Methylation status calling with METHimpute

Aaron Taudt*

September 28, 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. The confidence in the methylation status call is reported as well. Methimpute also outputs context-specific conversion rates, which might be used to optimize the experimental 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
```

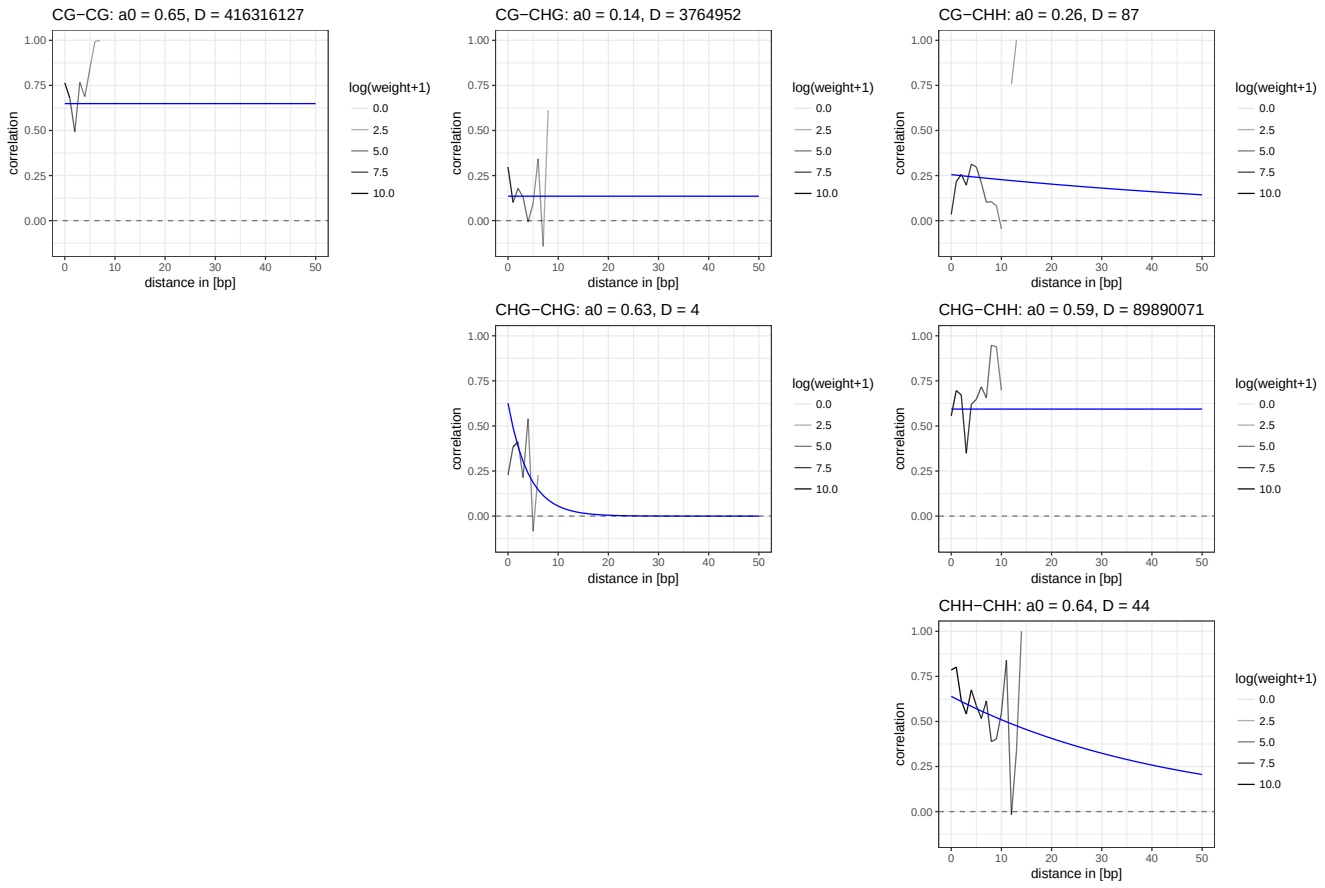
```
# Because most methylation extractor programs report only covered cytosines,
# we need to inflate the data to inlcude all cytosines (including non-covered sites)
fasta.file <- system.file("extdata","arabidopsis_sequence.fa.gz", package="methimpute")
cytosine.positions <- extractCytosinesFromFASTA(fasta.file, contexts = c('CG','CHG','CHH'))
seqlengths(cytosine.positions) <- seqlengths(bsseeker.data) # only necessary for our toy example
methylome <- inflateMethylome(bsseeker.data, cytosine.positions)
print(methylome)
```

```
## GRanges object with 199978 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
##         ...      ...             ...    ... .      ...      ...
##    [199974]     chr1 [533554, 533554]      - |       CG      2:2
##    [199975]     chr1 [533557, 533557]      + |      CHH      0:0
##    [199976]     chr1 [533560, 533560]      + |       CG      0:0
##    [199977]     chr1 [533561, 533561]      - |       CG      0:0
##    [199978]     chr1 [533565, 533565]      - |      CHH      0:0
##    -------
##    seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

```
# ===== 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 8.663764e+01 4.124081e+00 8.989007e+07 4.409969e+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       -40631.538364                inf              1
##          2       -26304.340570       14327.197794              1
##          3       -24210.680366        2093.660204              1
##          4       -23635.136829         575.543537              1
##          5       -23374.140150         260.996679              1
##          6       -23224.427402         149.712749              1
##          7       -23134.096529          90.330873              1
##          8       -23079.925890          54.170638              1
##          9       -23047.280470          32.645421              2
##         10       -23027.416339          19.864131              2
##         11       -23015.215444          12.200895              2
##         12       -23007.665702           7.549743              2
##         13       -23002.970956           4.694745              2
##         14       -23000.044292           2.926664              2
##         15       -22998.218552           1.825740              2
##         16       -22997.079038           1.139514              2
##         17       -22996.365767           0.713271              2
## 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 199978 ranges and 10 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>    0.6606448
##        [2]     chr1           [2, 2]      + |      CHH      0:0         0           CHH-CHH    0.7068376
##        [3]     chr1           [3, 3]      + |      CHH      0:0         0           CHH-CHH    0.7471539
##        [4]     chr1           [8, 8]      + |      CHH      0:0         4           CHH-CHH    0.7624978
##        [5]     chr1           [9, 9]      + |      CHH      0:0         0           CHH-CHH    0.7963517
##        ...      ...              ...    ... .      ...      ...       ...               ...          ...
##   [199974]     chr1 [533554, 533554]      - |       CG      2:2         0             CG-CG    0.9999923
##   [199975]     chr1 [533557, 533557]      + |      CHH      0:0         2           CG-CHH    0.5029823
##   [199976]     chr1 [533560, 533560]      + |       CG      0:0         2           CHH-CG    0.5294277
##   [199977]     chr1 [533561, 533561]      - |       CG      0:0         0             CG-CG    0.5595388
##   [199978]     chr1 [533565, 533565]      - |      CHH      0:0         3           CG-CHH    0.7689333
##            posteriorMeth posteriorUnmeth       status rc.meth.lvl rc.counts
##                <numeric>       <numeric>     <factor>   <numeric>  <matrix>
##        [1]     0.3393552       0.6606448 Unmethylated  0.11618207       0:0
##        [2]     0.2931624       0.7068376 Unmethylated  0.10047754       0:0
##        [3]     0.2528461       0.7471539 Unmethylated  0.08677089       0:0
##        [4]     0.2375022       0.7624978 Unmethylated  0.08155434       0:0
##        [5]     0.2036483       0.7963517 Unmethylated  0.07004476       0:0
##        ...           ...             ...          ...         ...       ...
##   [199974]     0.9999923    7.668704e-06   Methylated   0.8238826       6:7
##   [199975]     0.4970177    5.029823e-01 Unmethylated   0.1697838       0:0
##   [199976]     0.4705723    5.294277e-01 Unmethylated   0.3906849       0:0
##   [199977]     0.4404612    5.595388e-01 Unmethylated   0.3660465       0:0
##   [199978]     0.2310667    7.689333e-01 Unmethylated   0.0793664       0:0
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

2.1 Description of columns

- **context** The sequence context of the cytosine.
- **counts** Counts for methylated and total number of reads at each position.
- **distance** The distance in base-pairs from the previous to the current cytosine.
- **transitionContext** Transition context in the form "previous-current".
- **posteriorMax** Maximum posterior value of the methylation status call, can be interpreted as the confidence in the call.
- **posteriorMeth** Posterior value of the "methylated" component.
- **posteriorUnmeth** Posterior value of the "unmethylated" component.
- **status** Methylation status.
- **rc.meth.lvl** Recalibrated methylation level, calculated from the posteriors and the fitted parameters (see ?methimputeBinomialHMM for details).
- **rc.counts** Recalibrated counts for methylated and total number of reads at each position (see ?methimputeBinomialHMM for details).

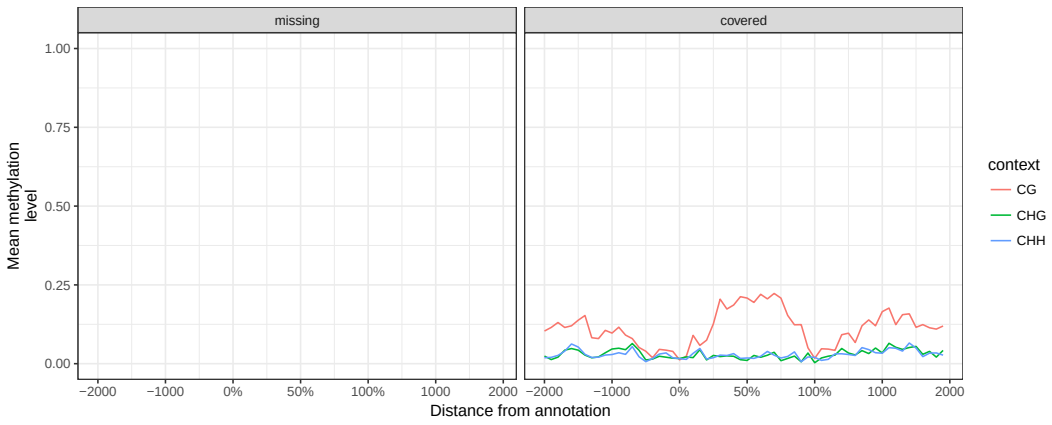
3 Plots and enrichment analysis

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. We would also like to see how the imputation works on cytosines with missing data compared to covered cytosines.

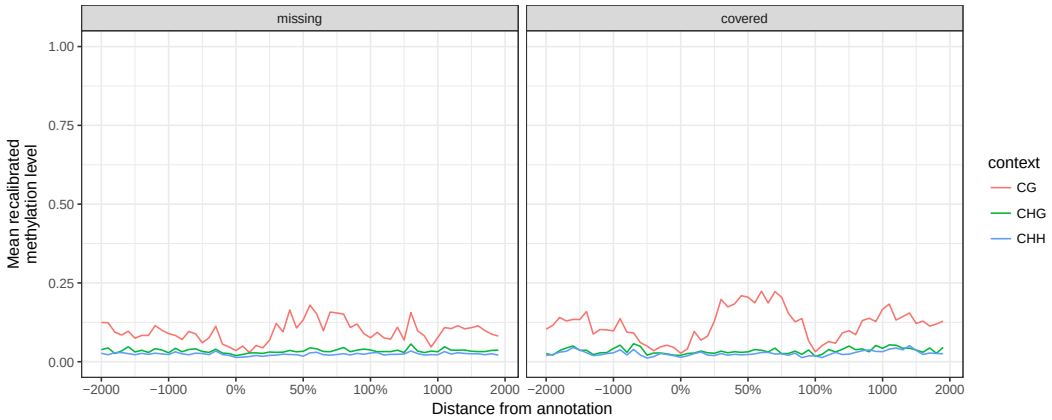
```
# Define categories to distinguish missing from covered cytosines
model$data$category <- factor('covered', levels=c('missing', 'covered'))
model$data$category[model$data$counts[,'total']>=1] <- 'covered'
model$data$category[model$data$counts[,'total']==0] <- 'missing'

# Note that the plots look a bit ugly because our toy data has only 200.000 datapoints.
data(arabidopsis_genes)
plotEnrichment(model$data, annotation=arabidopsis_genes, range = 2000, category.column = 'category')
```

```
## $meth.lvl
```

```
## Warning:  Removed 180 rows containing missing values (geom_path).
```



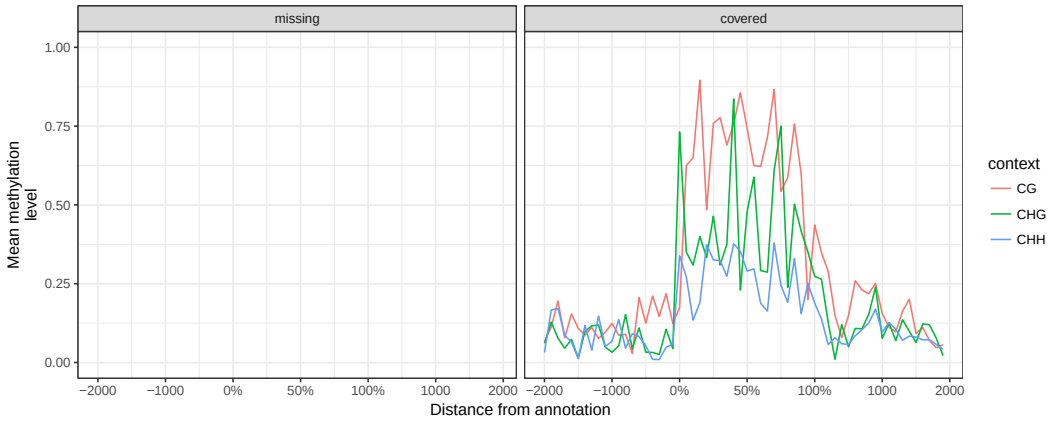

```
##
## $rc.meth.lvl
```



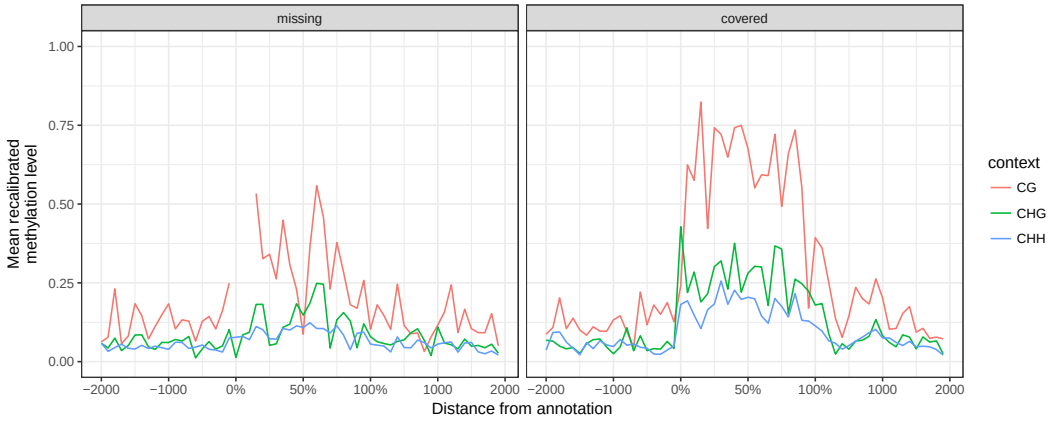

```
data(arabidopsis_TEs)
plotEnrichment(model$data, annotation=arabidopsis_TEs, range = 2000, category.column = 'category')
```

```
## $meth.lvl
```

```
## Warning:  Removed 180 rows containing missing values (geom_path).
```




```
##
## $rc.meth.lvl
```



4 Session Info

```
toLatex(sessionInfo())
```

- R version 3.4.1 (2017-06-30), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=de_DE.UTF-8, LC_COLLATE=C, 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
- Running under: Ubuntu 17.04
- Matrix products: default
- BLAS: /usr/local/lib/R/lib/libRblas.so
- LAPACK: /usr/local/lib/R/lib/libRlapack.so
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: BiocGenerics 0.22.0, GenomeInfoDb 1.12.2, GenomicRanges 1.28.5, IRanges 2.10.3, S4Vectors 0.14.4, ggplot2 2.2.1, methimpute 0.99.2
- Loaded via a namespace (and not attached): BiocStyle 2.4.1, Biostrings 2.44.2, GenomeInfoDbData 0.99.0, RCurl 1.95-4.8, Rcpp 0.12.12, XVector 0.16.0, backports 1.1.0, bitops 1.0-6, colorspace 1.3-2, compiler 3.4.1, cowplot 0.8.0, digest 0.6.12, evaluate 0.10.1, grid 3.4.1, gtable 0.2.0, highr 0.6, htmltools 0.3.6, knitr 1.17, labeling 0.3, lazyeval 0.2.0, magrittr 1.5, minpack.lm 1.2-1, munsell 0.4.3, plyr 1.8.4, reshape2 1.4.2, rlang 0.1.2, rmarkdown 1.6, rprojroot 1.2, scales 0.5.0, stringi 1.1.5, stringr 1.2.0, tibble 1.3.4, tools 3.4.1, yaml 2.1.14, zlibbioc 1.22.0