methylKit: how to use

Altuna Akalin

December 6, 2011

1 Introduction

In this example we show how to use the methylKit package. methylKit is an R package for analysis and annotation of DNA methylation from high-throughput bisulfate sequencing. The package is designed to deal with sequencing data from RRBS and its variants. But it can potentially handle whole-genome bisulfate sequencing data if proper input format is provided.

2 Basics

2.1 Reading the methylation call files

We start by reading in the methylation call data from bisulfate sequencing with read function. Reading in the data this way will return a methylRawList object which stores methylation information per sample.

```
> library(methylKit)
> file.list = list(system.file("data", "test1.myCpG.txt", package = "methylKit"),
+     system.file("data", "test2.myCpG.txt", package = "methylKit"),
+     system.file("data", "control1.myCpG.txt", package = "methylKit"),
+     system.file("data", "control2.myCpG.txt", package = "methylKit"))
> myobj = read(file.list, sample.id = list("test1", "test2", "ctrl1",
+     "ctrl2"), assembly = "hg18", pipeline = "amp", treatment = c(1,
+     1, 0, 0))
```

2.2 Basic statistics on samples

Since we read the data now we can check the basic stats about the methylation data such as coverage and percent methylation. We now have a methylRawList object which contains methylation information per sample. The following command prints out percent methylation statistics for second sample: "test2"

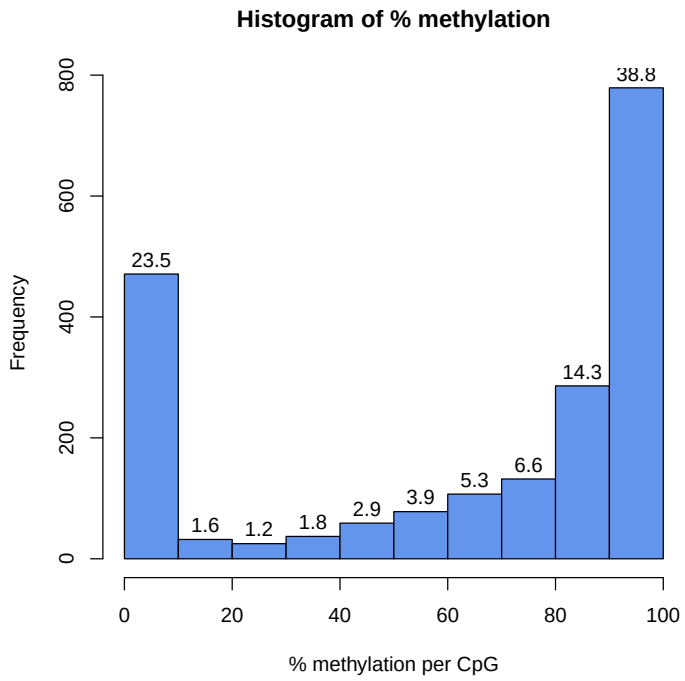
```
> getMethylationStats(myobj[[2]], plot = F, both.strands = F)
```

```
methylation statistics per base
summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   20.00   82.79   63.17   94.74  100.00
percentiles:
```

```
      0%      10%      20%      30%      40%      50%      60%      70%
 0.00000  0.00000  0.00000 48.38710 70.00000 82.78556 90.00000 93.33333
      80%      90%      95%      99%    99.5%    99.9%     100%
96.42857 100.00000 100.00000 100.00000 100.00000 100.00000 100.00000
```

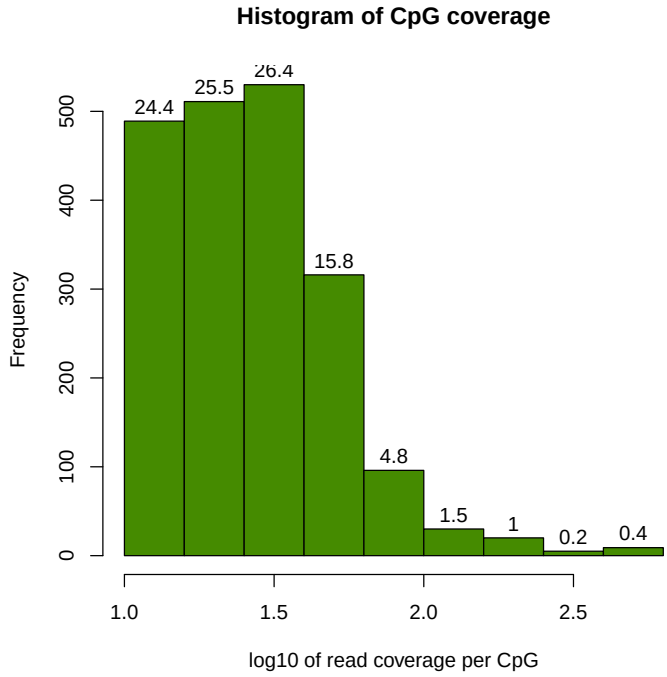
The following command plots the histogram for percent methylation distribution.

```
> library("graphics")
> getMethylationStats(myobj[[2]], plot = T, both.strands = F)
```



We can also plot the read coverage per base information in a similar way.

```
> library("graphics")
> getCoverageStats(myobj[[2]], plot = T, both.strands = F)
```



3 Comparative analysis

3.1 Merging samples

In order to do further analysis, we will need to get the bases covered in all samples. The following function will merge all samples to one object for base-pair locations that are covered in all samples. Setting destrand=TRUE (the deafult is FALSE) will merge reads on both strans of a CpG dinucleotide. This provides better coverage, but only advised when looking at CpG methylation (for CpH methylation this will cause in wrong results in subsequent analyses). This operation will return a methylBase object which will be our main object for all comparative analysis.

```
> methidh = unite(myobj, destrand = FALSE)
```

Let us take a look at the data content of methylBase object:

```
> head(methidh)
```

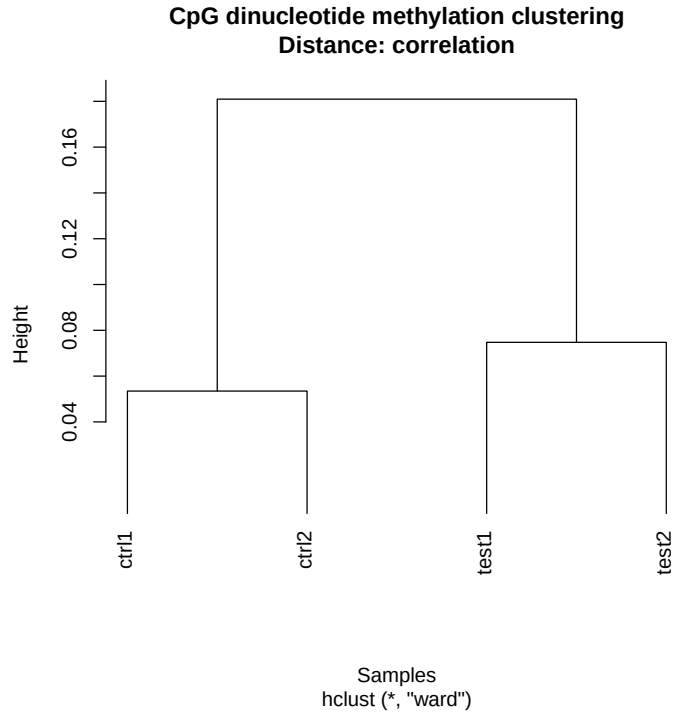
```
              id   chr    start      end strand coverage1 numCs1 numTs1
1 chr21.10011833 chr21 10011833 10011833      +       174    173       1
2 chr21.10011841 chr21 10011841 10011841      +       173    164       9
3 chr21.10011855 chr21 10011855 10011855      +       175    175       0
4 chr21.10011858 chr21 10011858 10011858      +       175    131      44
5 chr21.10011861 chr21 10011861 10011861      +       174    147      27
6 chr21.10011872 chr21 10011872 10011872      +       167    160       7
  coverage2 numCs2 numTs2 coverage3 numCs3 numTs3 coverage4 numCs4 numTs4
```

```
1        18      18      0       40      34      6       14      14      0
2        20      19      1       40      18      22      14      8       6
3        21      21      0       39      29      10      14      12      2
4        21      20      1       39      31      8       13      8       5
5        20      15      5       39      13      26      13      9       4
6        20      19      1       39      34      5       14      8       6
```

3.2 Clustering samples

We can cluster the samples based on the similarity of their methylation profiles. The following function will cluster the samples and draw a dendrogram.

```
> clusterSamples(methidh, dist = "correlation", method = "ward",
+     plot = TRUE)

Call:
hclust(d = d, method = HCLUST.METHODS[hclust.method])

Cluster method   : ward
Distance         : pearson
Number of objects: 4
```

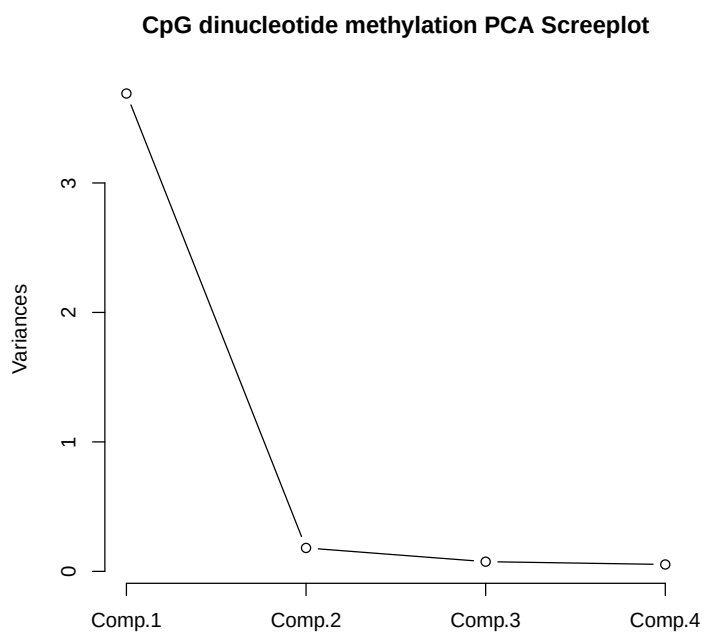


Setting the plot=FALSE will return a dendrogram object which can be manipulated by users or fed in to other user defined functions.

```
> hc = clusterSamples(methidh, dist = "correlation", method = "ward",
+     plot = FALSE)
```

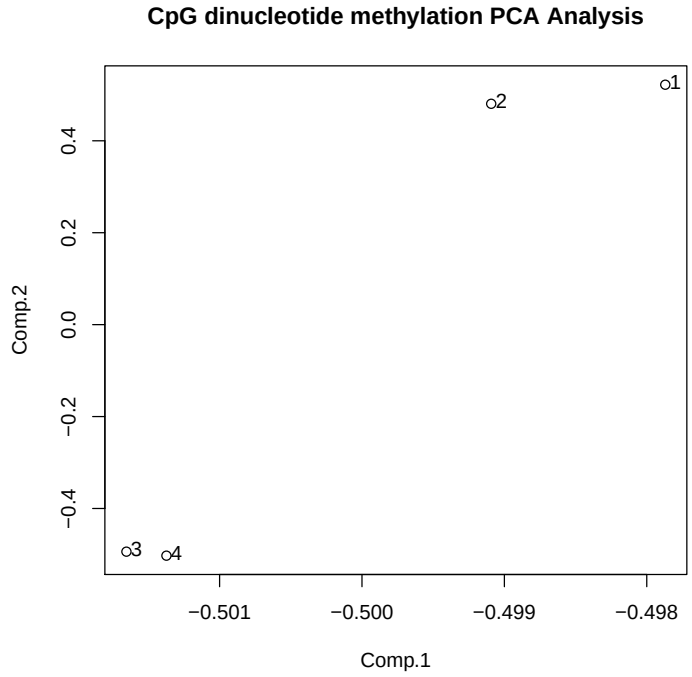
We can also do a PCA analysis on our samples. The following function will plot a scree plot for importance of components.

```
> PCASamples(methidh, screeplot = TRUE)

Importance of components:
                          Comp.1     Comp.2    Comp.3     Comp.4
Standard deviation     1.9211651 0.42545636 0.2735654 0.23081050
Proportion of Variance 0.9227188 0.04525328 0.0187095 0.01331837
Cumulative Proportion  0.9227188 0.96797213 0.9866816 1.00000000
```



We can also plot PC1 and PC2 axis and a scatter plot of our samples on those axis which will reveal how they cluster.

```
> PCASamples(methidh)

Importance of components:
                          Comp.1     Comp.2    Comp.3     Comp.4
Standard deviation     1.9211651 0.42545636 0.2735654 0.23081050
Proportion of Variance 0.9227188 0.04525328 0.0187095 0.01331837
Cumulative Proportion  0.9227188 0.96797213 0.9866816 1.00000000
```



3.3 Finding differentially methylated bases or regions

calculateDiffMeth() function is the main function to calculate differential methylation. Depending on the sample size per each set it will either use Fisher's exact or logistic regression to calculate P-values. P-values will be adjusted to Q-values using SLIM method[1].

```
> myDiff = calculateDiffMeth(methidh)
```

After q-value calculation, we can select the differentially methylated regions or bases based on q-value and percent methylation difference cutoffs. Following bit selects the bases that have q-value<0.01 and percent methylation difference larger than 25%.

```
> myDiff25p = get.methylDiff(myDiff, difference = 25, qvalue = 0.01)
```

3.4 Annotating differentially methylated bases or regions

We can annotate our differentially methylated regions/bases based on gene annotation. In this example, we read the gene annotation from a bed file and annotate our differentially methylated regions with that information. This will tell us what percentage of our differentially methylated regions are on promoters/introns/exons/intergenic region.

```
> gene.obj = read.transcript.features(system.file("tests", "refseq.hg18.bed.txt",
+     package = "methylKit"))
> annotate.WithGenicParts(myDiff25p, gene.obj)
```

```
summary of target set annotation with genic parts
133 rows in target set
--------------
--------------
percentage of target features overlapping with annotation :
  promoter       exon     intron intergenic
  27.81955   15.03759   34.58647   57.14286


percentage of target features overlapping with annotation (with promoter>exon>intron prece
  promoter       exon     intron intergenic
  27.81955    0.00000   15.03759   57.14286


percentage of annotation boundaries with feature overlap :
   promoter        exon      intron
0.018129079 0.001589593 0.010038738


summary of distances to the nearest TSS :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      5     828   45160   52030   94640  313500
```

References

[1] Hong-Qiang Wang, Lindsey K Tuominen, and Chung-Jui Tsai. SLIM: a sliding linear model for estimating the proportion of true null hypotheses in datasets with dependence structures. *Bioinformatics (Oxford, England)*, 27(2):225–31, January 2011.