segmentSeq: methods for detecting methylation loci and differential methylation

Thomas J. Hardcastle

October 15, 2013

1 Introduction

This vignette introduces analysis methods for data from high-throughput sequencing of bisulphite treated DNA to detect cytosine methylation. The `segmentSeq` package was originally designed to detect siRNA loci [1] and many of the methods developed for this can be used to detect loci of cytosine methylation from replicated (or unreplicated) sequencing data.

2 Preparation

Preparation of the segmentSeq package proceeds as in siRNA analysis. We begin by loading the `segmentSeq` package.

```
>   library(segmentSeq)
```

Note that because the experiments that `segmentSeq` is designed to analyse are usually massive, we should use (if possible) parallel processing as implemented by the `parallel` package. If using this approach, we need to begin by define a *cluster*. The following command will use eight processors on a single machine; see the help page for 'makeCluster' for more information.

```
> cl <- makeCluster(8)
```

If we don't want to parallelise, we can proceed anyway with a `NULL` cluster. Results may be slightly different depending on whether or not a cluster is used owing to the non-deterministic elements of the method.

```
>   cl <- NULL
```

The `segmentSeq` package is designed to read in output from the YAMA (Yet Another Methylome Aligner) program. This is a perl-based package using either bowtie or bowtie2 to align bisulphite treated reads (in an unbiased manner) to a reference and identify the number of times each cytosine is identified as methylated or unmethylated. Unlike most other aligners, YAMA does not require that reads that map to more than one location are discarded, instead it reports the number of alternate matches to the reference for each cytosine. This is then used by `segmentSeq` to weight the observed number of methylated/unmethylated cytosines at a location.

```
> datadir <- system.file("extdata", package = "segmentSeq")
> files <- c("short_18B_C24_C24_trim.fastq_CG_methCalls",
+ "short_Sample_17A_trimmed.fastq_CG_methCalls",
+ "short_13_C24_col_trim.fastq_CG_methCalls",
+ "short_Sample_28_trimmed.fastq_CG_methCalls")
> mD <- readMeths(files = files, dir = datadir,
+ libnames = c("A1", "A2", "B1", "B2"), replicates = c("A","A","B","B"),
+ nonconversion = c(0.004777, 0.005903, 0.016514, 0.006134))
```

We can begin by plotting the distribution of methylation for these samples.
The distribution can be plotted for each sample individually, or as an average
across multiple samples. We can also subtract one distribution from another to
visualise patterns of differential methylation on the genome.

```
> par(mfrow = c(2,1))
> distA <- plotMethDistribution(mD, samples = 1:2, main = "Distributions of methylation")
> distB <- plotMethDistribution(mD, samples = 3:4, add = TRUE, col = "red")
> plotMethDistribution(mD, samples = 1:2, subtract = distB[,2], main = "Differences betwee
```
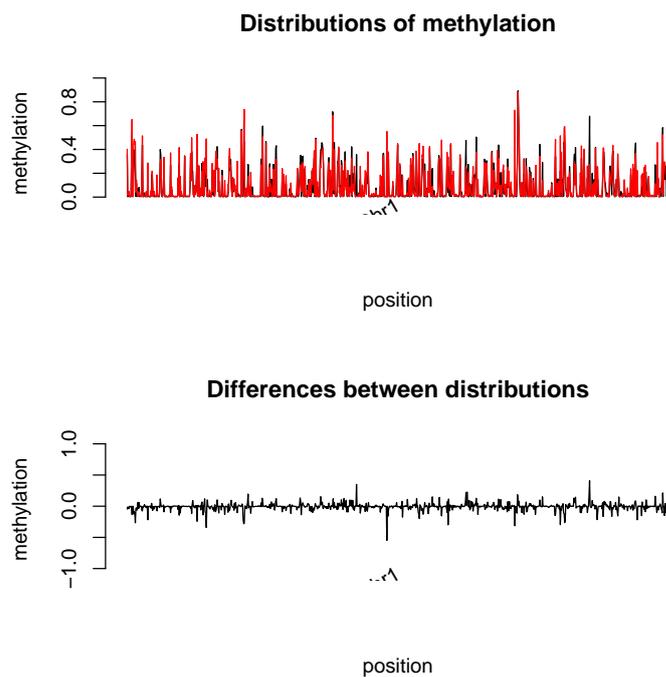

Figure 1: Distributions of methylation on the genome (first two million bases of
chromosome 1.

Next, we process this `alignmentData` object to produce a `segData` object.
This `segData` object contains a set of potential segments on the genome de-
fined by the start and end points of regions of overlapping alignments in the

`alignmentData` object. It then evaluates the number of tags that hit in each of these segments.

```
> sD <- processAD(mD, gap = 300, squeeze = 10, filterProp = 0.05, verbose = TRUE, strandSp
```

We can now construct a segment map from these potential segments.

Segmentation by heuristic Bayesian methods

A fast method of segmentation can be achieved by assuming a binomial distribution on the data with an uninformative beta prior, and identifying those potential segments which have a sufficiently large posterior likelihood that the proportion of methylation exceeds some critical value.

```
> hS <- heuristicSeg(sD, mD, prop = 0.2, cl = cl, gap = 100, getLikes = FALSE)
> hS

Slot "coordinates"
GRanges with 1580 ranges and 0 metadata columns:
        seqnames              ranges strand
           <Rle>           <IRanges>  <Rle>
     [1]    Chr1     [ 108,   948]      +
     [2]    Chr1     [ 150,   837]      -
     [3]    Chr1     [ 929,   929]      -
     [4]    Chr1     [5226,  5465]      +
     [5]    Chr1     [6452,  6452]      +
     ...     ...                 ...    ...
  [1576]    Chr1 [1990187, 1990187]      -
  [1577]    Chr1 [1990298, 1990298]      +
  [1578]    Chr1 [1993118, 1993335]      +
  [1579]    Chr1 [1994611, 1994611]      +
  [1580]    Chr1 [1994857, 1994886]      +
  ---
  seqlengths:
   Chr1
     NA
An object of class "methData"
1580 rows and 4 columns

Slot "replicates"
[1] A A B B
Levels: A B

Slot "libsizes"
      A1        A2        B1        B2
1.004777 1.005903 1.016514 1.006134

Slot "data":
      A1  A2  B1 B2
[1,] 166 145 128 22
[2,]  81  30  35  3
```

```
[3,]   0   0    0  3
[4,]   1   1   24  2
[5,]   2   5    0  0
1575 more rows...

Slot "pairData":
      A1     A2 B1 B2
[1,] 67  89.02 50 11
[2,] 90  33.00 61 15
[3,]  0   0.00  0  1
[4,] 86 120.00 36  9
[5,]  0   3.00  3  1
1575 more rows...

Slot "annotation":
data frame with 0 columns and 1580 rows

Slot "locLikelihoods" (stored on log scale):
Matrix with  1580  rows.
        A   B
1       1   1
2       1   1
3    <NA>   1
4       0   1
5       1   0
...   ... ...
1576    0   1
1577    1   1
1578    1   1
1579    1   0
1580    0   1
```

Within a methylation locus, it is not uncommon to find completely unmethylated cytosines. If the coverage of these cytosines is too high, it is possible that these will cause the locus to be split into two or more fragments. The `mergeMethSegs` function can be used to overcome this splitting by merging loci with identical patterns of expression that are not separated by too great a gap. Merging in this manner is optional, but recommended.

```
> hS <- mergeMethSegs(hS, mD, gap = 5000, cl = cl)
```

We can then estimate posterior likelihoods on the defined loci by applying empirical Bayesian methods. These will not change the locus definition, but will assign likelihoods that the identified loci represent a true methylation locus in each replicate group.

```
> hSL <- lociLikelihoods(hS, mD, cl = cl)

......
```

Visualising loci

By one of these methods, we finally acquire an annotated `methData` object, with the annotations describing the co-ordinates of each segment.

We can use this `methData` object, in combination with the `alignmentMeth` object, to plot the segmented genome.

```
> plotMeth(mD, hSL, chr = "Chr1", limits = c(1, 50000), cap = 10)
```
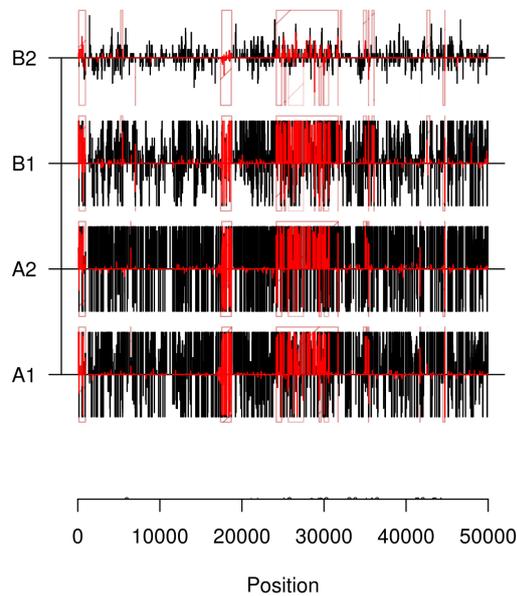

Figure 2: Methylation and identified loci on the first ten thousand bases of chromosome 1.

Differential Methylation analysis

We can also examine the `methData` object for differentially methylated regions using the beta-binomial methods [2] implemented in `baySeq`. We first define a group structure on the data.

```
> groups(hSL) <- list(NDE = c(1,1,1,1), DE = c("A", "A", "B", "B"))
```

We can then determine a prior distribution on the parameters of a beta-binomial model for the data.

```
> hSL <- getPriors.BB(hSL, cl = cl)
```

5

We can then find the posterior likelihoods of the models defined in the groups structure.

```
> hSL <- getLikelihoods.BB(hSL, cl = cl)
```

.

We can then retrieve the data for the top differentially methylated regions.

```
> topCounts(hSL, "DE")
```

	seqnames	start	end	width	strand	A1	A2	B1	B2	Likelihood
1	Chr1	958479	959092	614	+	0:284	0:702	1121:365	87:33	0.9999227
2	Chr1	1649924	1649957	34	+	56:6	43:2	1:49	0:8	0.9997544
3	Chr1	1509220	1509220	1	+	99:20	26:4	0:42	0:4	0.9997190
4	Chr1	1733868	1733987	120	+	0:23	0:85	48:5	10:1	0.9996601
5	Chr1	1181823	1181829	7	+	0:50	0:66	79:42	14:7	0.9996516
6	Chr1	880854	880896	43	+	22:5	69:11	1:72	0:4	0.9996231
7	Chr1	1439950	1440002	53	+	1:148	0:47	67:12	6:0	0.9995920
8	Chr1	526692	526853	162	+	68:77	64:72	0:145	0:5	0.9995839
9	Chr1	1655890	1656076	187	+	0:154	0:81	28:13	7:3	0.9995712
10	Chr1	76839	76944	106	+	0:40	0:46	128:33	6:1	0.9995650

	DE	FDR.DE
1	B>A	7.726324e-05
2	A>B	1.614383e-04
3	A>B	2.012903e-04
4	B>A	2.359420e-04
5	B>A	2.584383e-04
6	A>B	2.781753e-04
7	B>A	2.967254e-04
8	A>B	3.116472e-04
9	B>A	3.246610e-04
10	B>A	3.356967e-04

References

[1] Thomas J. Hardcastle and Krystyna A. Kelly and David C. Baulcombe. *Identifying small RNA loci from high-throughput sequencing data.* Bioinformatics (2012).

[2] Thomas J. Hardcastle and Krystyna A. Kelly. *Empirical Bayesian analysis of paired high-throughput sequencing data with a beta-binomial distribution.* BMC Bioinformatics (2013).