Methylation status calling with METHimpute

Aaron Taudt [*]

[*]aaron.taudt@gmail.com

May 2, 2019

Contents

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. The HMM can also be used to impute the methylation status for binned data instead of individual cytosines. Furthermore, *methimpute* 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 at https://doi.org/10.1186/s12864-018-4641-x.

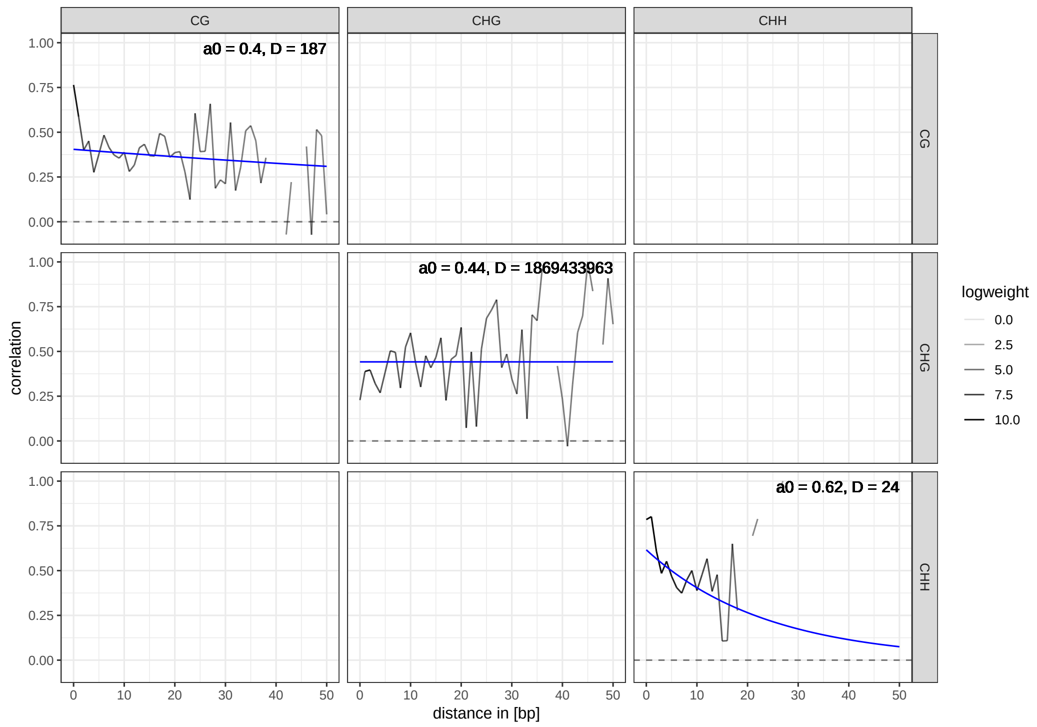
2 Methylation status calling on individual cytosines

The following examples explain 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, whereas it is high for well-covered cytosines (>=0.99).

2.1 Separate-context model

The separate-context model runs a separate HMM for each context. This assumes that only within-context correlations are important, and between-context correlations do not need to be considered.

```
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      - |      CHG      0:4
##          [2]     chr1        80      - |      CHH      2:9
##          [3]     chr1        84      + |      CHH      1:1
##          [4]     chr1        85      + |      CHH      1:1
##          [5]     chr1        86      + |      CHH      1:1
##          ...      ...       ...    ... .      ...      ...
##     [110923]     chr1    533552      - |       CG      2:2
##     [110924]     chr1    533554      - |       CG      2:2
##   [110925]     chr1    533595      + |      CHG      0:1
##   [110926]     chr1    533601      + |      CHG      0:2
##   [110927]     chr1    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'))
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      + |      CHH      0:0
##        [2]     chr1         2      + |      CHH      0:0
##        [3]     chr1         3      + |      CHH      0:0
##        [4]     chr1         8      + |      CHH      0:0
##        [5]     chr1         9      + |      CHH      0:0
##        ...      ...       ...    ... .      ...      ...
##   [199974]     chr1    533554      - |       CG      2:2
##   [199975]     chr1    533557      + |      CHH      0:0
##   [199976]     chr1    533560      + |       CG      0:0
##   [199977]     chr1    533561      - |       CG      0:0
##   [199978]     chr1    533565      - |      CHH      0:0
##   -------
##   seqinfo: 1 sequence 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, separate.contexts = TRUE)
fit <- estimateTransDist(distcor)
print(fit)

## $transDist
##        CG-CG      CHG-CHG      CHH-CHH
## 1.866779e+02 1.869434e+09 2.374003e+01
##
## $plot

## Warning:  Removed 23 rows containing missing values (geom_path).
```



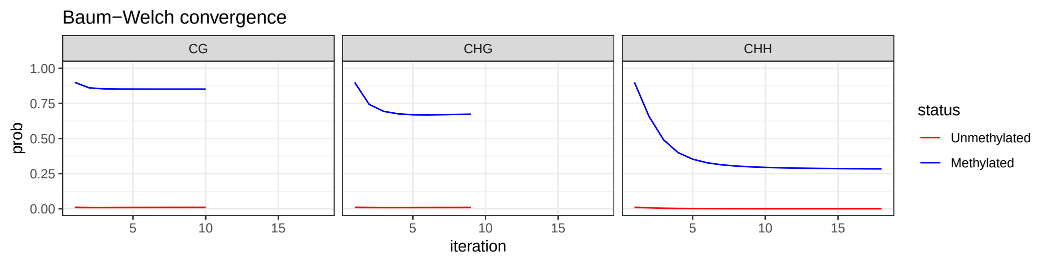
```
# ===== Step 3: Methylation status calling (and imputation) ===== #

model <- callMethylationSeparate(data = methylome, transDist = fit$transDist,
                                 verbosity = 0)
# 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 9 metadata columns:
##              seqnames    ranges strand |  context   counts  distance
##                 <Rle> <IRanges>  <Rle> | <factor> <matrix> <numeric>
##          [1]     chr1         1      + |      CHH      0:0       Inf
##          [2]     chr1         2      + |      CHH      0:0         0
##          [3]     chr1         3      + |      CHH      0:0         0
##          [4]     chr1         8      + |      CHH      0:0         4
##          [5]     chr1         9      + |      CHH      0:0         0
##          ...      ...       ...    ... .      ...      ...       ...
##     [199974]     chr1    533554      - |       CG      2:2         0
##     [199975]     chr1    533557      + |      CHH      0:0         8
##     [199976]     chr1    533560      + |       CG      0:0         5
##     [199977]     chr1    533561      - |       CG      0:0         0
##     [199978]     chr1    533565      - |      CHH      0:0         7
##              transitionContext      posteriorMax     posteriorMeth
##                       <factor>         <numeric>         <numeric>
##          [1]              <NA> 0.500403392500593 0.500403392500593
##          [2]           CHH-CHH 0.630124466026645 0.369875533974143
##          [3]           CHH-CHH 0.726585843137978  0.27341415686281
##          [4]           CHH-CHH 0.751636531740794 0.248363468259994
##          [5]           CHH-CHH 0.816371230118708  0.18362876988208
##        ...              ...              ...              ...
##   [199974]            CG-CG 0.999975488842536 0.999975488842536
##   [199975]          CHH-CHH  0.77422277474309  0.22577722525691
##   [199976]            CG-CG 0.905791249227651 0.905791249227651
##   [199977]            CG-CG 0.830499551294059 0.830499551294059
##   [199978]          CHH-CHH  0.74804177720675  0.25195822279325
##                posteriorUnmeth       status        rc.meth.lvl
##                      <numeric>     <factor>          <numeric>
##        [1]    0.499596607500194   Methylated  0.142145049642913
##        [2]    0.630124466026645 Unmethylated  0.105094723888744
##        [3]    0.726585843137978 Unmethylated 0.0777141670864091
##        [4]    0.751636531740794 Unmethylated 0.0706035304485416
##        [5]    0.816371230118708 Unmethylated 0.0522285897303228
##        ...                  ...          ...                ...
##   [199974] 2.45111574637451e-05   Methylated  0.851514345575296
##   [199975]     0.77422277474309 Unmethylated 0.0641924265640975
##   [199976]   0.0942087507723487   Methylated  0.772216179664385
##   [199977]    0.169500448705941   Methylated  0.708824538144899
##   [199978]     0.74804177720675 Unmethylated 0.0716239013330943
##   -------
##   seqinfo: 1 sequence from an unspecified genome
```

```
# Bisulfite conversion rates can be obtained with
1 - model$params$emissionParams$Unmethylated
```

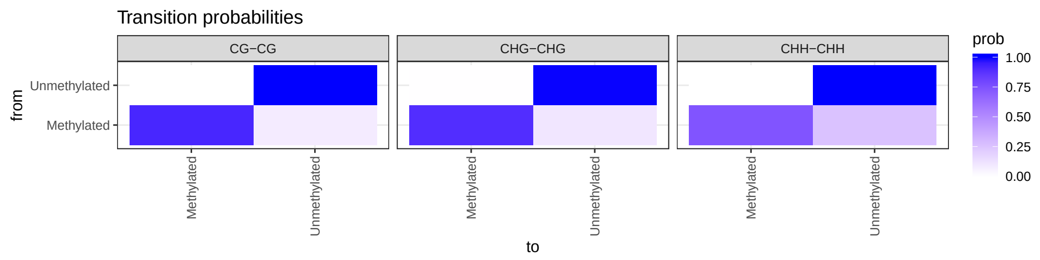
```
##          prob
## CG  0.9904123
## CHG 0.9907659
## CHH 0.9998944
```

You can also check several properties of the fitted Hidden Markov Model, such as convergence or transition probabilities, and check how well the fitted distributions describe the data.

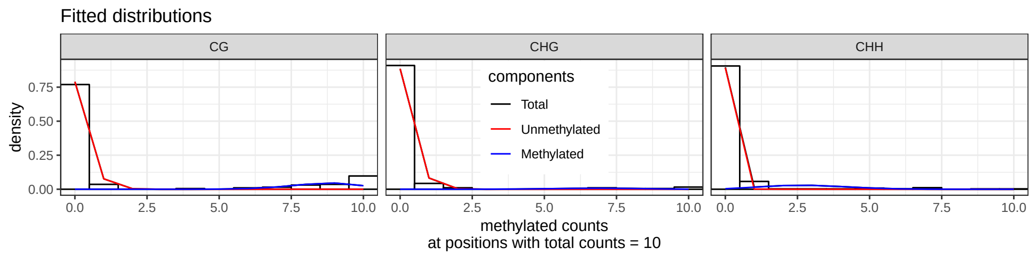
```
plotConvergence(model)
```



```
plotTransitionProbs(model)
```



```
plotHistogram(model, total.counts = 10)
```

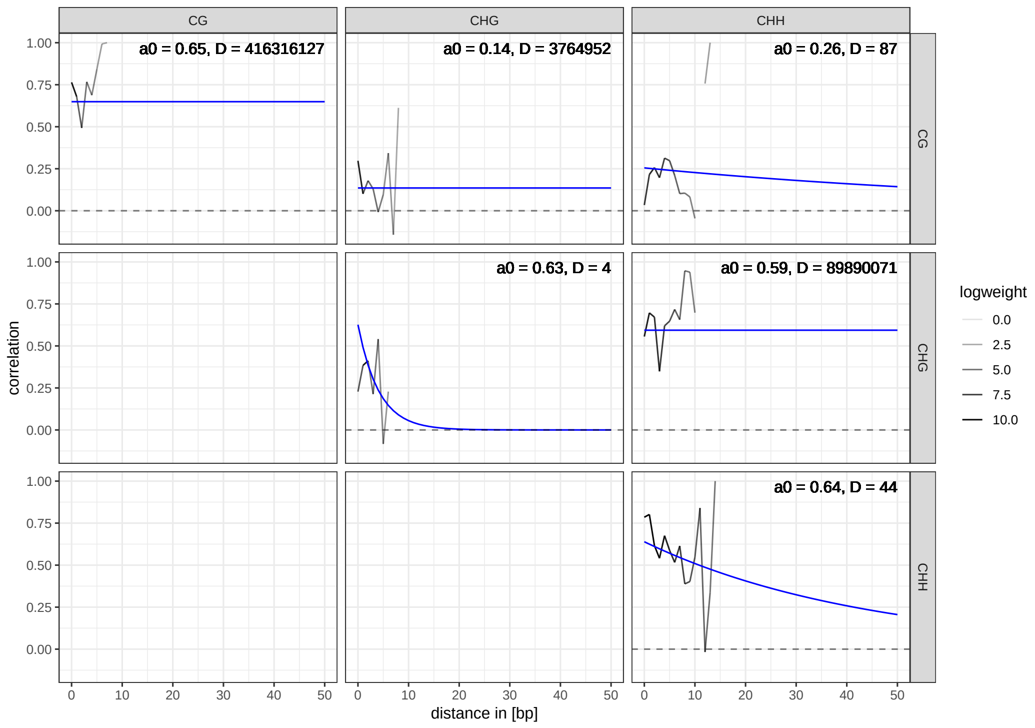


2.2 Interacting-context model

The interacting-context model runs a single HMM for all contexts. This takes into account the within-context and between-context correlations and should be more accurate than the separate-context model if sufficient data is available. However, we have observed that in low coverage settings too much information from well covered contexts is diffusing into the low covered contexts (e.g. CHH and CHG will look like CG with very low coverage). In this case, please use the separate-context model in section 2.1.

```
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      - |      CHG      0:4
##          [2]     chr1        80      - |      CHH      2:9
##          [3]     chr1        84      + |      CHH      1:1
##          [4]     chr1        85      + |      CHH      1:1
##          [5]     chr1        86      + |      CHH      1:1
##          ...      ...       ...    ... .      ...      ...
##     [110923]     chr1    533552      - |       CG      2:2
##     [110924]     chr1    533554      - |       CG      2:2
##     [110925]     chr1    533595      + |      CHG      0:1
##     [110926]     chr1    533601      + |      CHG      0:2
##     [110927]     chr1    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'))
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      + |      CHH      0:0
##        [2]     chr1         2      + |      CHH      0:0
##        [3]     chr1         3      + |      CHH      0:0
##        [4]     chr1         8      + |      CHH      0:0
##        [5]     chr1         9      + |      CHH      0:0
##        ...      ...       ...    ... .      ...      ...
##   [199974]     chr1    533554      - |       CG      2:2
##   [199975]     chr1    533557      + |      CHH      0:0
##   [199976]     chr1    533560      + |       CG      0:0
##   [199977]     chr1    533561      - |       CG      0:0
##   [199978]     chr1    533565      - |      CHH      0:0
##   -------
##   seqinfo: 1 sequence 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 8.663764e+01 4.124081e+00 8.989007e+07 4.409969e+01
##
## $plot

## Warning:  Removed 24 rows containing missing values (geom_path).
```



```
# ===== 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               0
##          2         -26304.340570        14327.197794               0
##          3         -24210.680366         2093.660204               0
##          4         -23635.136829          575.543537               0
##          5         -23374.140150          260.996679               0
##          6         -23224.427402          149.712749               0
##          7         -23134.096529           90.330873               0
##          8         -23079.925890           54.170638               1
##          9         -23047.280470           32.645421               1
##         10         -23027.416339           19.864131               1
##         11         -23015.215444           12.200895               1
##         12         -23007.665702            7.549743               1
##         13         -23002.970956            4.694745               1
##         14         -23000.044292            2.926664               1
##         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 9 metadata columns:
```

```
##            seqnames    ranges strand |  context   counts  distance
##               <Rle> <IRanges>  <Rle> | <factor> <matrix> <numeric>
##        [1]     chr1         1      + |      CHH      0:0       Inf
##        [2]     chr1         2      + |      CHH      0:0         0
##        [3]     chr1         3      + |      CHH      0:0         0
##        [4]     chr1         8      + |      CHH      0:0         4
##        [5]     chr1         9      + |      CHH      0:0         0
##        ...      ...       ...    ... .      ...      ...       ...
##   [199974]     chr1    533554      - |       CG      2:2         0
##   [199975]     chr1    533557      + |      CHH      0:0         2
##   [199976]     chr1    533560      + |       CG      0:0         2
##   [199977]     chr1    533561      - |       CG      0:0         0
##   [199978]     chr1    533565      - |      CHH      0:0         3
##            transitionContext      posteriorMax     posteriorMeth
##                     <factor>         <numeric>         <numeric>
##        [1]              <NA> 0.660644776579142 0.339355223421703
##        [2]           CHH-CHH 0.706837606307567 0.293162393693278
##        [3]           CHH-CHH 0.747153931606252 0.252846068394593
##        [4]           CHH-CHH 0.762497751475263 0.237502248525582
##        [5]           CHH-CHH 0.796351667673236 0.203648332327609
##        ...               ...               ...               ...
##   [199974]             CG-CG 0.999992331296328 0.999992331296328
##   [199975]            CG-CHH 0.502982261503229 0.497017738496771
##   [199976]            CHH-CG 0.529427694969344 0.470572305030656
##   [199977]             CG-CG 0.559538772790003 0.440461227209997
##   [199978]            CG-CHH 0.768933290456705 0.231066709543295
##                 posteriorUnmeth       status        rc.meth.lvl
##                       <numeric>     <factor>          <numeric>
##        [1]    0.660644776579142 Unmethylated  0.116182073397654
##        [2]    0.706837606307567 Unmethylated   0.10047754389566
##        [3]    0.747153931606252 Unmethylated 0.0867708945349779
##        [4]    0.762497751475263 Unmethylated 0.0815543387789686
##        [5]    0.796351667673236 Unmethylated 0.0700447640502754
##        ...                  ...          ...                ...
##   [199974] 7.66870367162537e-06   Methylated  0.823882576092641
##   [199975]    0.502982261503229 Unmethylated  0.169783804097243
##   [199976]    0.529427694969344 Unmethylated  0.390684902355937
##   [199977]    0.559538772790003 Unmethylated  0.366046527873539
##   [199978]    0.768933290456705 Unmethylated 0.0793663993865828
##   -------
##   seqinfo: 1 sequence from an unspecified genome
```

```
# Bisulfite conversion rates can be obtained with
1 - model$params$emissionParams$Unmethylated
```

```
##          prob
## CG  0.9943607
## CHG 0.9979215
## CHH 0.9991911
```

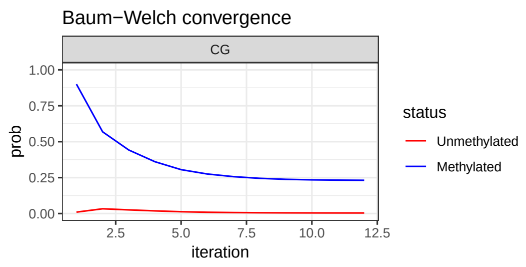
3 Methylation status calling on binned data

The following examples explain the necessary steps for methylation status calling (and imputation) on binned data, such as commonly used 100bp bins. To keep the calculation time short, it uses only the first 200.000 bp of the Arabidopsis genome. The example consists of four steps: 1) Data import, 2) binning and 3) methylation status calling.

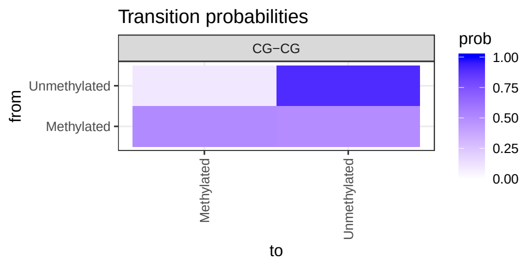
```
library(methimpute)

# ===== 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      - |      CHG      0:4
##          [2]     chr1        80      - |      CHH      2:9
##          [3]     chr1        84      + |      CHH      1:1
##          [4]     chr1        85      + |      CHH      1:1
##          [5]     chr1        86      + |      CHH      1:1
##          ...      ...       ...    ... .      ...      ...
##     [110923]     chr1    533552      - |       CG      2:2
##     [110924]     chr1    533554      - |       CG      2:2
##     [110925]     chr1    533595      + |      CHG      0:1
##     [110926]     chr1    533601      + |      CHG      0:2
##     [110927]     chr1    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'))
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      + |      CHH      0:0
##          [2]     chr1         2      + |      CHH      0:0
##          [3]     chr1         3      + |      CHH      0:0
##          [4]     chr1         8      + |      CHH      0:0
##          [5]     chr1         9      + |      CHH      0:0
##          ...      ...       ...    ... .      ...      ...
##     [199974]     chr1    533554      - |       CG      2:2
##     [199975]     chr1    533557      + |      CHH      0:0
##     [199976]     chr1    533560      + |       CG      0:0
##     [199977]     chr1    533561      - |       CG      0:0
##     [199978]     chr1    533565      - |      CHH      0:0
##   -------
##   seqinfo: 1 sequence from an unspecified genome

# ===== Step 2: Binning into 100bp bins ===== #
binnedMethylome <- binMethylome(methylome, binsize = 100, contexts = c('total','CG'))
print(binnedMethylome$CG)

## GRanges object with 5335 ranges and 3 metadata columns:
##          seqnames        ranges strand | cytosines  context   counts
##             <Rle>     <IRanges>  <Rle> | <integer> <factor> <matrix>
##      [1]     chr1         1-100      * |         0       CG     7:19
##      [2]     chr1       101-200      * |         6       CG    41:62
##      [3]     chr1       201-300      * |         0       CG     3:58
##      [4]     chr1       301-400      * |         2       CG     4:43
##      [5]     chr1       401-500      * |         1       CG     0:19
##      ...      ...           ...    ... .       ...      ...      ...
##   [5331]     chr1 533001-533100      * |         0       CG     0:55
##   [5332]     chr1 533101-533200      * |        14       CG    3:171
##   [5333]     chr1 533201-533300      * |        10       CG     0:16
##   [5334]     chr1 533301-533400      * |         2       CG     0:35
##   [5335]     chr1 533401-533500      * |         8       CG     1:44
##   -------
##   seqinfo: 1 sequence from an unspecified genome

# ===== Step 3: Methylation status calling (and imputation) ===== #

binnedModel <- callMethylation(data = binnedMethylome$CG)

##  Iteration              log(P)             dlog(P)    Time in sec
##          0                -inf                   -              0
##          1       -27231.884280                 inf              0
##          2       -16567.652577        10664.231703              0
##          3       -14617.459829         1950.192748              0
##          4       -13264.059065         1353.400764              0
##          5       -12272.683299          991.375766              0
##          6       -11836.442261          436.241038              0
##          7       -11645.945339          190.496922              0
##          8       -11556.833035           89.112304              0
##          9       -11524.535678           32.297356              0
##         10       -11514.865975            9.669703              0
##         11       -11512.066277            2.799698              0
##         12       -11511.303950            0.762327              0
## HMM: Convergence reached!

print(binnedModel)

## GRanges object with 5335 ranges and 10 metadata columns:
##          seqnames        ranges strand | cytosines  context   counts  distance
##             <Rle>     <IRanges>  <Rle> | <integer> <factor> <matrix> <numeric>
##      [1]     chr1         1-100      * |         0       CG     7:19       Inf
##      [2]     chr1       101-200      * |         6       CG    41:62         0
##      [3]     chr1       201-300      * |         0       CG     3:58         0
##      [4]     chr1       301-400      * |         2       CG     4:43         0
##      [5]     chr1       401-500      * |         1       CG     0:19         0
##      ...      ...           ...    ... .       ...      ...      ...       ...
##   [5331]     chr1 533001-533100      * |         0       CG     0:55         0
##   [5332]     chr1 533101-533200      * |        14       CG    3:171         0
##   [5333]     chr1 533201-533300      * |        10       CG     0:16         0
##   [5334]     chrl 533301-533400      * |         2       CG      0:35          0
##   [5335]     chrl 533401-533500      * |         8       CG      1:44          0
##         transitionContext      posteriorMax        posteriorMeth
##                  <factor>         <numeric>            <numeric>
##      [1]             <NA>  1.00000000000002     1.00000000000002
##      [2]           CG-CG  1.00000000000002     1.00000000000002
##      [3]           CG-CG 0.642214557789536     0.35778544221048
##      [4]           CG-CG 0.974229886049119    0.974229886049119
##      [5]           CG-CG 0.964947193718777    0.035052806281239
##      ...              ...               ...                  ...
##   [5331]           CG-CG 0.999999959766252 4.02337484841841e-08
##   [5332]           CG-CG 0.999999999999999 1.36231328639882e-15
##   [5333]           CG-CG 0.999017508882559 0.000982491117440949
##   [5334]           CG-CG  0.99999277818462 7.22181537982054e-06
##   [5335]           CG-CG 0.999908468202932 9.15317970682573e-05
##               posteriorUnmeth       status          rc.meth.lvl
##                     <numeric>     <factor>            <numeric>
##      [1] 3.27850556847044e-109   Methylated    0.231438411060913
##      [2]   6.6340963973634e-70   Methylated    0.231438411060913
##      [3]     0.642214557789536 Unmethylated   0.0854671890961127
##      [4]    0.0257701139508966   Methylated    0.225581030567185
##      [5]     0.964947193718777 Unmethylated   0.0121121449553183
##      ...                   ...          ...                  ...
##   [5331]     0.999999959766252 Unmethylated 0.00414487758160785
##   [5332]     0.999999999999999 Unmethylated 0.00414486843673694
##   [5333]     0.999017508882559 Unmethylated 0.00436818232341656
##   [5334]      0.99999277818462 Unmethylated 0.00414650990873848
##   [5335]     0.999908468202932 Unmethylated 0.00416567302315503
##   -------
##   seqinfo: 1 sequence from an unspecified genome
```

You can also check several properties of the fitted Hidden Markov Model, such as convergence or transition probabilities, and check how well the fitted distributions describe the data. This last point is important because the binomial distributions that the HMM uses were originally meant to describe individual cytosines and not bins. However, we have observed that they still capture the bimodal distributions of methylation levels in binned data quite well. Note that the histogram for our example looks quite sparse due to the very low number of bins that were used.

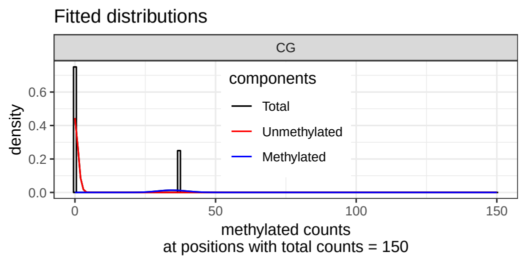
```
plotConvergence(binnedModel)
```



```
plotTransitionProbs(binnedModel)
```



```r
plotHistogram(binnedModel, total.counts = 150)
```



4 Description of columns in the output

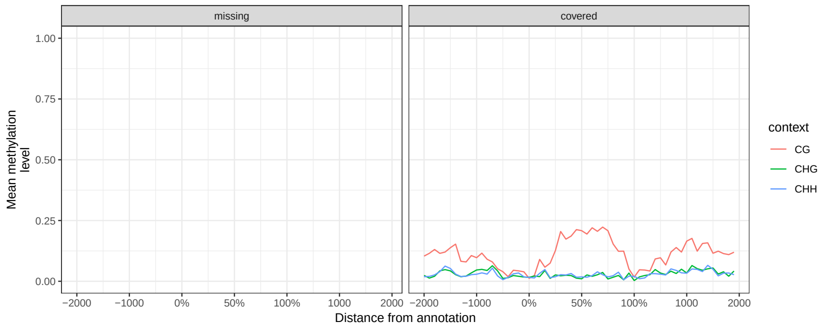
- **context** The sequence context of the cytosine.
- **counts** Counts for methylated and total number of reads at each position.
- **distance** The distance in base-pairs from the previous to the current cytosine.
- **transitionContext** Transition context in the form "previous-current".
- **posteriorMax** 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).

5 Plots and enrichment analysis

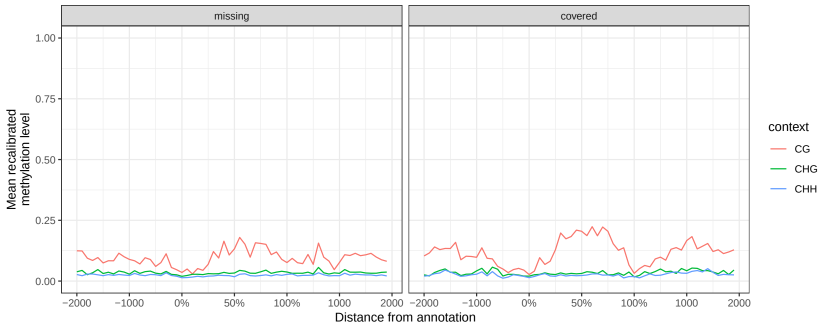
This package also offers plotting functions for a simple enrichment analysis. Let's say we are interested in the methylation level around genes and transposable elements. We would also like to see how the imputation works on cytosines with missing data compared to covered cytosines.

```r
# 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'
```

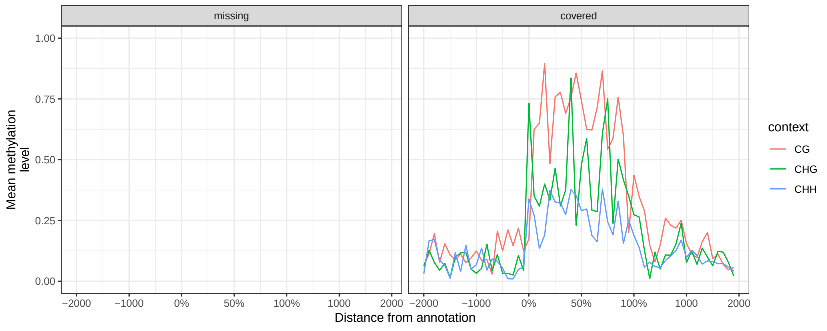
```
# Note that the plots look a bit ugly because our toy data has only 200.000 datapoints.
data(arabidopsis_genes)
seqlengths(model$data) <- seqlengths(arabidopsis_genes)[seqlevels(model$data)] # this
                          # line should only be necessary for our toy example
plotEnrichment(model$data, annotation=arabidopsis_genes, range = 2000,
               category.column = 'category')

## $meth.lvl

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



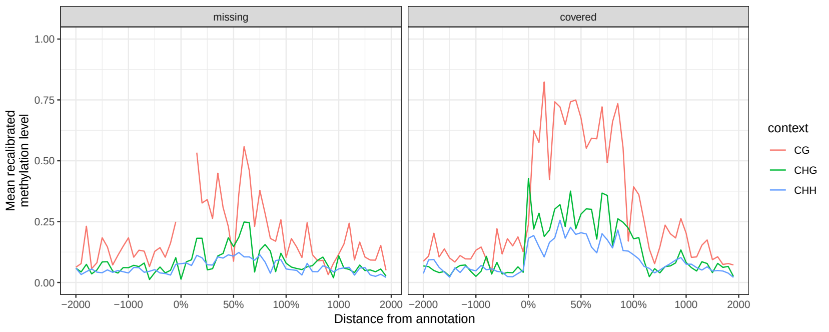
```
##
## $rc.meth.lvl
```



```
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
```



6 Export results

You can export the results as TSV file with the following columns:

- chromosome, position, strand, context, counts.methylated, counts.total, posteriorMax, posteriorMeth, posteriorUnmeth, status, rc.meth.lvl

```
exportMethylome(model, filename = tempfile())
```

Please see section 4 for a description of the columns.

7 Session Info

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

- R version 3.6.0 (2019-04-26), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 18.04.2 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.9-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.9-bioc/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: BiocGenerics 0.30.0, GenomeInfoDb 1.20.0, GenomicRanges 1.36.0, IRanges 2.18.0, S4Vectors 0.22.0, ggplot2 3.1.1, methimpute 1.6.0
- Loaded via a namespace (and not attached): BiocManager 1.30.4, BiocStyle 2.12.0, Biostrings 2.52.0, GenomeInfoDbData 1.2.1, R.methodsS3 1.7.1, R.oo 1.22.0, R.utils 2.8.0, R6 2.4.0, RCurl 1.95-4.12, Rcpp 1.0.1, XVector 0.24.0, assertthat 0.2.1, bitops 1.0-6, colorspace 1.4-1, compiler 3.6.0, crayon 1.3.4, data.table 1.12.2, digest 0.6.18, dplyr 0.8.0.1, evaluate 0.13, glue 1.3.1, grid 3.6.0, gtable 0.3.0, highr 0.8, htmltools 0.3.6, knitr 1.22, labeling 0.3, lazyeval 0.2.2, magrittr 1.5, minpack.lm 1.2-1, munsell 0.5.0, pillar 1.3.1, pkgconfig 2.0.2, plyr 1.8.4, purrr 0.3.2, reshape2 1.4.3, rlang 0.3.4, rmarkdown 1.12, scales 1.0.0, stringi 1.4.3, stringr 1.4.0, tibble 2.1.1, tidyselect 0.2.5, tools 3.6.0, withr 2.1.2, xfun 0.6, yaml 2.2.0, zlibbioc 1.30.0