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

Pan Du[‡*] and Richard Bourgon[‡]

May 3, 2016

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

Contents

[*]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 specificly 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, 10882029]      *
  cg06187584    chr21 [10883548, 10883548]      *
  cg12459059    chr21 [10884748, 10884748]      *
  cg25450479    chr21 [10884967, 10884967]      *
  cg23347501    chr21 [10884969, 10884969]      *
  cg03661019    chr21 [10885409, 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 correlaiton 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 oberseved 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
```

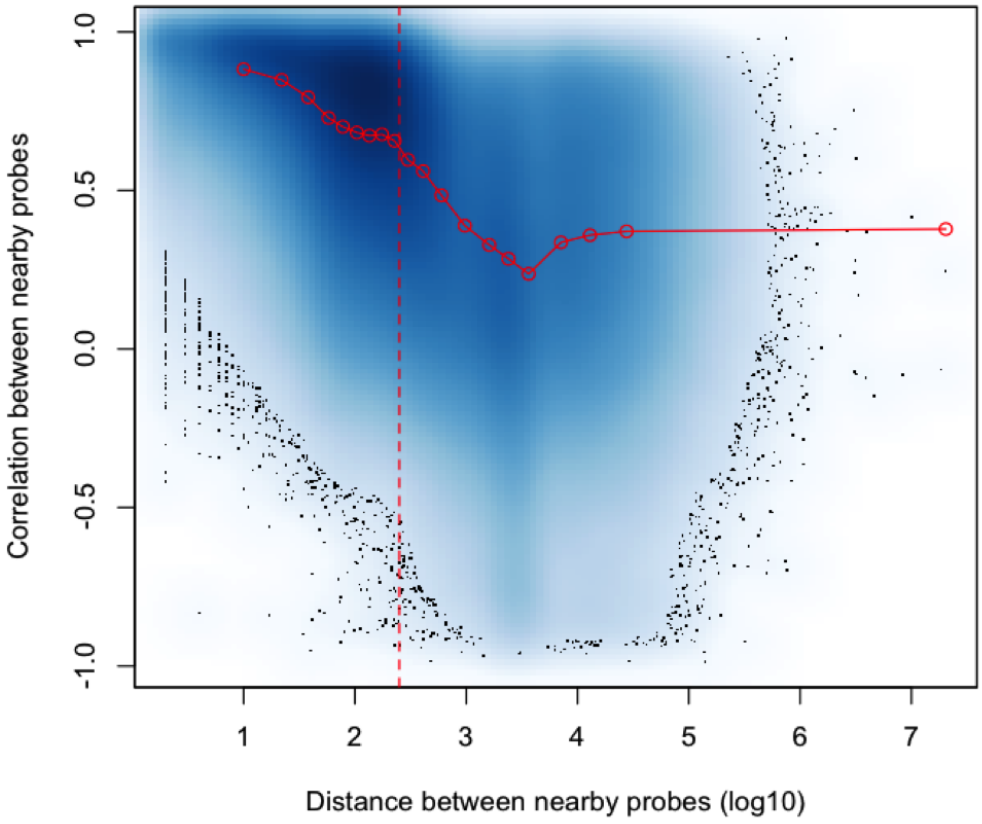

Figure 1: DNA methylation correlation between nearby CpG-sites

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, 10882029]      *
  cg06187584    chr21 [10883548, 10883548]      *
  cg12459059    chr21 [10884748, 10884748]      *
  cg25450479    chr21 [10884967, 10884967]      *
  cg23347501    chr21 [10884969, 10884969]      *
  cg03661019    chr21 [10885409, 10885409]      *
             |    PROBEID difference    p.value
             |   <factor>  <numeric>  <numeric>
  cg17035109 | cg17035109 -1.8411605 0.06276449
  cg06187584 | cg06187584 -0.4566059 0.41601486
  cg12459059 | cg12459059 -0.3591179 0.36542152
  cg25450479 | cg25450479 -0.3591179 0.36542152
  cg23347501 | cg23347501 -0.3591179 0.36542152
  cg03661019 | cg03661019 -0.3532662 0.38065600
              p.adjust     tscore startWinIndex
             <numeric>  <numeric>     <numeric>
  cg17035109 0.1888488 -2.2804091             1
  cg06187584 0.6149596 -0.8734252             2
  cg12459059 0.5627904 -0.9789270             3
  cg25450479 0.5627904 -0.9789270             3
  cg23347501 0.5627904 -0.9789270             3
  cg03661019 0.5782099 -0.9460314             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  mean_Type2
               <integer>  <numeric>   <numeric>
  cg17035109    10882029 -2.4183775 -0.57721699
  cg06187584    10883548 -2.2297567 -1.77315084
  cg12459059    10884969  0.2594151  0.61853304
  cg25450479    10884969  0.2594151  0.61853304
  cg23347501    10884969  0.2594151  0.61853304
  cg03661019    10885409 -0.4170363 -0.06377013
  -------
  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.knownGe 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 chromose 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 chomosome 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 methylaiton 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 version 3.3.0 (2016-05-03), `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`
- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.34.0, Biobase 2.32.0, BiocGenerics 0.18.0, GenomeInfoDb 1.8.0, GenomicFeatures 1.24.0, GenomicRanges 1.24.0, IRanges 2.6.0, S4Vectors 0.10.0, TxDb.Hsapiens.UCSC.hg19.knownGene 3.2.2, methyAnalysis 1.14.0, org.Hs.eg.db 3.3.0
- Loaded via a namespace (and not attached): AnnotationHub 2.4.0, BSgenome 1.40.0, BiocInstaller 1.22.0, BiocParallel 1.6.0, Biostrings 2.40.0, DBI 0.4, Formula 1.2-1, GEOquery 2.38.0, GenomicAlignments 1.8.0, Gviz 1.16.0, Hmisc 3.17-4, KernSmooth 2.23-15, MASS 7.3-45, Matrix 1.2-6, R6 2.1.2, RColorBrewer 1.1-2, RCurl 1.95-4.8, RSQLite 1.0.0, Rcpp 0.12.4.5, Rsamtools 1.24.0, SummarizedExperiment 1.2.0, VariantAnnotation 1.18.0, XML 3.98-1.4, XVector 0.12.0, acepack 1.3-3.3, affy 1.50.0, affyio 1.42.0, annotate 1.50.0, base64 1.1, beanplot 1.2, biomaRt 2.28.0, biovizBase 1.20.0, bitops 1.0-6, bumphunter 1.12.0,

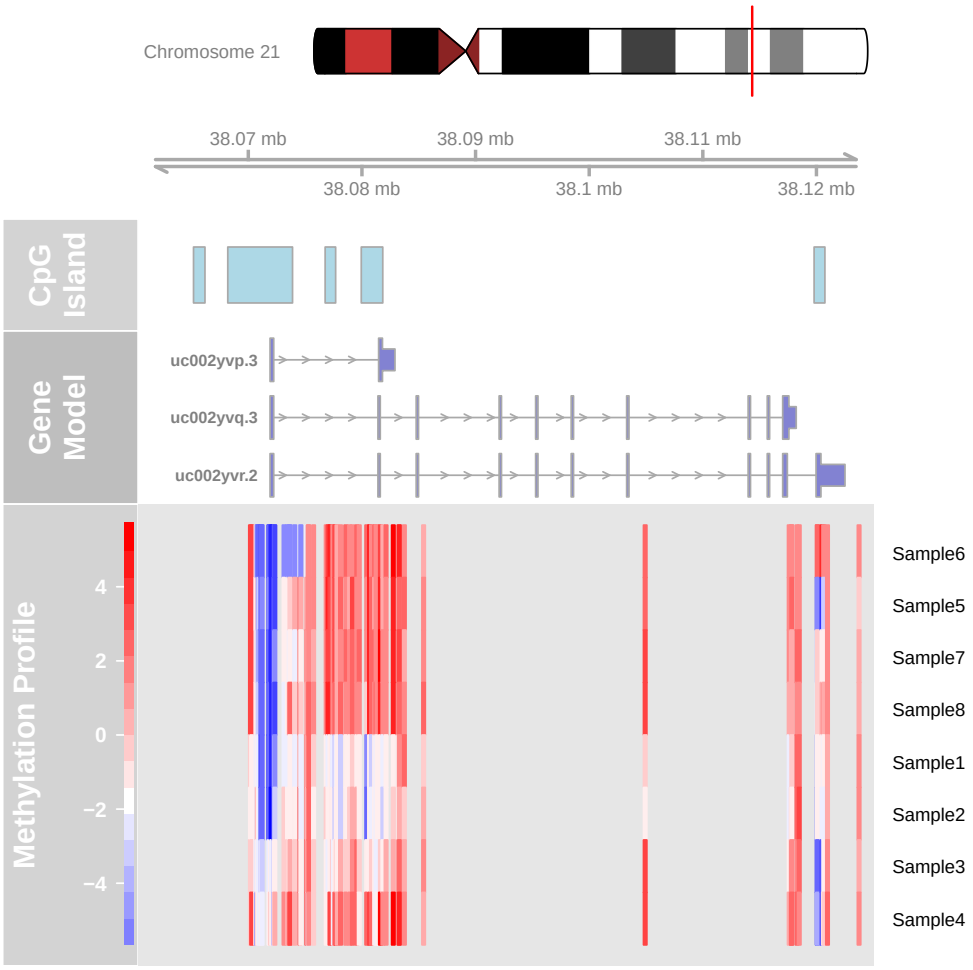

Figure 2: DNA methylation heatmap by chromosome location

```
Ploting  SIM2 (GeneID:6493)
```

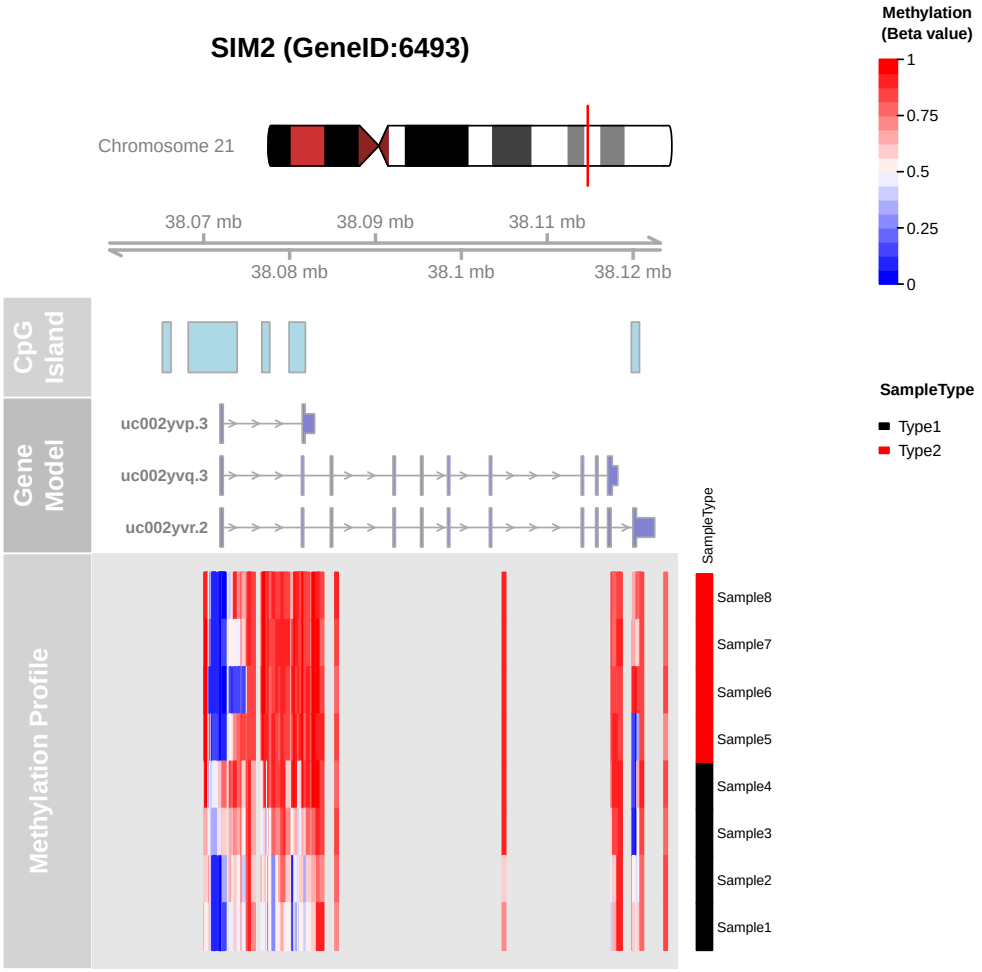

Figure 3: DNA methylation heatmap by chromosome location with phenotype information

chron 2.3-47, cluster 2.0.4, codetools 0.2-14, colorspace 1.2-6, data.table 1.9.6, dichromat 2.0-0, digest 0.6.9, doRNG 1.6, ensembldb 1.4.0, foreach 1.4.3, foreign 0.8-66, genefilter 1.54.0, genoset 1.28.0, ggplot2 2.1.0, gridExtra 2.2.1, gtable 0.2.0, htmltools 0.3.5, httpuv 1.3.3, httr 1.1.0, illuminaio 0.14.0, interactiveDisplayBase 1.10.0, iterators 1.0.8, lattice 0.20-33, latticeExtra 0.6-28, limma 3.28.0, locfit 1.5-9.1, lumi 2.24.0, magrittr 1.5, matrixStats 0.50.2, mclust 5.2, methylumi 2.18.0, mgcv 1.8-12, mime 0.4, minfi 1.18.0, multtest 2.28.0, munsell 0.4.3, nleqslv 3.0.1, nlme 3.1-127, nnet 7.3-12, nor1mix 1.2-1, pkgmaker 0.22, plyr 1.8.3, preprocessCore 1.34.0, quadprog 1.5-5, registry 0.3, reshape 0.8.5, rngtools 1.2.4, rpart 4.1-10, rtracklayer 1.32.0, scales 0.4.0, shiny 0.13.2, siggenes 1.46.0, splines 3.3.0, stringi 1.0-1, stringr 1.0.0, survival 2.39-2, tools 3.3.0, xtable 1.8-2, zlibbioc 1.18.0

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