Methylation status calling with methIMPUTE

Aaron Taudt[*]

February 6, 2017

Contents

[*] aaron.taudt@gmail.com

1 Introduction

Methimpute implements a powerful HMM-based binomial test for methylation status calling. Besides improved accuracy over the classical binomial test, the HMM allows imputation of the methylation status of **all cytosines** in the genome. It achieves this by borrowing information from neighboring covered cytosines. Of course, if there are long stretches of uncovered sequence, the imputation might not be very reliable. For this reason, *methimpute* also reports a confidence score for every position which allows judgement of the reliability of the imputation and methylation status calling procedure.

For the exact workings of *methimpute* we refer the interested reader to our publication TODO.

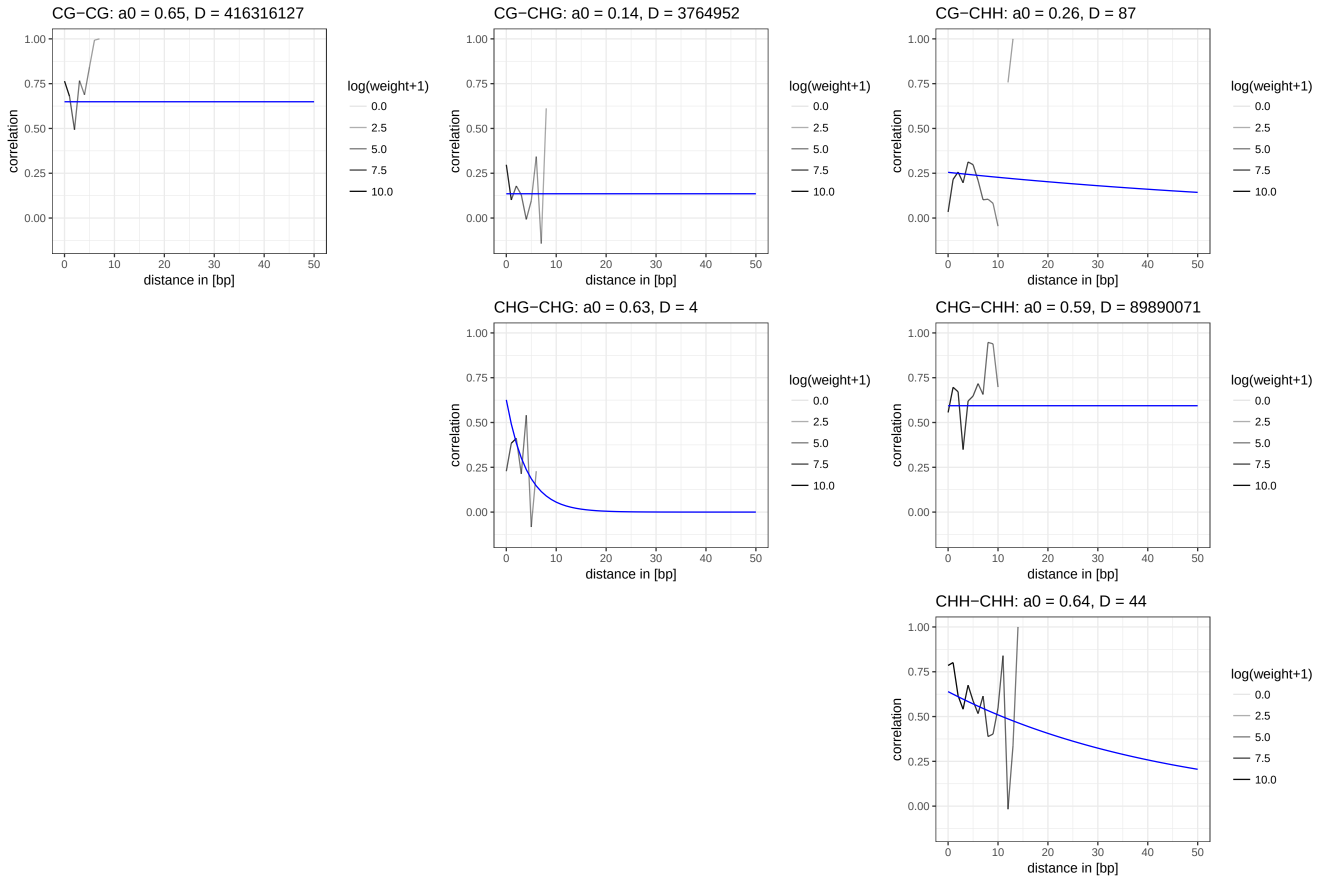
2 Methylation status calling

The following example explains the necessary steps for methylation status calling (and imputation). To keep the calculation time short, it uses only the first 200.000 bp of the Arabidopsis genome. The example consists of three steps: 1) Data import, 2) estimating the distance correlation and 3) methylation status calling. At the end of this example you will see that positions without counts are assigned a methylation status, but the confidence (column "posteriorMax") is generally quite low for those cytosines. Column "posteriorMeth" gives the HMM posterior probability for a cytosine being methylated, which can be interpreted as a methylation level for each site. Column "status" contains the imputed and non-imputed methylation status calls.

```
library(methimpute)

# === Step 1: Importing the data === #
# We load an example file in BSSeeker format that comes with the package
file <- system.file("extdata","arabidopsis_bsseeker.txt.gz", package="methimpute")
bsseeker.data <- importBSSeeker(file)
print(bsseeker.data)

## GRanges object with 110927 ranges and 2 metadata columns:
##            seqnames           ranges strand |  context   counts
##               <Rle>        <IRanges>  <Rle> | <factor> <matrix>
##        [1]     chr1         [34, 34]      - |      CHG      0:4
##        [2]     chr1         [80, 80]      - |      CHH      2:9
##        [3]     chr1         [84, 84]      + |      CHH      1:1
##        [4]     chr1         [85, 85]      + |      CHH      1:1
##        [5]     chr1         [86, 86]      + |      CHH      1:1
##        ...      ...              ...    ... .      ...      ...
##   [110923]     chr1 [533552, 533552]      - |       CG      2:2
##   [110924]     chr1 [533554, 533554]      - |       CG      2:2
##   [110925]     chr1 [533595, 533595]      + |      CHG      0:1
##   [110926]     chr1 [533601, 533601]      + |      CHG      0:2
##   [110927]     chr1 [533614, 533614]      + |       CG      0:2
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths

# Because most methylation extractor programs report only covered cytosines,
# we need to inflate the data to inlcude all cytosines (including non-covered sites)
data(arabidopsis_toydata)
methylome <- inflateMethylome(bsseeker.data, arabidopsis_toydata)
print(methylome)

## GRanges object with 200000 ranges and 2 metadata columns:
##            seqnames           ranges strand |  context   counts
##               <Rle>        <IRanges>  <Rle> | <factor> <matrix>
##        [1]     chr1           [1, 1]      + |      CHH      0:0
##        [2]     chr1           [2, 2]      + |      CHH      0:0
##        [3]     chr1           [3, 3]      + |      CHH      0:0
##          [4]      chr1             [8, 8]       + |       CHH       0:0
##          [5]      chr1             [9, 9]       + |       CHH       0:0
##          ...       ...                ...     ... .       ...       ...
##     [199996]      chr1 [533605, 533605]       - |       CHH       0:0
##     [199997]      chr1 [533608, 533608]       - |       CHH       0:0
##     [199998]      chr1 [533609, 533609]       - |       CHH       0:0
##     [199999]      chr1 [533611, 533611]       - |       CHH       0:0
##     [200000]      chr1 [533614, 533614]       + |        CG       0:2
##     -------
##     seqinfo: 7 sequences from an unspecified genome
```

```
# === Step 2: Obtain correlation parameters === #
# The correlation of methylation levels between neighboring cytosines is an important
# parameter for the methylation status calling, so we need to get it first. Keep in mind
# that we only use the first 200.000 bp here, that's why the plot looks a bit meagre.
distcor <- distanceCorrelation(methylome)
fit <- estimateTransDist(distcor)
print(fit)
```

```
## $transDist
##        CG-CG       CG-CHG       CG-CHH      CHG-CHG      CHG-CHH      CHH-CHH
## 4.163161e+08 3.764952e+06 8.661062e+01 4.124081e+00 8.989007e+07 4.410062e+01
##
## $plot
```



```
# === Step 3: Methylation status calling (and imputation) === #
model <- callMethylation(data = methylome, transDist = fit$transDist)
```

```
##  Iteration                log(P)              dlog(P)     Time in sec
##          0                -inf                   -              0
##          1       -39690.988136                 inf              0
##          2       -24554.965405        15136.022731              0
##          3       -22835.947766         1719.017639              1
##          4       -22425.161398          410.786368              1
##          5       -22261.590523          163.570876              1
##          6       -22175.430276           86.160247              1
##          7       -22124.297165           51.133111              2
##          8       -22092.230088           32.067077              2
##          9       -22071.383277           20.846811              2
##         10       -22057.398106           13.985171              2
##         11       -22047.724189            9.673917              2
##         12       -22040.827759            6.896430              2
##         13       -22035.766704            5.061055              3
##         14       -22031.950034            3.816669              3
##         15       -22028.998772            2.951263              3
##         16       -22026.664449            2.334323              3
##         17       -22024.780663            1.883786              4
##         18       -22023.233679            1.546984              4
##         19       -22021.944170            1.289508              4
##  Iteration              log(P)             dlog(P)    Time in sec
##         20       -22020.855646            1.088524              4
##         21       -22019.927000            0.928646              4
## HMM: Convergence reached!
```

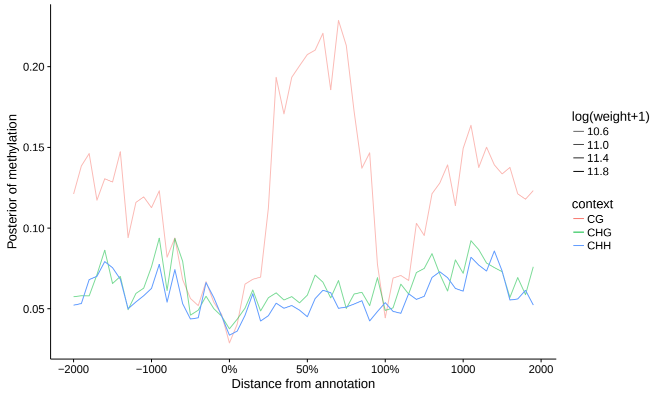
```
# The confidence in the methylation status call is given in the column "posteriorMax".
# For further analysis one could split the results into high-confidence (posteriorMax >= 0.98)
# and low-confidence calls (posteriorMax < 0.98) for instance.
print(model)
```

```
## GRanges object with 200000 ranges and 7 metadata columns:
##            seqnames           ranges strand |  context   counts  distance transitionContext posteriorMax
##               <Rle>        <IRanges>  <Rle> | <factor> <matrix> <numeric>          <factor>    <numeric>
##         [1]    chr1           [1, 1]      + |      CHH      0:0       Inf              <NA>    0.5401411
##         [2]    chr1           [2, 2]      + |      CHH      0:0         0           CHH-CHH    0.6137591
##         [3]    chr1           [3, 3]      + |      CHH      0:0         0           CHH-CHH    0.6760382
##         [4]    chr1           [8, 8]      + |      CHH      0:0         4           CHH-CHH    0.6983594
##         [5]    chr1           [9, 9]      + |      CHH      0:0         0           CHH-CHH    0.7481762
##         ...     ...              ...    ... .      ...      ...       ...               ...          ...
##    [199996]    chr1 [533605, 533605]      - |      CHH      0:0         1           CHG-CHH    0.9265557
##    [199997]    chr1 [533608, 533608]      - |      CHH      0:0         2           CHH-CHH    0.9372669
##    [199998]    chr1 [533609, 533609]      - |      CHH      0:0         0           CHH-CHH    0.9592129
##    [199999]    chr1 [533611, 533611]      - |      CHH      0:0         1           CHH-CHH    0.9757160
##    [200000]    chr1 [533614, 533614]      + |       CG      0:2         2            CHH-CG    0.9759694
##            posteriorMeth       status
##                <numeric>     <factor>
##         [1]    0.3282551 Unmethylated
##         [2]    0.2762159 Unmethylated
##         [3]    0.2320815 Unmethylated
##         [4]    0.2177778 Unmethylated
##         [5]    0.1820458 Unmethylated
##         ...          ...          ...
##    [199996]   0.04683761 Unmethylated
##    [199997]   0.03998640 Unmethylated
##    [199998]   0.02519109 Unmethylated
##    [199999]   0.01379039 Unmethylated
##    [200000]   0.01283037 Unmethylated
##   -------
##   seqinfo: 7 sequences from an unspecified genome
```

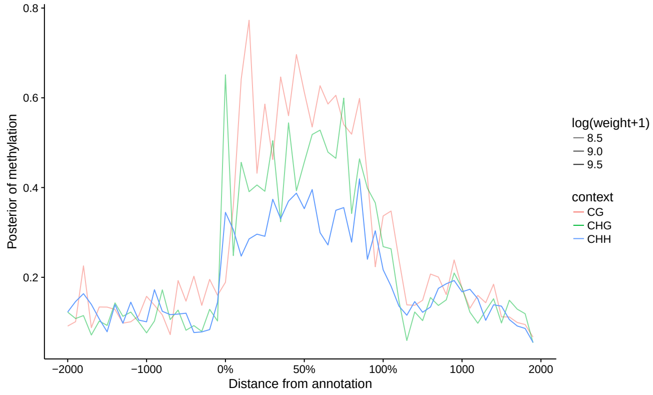
3 Plots and enrichment analysis

This package also offers plotting functions for a simple enrichment analysis. Let's say we are interested in the methylation level around genes and transposable elements.

```r
# Note that the plots look a bit ugly because our toy data has only 200000 datapoints.
data(arabidopsis_genes)
plotEnrichment(model$data, annotation=arabidopsis_genes, range = 2000)
```



```r
data(arabidopsis_TEs)
plotEnrichment(model$data, annotation=arabidopsis_TEs, range = 2000)
```



4 Session Info

```r
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=de_DE.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=de_DE.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=de_DE.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=de_DE.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, stats4, utils
- Other packages: BiocGenerics 0.20.0, cowplot 0.7.0, devtools 1.12.0, GenomeInfoDb 1.10.2, GenomicRanges 1.26.2, ggplot2 2.2.1, IRanges 2.8.1, knitr 1.15.1, methimpute 0.99.0, Rcpp 0.12.9, S4Vectors 0.12.1
- Loaded via a namespace (and not attached): assertthat 0.1, BiocStyle 2.1.14, colorspace 1.2-6, digest 0.6.10, evaluate 0.10, grid 3.3.0, gtable 0.2.0, highr 0.6, labeling 0.3, lazyeval 0.2.0, magrittr 1.5, memoise 1.0.0, minpack.lm 1.2-1, munsell 0.4.3, plyr 1.8.4, reshape2 1.4.2, scales 0.4.1, stringi 1.1.2, stringr 1.1.0, tibble 1.2, tools 3.3.0, withr 1.0.2, XVector 0.14.0, zlibbioc 1.20.0