methylPipe

Kamal Kishore, Mattia Pelizzola

March 16, 2016

Contents

1 Introduction

In this document you can find a brief tutorial on the ***methylPipe*** package for the analysis of base-pair resolution DNA methylation data. DNA methylation is a potentially heritable epigenetic modification of the genomic DNA typical of most eukaryotic organisms and critical for the regulation of gene transcription. The level of DNA methylation varies according to age, diet and environment and an appropriate control of methylation patterns is important for the onset of cellular differentiation processes and can be deregulated in diseases as cancer. In eukaryotic genomes the methyl group can be added to Cytosines in the CG, CHG and CHH sequence context (H being one of A, C or T). It is nowadays possible to generate genome-wide base-pair resolution DNA methylation maps (see Lister R et al Nature 2009). ***methylPipe*** is an R package that includes a series of objects and methods for a memory efficient management, query, analysis and visualization of DNA methylation data and their integration with heterogeneous data types. This package has been developed alongwith ***compEpiTools*** which provide functions and methods for the analysis of epigenomics data. The data package ***ListerEtAlBSseq*** has also been developed which consists of base resolution Bisulfite sequencing data of H1 and IMR90 cell line (Lister R et al Nature 2009).

The overview section of this document will briefly walk you through the main available functionalities, while the following sections will provide additional details on a selection of available classes and methods.

2 Quick overview

A number of **classes** are defined in *methylPipe*: *BSdata*, *BSdataSet*, *GEcollection* and *GElist*.

- ***BSdata*** is a reference class to store base resolution DNA methylation data generated from a high-thoughput sequencing experiment for a given biological sample.
- The ***BSdataSet*** class allows combining DNA methylation data for several samples for the same organism.
- The ***GEcollection*** class is used to store the DNA methylation status of a collection of genomic regions.
- The ***GElist*** class is the list of multiple objects of class *GEcollection*.

methylPipe consists of methods which allows analysis and visualisation of genome wide DNA methylation data. The `meth.call` function processes methylation information from aligned files and creates tabular data file. `BSprepare` processes and tabix compresses such tabular data files for creation of a *BSdata* object for each sample. The *BSdataSet* classe stores multiple such samples of class *BSdata*. The `mCsmoothing` methods extracts DNA methylation data for a given genomic region and smooths/plots it.

For methylation analysis on your regions of interest *GEcollection* and *GElist* acts as a repository of methylation data across various genomic ranges. `getCpos` and `getCposDensity` retrieves all potential genomic cytosine positions and their density across genomic regions, repectively. `mapBSdata2GRanges` allows extraction of DNA methylation information across multiple genomic ranges and `profileDNAmetBin` determines the absolute and relative methylation to create a *GEcollection* object. Different methods for the detection of differentially methylated regions are implemented, according to the number of samples (see `findDMR` and `consolidateDMRs`). In addition the methylation profile can be visualised using `plotMeth`. Morover, integrative analysis with other epignomics dataset can be performed by using `heatmapdata`, `heatmapPlot` functions of the package ***compEpiTools***.

The methods that are most demanding in terms of computational resources are optimized for low memory fingerprint and multi-processor support. *methylPipe* includes a series of objects and methods that can be used as building blocks for the creation of pipelines for the data analysis of epigenomics data, with particular emphasis on DNA methylation, and their integration with any kind of annotation or additional data type.

3 Profiling Genome wide DNA methylation

First of all the *methylPipe* and the genome sequence libraries are loaded:

```
library(methylPipe)
library(BSgenome.Hsapiens.UCSC.hg18)
```

The DNA methylation information can be read from a text file. See the documentation of the `BSprepare` on how to build data suitable to populate a *BSdata* object. Moreover, the methylation information can also be read directly from the sorted SAM file(s) generated from the BISMARK aligner. The function `meth.call` process the sorted SAM file. The user can specify the sequence context in which the methylation information is read from these files either "CpG" or "All". In case of whole genome Bisulfite data especially in case of non-CG methylation the number of potential methylation sites are enormous and vast majority of them are unmethylated. In order to avoid storing too much data while maintaining the ability to identify methylated, unmethylated and uncovered Cytosines, *methylPipe* only stores and access C positions with at least 1 mC read. Genomic regions not covered by sequencing are stored as a *GRanges* object. The function `meth.call` produces methylation call text file and uncovered genomic regions file for each sample in the output folder. Finally, when profiling DNA methylation unmethylated Cs are determined as those genomic cytosines not having any methylated reads and not belogning to uncovered genomic regions.

```
file_loc <- system.file('extdata', 'test_methcall', package='methylPipe')
meth.call(files_location=file_loc, output_folder=tempdir(), no_overlap=TRUE,
          read.context="CpG", Nproc=1)
```

methylPipe adopts TABIX compressed indexing of flatfiles to reduce disk space which allows fast access. The *methylPipe* library includes a small subset of the first two base resolution human DNA methylomes (Lister R et al Nature 2009) for two well known human cell lines: embryonic stem cells (H1) and fetal lung fibroblasts (IMR90). The methylation data can be stored into *BSdata* objects and then collected together within a *BSdataSet* object. The *BSprepare* command is not run in the following example since it requires a local copy of tabix software. Hence, for this example we use pre-generated tabix indexed files from *methylPipe* for H1 and IMR90.

```
file_loc <- system.file('extdata', 'H1_chr20_CG_10k.txt', package='methylPipe')
#BSprepare(files_location=file_loc,output_folder=file_loc, tabix="/path-to-tabix/")
uncov_GR <- GRanges(Rle('chr20'), IRanges(c(14350,69251,84185), c(18349,73250,88184)))
H1data <- system.file('extdata', 'H1_chr20_CG_10k_tabix_out.txt.gz',
                      package='methylPipe')
H1.db <- BSdata(file=H1data, uncov=uncov_GR, org=Hsapiens)
H1.db

## GRanges object with 3 ranges and 0 metadata columns:
##       seqnames         ranges strand
##          <Rle>      <IRanges>  <Rle>
##   [1]    chr20 [14350, 18349]      *
##   [2]    chr20 [69251, 73250]      *
##   [3]    chr20 [84185, 88184]      *
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
## [1] "chr20"
## GRanges object with 6 ranges and 4 metadata columns:
##       seqnames       ranges strand |     Context         C         T
##          <Rle>    <IRanges>  <Rle> | <character> <numeric> <numeric>
##   [1]    chr20 [8179, 8179]      + |          CG         2         4
##   [2]    chr20 [8180, 8180]      - |          CG         4         4
##   [3]    chr20 [8426, 8426]      + |          CG         1         0
##   [4]    chr20 [8427, 8427]      - |          CG         5         0
##   [5]    chr20 [8432, 8432]      + |          CG         1         0
##   [6]    chr20 [8433, 8433]      - |          CG         6         0
##       Significance
##          <numeric>
##   [1]           20
##   [2]           48
##   [3]           14
##   [4]           84
##   [5]           14
##   [6]          102
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Multiple *BSdata* objects can be stored in *BSdataSet* object by specifying group name for each sample either as "C" (control) or "E" (Experiment):

```
IMR90data <- system.file('extdata', 'IMR90_chr20_CG_10k_tabix_out.txt.gz',
                         package='methylPipe')
IMR90.db <- BSdata(file=IMR90data, uncov=uncov_GR, org=Hsapiens)
H1.IMR90.set <- BSdataSet(org=Hsapiens, group=c("C","E"), IMR90=IMR90.db, H1=H1.db)
H1.IMR90.set

## S4 Object of class BSdataSet
##
## BSdata objects contained:

## [1] "IMR90" "H1"
```

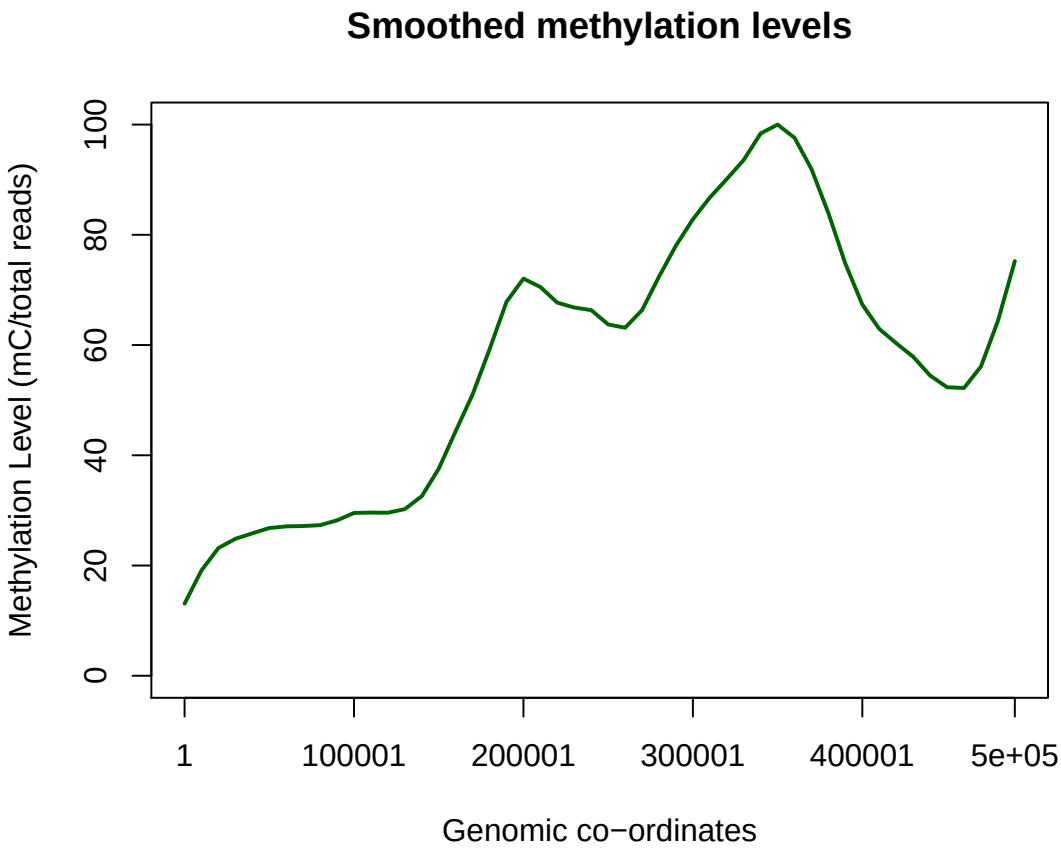
```
##
## Groups of objects:

## [1] "C" "E"

##
## Associated organism genome:
## Homo sapiens
##
```

The data includes the chromosome, the genomic position, strand and sequence context for each methyl-Cytosine (mC). In addition, for each mC the number of reads with C at that genomic position and strand (supporting DNA methylation), the number of reads with T (evidence of an unmethylated C) and the binomial p-value supporting the mC call are attached. Importantly, these data are referenced into the *BSdata* but they are not actually loaded into the memory. The `mCsmoothing` method allow uploading into memory the DNA methylation data for a given genomic region and to display their methylation profile over it. The smoothing can be performed by either "sum" or "mean" of methylation level for each regions.

```
gr <- GRanges("chr20",IRanges(1,5e5))
sres <- mCsmoothing(H1.db, gr, Scorefun='sum', Nbins=50, plot=TRUE)
```



4 Descriptive statistics of DNA methylation

methylPipe allows checking the basic stats about the methylation data such as range, mean and quantile distribution of methylation and assess sample similarity with correlation and clustering analysis. The `methstats` method computes pairwise correlation coefficients (Pearson) between the methylation profiles across all the samples in *BSdataSet* object. It outputs scatter plot matrix of correlation coefficients. Finally, it performs (eucledian distance based) hierarchical clustering of samples and outputs the dendrogram. In the example below, the analysis is performed on *BSdataSet* object of artificially replicated H1 and IMR90.

```
stats.set <- BSdataSet(org=Hsapiens, group=c("C","C","E","E"), IMR_1=IMR90.db,
IMR_2=IMR90.db, H1_1=H1.db,H1_2=H1.db)
stats_res <- methstats(stats.set,chrom="chr20",mcClass='mCG', Nproc=1)
stats_res

## $descriptive_stats
##      IMR_1            IMR_2            H1_1            H1_2
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.5360   1st Qu.:0.5360   1st Qu.:0.6670   1st Qu.:0.6670
##  Median :0.8570   Median :0.8570   Median :0.8480   Median :0.8480
##  Mean   :0.7136   Mean   :0.7136   Mean   :0.7497   Mean   :0.7497
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.9700   3rd Qu.:0.9700
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##
## $correlation_mat
##           IMR_1     IMR_2      H1_1      H1_2
## IMR_1 1.0000000 1.0000000 0.1653737 0.1653737
## IMR_2 1.0000000 1.0000000 0.1653737 0.1653737
## H1_1  0.1653737 0.1653737 1.0000000 1.0000000
## H1_2  0.1653737 0.1653737 1.0000000 1.0000000
```

5 Profiling DNA methylation on genomic ranges

methylPipe allows profiling DNA methylation over a set of genomic regions and determining their absolute or relative methylation profiles. Absolute methylation is here defined as the genomic density of mC calls per bp, while relative methylation is the proportion of density of methylated genomic sites over the density of potential methylation sites.

The `mapBSdata2GRanges` method retrieves mC calls given a *BSdata* object of a sample and a set of genomic regions. In the following example we will extract the CG sites for a set of genomic regions. In particular we will consider a *GRanges* object of genomic regions that spans 2Kb upstream and downstream the TSS of the transcript ids on the first 500Kb of chromosome 20:

```
gr_file <- system.file('extdata', 'GR_chr20.Rdata', package='methylPipe')
load(gr_file)
resmC <- mapBSdata2GRanges(GenoRanges=GR_chr20, Sample=H1.db, context='CG')
head(resmC[[4]])

## GRanges object with 6 ranges and 4 metadata columns:
##       seqnames           ranges strand |     Context         C
```

```
##          <Rle>        <IRanges>  <Rle> | <character> <numeric>
##   [1]    chr20 [116504, 116504]      + |          CG        18
##   [2]    chr20 [116757, 116757]      + |          CG        15
##   [3]    chr20 [116758, 116758]      - |          CG        24
##   [4]    chr20 [116796, 116796]      + |          CG        32
##   [5]    chr20 [116797, 116797]      - |          CG         8
##   [6]    chr20 [117132, 117132]      - |          CG         9
##               T Significance
##       <numeric>    <numeric>
##   [1]        25          202
##   [2]         4          227
##   [3]        12          334
##   [4]         0          564
##   [5]         1          128
##   [6]         3          132
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

To profile DNA methylation on genomic regions of interest the *GEcollection* class (collection of genomic regions) is used. These genomic regions can be any regions of interest such as one or more gene loci, promoter regions, a set of transcription factor binding sites, a set of regions enriched by some histone mark or a set of transposable elements. Most of these regions can be usually retrieved from either publications or public databases like the UCSC Genome Browser and its Table Browser tools (see *compEpiTools* for methods for generation/manipulation of these regions). The *GEcollection* class has data slots and methods tailored for storing and analyzing DNA methylation data. It is an extension of the *RangedSummarizedExperiment* class from the *SummarizedExperiment* package.

For the creation of the *GEcollection*, the `getCpos` and `mapBSdata2GRanges` methods are called automatically to populate its various data slots. Further arguments of the latter function allows to filter the mC sites based on the depth of sequencing, number of reads with C at that genomic position and their binomial p-value. The `profileDNAmetBin` method acts as a wrapper for these methods to profile absolute and relative DNA methylation for a set of regions within a *GEcollection*. In particular for each genomic region, and possibly each bin of it, the absolute methylation density (mC/bp), the density of possible methylation sites (C/bp) and the relative methylation level (mC/C) will be determined. For example, in case of the CG context: mCG/bp, CG/bp and mCG/CG densities will be stored for each bin of each genomic region in the binmC, binC and binrC data slots, respectively.

```
gec.H1 <- profileDNAmetBin(GenoRanges=GR_chr20, Sample=H1.db, mcCLASS='mCG', nbins=3)
binmC(gec.H1)[4:5,]

##                            1       2       3
## chr20:116264-120263 0.00378 0.00911 0.01440
## chr20:153924-157923 0.00969 0.01170 0.00825

binC(gec.H1)[4:5,]

##                           1      2      3
## chr20:116264-120263 0.0075 0.0165 0.0210
## chr20:153924-157923 0.0150 0.0180 0.0105

binrC(gec.H1)[4:5,]
```

```
##                         1    2    3
## chr20:116264-120263 50.4 55.2 68.8
## chr20:153924-157923 64.6 64.8 78.6
```

GEcollection objects can be **subsetted and combined**. Subset can be useful to work only on the regions on a particular strand and/or chromosome or chromosomal region.

```
gec1 <- gec.H1[start(gec.H1) < 153924]
gec2 <- gec.H1[start(gec.H1) > 153924]
```

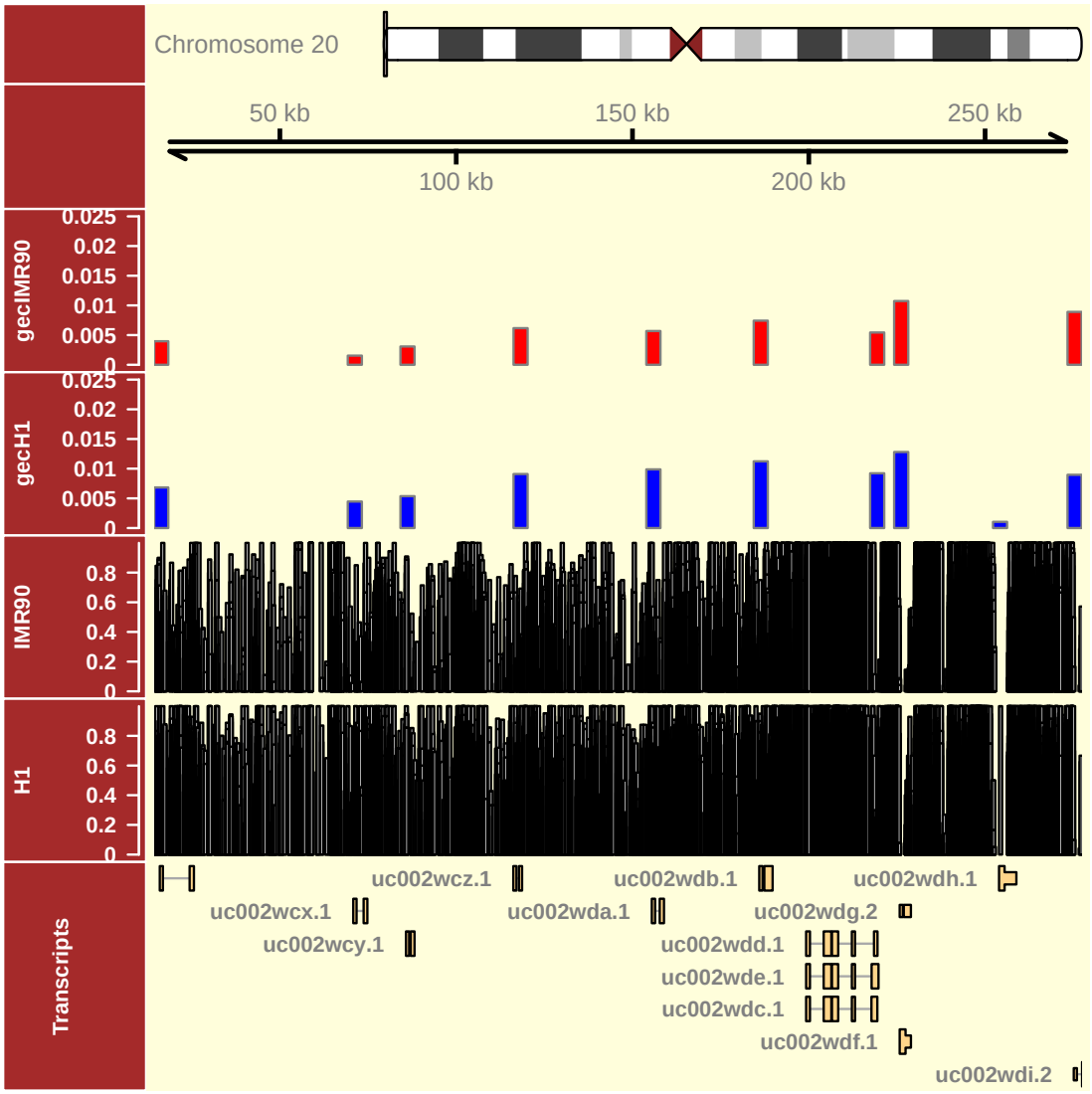
Multiple *GEcollection* objects can be saved into a *GElist* object. This can be convenient if there are many of them and if one would like to iteratively apply the same method to them, for example profiling DNA methylation for their genomic regions in the same sample.

```
gecIMR_file <- system.file('extdata', 'gec.IMR90.Rdata', package='methylPipe')
load(gecIMR_file)
gel <- GElist(gecIMR90=gec.IMR90, gecH1=gec.H1)
print(names(gel))

## [1] "gecIMR90" "gecH1"
```

The *GElist* objects can be visualized using `plotMeth`. This allows methylation data of various samples to be displayed together with the annotation information for the genomic regions of interest. Morover, various epigenomics data tracks can also be visualized together with the methylation information. See the documentation of the `plotMeth` for more details.

```
library(TxDb.Hsapiens.UCSC.hg18.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg18.knownGene
gel <- GElist(gecIMR90=gec.IMR90[1:10], gecH1=gec.H1[1:10])
plotMeth(gel, colors=c("red","blue"), datatype=c("mC","mC"), yLim=c(.025, .025), brmeth=list(IMR9
         mcContext="CG", transcriptDB=txdb, chr="chr20", start=14350, end=277370, org=Hsapiens)
```



6 Differential DNA methylation

The `findDMR` function can be used to identify differentially methylated regions (DMRs) for any sequence context (mCG, mCHG or mCHH). DMRs can be identified comparing the mC methylation levels (the proportion of methylated over total reads for a set of mC) between **two samples** or within a group of **N samples**, with the Wilcoxon or the Kruskal-Wallis non parametric test, respectively. Only cytosine positions covered in both the samples (pairwise analysis) or all the samples (multiple sample analysis) are considered for DMR identification. The algorithm uses dynamic sliding window approach which identifies DMRs depending on methylated cytosine frequency and their relative distance. Moreover, to perform regions specific analysis (such as promoter or CpG island centric) genomic regions could be specified and the alogirithm will report DMRs only within those regions, if any. In a first step, all the evaluated regions are reported in the output with their corresponding statistical significance. The methylation difference is determined in terms of MethDiffPerc (percentage difference between the mean methylation of experiment and control) and log2Enrichment (log2 of mean methylation of experiment over control).

```
DMRs <- findDMR(object= H1.IMR90.set, Nproc=1, ROI=GR_chr20, MCClass='mCG',
     dmrSize=6, dmrBp=800)
```

```
head(DMRs)

## GRanges object with 6 ranges and 3 metadata columns:
##       seqnames         ranges strand |    pValue MethDiff_Perc
##          <Rle>      <IRanges>  <Rle> | <numeric>     <numeric>
##   [1]    chr20 [14404, 15060]      * |     0.036        37.133
##   [2]    chr20 [15001, 15095]      * |     0.036        41.817
##   [3]    chr20 [15760, 15906]      * |     0.787           8.8
##   [4]    chr20 [16170, 16792]      * |     0.059        40.033
##   [5]    chr20 [16721, 17355]      * |     0.059        35.133
##   [6]    chr20 [69287, 69975]      * |     0.059        40.033
##       log2Enrichment
##            <numeric>
##   [1]          1.486
##   [2]          1.341
##   [3]          0.181
##   [4]          1.188
##   [5]          0.917
##   [6]          1.277
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

The `consolidateDMRs` function can be used to multiple-testing correct the DMRs and consolidate them according to their relative distance, type of DMRs and thresholds of p-value/Methylation difference/log enrichment. A final *GRanges* object with the set of DMRs including p-value, methylation difference and log enrichment is provided:

```
hyper.DMRs.conso <- consolidateDMRs(DmrGR=DMRs, pvThr=0.05, GAP=100, type="hyper")
hyper.DMRs.conso[1:4]

## GRanges object with 4 ranges and 3 metadata columns:
##       seqnames         ranges strand |    pValue MethDiff_Perc
##          <Rle>      <IRanges>  <Rle> | <numeric>     <numeric>
##   [1]    chr20 [14404, 15095]      * |      0.01        39.475
##   [2]    chr20 [69927, 70629]      * |     0.036        32.717
##   [3]    chr20 [72059, 72852]      * |      0.01        52.458
##   [4]    chr20 [84742, 85118]      * |     0.036         36.75
##       log2Enrichment
##            <numeric>
##   [1]          1.414
##   [2]           1.32
##   [3]          2.037
##   [4]          0.921
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

7 Session Information

```
sessionInfo()

## R version 3.2.4 (2016-03-10)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.9.5 (Mavericks)
##
## locale:
## [1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils
## [7] datasets  methods   base
##
## other attached packages:
##  [1] TxDb.Hsapiens.UCSC.hg18.knownGene_3.2.2
##  [2] GenomicFeatures_1.22.13
##  [3] AnnotationDbi_1.32.3
##  [4] BSgenome.Hsapiens.UCSC.hg18_1.3.1000
##  [5] BSgenome_1.38.0
##  [6] rtracklayer_1.30.3
##  [7] methylPipe_1.4.5
##  [8] Rsamtools_1.22.0
##  [9] Biostrings_2.38.4
## [10] XVector_0.10.0
## [11] SummarizedExperiment_1.0.2
## [12] Biobase_2.30.0
## [13] GenomicRanges_1.22.4
## [14] GenomeInfoDb_1.6.3
## [15] IRanges_2.4.8
## [16] S4Vectors_0.8.11
## [17] BiocGenerics_0.16.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.3              biovizBase_1.18.0
##  [3] lattice_0.20-33          gtools_3.5.0
##  [5] digest_0.6.9             plyr_1.8.3
##  [7] chron_2.3-47             futile.options_1.0.0
##  [9] acepack_1.3-3.3          RSQLite_1.0.0
## [11] evaluate_0.8.3           ggplot2_2.1.0
## [13] highr_0.5.1              gplots_2.17.0
## [15] zlibbioc_1.16.0          data.table_1.9.6
## [17] gdata_2.17.0             rpart_4.1-10
## [19] splines_3.2.4            BiocParallel_1.4.3
## [21] stringr_1.0.0            foreign_0.8-66
## [23] RCurl_1.95-4.8           biomaRt_2.26.1
## [25] munsell_0.4.3            marray_1.48.0
## [27] Gviz_1.14.4              nnet_7.3-12
```

```
## [29] gridExtra_2.2.1          Hmisc_3.17-2
## [31] codetools_0.2-14         matrixStats_0.50.1
## [33] XML_3.98-1.4             GenomicAlignments_1.6.3
## [35] bitops_1.0-6             grid_3.2.4
## [37] gtable_0.2.0             DBI_0.3.1
## [39] magrittr_1.5             formatR_1.3
## [41] scales_0.4.0             KernSmooth_2.23-15
## [43] stringi_1.0-1            limma_3.26.8
## [45] latticeExtra_0.6-28      futile.logger_1.4.1
## [47] Formula_1.2-1            lambda.r_1.1.7
## [49] RColorBrewer_1.1-2       tools_3.2.4
## [51] dichromat_2.0-0          survival_2.38-3
## [53] colorspace_1.2-6         cluster_2.0.3
## [55] caTools_1.17.1           knitr_1.12.3
## [57] VariantAnnotation_1.16.4
```