

methyAnalysis: an R package for DNA methylation data analysis and visualization

Pan Du^{†*} and Richard Bourgon[‡]

October 30, 2018

[‡]Department of Bioinformatics and Computational Biology
Genentech Inc., South San Francisco, CA, 94080, USA

Contents

1	Introduction	2
2	MethyGenoSet-class	2
2.1	Example dataset	2
2.2	Input methylation data from other packages	4
3	Identifying Differentially Methylated Regions (DMR)	4
3.1	DNA methylation correlation between nearby CpG-sites	4
3.2	Differential methylation test	6
3.3	Define differentially methylated regions	7
4	Annotating DMRs	8
5	Visualizing DNA methylation data	8
5.1	Export data for external visualization	8
5.2	Plot methylation heatmap by chromosome location	9
6	sessionInfo	9
7	References	13

*dupan.mail (at) gmail.com

1 Introduction

The `methyAnalysis` package aims to provide functionalities of analyzing and visualizing the DNA methylation data.

As most DNA-methylation data is still array-based, most public analysis tools use traditional probe-based analysis methods. However, with the increase of probe density, considering the probe spatial information becomes more and more important for better understanding the data. To meet this need, we developed this package for chromosome location based DNA methylation analysis. The current version of the package mainly focus on analyzing the Illumina Infinium methylation array data preprocessed by the `lumi` package [1], but most methods can be generalized to other methylation array or sequencing data. Functions specifically designed for DNA methylation sequencing data will be added in the near future.

The package mainly provides functions in the following aspects:

1. Defines a new class, *MethyGenoSet*, and related methods for the chromosome-location based DNA methylation analysis.
2. Provides functions related with differential methylation analysis, slide-window smoothing of DNA methylation levels, DMR (Differentially Methylation Region) detection and annotation.
3. Visualization of the DNA methylation data.

2 MethyGenoSet-class

In order to keep the chromosome location information together with the data, we defined a new *MethyGenoSet* class as a direct extension of the *GenoSet* class in the `genoset` package. The *GenoSet* class is an extension of *eSet* class. It keeps the chromosome location information in an additional `rowRanges` slot, a *GRanges* or *RangedData* object. For convenience of retrieving the methylation data, we keeps the DNA methylation data (using M-value [2] by default) in the `exprs assayData` element. Users can easily retrieve the methylation data by using *exprs* method.

2.1 Example dataset

For better understanding the package, we created a small example dataset, `exampleMethyGenoSet`. The `exampleMethyGenoSet` consists of eight random selected cancer cell line samples from two tissues. To save space, only probes in chromosome 21 were included.

```
> library(methyAnalysis)
> ## load the example data
> data(exampleMethyGenoSet)
> ## show MethyGenoSet class
> slotNames(exampleMethyGenoSet)
```

```

[1] "history"          "annotation"
[3] "rowRanges"       "colData"
[5] "assays"          "NAMES"
[7] "elementMetadata" "metadata"

> # showClass('MethyGenoSet')
>
> ## get chromosome location information
> head(rowRanges(exampleMethyGenoSet))

GRanges object with 6 ranges and 1 metadata column:
      seqnames      ranges strand |
      <Rle> <IRanges> <Rle> |
cg17035109 chr21 10882029      * |
cg06187584 chr21 10883548      * |
cg12459059 chr21 10884748      * |
cg25450479 chr21 10884967      * |
cg23347501 chr21 10884969      * |
cg03661019 chr21 10885409      * |
      ID
      <factor>
cg17035109 cg17035109
cg06187584 cg06187584
cg12459059 cg12459059
cg25450479 cg25450479
cg23347501 cg23347501
cg03661019 cg03661019
-----
seqinfo: 1 sequence from hg19 genome; no seqlengths

> ## retrieve methylation data
> dim(exprs(exampleMethyGenoSet))

[1] 4243      8

> ## Sample information
> colData(exampleMethyGenoSet)

DataFrame with 8 rows and 1 column
      SampleType
      <character>
Sample1      Type1
Sample2      Type1
Sample3      Type1

```

Sample4	Type1
Sample5	Type2
Sample6	Type2
Sample7	Type2
Sample8	Type2

2.2 Input methylation data from other packages

Lumi or methylumi package

3 Identifying Differentially Methylated Regions (DMR)

One common DNA methylation analysis task is to identify Differentially Methylated Regions (DMR) between two comparison groups. Similar as the expression microarray analysis, many existing differential test methods can be used here. However, most of these methods do not consider the probe spatial information and assuming probe measurements are independent to each other.

3.1 DNA methylation correlation between nearby CpG-sites

For DNA methylation data, we observed strong correlation between nearby CpG-sites. Figure 1 shows the correlation between nearby CpG-sites. The x-axis is the distance between nearby CpG-sites and the y-axis is the Pearson correlation of the related methylation profiles of 49 cell line samples (data not shown). The red dots are the median correlation of the 5 percentile cut (ranked by the distance between nearby CpG-sites (x-axis)). We can see the correlation is very strong when the CpG-sites are close to each other.

On the other hand, due to the sequence variation across samples and fixed probe designs, the array-based DNA-methylation data also tends to be noisy. By considering the observed strong correlation between nearby CpG-sites, we can reduce the measurement noise by using sliding-window smoothing. `smoothMethyData` function is designed for this purpose. By default, we set `winSize` (half-window size) as 250bp, which is selected based on Figure 1.

```
> methyGenoSet.sm <- smoothMethyData(exampleMethyGenoSet, winSize = 250)
```

```
Smoothing Chromosome chr21 ...
```

```
> ## winsize is kept as an attribute  
> attr(methyGenoSet.sm, 'windowSize')
```

```
[1] 250
```

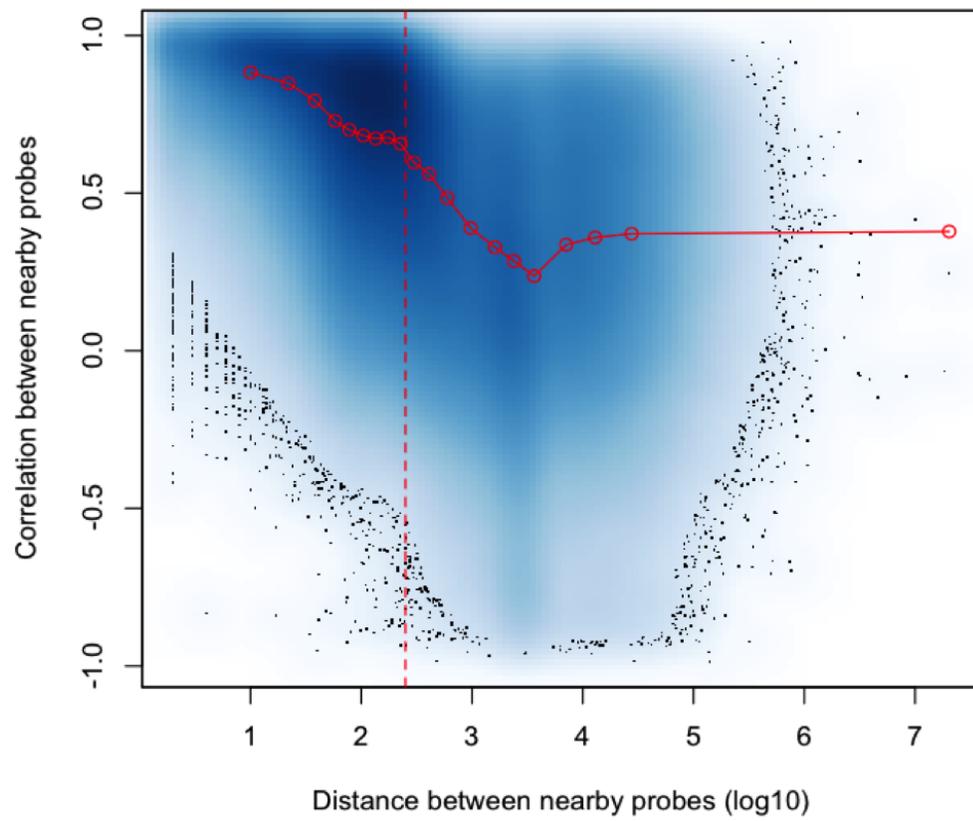


Figure 1: DNA methylation correlation between nearby CpG-sites

3.2 Differential methylation test

Function `detectDMR.slideWin` is designed to perform differential methylation test. The function will automatically check whether the methylation data has been smoothed or not. If not, slide window smoothing will be performed first. The current version only implement 't-test' and wilcox test for the differential methylation test. A more flexible (lm)-based method will be added in the future version.

```
> ## get sample type information
> sampleType <- colData(exampleMethyGenoSet)$SampleType
> ## Do differential test
> allResult <- detectDMR.slideWin(exampleMethyGenoSet, sampleType=sampleType)
```

Smoothing Chromosome chr21 ...

```
> head(allResult)
```

GRanges object with 6 ranges and 11 metadata columns:

	seqnames	ranges	strand	
	<Rle>	<IRanges>	<Rle>	
cg17035109	chr21	10882029	*	
cg06187584	chr21	10883548	*	
cg12459059	chr21	10884748	*	
cg25450479	chr21	10884967	*	
cg23347501	chr21	10884969	*	
cg03661019	chr21	10885409	*	
	PROBEID		difference	
	<factor>		<numeric>	
cg17035109	cg17035109	-1.84116050891237		
cg06187584	cg06187584	-0.456605878544024		
cg12459059	cg12459059	-0.359117942730594		
cg25450479	cg25450479	-0.359117942730594		
cg23347501	cg23347501	-0.359117942730594		
cg03661019	cg03661019	-0.353266150910433		
		p.value		p.adjust
		<numeric>		<numeric>
cg17035109		0.0627644943115837		0.18884880818101
cg06187584		0.416014860951571		0.61495957661053
cg12459059		0.365421521707119		0.562790422031127
cg25450479		0.365421521707119		0.562790422031127
cg23347501		0.365421521707119		0.562790422031127
cg03661019		0.380656000521138		0.578209920562373
		tscore		startWinIndex
		<numeric>		<numeric>

cg17035109	-2.28040907618597		1
cg06187584	-0.873425189908341		2
cg12459059	-0.978927000428536		3
cg25450479	-0.978927000428536		3
cg23347501	-0.978927000428536		3
cg03661019	-0.946031414573684		6
	endWinIndex	startLocation	
	<numeric>	<integer>	
cg17035109	1	10882029	
cg06187584	2	10883548	
cg12459059	5	10884748	
cg25450479	5	10884748	
cg23347501	5	10884748	
cg03661019	6	10885409	
	endLocation	mean_Type1	
	<integer>	<numeric>	
cg17035109	10882029	-2.41837749804746	
cg06187584	10883548	-2.22975671527783	
cg12459059	10884969	0.259415093217181	
cg25450479	10884969	0.259415093217181	
cg23347501	10884969	0.259415093217181	
cg03661019	10885409	-0.417036279735772	
	mean_Type2		
	<numeric>		
cg17035109	-0.577216989135089		
cg06187584	-1.7731508367338		
cg12459059	0.618533035947775		
cg25450479	0.618533035947775		
cg23347501	0.618533035947775		
cg03661019	-0.0637701288253387		

seqinfo: 1 sequence from hg19 genome; no seqlengths

3.3 Define differentially methylated regions

We define a differentially methylated region (DMR) as a region, in which most measured CpG-sites are differentially methylated. To identify DMRs, we first determine the differential methylation status of each probe, then merge them as a continuous region. The `identifySigDMR` function is a wrapper function for all of these. The `getContinuousRegion` function is called by `identifySigDMR` to detection continous regions. Its input is a `GRanges` object with a "status" column to show whether the probe is differentially methylated or not. Its output is also a `GRanges` object indicating the identified DMRs. The `identifySigDMR` function returns a list of two `GRanges` objects. `sigDMRInfo` includes the identified DMRs,

and `sigDataInfo` includes all differentially methylated probe information.

```
> ## Identify the DMR (Differentially Methylated Region) by setting proper parameters.
> ## Here we just use default ones
> allDMRInfo = identifySigDMR(allResult)
> names(allDMRInfo)

[1] "sigDMRInfo" "sigDataInfo"
```

4 Annotating DMRs

To understand what genes or gene elements (promoters or exons) are overlapping with these identified DMRs, we need to do annotate. The `annotateDMRInfo` is defined for this purpose. A `TxDb` annotation package is required for the annotation process. Here we use the `TxDb.Hsapiens.UCSC.hg19.knownGene` package for the annotation. The `TxDb.Hsapiens.UCSC.hg19.knownGene` package includes the Homo Sapiens data from UCSC build hg19 based on the knownGene Track. Other `TxDb` annotation packages, `TxDb` or `GRanges` objects can also be used as `annotationDatabase`. The `export.DMRInfo` function is to output the annotated DMR information as `.csv` files.

```
> ## Annotate significant DMR info
> DMRInfo.ann <- annotateDMRInfo(allDMRInfo, 'TxDb.Hsapiens.UCSC.hg19.knownGene')
> ## output the DMR information
> export.DMRInfo(DMRInfo.ann, savePrefix='testExample')
```

5 Visualizing DNA methylation data

As the DNA methylation levels are chromosome location dependent. The methylation patterns can be pretty different between different gene elements, like promoter, exon1, intron and exons. The methylation patterns within the CpG-islands usually are also different from other regions. In order to better understanding these difference, we need to visualize the DNA methylation data. Two visualization options are supported in the `methyAnalysis` package.

5.1 Export data for external visualization

One easier option is to export the DNA methylation data in certain formats, and visualize these files using some genome browser tools, like IGV (<http://www.broadinstitute.org/igv/>) and IGB (<http://bioviz.org/igb/index.html>). Users can use `export.methyGenoSet` to output the `MethyGenoSet` object. The current implementation supports two output formats: `".gct"` and `".bw"` files. `".gct"` includes all samples in a single file. It is only supported by IGV genome browser. The BigWig format (`".bw"`) is a more general format supported by many visualization tools. Each BigWig file represents one single sample. So it is more flexible for the users only interested in a subset of samples.

```

> ## output in IGV supported "gct" file
> export.methyGenoSet(exampleMethyGenoSet, file.format='gct', savePrefix='test')
> ## output in BigWig files
> export.methyGenoSet(exampleMethyGenoSet, file.format='bw', savePrefix='test')

```

5.2 Plot methylation heatmap by chromosome location

Another visualization option is to show a focused regions, like DMRs, as a chromosome location based heatmap. `heatmapByChromosome` is designed for this. It is adapted based on the `plotTracks` function in `Gviz` package. The function is designed for different types of data with chromosome location information. Figure 2 shows an example plot of gene SIM2 (Entrez Gene ID:6493), which overlaps with the identified DMRs shown above. Users can also provide a `GRanges` object to specify a plot region.

```

> ## plot the DNA methylation heatmap by chromosome location
> heatmapByChromosome(exampleMethyGenoSet, gene='6493',
  genomicFeature='TxDb.Hsapiens.UCSC.hg19.knownGene', includeGeneBody=TRUE)

```

Another wrapper function, `plotMethylationHeatmapByGene`, is specifically designed for the methylation data. Users can add phenotypes or matched gene expression data to the right panel of the plot. Figure legends can be also added, as shown in Figure 3. By default, the `plotMethylationHeatmapByGene` plots methylation Beta-values [2] (in the range of 0 to 1) instead of M-values. Users can set `useBetaValue` as `FALSE` if they want to change to M-values.

```

> ## plot the DNA methylation heatmap by gene of selected GRanges
> plotMethylationHeatmapByGene('6493', methyGenoSet=exampleMethyGenoSet,
  phenoData=colData(exampleMethyGenoSet), includeGeneBody=TRUE,
  genomicFeature='TxDb.Hsapiens.UCSC.hg19.knownGene')

```

6 sessionInfo

```

> toLatex(sessionInfo())

```

- R Under development (unstable) (2018-10-16 r75448), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=C, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Running under: Ubuntu 18.04.1 LTS

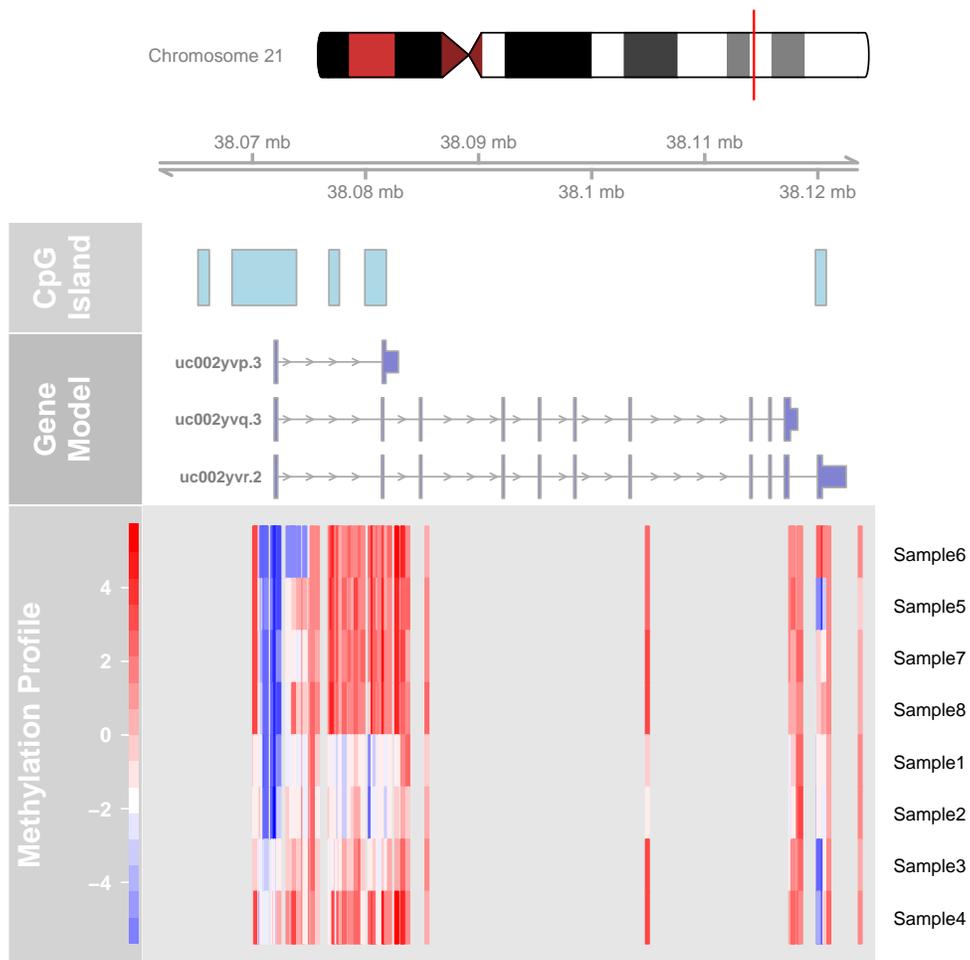


Figure 2: DNA methylation heatmap by chromosome location

Plotting SIM2 (GeneID:6493)

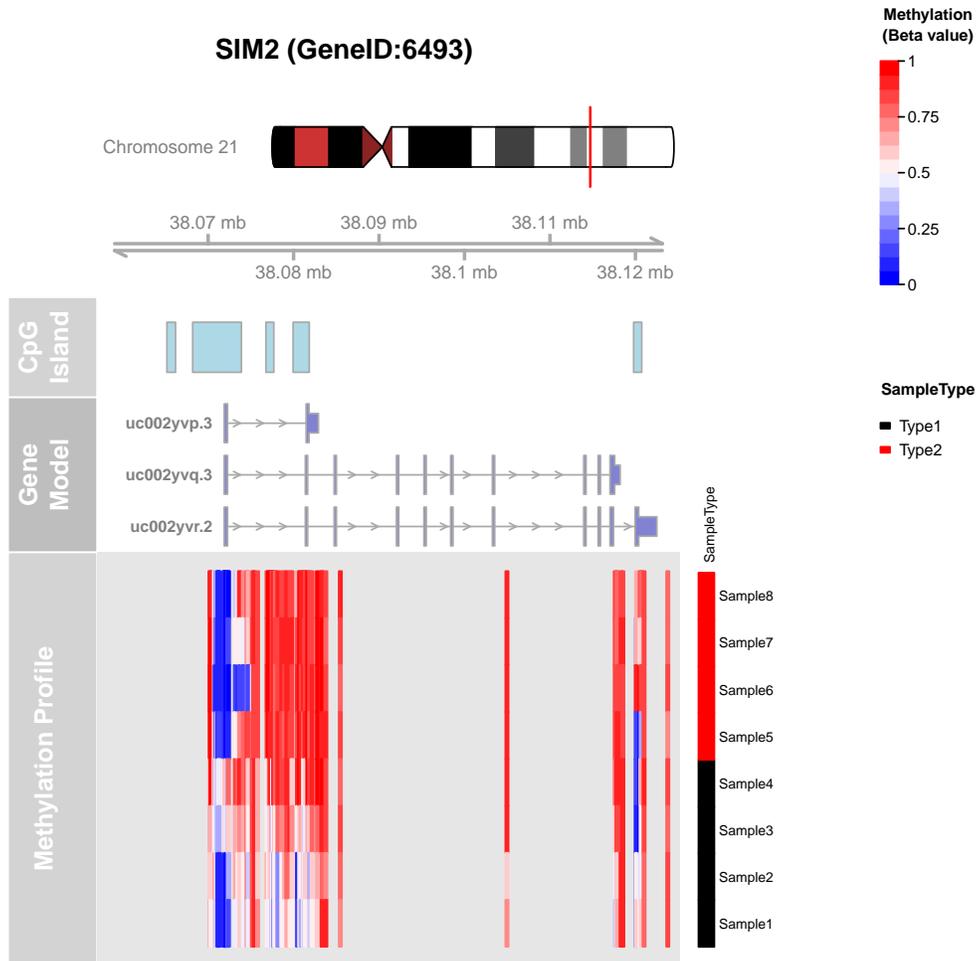


Figure 3: DNA methylation heatmap by chromosome location with phenotype information

- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.9-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.9-bioc/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.45.0, Biobase 2.43.0, BiocGenerics 0.29.0, GenomeInfoDb 1.19.0, GenomicFeatures 1.35.0, GenomicRanges 1.35.0, IRanges 2.17.0, S4Vectors 0.21.0, TxDb.Hsapiens.UCSC.hg19.knownGene 3.2.2, methyAnalysis 1.25.0, org.Hs.eg.db 3.7.0
- Loaded via a namespace (and not attached): AnnotationFilter 1.7.0, BSgenome 1.51.0, BiocManager 1.30.3, BiocParallel 1.17.0, Biostrings 2.51.0, DBI 1.0.0, DelayedArray 0.9.0, DelayedMatrixStats 1.5.0, Formula 1.2-3, GEOquery 2.51.0, GenomeInfoDbData 1.2.0, GenomicAlignments 1.19.0, Gviz 1.27.0, HDF5Array 1.11.0, Hmisc 4.1-1, KernSmooth 2.23-15, MASS 7.3-51, Matrix 1.2-14, ProtGenerics 1.15.0, R6 2.3.0, RColorBrewer 1.1-2, RCurl 1.95-4.11, RSQLite 2.1.1, Rcpp 0.12.19, Rhdf5lib 1.5.0, Rsamtools 1.35.0, SummarizedExperiment 1.13.0, VariantAnnotation 1.29.0, XML 3.98-1.16, XVector 0.23.0, acepack 1.4.1, affy 1.61.0, affyio 1.53.0, annotate 1.61.0, assertthat 0.2.0, backports 1.1.2, base64 2.0, base64enc 0.1-3, beanplot 1.2, bibtex 0.4.2, bindr 0.1.1, bindrcpp 0.2.2, biomaRt 2.39.0, biovizBase 1.31.0, bit 1.1-14, bit64 0.9-7, bitops 1.0-6, blob 1.1.1, bumphunter 1.25.0, checkmate 1.8.5, cluster 2.0.7-1, codetools 0.2-15, colorspace 1.3-2, compiler 3.6.0, crayon 1.3.4, curl 3.2, data.table 1.11.8, dichromat 2.0-0, digest 0.6.18, doRNG 1.7.1, dplyr 0.7.7, ensemblDb 2.7.0, foreach 1.4.4, foreign 0.8-71, genefilter 1.65.0, genoset 1.39.0, ggplot2 3.1.0, glue 1.3.0, gridExtra 2.3, gtable 0.2.0, hms 0.4.2, htmlTable 1.12, htmltools 0.3.6, htmlwidgets 1.3, httr 1.3.1, illuminaio 0.25.0, iterators 1.0.10, knitr 1.20, lattice 0.20-35, latticeExtra 0.6-28, lazyeval 0.2.1, limma 3.39.0, locfit 1.5-9.1, lumi 2.35.0, magrittr 1.5, matrixStats 0.54.0, mclust 5.4.1, memoise 1.1.0, methylumi 2.29.0, mgcv 1.8-25, minfi 1.29.0, multtest 2.39.0, munsell 0.5.0, nleqslv 3.3.2, nlme 3.1-137, nnet 7.3-12, nor1mix 1.2-3, openssl 1.0.2, pillar 1.3.0, pkgconfig 2.0.2, pkgmaker 0.27, plyr 1.8.4, preprocessCore 1.45.0, prettyunits 1.0.2, progress 1.2.0, purrr 0.2.5, quadprog 1.5-5, readr 1.1.1, registry 0.5, reshape 0.8.8, rhdf5 2.27.0, rlang 0.3.0.1, rngtools 1.3.1, rpart 4.1-13, rstudioapi 0.8, rtracklayer 1.43.0, scales 1.0.0, siggenes 1.57.0, splines 3.6.0, stringi 1.2.4, stringr 1.3.1, survival 2.43-1, tibble 1.4.2, tidyr 0.8.2, tidyselect 0.2.5, tools 3.6.0, withr 2.1.2, xml2 1.2.0, xtable 1.8-3, zlibbioc 1.29.0

7 References

1. Du P, Kibbe WA and Lin SM: "lumi: a Bioconductor package for processing Illumina microarray" *Bioinformatics* 2008 24(13):1547-1548
2. Du P, Zhang X, Huang CC, Jafari N, Kibbe WA, Hou L, and Lin SM: "Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis", *BMC Bioinformatics* 2010, 11:587