```r
> library(edgeR)
Loading required package: limma
> counts <- matrix(c(2,12,11,0),1,4)
> dimnames(counts) <- list("Locus", c("A.Me","A.Un","B.Me","B.Un"))
> counts
   A.Me A.Un B.Me B.Un
Locus  2   12   11    0
```

```r
> design <- cbind(Sample1 = c(1,1,0,0),
+                  Sample2 = c(0,0,1,1),
+                  MvsU = c(1,0,1,0),
+                  BvsA.MvsU = c(0,0,1,0))
> fit <- glmFit(counts, design, lib.size=c(100,100,100,100), dispersion=0.0247)
> lrt <- glmLRT(fit, coef="BvsA.MvsU")
> topTags(lrt)

<table>
<thead>
<tr>
<th>Coefficient: BvsA.MvsU</th>
</tr>
</thead>
<tbody>
<tr>
<td>logFC</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>A.Me</td>
</tr>
</tbody>
</table>
```

In this analysis, the first two coefficients are used to model the total number of reads (methylated or unmethylated) for samples 1 and 2, respectively. Coefficient 3 ($\text{MvsU}$) estimates the log ratio of methylated to unmethylated reads for condition A, a quantity that can also be viewed as the logit proportion of methylated reads in condition A. Coefficient 4 ($\text{BvsA.MvsU}$) estimates the difference in logit proportions of methylated reads between conditions B and A. The difference in logits is estimated here as 8.99 on the log2 scale. The $P$-value for differential methylation in condition B vs condition A is $P = 5.27 \times 10^{-6}$.

Note that the specific parametrization for the sample effects does not matter. The test for differential methylation would be unchanged if we represented the two sample effects using an intercept model. Indeed, the first two columns of the design matrix could be any two columns spanning the sample effects:
**Table 2.** A slightly less small example data set with replicates.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Condition</th>
<th>Methylated Count</th>
<th>Unmethylated Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>15</td>
<td>3</td>
</tr>
</tbody>
</table>

**Coefficient:** BvsA.MvsU

<table>
<thead>
<tr>
<th>logFC</th>
<th>logCPM</th>
<th>LR</th>
<th>PValue</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.99</td>
<td>16.3</td>
<td>20.7</td>
<td>5.27e-06</td>
<td>5.27e-06</td>
</tr>
</tbody>
</table>

The dispersion parameter controls the degree of biological variability [32]. If we had set `dispersion=0` in the above code, then the above analysis would be exactly equivalent to a logistic binomial regression, with the methylated counts as responses and the total counts as sizes, and with a likelihood ratio test for a difference in proportions between conditions A and B. Positive values for the dispersion produce over-dispersion relative to the binomial distribution. We have set the dispersion here equal to the value that is estimated below for the mammary epithelial data.

In the above code, the two library sizes for each sample should be equal. Otherwise, the library size values are arbitrary and any settings would lead to the same P-value.

**Relationship to beta-binomial modeling**

It is interesting to compare this approach with beta-binomial modeling. It is well known that if \( m \) and \( u \) are independent Poisson random variables with means \( \mu_m \) and \( \mu_u \), then the conditional distribution of \( m \) given \( m + u \) is binomial with success probability \( p = \mu_m / (\mu_m + \mu_u) \). If the Poisson means \( \mu_m \) and \( \mu_u \) themselves follow gamma distributions, then the marginal distributions of \( m \) and \( u \) are NB instead of Poisson. If the two NB distributions have different dispersions, and have expected values in inverse proportion to the dispersions, then the conditional distribution of \( m \) given \( m + u \) follows a beta-binomial distribution. The approach taken in this article is closely related to the beta-binomial approach but makes different and seemingly more natural assumptions about the NB distributions. We instead assume the two NB distributions to have the same dispersion but different means. The NB linear modeling approach allows the means and dispersions of the two NB distributions to be estimated separately, in concordance with the data instead of being artificially linked.

**A small example with replicates**

`edgeR` linear modeling takes advantage of replicate libraries for each condition. Now we augment the small example above to include two replicates for each experimental condition. There are now four samples and eight counts (Table 2). The eight counts can be represented in R as follows:

```r
> counts <- matrix(c(2,12,4,20,11,0,15,3),1,8)
> row.names(counts) <- "Locus"
> counts
Locus  2   12  4   20  11   0  15   3
```

We can represent the sample information associated with the counts as follows:

```r
> Sample <- gl(4,2,8)
> Methylation <- rep(c("Me","Un"), 4)
> Condition <- gl(2,4,8, labels=c("A","B"))
> SampleInfo <- data.frame(Sample,Methylation,Condition)
> row.names(SampleInfo) <- colnames(counts)
> SampleInfo

<table>
<thead>
<tr>
<th>Sample Methylation Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1.Me</td>
</tr>
<tr>
<td>A1.Un</td>
</tr>
<tr>
<td>A2.Me</td>
</tr>
<tr>
<td>A2.Un</td>
</tr>
<tr>
<td>B1.Me</td>
</tr>
<tr>
<td>B1.Un</td>
</tr>
</tbody>
</table>
```

We can represent the sample information associated with the counts as follows:
The design matrix is a simple combination of two parts. First there is a matrix to model the sample coverages:

```
> design.samples <- model.matrix(~0+Sample)
```

Then there is a matrix to model the two treatment conditions.

```
> design.conditions <- model.matrix(~Condition)
```

The full matrix is constructed from the two parts, with the condition design matrix mediating the methylation effects:

```
> design <- cbind(design.samples, (Methylation=="Me") * design.conditions)
> design

Sample1 Sample2 Sample3 Sample4 (Intercept) ConditionB
1 1 0 0 0 1 0
2 1 0 0 0 0 0
3 0 1 0 0 1 0
4 0 1 0 0 0 0
5 0 0 1 0 1 1
6 0 0 1 0 0 0
7 0 0 0 1 1 1
8 0 0 0 1 0 0
```

Now we can fit the GLM and conduct the statistical test for differential methylation:

```
> fit <- glmFit(counts, design, lib.size=100, dispersion=0.0247)
> lrt <- glmLRT(fit, coef="ConditionB")
> topTags(lrt)

Coefficient: ConditionB

<table>
<thead>
<tr>
<th>Locus</th>
<th>logFC</th>
<th>logCPM</th>
<th>LR</th>
<th>PValue</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Me</td>
<td>5.4</td>
<td>16.6</td>
<td>34.3</td>
<td>4.84e-09</td>
<td>4.84e-09</td>
</tr>
</tbody>
</table>
```

The results for each condition have been averaged over the replicates. The change in methylation is now of smaller magnitude than for Table 1, as shown by the smaller log FC value, but the P-value is more significant because of the greater confidence that comes from the extra replicates.

**Automatic construction of the design matrix**

In this expository introduction we have demonstrated the construction of the design matrix from basic principles. In practice, edgeR provides a function `modelMatrixMeth` to automate the construction of the design matrix. The user can simply specify the treatment conditions associated with each DNA sample (as in the second column of Table 2):

```
> Condition <- factor(c("A","A","B","B"))
```

Then the design matrix is constructed by:

```
> design <- modelMatrixMeth(~Condition)
> design

Sample1 Sample2 Sample3 Sample4 (Intercept) ConditionB
1 1 0 0 0 1 0
2 1 0 0 0 0 0
3 0 1 0 0 1 0
4 0 1 0 0 0 0
5 0 0 1 0 1 1
6 0 0 1 0 0 0
7 0 0 0 1 1 1
8 0 0 0 1 0 0
```
Table 3. A hypothetical paired-samples RNA-seq experiment.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Group</th>
<th>Treated</th>
<th>Untreated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>15</td>
<td>3</td>
</tr>
</tbody>
</table>

Alternatively, a user can construct any appropriate design matrix at the sample level (with 4 rows), exactly as one would do for a RNA-seq differential expression analysis, then the function will expand the sample-level design matrix to the appropriate observation-level matrix with 8 rows:

```r
> designSL <- model.matrix(~Condition)
> design <- modelMatrixMeth(designSL)
```

This gives exactly the same result as above.

**Analogy with paired-samples expression analyses**

The above design matrix might seem as if it is very special to methylation data, but in fact the same sort of design matrix would be used for any gene expression analysis when there are paired-samples and we wish to compare treatment effects between groups. To see this, consider the small RNA-seq experiment shown in Table 3. This experiment has four subjects, each of whom contributes both treated and untreated RNA samples. The subjects belong to two groups (A and B) and we wish to test whether the treatment effect differs between the groups.

Readers will note that Table 3 is the same as Table 2 except that the columns have been relabeled. Here, the four subjects are analogous to the four DNA samples in Table 2, the groups are analogous to conditions, and treatment is analogous to methylation. Exactly the same design matrix, as used above for the methylation experiment, would be appropriate here for the expression experiment. In edgeR, the only difference between the methylation and expression analyses would be in the normalization steps. The linear modeling and testing steps would be identical.

This shows that BS-seq data has a structure that is already familiar from RNA-seq experiments with paired-samples. The theme of this article is that statistical analysis methods developed for RNA-seq can be beneficially applied to BS-seq data.

**Use of contrasts**

edgeR has the ability to make very general comparisons between the linear model parameters. Another way to construct the design matrix is as follows:

```r
> design <- modelMatrixMeth(~0+Condition)
> design
```

| Sample1 Sample2 Sample3 Sample4 ConditionA ConditionB |
|-----------|-----------|-----------|-----------|-----------|-----------|
| 1         | 1         | 0         | 0         | 0         | 1         | 0         |
| 2         | 1         | 0         | 0         | 0         | 0         | 0         |
| 3         | 0         | 1         | 0         | 0         | 1         | 0         |
| 4         | 0         | 1         | 0         | 0         | 0         | 0         |
| 5         | 0         | 0         | 1         | 0         | 0         | 1         |
| 6         | 0         | 0         | 1         | 0         | 0         | 0         |
| 7         | 0         | 0         | 0         | 1         | 0         | 1         |
| 8         | 0         | 0         | 0         | 1         | 0         | 0         |

In this formulation, the linear model parameters **conditionA** and **conditionB** now estimate the average methylation levels in conditions A and B separately. We can then take an explicit contrast between conditions A and B to compare the methylation levels:

```r
> fit <- glmFit(counts, design, lib.size=100, dispersion=0.0247)
> Contrast <- makeContrasts(ConditionB - ConditionA, levels=design)
> lrt <- glmLRT(fit, contrast=Contrast)
> topTags(lrt)
```

<table>
<thead>
<tr>
<th>Coefficient: -1<em>ConditionA 1</em>ConditionB</th>
</tr>
</thead>
<tbody>
<tr>
<td>logPC</td>
</tr>
<tr>
<td>Locus</td>
</tr>
</tbody>
</table>

Again, this gives the same results as the previous analysis.
NB GLMs
In practice, the data will consist of many thousands of genomic loci and the counts matrix will have a row for each locus. The function `glmFit` fits a NB GLM to each row of read counts. If \( y_{gi} \) is the \( i \)th read count for genomic locus \( g \), then \( y_{gi} \) is assumed to be NB distributed with expected value \( \mu_{gi} \) and dispersion \( \phi_g \). The expected values are modelled by the design matrix,

\[
\log \mu_{gi} = \sum_{j=1}^{p} x_{ij} \beta_{gj} + \log N_i \tag{1}
\]

where the \( x_{ij} \) are the \( i \)th row of the design matrix, \( \beta_{gj} \) are the regression coefficients and \( N_i \) are the library sizes. In the small example above with four samples, the design matrix had 6 columns, so \( p = 6 \). In general, if we have an experiment comparing \( k \) treatment conditions using \( n \) samples, then the count matrix will have \( 2n \) columns and the design matrix will have \( p = n + k \) columns. The first \( n \) coefficients capture sample effects in read abundance, while the remaining \( k \) coefficients compare the methylation proportions between the conditions.

Dispersion estimation
In the small example above we did not estimate the dispersion but simply used a preset value. In practice, there will not only be replicates for each genomic locus but also many thousands of loci. The `edgeR` package estimates a dispersion for each locus from the replicate variability, using an empirical Bayes procedure to pool information across all the loci while estimating locus-specific dispersion estimates\[33, 32, 34\]. The full data example on mouse mammary epithelial cells will illustrate this.

The variance of the read counts is assumed to follow a quadratic mean-variance relationship

\[
\text{var}(y_{gi}) = \mu_{gi} + \phi_g \mu_{gi}^2
\]

where \( \phi_g \) is the dispersion for locus \( g \). The first term follows from sequencing variability and the second from biological variability \[32\]. The NB variance function captures the fact that larger counts have larger variances. The quadratic relationship captures the technical factors that affect variability, so that \( \phi_g \) itself reflects only the biological characteristics of each locus.

RRBS profiling of mammary epithelial cells
Aim of the study
We now describe the main data set to be analyzed in this article. The epithelium of the mammary gland exists in a highly dynamic state, undergoing dramatic morphogenetic changes during puberty, pregnancy, lactation and regression [35]. Characterization of the lineage hierarchy of cells in the mammary epithelium is an important step toward understanding which cells are predisposed to oncogenesis. In this study, we profiled the methylation status of the two major functionally distinct epithelial compartments: basal and luminal cells. The basal cells were further divided into those showing high or low expression of the surface marker Itga5 as part of our investigation of heterogeneity within the basal compartment. We carried out global RRBS DNA methylation assays on two biological replicates of each of the three cell populations to determine whether the epigenetic machinery played a potential role in (i) differentiation of luminal cells from basal and (ii) any compartmentalization of the basal cells associated with Itga5.

Sample preparation
Inguinal mammary glands (minus lymph node) were harvested from FVB/N mice. All animal experiments were conducted using mice bred at and maintained in our animal facility according to the Walter and Eliza Hall Institute of Medical Research Animal Ethics Committee guidelines. Epithelial cells were suspended and fluorescence-activated cell sorting (FACS) was used to isolate basal and luminal cell populations [36]. Genomic DNA (gDNA) was extracted from freshly sorted cells using the Qiagen DNeasy kit. Around 25ng gDNA input was subjected to DNA methylation analysis by BS-seq using the Ovation RRBS Methyl-seq kit from NuGEN. The process includes MspI digestion of gDNA, sequencing adapter ligation, end repair, bisulfite conversion, and PCR amplification to produce the final sequencing library. The Qiagen EpiTect Bisulfite kit was used for bisulfite-mediated conversion of unmethylated cytosines.

Experimental design
There are three groups of samples: luminal population, Itga5- basal population and Itga5+ basal population. Two biological replicates were collected for each group. This experimental design is summarized in the table below.

```r
> targets <- read.delim("targets.txt", row.names="Sample", stringsAsFactors=FALSE)
> targets
```
Population Description
P6_1  P6  Luminal
P6_4  P6  Luminal
P7_2  P7  Basal_Itga5_neg
P7_5  P7  Basal_Itga5_neg
P8_3  P8  Basal_Itga5_pos
P8_6  P8  Basal_Itga5_pos

The experiment has a simple one-way layout with three groups. The sequencing was carried out on the Illumina NextSeq 500 platform. About 30 million 75bp paired-end reads were generated for each sample.

Differential methylation analysis at CpG loci

Processing the BS-seq FASTQ files with Bismark

The first step of the analysis is to map the sequence reads from the FASTQ files to the mouse genome and to perform methylation calls. This is the only step of the analysis that cannot currently be done in R. Though many options are available, we used Bismark software (https://www.bioinformatics.babraham.ac.uk/projects/bismark/) to count the methylated and unmethylated reads at each genomic locus. Bismark was run using recommended default settings. We first ran trim_galore (https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/) to remove adapters and to trim poor quality reads. Then Bismark version v0.13.0 was used to align the reads to the mouse mm10 genome using Bowtie2 [21]. Finally, methylation calls were made using bismark_methylation_extractor.

The Bismark output consists of one coverage file for each sample. Readers wishing to reproduce the analysis presented in this article can download the coverage files produced by Bismark from http://bioinf.wehi.edu.au/edgeR/F1000Research2017.

Reading in the data

The Bismark coverage files are just tab-delimited text files and so can be read into a dataframe in R using read.delim. Each of the files has the following format:

```r
> P6_1 <- read.delim("P6_1.bismark.cov.gz", header=FALSE, nrows=6)
> P6_1
          V1  V2  V3  V4  V5  V6
1  chr6 3052156 3052156 87.9 51  7
2  chr6 3052157 3052157 85.7  6  1
3  chr6 3052246 3052246   0  0  1
4  chr6 3052415 3052415 100.0 57  0
5  chr6 3052416 3052416 100.0  7  0
6  chr6 3052434 3052434  94.7 54  3
```

The columns in the coverage file represent: V1: chromosome number; V2: start position of the CpG site; V3: end position of the CpG site; V4: methylation proportion; V5: number of methylated Cs; V6: number of unmethylated Cs.

In practice, we only need to read in columns V1, V2, V5 and V6. Column V3 is the same as V2, and the methylation proportion V4 is just a function of V5 and V6. The coverage file can be read a little more efficiently by specifying which columns to read as well as the format of each column:

```r
> P6_1 <- read.delim("P6_1.bismark.cov.gz", header=FALSE, nrows=6,
+  colClasses=c("character","integer","NULL","NULL","integer","integer"))
> P6_1
          V1  V2  V5  V6
1  chr6 3052156 51  7
2  chr6 3052157  6  1
3  chr6 3052246  0  1
4  chr6 3052415 57  0
5  chr6 3052416  7  0
6  chr6 3052434 54  3
```

We now read in the Bismark coverage files for all the samples. The resulting dataframes are stored in an R list:
> Sample <- row.names(targets)
> fn <- paste0(Sample,".bismark.cov.gz")
> DataList <- list()
> for(i in 1:length(fn)) {
+ x <- read.delim(fn[i], header=FALSE,
+ colClasses=c("character","integer","NULL","NULL","integer","integer"))
+ names(x) <- c("Chr", "Locus", "Meth", "Un")
+ row.names(x) <- paste(x$Chr, x$Locus, sep="-")
+ DataList[[i]] <- as.matrix(x[,3:4])
+ }
> head(DataList[[1]])

<table>
<thead>
<tr>
<th>Meth</th>
<th>Un</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>57</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>3</td>
</tr>
</tbody>
</table>

DataList is a list containing six dataframes, each of which represents one sample. The first and second columns of each dataframe are the chromosome numbers and positions of all the CpG loci observed in that sample. The last two columns contain the numbers of methylated and unmethylated Cs detected at those loci.

**Assembling the counts into a combined matrix**

The next step is to consolidate the counts for all six samples into one matrix. Some care is required in combining the different dataframes because the number of reported CpG loci varies across different samples. We first obtain all unique CpG loci observed in at least one of the six samples:

> Loci_all <- unique(unlist(lapply(DataList,row.names)))

Then we extract read counts of methylated and unmethylated Cs at these locations across all the samples and combine them into a count matrix:

> counts <- matrix(0L, nrow=length(Loci_all), ncol=2*length(Sample))
> rownames(counts) <- Loci_all
> j <- 1:2
> for(i in 1:length(Sample)) {
+ counts[rownames(DataList[[i]]), j] <- DataList[[i]]
+ j <- j+2
+ }

counts is a matrix of integer counts with 12 columns, two for each sample. The odd-numbered columns contain the counts for methylated Cs, whereas the even-numbered columns contain the counts for unmethylated Cs.

> Sample2 <- rep(Sample, each=2)
> Sample2 <- factor(Sample2)
> Methylation <- rep(c("Me","Un"), length(Sample))
> colnames(counts) <- paste(Sample2, Methylation, sep="-")
> head(counts)

<table>
<thead>
<tr>
<th>Sample2</th>
<th>Methylation</th>
</tr>
</thead>
<tbody>
<tr>
<td>P6_1-Me</td>
<td>P6_1-Un</td>
</tr>
<tr>
<td>P6_4-Me</td>
<td>P6_4-Un</td>
</tr>
<tr>
<td>P7_2-Me</td>
<td>P7_2-Un</td>
</tr>
<tr>
<td>P7_5-Me</td>
<td>P7_5-Un</td>
</tr>
<tr>
<td>chr6-3052156</td>
<td>51 7 62 13 48 3 31 8</td>
</tr>
<tr>
<td>chr6-3052157</td>
<td>6 1 5 0 0 0 3 1</td>
</tr>
<tr>
<td>chr6-3052246</td>
<td>0 1 0 0 0 2 0 0</td>
</tr>
<tr>
<td>chr6-3052415</td>
<td>57 0 75 1 50 1 36 1</td>
</tr>
<tr>
<td>chr6-3052416</td>
<td>7 0 5 0 0 0 4 0</td>
</tr>
<tr>
<td>chr6-3052434</td>
<td>54 3 72 4 48 3 36 1</td>
</tr>
<tr>
<td>chr6-3052156</td>
<td>40 9 28 10</td>
</tr>
</tbody>
</table>
Making a DGEList data object

The edgeR package stores the counts and associated annotation in a simple list-based data object called a DGEList. We first create a DGEList object using the count matrix generated before. The information of CpG sites is converted into a dataframe and stored in the genes component of the DGEList object.

```r
> Chr <- gsub("-.*$", "", Loci_all)
> Locus <- gsub("^.*-", "", Loci_all)
> Genes <- data.frame(Chr=Chr, Locus=as.integer(Locus), stringsAsFactors=FALSE)
> head(Genes)

     Chr  Locus
   1 chr6 3052156
   2 chr6 3052157
   3 chr6 3052246
   4 chr6 3052415
   5 chr6 3052416
   6 chr6 3052434
```

Filtering unassembled chromosomes

It is convenient to remove genomic segments that have not been assembled into any of the recognized chromosomes:

```r
> keep <- rep(TRUE, nrow(yall))
> keep[ grep("random",Chr) ] <- FALSE
> keep[ grep("chrUn",Chr) ] <- FALSE
```

For this analysis, we also remove the Y chromosome and mitochondrial DNA:

```r
> keep[Chr=="chrY"] <- FALSE
> keep[Chr=="chrM"] <- FALSE
> table(keep)

   FALSE   TRUE
  6727 3531359
```

Then we can subset the DGEList data object to remove the rows corresponding to the unwanted chromosomes:

```r
> yall <- yall[keep,, keep.lib.sizes=FALSE]
```

The option keep.lib.sizes=FALSE causes the library sizes to be recomputed.

Gene annotation

We now annotate the CpG loci with the identity of the nearest gene. We search for the gene transcriptional start site (TSS) closest to each our CpGs:

```r
> TSS <- nearestTSS(yall$genes$Chr, yall$genes$Locus, species="Mm")
> yall$genes$EntrezID <- TSS$gene_id
> yall$genes$Symbol <- TSS$symbol
> yall$genes$Strand <- TSS$strand
> yall$genes$Distance <- TSS$distance
> yall$genes$Width <- TSS$width
> head(yall$genes)

     Chr  Locus EntrezID Symbol Strain Distance  Width
  chr6-3052156  chr6 3052156 667335 Gm8579   + 236363 285.01
  chr6-3052157  chr6 3052157 667335 Gm8579   + 236362 285.01
  chr6-3052246  chr6 3052246 667335 Gm8579   + 236273 285.01
  chr6-3052415  chr6 3052415 667335 Gm8579   + 236104 285.01
  chr6-3052416  chr6 3052416 667335 Gm8579   + 236103 285.01
  chr6-3052434  chr6 3052434 667335 Gm8579   + 236085 285.01
```
Here **EntrezID**, **Symbol**, **Strand** and **Width** are the Entrez Gene ID, symbol, strand and width of the nearest gene. **Distance** is the genomic distance from the CpG to the TSS. Positive values mean the TSS is downstream of the CpG and negative values mean the TSS is upstream.

### Filtering to remove low counts

CpG loci that have low coverage are removed prior to downstream analysis as they provide little information for assessing methylation levels. Filtering low-coverage CpGs also simplifies the subsequent analysis because of the reduction in data rows.

We sum up the counts of methylated and unmethylated reads to get the total read coverage at each CpG site for each sample:

```r
> Coverage <- yall$counts[, Methylation=="Me"] + yall$counts[, Methylation=="Un"]
> head(Coverage)
```

<table>
<thead>
<tr>
<th></th>
<th>P6_1-Me</th>
<th>P6_4-Me</th>
<th>P7_2-Me</th>
<th>P7_5-Me</th>
<th>P8_3-Me</th>
<th>P8_6-Me</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr6-3052156</td>
<td>58</td>
<td>75</td>
<td>51</td>
<td>39</td>
<td>49</td>
<td>38</td>
</tr>
<tr>
<td>chr6-3052157</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>chr6-3052246</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>chr6-3052415</td>
<td>57</td>
<td>76</td>
<td>51</td>
<td>37</td>
<td>47</td>
<td>36</td>
</tr>
<tr>
<td>chr6-3052416</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>chr6-3052434</td>
<td>57</td>
<td>76</td>
<td>51</td>
<td>37</td>
<td>47</td>
<td>36</td>
</tr>
</tbody>
</table>
```

As a conservative rule of thumb, we require a CpG site to have a total count (both methylated and unmethylated) of at least 10 across all the samples before it is considered in the study.

```r
> keep <- rowSums(Coverage >= 10) == 6
> table(keep)

keep
FALSE TRUE
3132913 398446
```

If necessary, this filtering criterion could be relaxed somewhat, but here the number of CpGs kept in the analysis is large enough for our purposes.

The `DGEList` object is subsetted to retain only the non-filtered loci:

```r
> y <- yall[keep,, keep.lib.sizes=FALSE]
```

Again, the option `keep.lib.sizes=FALSE` causes the library sizes to be recomputed after the filtering. We generally recommend this, although the effect on the downstream analysis is usually small.

### Normalization

A key difference between BS-seq and other sequencing data is that the pair of libraries holding the methylated and unmethylated reads for a particular sample are treated as a unit. To ensure that the methylated and unmethylated reads for the same sample are treated on the same scale, we need to set the library sizes to be equal for each pair of libraries. We set the library sizes for each sample to be the average of the total read counts for the methylated and unmethylated libraries:

```r
> TotalLibSize <- y$samples$lib.size[Methylation=="Me"] + y$samples$lib.size[Methylation=="Un"]
> y$samples$lib.size <- rep(TotalLibSize, each=2)
```

```r
> y$samples

<table>
<thead>
<tr>
<th>group</th>
<th>lib.size</th>
<th>norm.factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>P6_1-Me</td>
<td>P6 16119711</td>
<td>1</td>
</tr>
<tr>
<td>P6_1-Un</td>
<td>P6 16119711</td>
<td>1</td>
</tr>
<tr>
<td>P6_4-Me</td>
<td>P6 25813898</td>
<td>1</td>
</tr>
<tr>
<td>P6_4-Un</td>
<td>P6 25813898</td>
<td>1</td>
</tr>
<tr>
<td>P7_2-Me</td>
<td>P7 13437577</td>
<td>1</td>
</tr>
<tr>
<td>P7_2-Un</td>
<td>P7 13437577</td>
<td>1</td>
</tr>
<tr>
<td>P7_5-Me</td>
<td>P7 14882178</td>
<td>1</td>
</tr>
<tr>
<td>P7_5-Un</td>
<td>P7 14882178</td>
<td>1</td>
</tr>
<tr>
<td>P8_3-Me</td>
<td>P8 11823724</td>
<td>1</td>
</tr>
<tr>
<td>P8_6-Me</td>
<td>P8 11823724</td>
<td>1</td>
</tr>
</tbody>
</table>
```
Figure 1. MDS plots showing overall differences in methylation levels between the samples. Replicate samples from the same population cluster together. The first dimension separates the luminal cells from the basal cells. The second dimension separates the Itga+ basal cells at the bottom from the Itga- basal cells at the top.

Other normalization methods developed for RNA-seq data, such as TMM [37], are not required for BS-seq data.

Exploring differences between samples
In microarray methylation studies, a common measure of methylation level is the M-value, which is defined as

$$M = \log_2 \left( \frac{Me + \alpha}{Un + \alpha} \right)$$

where Me and Un are the methylated and unmethylated intensities and \(\alpha\) is some suitable offset to avoid taking logarithms of zero [38]. The M-value can be interpreted as the base2 logit transformation of the proportion of methylated signal at each locus.

We compute the corresponding methylation summary from the methylated and unmethylated counts.

```r
> Me <- y$counts[, Methylation=="Me"]
> Un <- y$counts[, Methylation=="Un"]
> M <- log2(Me + 2) - log2(Un + 2)
> colnames(M) <- Sample
```

Here \(M\) contains the empirical logit methylation level for each CpG site in each sample. We have used a prior count of 2 to avoid logarithms of zero. The exact value of the prior count is unimportant, but a value of 2 is common in other contexts such as RNA-seq.

Now we can generate a multi-dimensional scaling (MDS) plot to explore the overall differences between the methylation levels of the different samples (Figure 1):

```r
> plotMDS(M)
```

In this plot the distance between each pair of samples represents the average logit change between the samples for the top most differentially methylated CpG loci between that pair of samples. (We call this average the leading log-fold-change.) The two replicate samples from the luminal population (P6) are seen to be well separated from the four basal samples (populations P7 and P8), with a leading change of about 6 logit units.

Design matrix
One aim of this study is to identify differentially methylated (DM) loci between the different cell populations. In edgeR, this can be done by fitting linear models under a specified design matrix and testing for corresponding coefficients or contrasts. A basic sample-level design matrix can be made as follows:
> designSL <- model.matrix(~0+Population, data=targets)
> colnames(designSL) <- c("P6", "P7", "P8")
> designSL

    P6 P7 P8
  P6_1  1  0  0
  P6_4  1  0  0
  P7_2  0  1  0
  P7_5  0  1  0
  P8_3  0  0  1
  P8_6  0  0  1

attr(,"assign")
  [1] 1 1 1

attr(,"contrasts")
attr(,"contrasts")$Population
  [1] "contr.treatment"

The we expand this to the full design matrix modeling the sample and methylation effects:

> design <- modelMatrixMeth(designSL)
> design

       Sample1 Sample2 Sample3 Sample4 Sample5 Sample6 P6 P7 P8
  1      1      0      0      0      0      0 1  0  0
  2      1      0      0      0      0      0 0  0  0
  3      0      1      0      0      0      0 0  1  0
  4      0      1      0      0      0      0 0  0  0
  5      0      0      1      0      0      0 0  0  1
  6      0      0      1      0      0      0 0  0  0
  7      0      0      0      1      0      0 0  1  0
  8      0      0      0      1      0      0 0  0  0
  9      0      0      0      0      1      0 0  0  1
 10      0      0      0      0      0      1 0  0  0
 11      0      0      0      0      0      0 1  0  1
 12      0      0      0      0      0      0 0  0  0

The first six columns represent the sample coverage effects. The last three columns represent the methylation levels (in logit units) in the three cell populations.

Dispersion estimation
With the design matrix specified, we can now proceed to the standard edgeR pipeline and analyze the data in the same way as for RNA-seq data. Similar to the RNA-seq data, the variability between biological replicates has also been observed in bisulfite sequencing data. This variability can be captured by the NB dispersion parameter under the generalized linear model (GLM) framework in edgeR. The mean-dispersion relationship of BS-seq data has been studied in the past and no apparent mean-dispersion trend was observed\(^{[26]}\). This is also verified through our own practice. Therefore, we would not consider a mean-dependent dispersion trend as we normally would for RNA-seq data. A common dispersion estimate for all the loci, as well as an empirical Bayes moderated dispersion for each individual locus, can be obtained from the `estimateDisp` function in edgeR:

> y <- estimateDisp(y, design, trend="none")
> y$common.dispersion

[1] 0.0247

> y$prior.df

[1] Inf

This returns a DGEList object with additional components (`common.dispersion` and `tagwise.dispersion`) added to hold the estimated dispersions. Here the estimation of trended dispersion has been turned off by setting `trend="none"`. For this data, the estimated prior degrees of freedom (df) are infinite for all the loci, which implies all the CpG-wise dispersions are exactly the same as the common dispersion. A BCV plot is often useful to visualize the dispersion estimates, but is not informative in this case.
Testing for differentially methylated CpG loci

We first fit NB GLMs for all the CpG loci using the `glmFit` function in `edgeR`.

```R
> fit <- glmFit(y, design)
```

Then we can proceed to testing for differentially methylated CpG sites between different populations. One of the most interesting comparisons is between the basal (P7 and P8) and luminal (P6) populations. The contrast corresponding to any specified comparison can be constructed conveniently using the `makeContrasts` function:

```R
> contr <- makeContrasts(LvsB=P6-0.5*(P7+P8), levels=design)
```

The actual testing is performed using likelihood ratio tests (LRT) in `edgeR`:

```R
> lrt <- glmLRT(fit, contrast=contr)
```

The top set of most differentially methylated (DM) CpG sites can be viewed with `topTags`:

```R
> topTags(lrt)
```

The differential methylation results can be visualized with an MD plot (see Figure 2):

```R
> plotMD(lrt)
```
Figure 2. MD plot showing the log-fold-change of the methylation level and average abundance of each CpG site. Significantly hyper- and hypomethylated CpGs are highlighted in red and blue, respectively.

Differential methylation by chromosome

We now explore overall methylation patterns by chromosome. To do this we make a list of index vectors, of which each vector identifies which loci belong to that chromosome:

```r
ChrLev <- unique(y$genes$Chr)
ChrIndices <- list()
for (a in ChrLev) ChrIndices[[a]] <- which(y$genes$Chr==a)
```

Then we can conduct “fry” gene set tests to evaluate overall changes in methylation in each chromosome:

```r
> fry(y, index=ChrIndices, design=design, contrast=contr)
```

<table>
<thead>
<tr>
<th>Chr</th>
<th>NGenes</th>
<th>Direction</th>
<th>PValue</th>
<th>FDR</th>
<th>PValue.Mixed</th>
<th>FDR.Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr18</td>
<td>13405</td>
<td>Up</td>
<td>0.00151</td>
<td>0.00502</td>
<td>1.38e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr15</td>
<td>17322</td>
<td>Up</td>
<td>0.00164</td>
<td>0.00502</td>
<td>1.06e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr10</td>
<td>21963</td>
<td>Up</td>
<td>0.00201</td>
<td>0.00502</td>
<td>1.32e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr2</td>
<td>27787</td>
<td>Up</td>
<td>0.00238</td>
<td>0.00502</td>
<td>1.13e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr19</td>
<td>12158</td>
<td>Up</td>
<td>0.00284</td>
<td>0.00502</td>
<td>1.04e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr3</td>
<td>19637</td>
<td>Up</td>
<td>0.00286</td>
<td>0.00502</td>
<td>1.48e-04</td>
<td>1.56e-04</td>
</tr>
<tr>
<td>chr8</td>
<td>22505</td>
<td>Up</td>
<td>0.00299</td>
<td>0.00502</td>
<td>1.91e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr1</td>
<td>24190</td>
<td>Up</td>
<td>0.00305</td>
<td>0.00502</td>
<td>1.31e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr7</td>
<td>23049</td>
<td>Up</td>
<td>0.00305</td>
<td>0.00502</td>
<td>1.16e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr6</td>
<td>19442</td>
<td>Up</td>
<td>0.00319</td>
<td>0.00502</td>
<td>1.61e-04</td>
<td>1.61e-04</td>
</tr>
<tr>
<td>chr14</td>
<td>14613</td>
<td>Up</td>
<td>0.00329</td>
<td>0.00502</td>
<td>1.20e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr13</td>
<td>17883</td>
<td>Up</td>
<td>0.00351</td>
<td>0.00502</td>
<td>1.16e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr9</td>
<td>20490</td>
<td>Up</td>
<td>0.00351</td>
<td>0.00502</td>
<td>9.67e-05</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr11</td>
<td>27020</td>
<td>Up</td>
<td>0.00383</td>
<td>0.00511</td>
<td>7.77e-05</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr5</td>
<td>28884</td>
<td>Up</td>
<td>0.00415</td>
<td>0.00519</td>
<td>1.14e-04</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr12</td>
<td>26704</td>
<td>Up</td>
<td>0.00461</td>
<td>0.00542</td>
<td>9.71e-05</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chr7</td>
<td>18500</td>
<td>Up</td>
<td>0.01054</td>
<td>0.01110</td>
<td>7.76e-05</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>chrX</td>
<td>13407</td>
<td>Up</td>
<td>0.81739</td>
<td>0.81739</td>
<td>1.28e-06</td>
<td>2.57e-05</td>
</tr>
</tbody>
</table>

This shows that methylation increases overall in committed luminal cells for every chromosome except for Chromosome X. This is consistent with the expectation that methylation is associated with silencing of genes not needed in a committed lineage. The column heading NGenes gives the number of CpG loci in each chromosome.
The results for Chromosome X are interesting. The mixed P-value shows that the X chromosome is actually enriched for DM CpGs, but the direction of change is not consistent with hyper and hypomethylated CpGs balancing each other.

Differential methylation in gene promoters

Pre-defined gene promoters

The majority of CpGs are methylated in mammals. On the other hand, unmethylated CpGs tend to group into clusters of CpG islands, which are often enriched in gene promoters. CpG methylation in promoter regions is often associated with silencing of transcription and gene expression [3]. Therefore it is of great biological interest to examine the methylation level within the gene promoter regions. For simplicity, we define the promoter of a gene as the region from 2kb upstream to 1kb downstream of the transcription start site of that gene.

```r
> InPromoter <- yall$genes$Distance >= -1000 & yall$genes$Distance <= 2000
```

We subset the CpGs to those contained in a promoter region:

```r
> yIP <- yall[InPromoter,,keep.lib.sizes=FALSE]
```

Summarizing counts in promoter regions

One simple and effective way to conduct a gene-orientated analysis of the methylation changes is to collapse all the CpGs in each promoter into one locus, i.e., to compute the total number of methylated and unmethylated reads within each promoter. This strategy does not consider any possible nuances within each promoter, but instead simply looks for an overall increase or decrease in methylation for each promoter.

First we compute the total counts for each gene promoter:

```r
> o <- order(yIP$genes$Symbol)
> yIP <- yIP[o,]
> notdup <- !duplicated(yIP$genes$Symbol)
> ypr <- yIP[notdup,]
> ypr$counts <- rowsum(yIP$counts, yIP$genes$Symbol, reorder=FALSE)
> row.names(ypr) <- ypr$genes$EntrezID
> ypr$genes$Locus <- ypr$genes$Distance <- ypr$genes$EntrezID <- NULL
```

The integer matrix `counts2` contains the total numbers of methylated and unmethylated CpGs observed within the promoter of each gene. Same as before, `counts2` has 12 columns, two for each sample. The odd-numbered columns contain the numbers of methylated Cs, whereas the even-numbered columns contain the numbers of unmethylated Cs. The only difference is that each row of `counts2` now represents a gene promoter instead of an individual CpG site.

Filtering to remove low counts

Filtering is performed in the same way as before. We sum up the read counts of both methylated and unmethylated Cs at each gene promoter within each sample.

```r
> Coveragepr <- ypr$counts[,Methylation=="Me"] +
  + ypr$counts[,Methylation=="Un"]
```

Since each row represents a 3,000-bps-wide promoter region that contains multiple CpG sites, we would expect less filtering than before.

```r
> keeppr <- rowSums(Coveragepr >= 10) == 6
> table(keeppr)

keeppr
FALSE   TRUE
1906 16639
```

The same as before, we do not perform normalization but set the library sizes for each sample to be the average of the total read counts for the methylated and unmethylated libraries.
Figure 3. MDS plot showing differences in methylation profiles at gene promoters. Basal and luminal cell populations are well separated by the first dimension.

```r
> TotalLibSizepr <- 0.5*ypr$samples$lib.size[Methylation=="Me"] +
  + 0.5*ypr$samples$lib.size[Methylation=="Un"]
> ypr$samples$lib.size <- rep(TotalLibSizepr, each=2)
> ypr$samples

<table>
<thead>
<tr>
<th>group</th>
<th>lib.size</th>
<th>norm.factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>P6_1-Me</td>
<td>11502651</td>
<td>1</td>
</tr>
<tr>
<td>P6_1-Un</td>
<td>11502651</td>
<td>1</td>
</tr>
<tr>
<td>P6_4-Me</td>
<td>11619038</td>
<td>1</td>
</tr>
<tr>
<td>P6_4-Un</td>
<td>11619038</td>
<td>1</td>
</tr>
<tr>
<td>P7_2-Me</td>
<td>4733320</td>
<td>1</td>
</tr>
<tr>
<td>P7_2-Un</td>
<td>4733320</td>
<td>1</td>
</tr>
<tr>
<td>P7_5-Me</td>
<td>10321570</td>
<td>1</td>
</tr>
<tr>
<td>P7_5-Un</td>
<td>10321570</td>
<td>1</td>
</tr>
<tr>
<td>P8_3-Me</td>
<td>3815569</td>
<td>1</td>
</tr>
<tr>
<td>P8_3-Un</td>
<td>3815569</td>
<td>1</td>
</tr>
<tr>
<td>P8_6-Me</td>
<td>3295809</td>
<td>1</td>
</tr>
<tr>
<td>P8_6-Un</td>
<td>3295809</td>
<td>1</td>
</tr>
</tbody>
</table>
```

Exploring differences between samples

Same as before, we measure the methylation levels of gene promoter regions using M-values. A prior count of 2 is added to the calculation to avoid undefined values and to reduce the variability of M-values for gene promoters with low counts. Then a MDS plot is produced to examine the overall differences between the methylation levels of the different samples.

```r
> Me <- ypr$counts[, Methylation=="Me"]
> Un <- ypr$counts[, Methylation=="Un"]
> M2 <- log2(Me + 2) - log2(Un + 2)
> colnames(M2) <- Sample
> plotMDS(M2)
```

The resulting Figure 3 shows that the two replicate samples from the luminal population (P6) are well separated from the four replicate samples from the basal population (P7 and P8).

Dispersion estimation

We estimate the NB dispersions using the `estimateDisp` function in `edgeR`. For the same reason, we do not consider a mean-dependent dispersion trend as we normally would for RNA-seq data.
Figure 4. Scatterplot of the biological coefficient of variation (BCV) against the average abundance of each gene. The plot shows the square-root estimates of the common and tagwise NB dispersions.

```r
> ypr <- estimateDisp(ypr, design, trend="none")
> ypr$common.dispersion
[1] 0.0307
> ypr$prior.df
[1] 10.8
```

The dispersion estimates ($\phi_g$) can be visualized with a BCV plot (see Figure 4):

```r
> plotBCV(ypr)
```

**Testing for differential methylation in gene promoters**

We first fit NB GLMs for all the gene promoters using `glmFit`.

```r
> fitpr <- glmFit(ypr, design)
```

Then we can proceed to testing for differential methylation in gene promoter regions between different populations. Suppose the comparison of interest is same as before. The same contrast can be used for the testing.

```r
> lrtpr <- glmLRT(fitpr, contrast=contr)
```

The top set of most differentially methylated gene promoters can be viewed with `topTags`:

```r
> topTags(lrtpr, n=20)
```

<table>
<thead>
<tr>
<th>Coefficient: 1<em>P6 -0.5</em>P7 -0.5*P8</th>
<th>Chr</th>
<th>Symbol</th>
<th>Strand</th>
<th>Width</th>
<th>logFC</th>
<th>logCPM</th>
<th>LR</th>
<th>FValue</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chr5</td>
<td>Lnx1</td>
<td>-105800</td>
<td>-6.85</td>
<td>311</td>
<td>1.20e-69</td>
<td>1.99e-65</td>
<td>1.99e-65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chr12</td>
<td>Akap6</td>
<td>451633</td>
<td>-5.26</td>
<td>247</td>
<td>1.49e-55</td>
<td>1.24e-51</td>
<td>1.24e-51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chr16</td>
<td>Popdc2</td>
<td>16027</td>
<td>4.91</td>
<td>223</td>
<td>1.58e-50</td>
<td>8.77e-47</td>
<td>8.77e-47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chr8</td>
<td>Angpt2</td>
<td>51300</td>
<td>5.58</td>
<td>207</td>
<td>1.58e-47</td>
<td>1.24e-43</td>
<td>1.24e-43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chr5</td>
<td>Cldn4</td>
<td>1811</td>
<td>-5.56</td>
<td>223</td>
<td>1.58e-50</td>
<td>8.77e-47</td>
<td>8.77e-47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chr11</td>
<td>Krt19</td>
<td>2089</td>
<td>-5.05</td>
<td>186</td>
<td>2.06e-42</td>
<td>3.07e-39</td>
<td>1.01e-39</td>
<td>3.07e-39</td>
<td></td>
</tr>
<tr>
<td>Chr12</td>
<td>2210039B01Rik</td>
<td>3504</td>
<td>-4.31</td>
<td>178</td>
<td>2.06e-42</td>
<td>7.10e-39</td>
<td>7.10e-39</td>
<td>7.10e-39</td>
<td></td>
</tr>
</tbody>
</table>
Here positive log-fold-changes represent gene promoters that have higher methylation level in the luminal population compared to the basal population. The Benjamini-Hochberg multiple testing correction is applied to control the false discovery rate (FDR).

The total number of DM gene promoters identified at an FDR of 5% can be shown with `decideTests`. There are in fact about 1,200 differentially methylated gene promoters in this comparison:

```r
> summary(decideTests(lrtpr))
  Down 1*P6 -0.5*P7 -0.5*P8 355
  NotSig 15475
  Up 809
```

For future reference we make a dataframe of all the DM genes:

```r
> topME <- topTags(lrtpr, n=Inf, p=0.05)$table
> dim(topME)
[1] 1164 9
```

The differential methylation results can be visualized with an MD plot (see Figure 5):

```r
> plotMD(lrtpr)
```

### Correlate with RNA-seq profiles

**RNA-seq profiles of mouse epithelium luminal and basal cells**

To explore whether hypermethylation of promoter regions is associated with repressed gene expression, we relate the differential methylation results to differential expression results from RNA-seq for similar cell populations. The RNA-seq data used here is from a study of the epithelial cell lineage in the mouse mammary gland, in which the expression profiles were generated from basal stem-cell enriched cells and committed luminal cells in the mammary glands of virgin, pregnant and lactating mice [39]. The complete differential expression analysis of the data is described in Chen et al [31]. The RNA-seq data is stored as a `DGEList` object `y_rna` and saved in a `RData` file `rna.RData`. The object `y_rna` contains the count matrix, sample information, gene annotation, design matrix and dispersion estimates of the RNA-seq data. The gene filtering, normalization and dispersion estimation were performed in the same way as described in Chen et al [31]. The `rna.RData` file is available for download at [http://bioinf.wehi.edu.au/edgeR/F1000Research2017](http://bioinf.wehi.edu.au/edgeR/F1000Research2017).

We load the `RData` file:

```r
> load("rna.RData")
> dim(y_rna)
[1] 15641 12
> y_rna$samples
   group   lib.size norm.factors
MCL1.DG B.virgin 23137472  1.235
MCL1.DH B.virgin 21687755  1.213
MCL1.DI B.pregnant 23974787  1.125
MCL1.DJ B.pregnant 22845375  1.069
```
Figure 5. MD plot showing the log-fold-change of the methylation level and average abundance of CpG sites in each gene promoter. Significantly hyper and hypomethylated gene promoters are highlighted in red and blue, respectively.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Sample Type</th>
<th>Methylation Level</th>
<th>log Fold Change</th>
<th>log Fold Change (RNA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCL1.DK</td>
<td>B.lactating</td>
<td>21420532</td>
<td>1.036</td>
<td></td>
</tr>
<tr>
<td>MCL1.DL</td>
<td>B.lactating</td>
<td>19916885</td>
<td>1.087</td>
<td></td>
</tr>
<tr>
<td>MCL1.LA</td>
<td>L.virgin</td>
<td>20273585</td>
<td>1.370</td>
<td></td>
</tr>
<tr>
<td>MCL1.LB</td>
<td>L.virgin</td>
<td>21568458</td>
<td>1.368</td>
<td></td>
</tr>
<tr>
<td>MCL1.LC</td>
<td>L.pregnant</td>
<td>22117517</td>
<td>1.006</td>
<td></td>
</tr>
<tr>
<td>MCL1.LD</td>
<td>L.pregnant</td>
<td>21877287</td>
<td>0.924</td>
<td></td>
</tr>
<tr>
<td>MCL1.LE</td>
<td>L.lactating</td>
<td>24657903</td>
<td>0.529</td>
<td></td>
</tr>
<tr>
<td>MCL1.LF</td>
<td>L.lactating</td>
<td>24600304</td>
<td>0.535</td>
<td></td>
</tr>
</tbody>
</table>

We keep only the genes that are also included in our methylation analysis:

```r
> haveME <- row.names(y_rna) %in% row.names(ypr)
> y_rna <- y_rna[haveME,]
> dim(y_rna)
[1] 13141 12
```

We assess differential expression between the luminal and basal virgin samples:

```r
> fitrna <- glmFit(y_rna)
> Contrastrna <- makeContrasts(L.virgin-B.virgin, levels=y_rna$design)
> lrtrna <- glmLRT(fitrna, contrast=Contrastrna)
```

We correlate methylation and expression:

```r
> topME$logFC.RNA <- lrtrna$table[row.names(topME),"logFC"]
> topME[1:30,c(1:5,10)]
```

<table>
<thead>
<tr>
<th>Chr</th>
<th>Symbol</th>
<th>Strand</th>
<th>Width</th>
<th>logFC</th>
<th>logFC.RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>16924</td>
<td>chr5</td>
<td>Lnx1</td>
<td>-105800</td>
<td>-6.85</td>
<td>2.2748</td>
</tr>
<tr>
<td>238161</td>
<td>chr12</td>
<td>Akap6</td>
<td>451633</td>
<td>-5.26</td>
<td>-3.2258</td>
</tr>
<tr>
<td>64082</td>
<td>chr16</td>
<td>Pordc2</td>
<td>16027</td>
<td>4.91</td>
<td>-7.6743</td>
</tr>
<tr>
<td>11601</td>
<td>chr3</td>
<td>Angpt2</td>
<td>51300</td>
<td>5.58</td>
<td>2.1021</td>
</tr>
<tr>
<td>12740</td>
<td>chr5</td>
<td>Gldn4</td>
<td>1811</td>
<td>-5.66</td>
<td>5.0988</td>
</tr>
</tbody>
</table>
The negative correlation between methylation and expression is immediately apparent. Of the top 30 DM genes, seven have NA expression fold-changes because the genes were not expressed at a high enough level to be included in the analysis. Of the rest, all but Akap6 have methylation and expression logFCs of opposite signs.

We can explore this correlation further by plotting the expression logFC vs the methylation logFC for all DM genes (see Figure 6):

```r
> plot(topME$logFC, topME$logFC.RNA, main="Lumina vs Basal",
+ xlab="Methylation logFC", ylab="Expression logFC",
+ pch=16, cex=0.8, col="gray30")
> abline(h=0, v=0, col="gray10", lty=2, lwd=2)
```

The horizontal axis of the scatterplot shows the log-fold-change in methylation level for each gene promoter while the vertical axis shows the log-fold-change in gene expression. To assess the correlation, we fit a least squares regression line through the origin and compute the P-value:

```r
> RNAvsME <- lm(topME$logFC.RNA ~ 0 + topME$logFC)
> coef(summary(RNAvsME))

             Estimate Std. Error t value Pr(>|t|)
topME$logFC  -0.745    0.0473  -15.7  3.83e-48
```

The negative association is highly significant ($P = 4 \times 10^{-48}$). The last line of code adds the regression line to the plot (Figure 6).

**Gene set testing**

The correlation in Figure 6 is convincing, but the above P-value computation assumes that genes are statistically independent of one another. We can perform a gene set test to get around this assumption.

First we make a dataframe of the logFCs of the DM genes, keeping only the genes for which we have RNA-seq results:

```r
> ME <- data.frame(ID=row.names(topME), logFO=topME$logFC, stringsAsFactors=FALSE)
> inRNA <- ME$ID %in% row.names(y_rna)
> ME <- ME[inRNA,]
```

Then we test whether the DM genes are up or down-regulated using a “fry” gene set test. This analysis weights the DM genes by their methylation logFCs so that the test evaluates whether the expression changes are positively or negatively associated with the methylation changes:
Figure 6. Scatter plot of the log-fold-changes of methylation levels in gene promoters (x-axis) vs the log fold-changes of gene expression (y-axis). The plot shows results for the genes of which the promoters are significantly differentially methylated between basal and luminal. The red line shows the least squares line with zero intercept. A strong negative correlation is observed.

```r
> fry(y_rna, index=ME, contrast=Contrastrna)

<table>
<thead>
<tr>
<th>NGenes</th>
<th>Direction</th>
<th>PValue</th>
<th>PValue.Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Down</td>
<td>1.38e-09</td>
<td>6.39e-11</td>
</tr>
</tbody>
</table>
```

The result **Down** in the **Direction** column indicates negative correlation between the methylation and expression changes. The small **PValue** confirms a highly significant result.

We can visualize the gene set test result with a barcode plot (see Figure 7):

```r
> logFC.ME <- rep_len(0,nrow(y_rna))
> names(logFC.ME) <- row.names(y_rna)
> logFC.ME[ME$ID] <- ME$logFC
> barcodeplot(lrtrna$table$logFC, gene.weights=logFC.ME,
+             labels=c("Basal","Luminal"), main="Luminal vs Basal")
> legend("topright", col=c("red","blue"), lty=1, lwd=2,
+         legend=c("Hypermethylation in Luminal", "Hypomethylation in Luminal"))
```

In the barcode plot, genes are sorted left to right from most down-regulated to most up-regulated in luminal vs basal. The x-axis shows the expression log2-fold-change. The vertical red bars indicate genes hypermethylated in luminal and vertical blue bars indicate genes hypomethylated in luminal. The variable-height vertical bars show the methylation log-fold-changes. The red and blue worms measure relative enrichment, showing that hypermethylation is associated with decreased regulation and hypomethylation is associated with up-regulation. In other words, there is a strong negative association between methylation of promoter regions and expression of the corresponding genes.

**Packages used**

This workflow depends on various packages from version 3.6 of the Bioconductor project, running on R version 3.4.0 or higher. For all the code to work as presented, edgeR 3.20.5 or later is required. A complete list of the packages used for this workflow is shown below:

```r
> sessionInfo()

R version 3.4.3 (2017-11-30)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 15063)
**Figure 7.** Barcode plot showing strong negative correlation between gene expression and DNA methylation in gene promoters.

Data and software availability

All software packages used in this workflow are publicly available as part of Bioconductor 3.6. The data and analysis code used in this workflow are available from [http://bioinf.wehi.edu.au/edgeR/F1000Research2017](http://bioinf.wehi.edu.au/edgeR/F1000Research2017). The data and analysis code as at time of publication of this article has been archived at [http://doi.org/](http://doi.org/).
Competing interests
No competing interests were disclosed.

Grant information
This work was supported by the National Health and Medical Research Council (Fellowship 1058892 and Program 1054618 to G.K.S, Independent Research Institutes Infrastructure Support to the Walter and Eliza Hall Institute) and by a Victorian State Government Operational Infrastructure Support Grant.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgments
The authors thank Andrew Keniry for help on Bismark.

References


