Supplemental Material

metilene: Fast and sensitive detection of differentially methylated regions from bisulfite sequencing data

Frank Jühling[†,1,2], Helene Kretzmer[†,1,2], Stephan H. Bernhart[1,2], Christian Otto[1,2], Peter F. Stadler[2−6] & Steve Hoffmann[*1,2]

Contents

1 Supplemental Information

1.1 Simulation of Artificial Data

For benchmarking purposes we simulated DMRs on the human chromosome 10 (hg19). To do this we used the chromatin segmentations of the GM12878 cell line. The scripts for simulating the DMRs from the scratch are given in an additional Supplemental File. This file also contains the chromatin annotations used here.

In brief, we simulated the background and methylation rates for 20 input WGBS input files using the beta distributions given in (Tab. 11) calling the script

```
>Rscript simulate_background.R α β chromatin_annotation_chr10.txt
```

where the parameter α and β refer to the shapes of the beta distribution and the txt file annotates promotor and non-promotor CpGs. To simulate the DMRs inside these backgrounds we used the same beta distributions (cf. Tab. 11). To obtain four different classes of DMRs the mixture factors in (Tab.12) were used.

```
>Rscript simulate_DMRs_WGBS.R α β c <path-to-background-files> <outputpath>
```

where c is the mixture factor. The script will write the input files for `BSmooth` and `metilene` to `<outputpath>`. We provide an additional script to convert the `BSmooth` input to `MOABS` input.

For the RRBS data simulation we extract the RRBS regions using `bedtools intersect` and finally call

```
>Rscript simulate_DMRs_RRBS.R α β c <path-to-background-files> <outputpath>
```

to generate the DMRs.

1.2 Tool parameter

The calls to the benchmarked tools are given below. In case of the `R`-tools `BiSeq` and `BSmooth` the parameters of the used functions are stated instead.

`MOABS`

For `MOABS` (v1.2.9) we used the following calls:

```
>mcomp -p <threads> -r <list-of-files-groupA> -r <list-of-files-groupB> -m <mergedratios-groupA>
-m <mergedratios-groupB> -c <compfile> -maxDistConsDmcs 300 > <outfile>
```

`metilene`

For `metilene` (v0.2-4) we used:

```
>metilene -maxdist 300 -t <threads> -a <prefix_groupA> -b <prefix_groupB> <inputfile> >
<outfile>
```

`BSmooth`

We loaded `BSmooth` (v.1.0.0) input into `R` using `read.lister` function and combined data using the `combine` function.
`BSmooth.tstat`: estimate.var="same", local.correct=T (for RRBS data local.correct=F)
`dmrFinder`: cutoff=NULL, qcutoff=c(0.025, 0.975), maxGap=300, stat="tstat.corrected"

`BiSeq`

For `BiSeq` (v1.2.5), we loaded `BSmooth` input into `R` using `read.table` function and combined data using `BSraw` and `GRanges`.
`clusterSites`: perc.samples=1, min.sites=10, max.dist=300
`limitCov`: maxCov used 90%-quantile
`betaRegression`: link="probit", type="BR"
`smoothVariogram`: sill=0.9
`testClusters`: FDR.cluster=0.1
`trimClusters`: FDR.loc=0.05
`findDMRs`: max.dist=300, diff.dir=TRUE

2 Supplemental Tables

2.1 TPR, PPV Artificial Data – WGBS

		TPR			PPV		
background	DMR class	metilene	BSmooth	MOABS	metilene	BSmooth	MOABS
1	1	0.999	0.485	0.999	1	0.356	0.953
1	2	0.999	0.472	0.999	1	0.369	0.953
1	3	0.999	0.414	0.970	1	0.419	0.984
1	4	0.998	0.090	NA	0.999	0.651	NA
2	1	0.999	0.446	0.985	1	0.386	0.969
2	2	0.999	0.397	0.963	1	0.428	0.989
2	3	0.999	0.250	0.380	0.999	0.522	1
2	4	0.526	0.011	NA	0.989	0.702	NA

Table 1: TPR and PPV values based on the CpG-wise comparisons of predicted and simulated DMRs in the human chromosome 10 of a WGBS data set.

		TPR			PPV		
background	DMR class	metilene	BSmooth	MOABS	metilene	BSmooth	MOABS
1	1	1	0.116	0.995	1	0.231	0.998
1	2	1	0.112	0.995	1	0.233	0.998
1	3	1	0.116	0.996	1	0.289	1
1	4	1	0.081	NA	1	1	NA
2	1	1	0.112	0.997	1	0.259	1
2	2	1	0.114	0.988	1	0.305	1
2	3	1	0.127	0.375	1	0.587	1
2	4	0.527	0.010	NA	1	0.500	NA

Table 2: TPR and PPV values based on the segment-wise comparisons of predicted and simulated DMRs in the human chromosome 10 of a WGBS data set.

2.2 TPR, PPV Artificial Data – RRBS

back-	DMR	TPR				PPV			
ground	class	metilene	BSmooth	MOABS	BiSeq	metilene	BSmooth	MOABS	BiSeq
1	1	0.993			1	0.998			0.696
1	2	0.993			1	0.998			0.700
1	3	0.993			1	0.999			0.714
1	4	0.992			0.954	0.999			0.768
2	1	0.998			0.997	0.998			0.747
2	2	0.999			0.995	0.999			0.762
2	3	0.998			0.980	0.999			0.810
2	4	0.533			0.274	0.991			0.933

Table 3: TPR and PPV values based on the CpG-wise comparisons of predicted and simulated DMRs in the human chromosome 10 of a RRBS data set.

back-	DMR	TPR				PPV			
ground	class	metilene	BSmooth	MOABS	BiSeq	metilene	BSmooth	MOABS	BiSeq
1	1	1			0.949	1			0.960
1	2	1			0.955	1			0.965
1	3	1			0.955	1			0.965
1	4	1			1	1			0.945
2	1	1			0.952	1			0.975
2	2	1			0.957	1			0.980
2	3	1			0.978	1			0.975
2	4	1			0.987	0.530			0.280

Table 4: TPR and PPV values based on the segment-wise comparisons of predicted and simulated DMRs in the human chromosome 10 of a RRBS data set.

2.3 Runtime and Memory

	cores	metilene	MOABS	BSmooth	speedup
real time	1	0h4m7s	65h35m11s	2h20m3s	34x–956x
	10	0h1m14s	9h11m51s	0h23m19s	19x–447x
RAM	1	0.7 GB	5.4 GB	10.7 GB	
	10	1.2 GB	6.8 GB	90.2 GB	

Table 5: **WGBS** running time and memory requirements for `metilene`, `MOABS`, and `BSmooth` for calling DMRs on the human chromosome 10 (hg19) with 10 vs. 10 **simulated** samples. In the simulations a total of 2.7M CpG positions was evaluated.

	cores	metilene	MOABS	BSmooth	BiSeq	speedup
real time	1	4s	SF*	2m20s	8h21m35s	35x–7.524x
	10	2s	20m27s	0m52s	8h19m18s	26x–14.979x
RAM	1	0.08 GB	SF*	1.12 GB	1.42 GB	
	10	0.75 GB	7.54 GB	7.31 GB	1.42 GB	

Table 6: **RRBS** running time and memory requirements for `metilene`, `MOABS`, and `BSmooth` for calling DMRs on the human chromosome 10 (hg19) with 10 vs. 10 **simulated** samples. In the simulations a total of 57,8k CpG positions was evaluated. SF*: `MOABS` did not finish any of several test runs (segmentation faults) on one core while we observed no problems for the same input data when running on more than one core. E.g., the running time of `MOABS`on two cores was 73m35s with 7.03 GB RAM.

	metilene	MOABS	BSmooth	speedup
chromosome 10:				
real time	0h0m52s	5h29m35s	0h17m25s	20x–380x
RAM	0.02 GB	6.8 GB	73.3 GB	
whole genome:				
real time	0h9m55s	NA	NA	
RAM	0.09 GB	NA	NA	

Table 7: **WGBS** running time and memory requirements for `metilene`, `MOABS`, and `BSmooth`, each running on **10 cores**, for calling DMRs on the human chromosome 10 and the whole human genome (hg19) with 8 vs. 12 **real** samples. Due to missing values this data set is not directly comparable to the simulations. For chromosome 10 a total of 1.1M CpG positions was evaluated for all samples.

	samples	`metilene`	`MOABS`	`BSmooth`	speedup
real time	2 vs. 2	04m21s	1d01h54m32s	2h01m26s	28x–357x
	4 vs. 4	05m18s	3d04h30m28s	2h24m10s	27x–866x
	8 vs. 8	08m21s	4d10h47m05s	18d18h29m33s	726x–3,065x
	16 vs. 16	14m12s	NA	NA	NA
	50 vs. 50	50m15s	NA	NA	NA
RAM	2 vs. 2	0.12 GB	17.85 GB	67.99 GB	
	4 vs. 4	0.09 GB	17.85 GB	176.34 GB	
	8 vs. 8	0.08 GB	17.85 GB	300.00 GB	
	16 vs. 16	0.12 GB	NA	NA	
	50 vs. 50	0.08 GB	NA	NA	

Table 8: **WGBS** running time and memory requirements for `metilene`, `MOABS`, and `BSmooth`, each running on **10 cores**, for calling DMRs on the human genome (hg19) with different sample sizes, i.e., 2 vs. 2, 4 vs. 4, 8 vs. 8, 16 vs. 16, and 50 vs. 50 **real** samples. All "NA" entries were not evaluated due to run time/memory issues. Test input data sets with more than 8 vs. 8 samples contained duplicates.

		WGBS		RRBS	
	t_{dist}	real time	memory	real time	memory
	50	1m12s	0.08 GB	3s	0.08 GB
	100	1m35s	0.21 GB	3s	0.08 GB
	250	3m06s	0.27 GB	3s	0.08 GB
(default setting)	300	4m07s	0.66 GB	4s	0.08 GB
	500	9m49s	2.18 GB	4s	0.08 GB
	750	25m55s	4.30 GB	4s	0.08 GB
	1.000	1h10m13s	31.92 GB	4s	0.08 GB

Table 9: Running time and memory requirements for `metilene` with different t_{dist} settings.

		WGBS		RRBS	
	t_{min}	real time	memory	real time	memory
	3	6m28s	0.74 GB	6s	0.08 GB
	5	4m11s	0.73 GB	5s	0.08 GB
	7	3m49s	0.73 GB	4s	0.08 GB
(default setting)	10	3m55s	0.73 GB	4s	0.08 GB
	15	6m21s	0.73 GB	3s	0.08 GB
	25	14m41s	0.73 GB	3s	0.08 GB
	50	13m39s	0.73 GB	3s	0.08 GB
	100	28m14s	0.73 GB	2s	0.08 GB
	150	24m13s	0.73 GB	2s*	0.08 GB*
	200	21m41s	0.73 GB	2s*	0.08 GB*

Table 10: Running time and memory requirements for `metilene` with different t_{min} settings. No DMRs were found anymore for settings flagged with $*$.

2.4 Simulation parameter

	non-promoter		promoter	
	α	β	α	β
background 1	40	3	3	40
background 2	15	5	5	15

Table 11: Parameters for the beta distributions to simulate background methylation rates.

DMR class	Mixture factors c
1	1
2	0.87
3	0.73
4	0.60

Table 12: Mixture factors of random variables sampled from beta distributions for the simulation.

3 Supplemental Figures

3.1 Distribution of Background and DMR Methylation

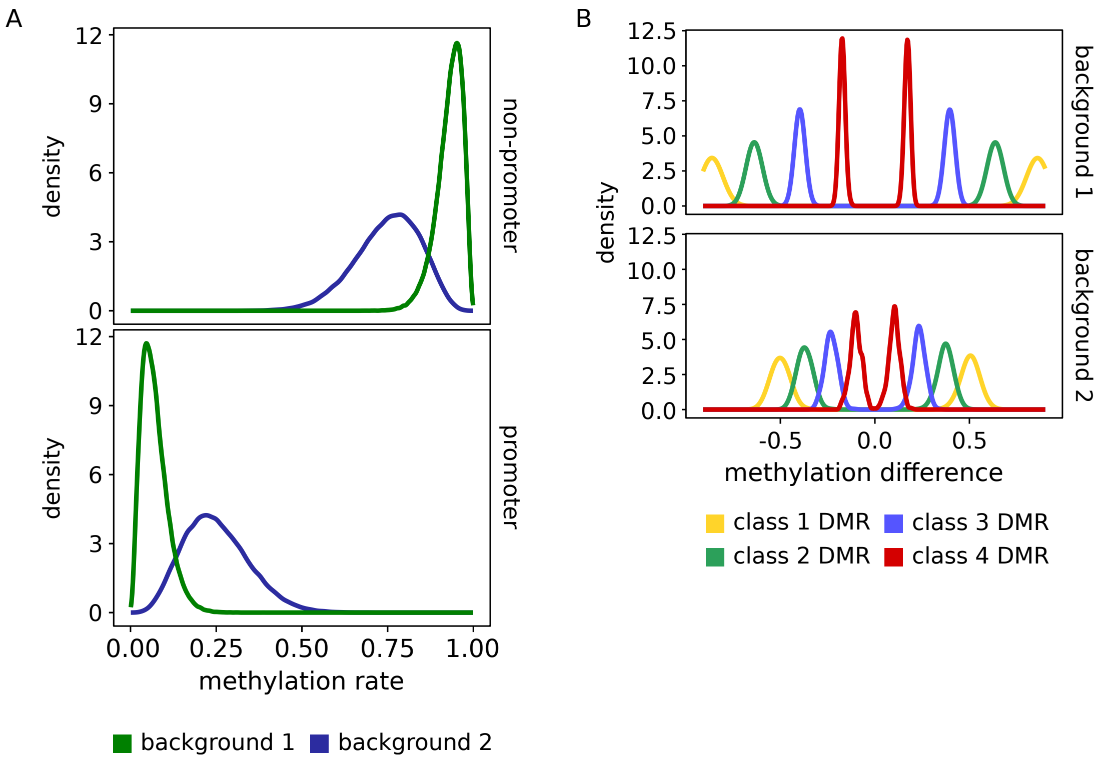

Figure 1: Distributions of methylation rates for backgrounds and DMRs. A) Two different background distributions were used to simulate non-promoter (top) and promoter (bottom) regions. B) The distributions of mean methylation differences in DMR regions for the combination of the two simulated backgrounds with four different mixture ratios. This allows to simulate a comprehensive grading set of DMRs between easily (class 1 DMRs – background 1, top – yellow) and difficultly (class 4 DMR on background 2, bottom – red) distinguishable.

3.2 Boundary Detection – WGBS

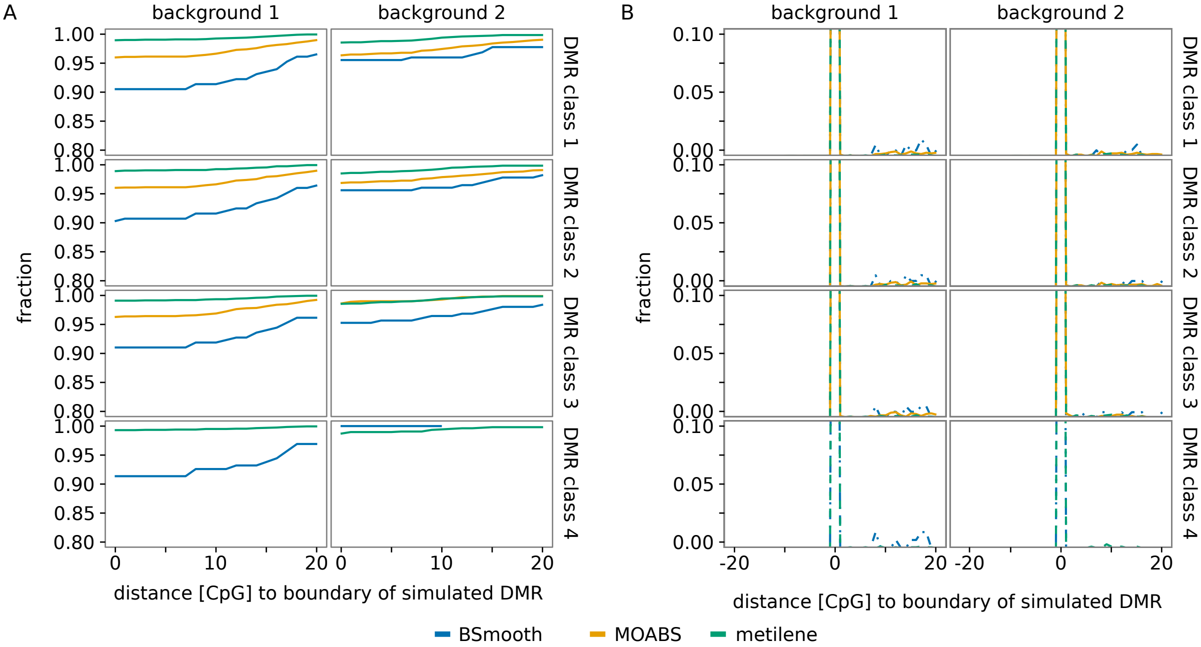

Figure 2: **WGBS** boundary detection analyses for background 1+2 and DMRs of class 1-4. `MOABS` did not predict any class 4 DMR and is therefore missing in the corresponding figures. The fraction of predicted DMR boundaries of `metilene`, `MOABS`, and `BSmooth` within different maximum absolute distances, ranging from 0 (no difference between simulated and predicted boundary) to 20 CpGs. B) The fraction of distances (in CpGs) between predicted and simulated boundaries for the three tools. Negative distances indicate that the predictions were too short compared to the simulated ones while positive values indicate predictions extending beyond the simulated DMRs.

3.3 Noisy Data – WGBS

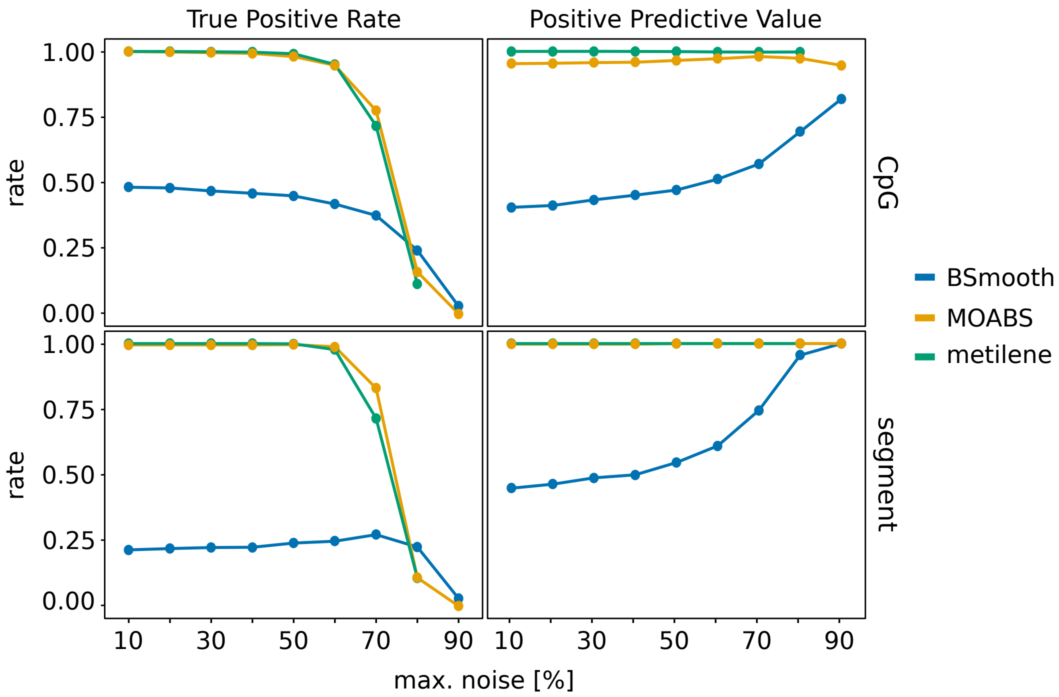

Figure 3: Simulations with different percentages of noise introduced into **WGBS** DMR regions. TPRs and PPVs on the CpG level (top) and the DMR level (bottom) were measured. metilene and MOABS showed a very stable detection of DMRs also with high levels of noise. For DMRs with almost $^2/_3$ noise and only $^1/_3$ signal both tools miss DMRs. BSmooth reports less than $^1/_3$ DMRs in general.

3.4 TPR, PPV Artificial Data – RRBS

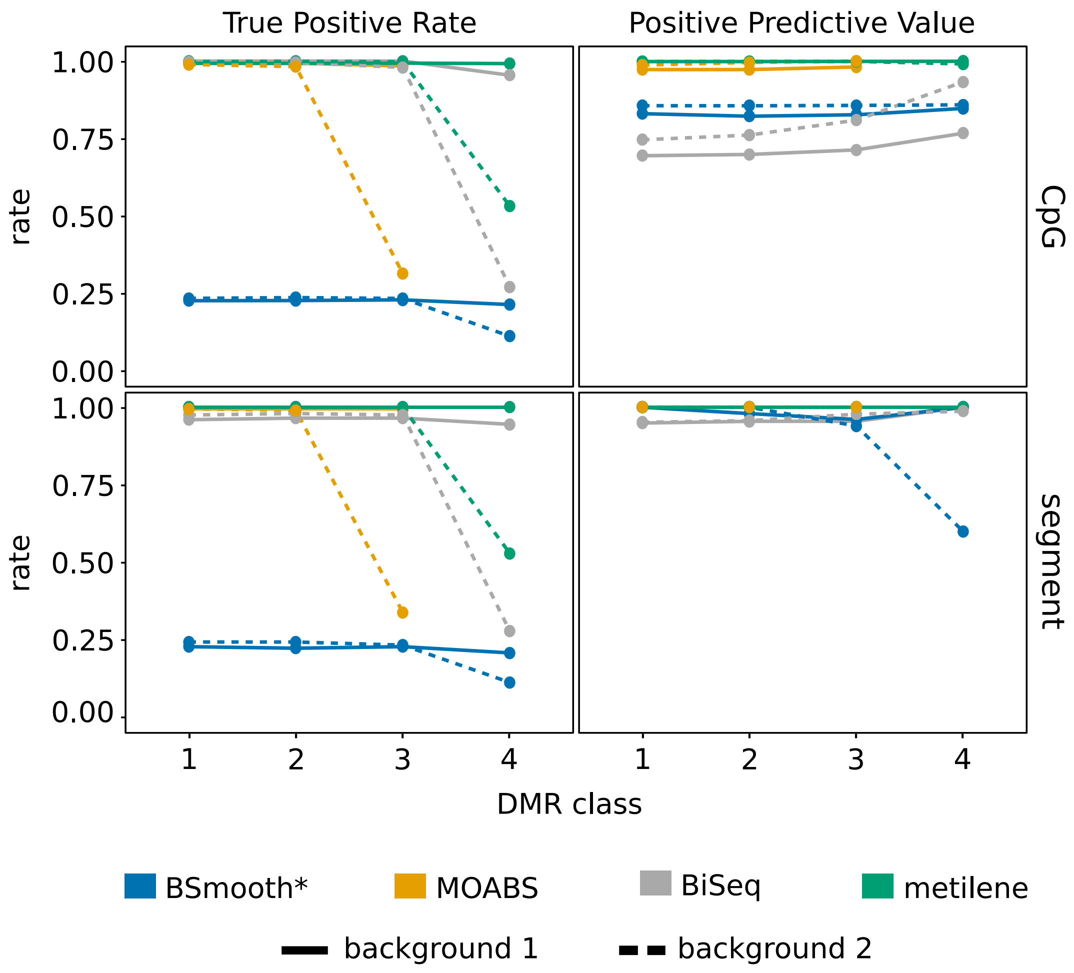

Figure 4: The performance of `metilene`, `MOABS`, `BSmooth`, and `BiSeq` in terms of true positive rates and positive predictive values (PPVs) for different classes of DMRs on the RRBS simulations.

3.5 Boundary Detection – RRBS

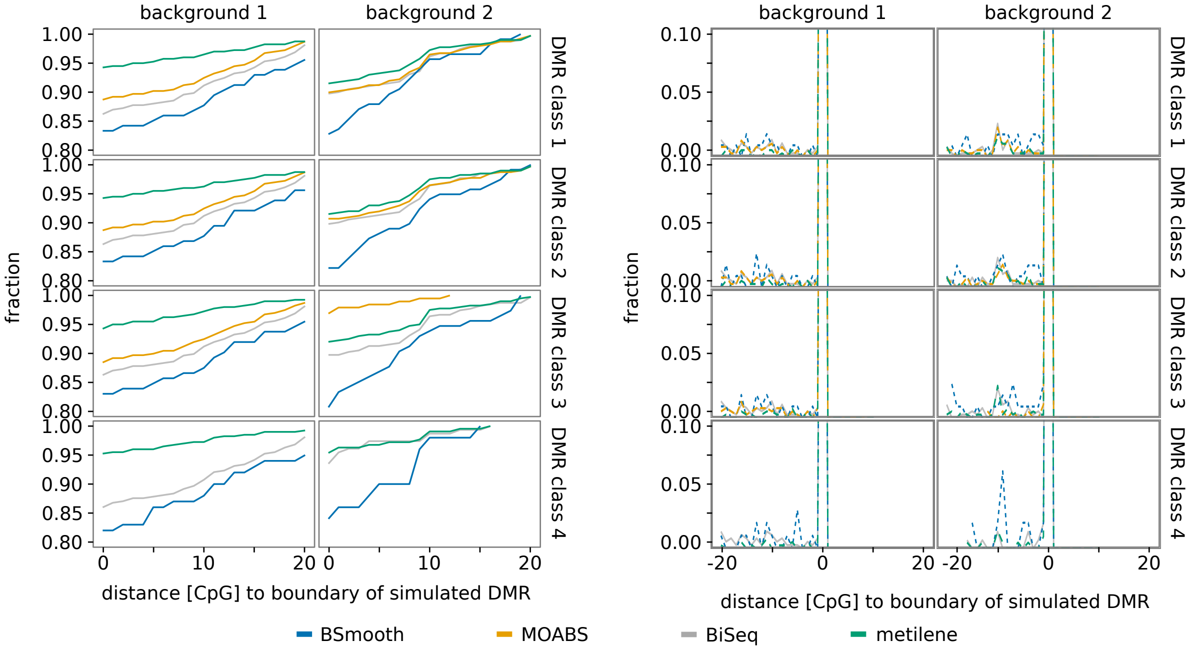

Figure 5: **RRBS** boundary detection analyses for background 1+2 and DMRs of class 1-4. MOABS did not predict any class 4 DMR and is therefore missing in the corresponding figures. The fraction of predicted DMR boundaries of metilene, MOABS, BSmooth, and BiSeq within different maximum absolute distances, ranging from 0 (no difference between simulated and predicted boundary) to 20 CpGs. B) The fraction of distances (in CpGs) between predicted and simulated boundaries for the three tools. Negative distances indicate that the predictions were too short compared to the simulated ones while positive values indicate predictions extending beyond the simulated DMRs.

3.6 Low Number of Samples

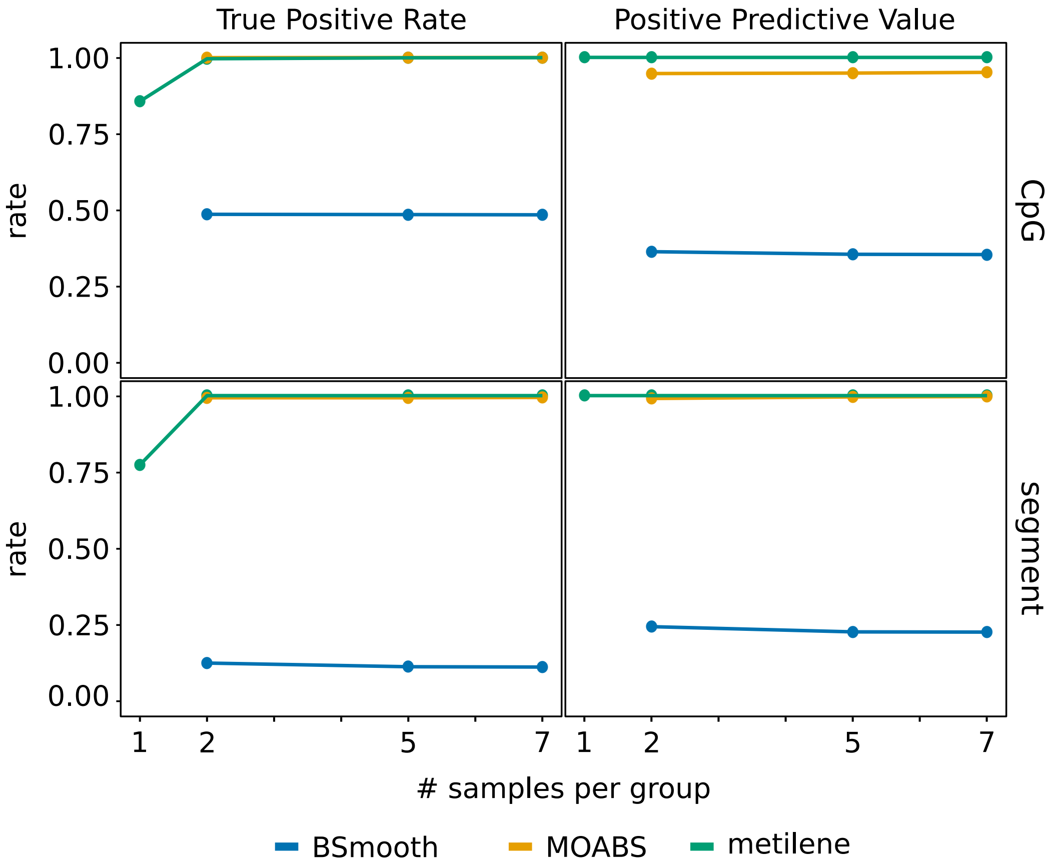

Figure 6: **WGBS** simulations with low number of samples. TPRs and PPVs on the CpG level (top) and the DMR level (bottom) were measured while comparing groups consisting of only 1, 3, 5 or 7 samples. Only `metilene` is able to compare 1 vs. 1 sample while both other tools need at least 2 samples within each group.

3.7 Missing Data

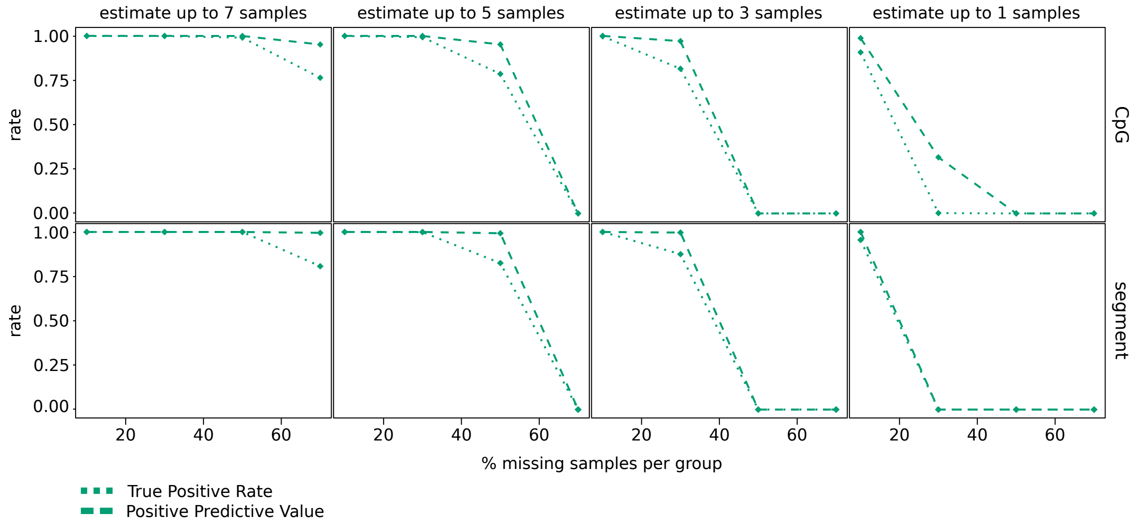

Figure 7: **WGBS** simulations with different levels of missing data as well as different amounts of estimated samples. The data set consisted of 10 vs. 10 samples while a certain amount of values was removed from the data, and different numbers of samples (7, 5, 3, and 1) per CpG position were allowed to be estimated by `metilene` using a beta distribution estimated from the existing methylation rates. TPRs and PPVs were measured on the CpG level (top) and the DMR level (bottom).

3.8 The algorithm implemented in metilene for *de-novo* DMR prediciton as pseudocode

Algorithm 1 metilene

```
diff=mean(group1)-mean(group2)
procedure Sub-regions(distCpGs)
    for all CpGs x,y do
        if dist(x,y) > $t_{dist}$ then
            subregion1 = [.,x]
            subregion2 = [y,.]
        end if
    end for
end procedure
Phase 1 - Segment each sub-region [s,t]
for all $s \leq a < b \leq t$ do
    Calculate $Z_{s,t}(a,b)$
end for
$Z_{max}(a,b) = max_{s \leq a < b \leq t}|Z_{s,t}(a,b)|$
Define pre-segments as [s,a), [a,b], (b,t]
Phase 2 - Filter pre-segments
for all pre-segments do
    if #CpGs $\leq$ minCpGs then
        Do 2D KS-test and calculate $p_{new}$
        Label as potential DMR
    else if low variation filter passed then
        if majority filter passed then
            Do 2D KS-test and calculate $p_{new}$
            if exists($p_{[s,t]}$) AND $p_{new} > p_{[s,t]}$ then
                Label as potential DMR
            else
                Goto Phase 1
            end if
        else
            Goto Phase 1
        end if
    else
        Goto Phase 1
    end if
end for
Phase 3 - Call DMRs
DMR = $argmin_{potentialDMRs}(p\_value)$
Goto Phase 1 for [s,start$_{DMR}$), (end$_{DMR}$,t]
Phase 4 - Output DMRs
Merge all regions without p-value
for all regions labeled as potential DMR do
    if diff $\geq diff_{min}$ then
        Do Mann-Whitney U test
    end if
end for
Output all segments
```

Figure 8: The algorithm implemented in metilene for *de-novo* DMR predicition as pseudocode.