PycoMeth: A toolbox for differential methylation testing from Nanopore methylation calls

Rene Snajder[1,2,4,*], Oliver Stegle[1,3,*], Marc Jan Bonder[1]

1 Division of Computational Genomics and Systems Genetics, German Cancer Research Center (DKFZ), Heidelberg, Germany
2 Faculty for Biosciences, Heidelberg University, Heidelberg, Germany
3 Genome Biology Unit, European Molecular Biology Laboratory (EMBL), Heidelberg, Germany
4 HIDSS4Health, Helmholtz Information and Data Science School for Health, Heidelberg, Germany

* Corresponding authors: r.snajder@dkfz-heidelberg.de, o.stegle@dkfz-heidelberg.de

ABSTRACT

Advances in Nanopore sequencing have opened up the possibility for the simultaneous analysis of genomic and epigenetic variation by way of base-calling and methylation-calling of the same long reads. Methylation analysis based on long read technologies requires a re-evaluation of data storage and analysis approaches previously developed for either CpG-methylation arrays or short-read bisulfite sequencing data. To address this, we here present a toolbox for the segmentation and differential methylation analysis of (haplotyped) methylation calls from Nanopore data. Additionally, we describe a storage format for read-level and reference-anchored methylation call data, which simultaneously allows for efficient storage and rapid data access.

Keywords nanopore · methylation · meth5 · pycometh

Background

The analysis of DNA base modifications, such as cytosine methylation, has become a crucial component of understanding epigenetic transcriptional regulation. In mammalian cells, the predominant and most well studied type of base modification is the methylation of cytosine in the 5'CpG3' context (often abbreviated 5mC or simply CpG-methylation). Genomic regions enriched with this CpG motif (often referred to as CpG-islands, CGI) are found to be less tightly associated with nucleosomes, hence more accessible to DNA-binding proteins such as transcription factors [1]. Methylation of CpG in regulatory regions can then influence gene expression in a variety of ways. Primarily, CpG-methylation is associated with the silencing of related genes, either through direct interference with transcription factor binding or via recruitment of binding proteins attracted to methylated CpG [1, 2]. Other, arguably less well studied, types of DNA base modifications include the methylation of adenine in 5'GATC3' (5mA) context or any of the oxidative derivatives of 5mC (5hmC, 5fC, and 5caC) [3].

There exists a growing repertoire of high-throughput assays for the profiling of CpG-methylation states. Among the most popular methods are bead array based techniques, which cover a specific set of CpG sites on the human genome and allow for rapid quantitative evaluation of methylation rates in these regions. For other organisms and for a genome-wide evaluation of cytosine methylation, whole genome bisulfte sequencing (WGBS) has long been considered a gold standard, offering single-base resolution and genome-wide coverage[4, 5], and more recently, enzymatic methylation sequencing has gained popularity as an alternative to bisulfite conversion, promising lower DNA degradation and more balanced base representation [6]. Both methods are based on short-read sequencing techniques and can also be used in a single-cell setting.

Demand for longer reads have given rise to long-read sequencing techniques such as the sequencing technology by Pacific Biosciences (PacBio) as well as Oxford Nanopore Technologies (ONT), which can directly sequence native DNA and RNA molecules. Sequencing long single molecules can aide in the reliable detection of structural variations [7], phasing of variants into paternal and maternal haplotype [8], as well as the assembly of an entire human genome including low-complexity regions, that otherwise remain difficult to resolve with short-read sequencing [9]. Additionally, the same ONT sequencing datasets can be reprocessed to obtain measurements of modification states of sequenced bases [10]. This allowed for a host of applications for profiling the epigenome in a haplotype resolved whole-genome single-molecule setting [11], which requires software tools for the inference of methylation state as well as interpretation of methylation rates. A number of methylation callers have already been published, including Nanopolish [11], DeepSignal [12] and ONT's own Megalodon [13]. These methods have been compared and benchmarked elsewhere [14].

In this work we focus on the data management, the interpretation and the downstream analysis of ONT methylation calls. To facilitate these tasks, we propose a software toolbox for the storage, segmentation, and differential methylation calling between samples and/or haplotypes. Specifically, we present **MetH5**, an HDF5-based container format for efficient storage of read-level reference-anchored methylation calls, and **PycoMeth**, a software suite for sequence based, and methylation based segmentation as well as differential methylation testing and reporting.

Results

MetH5 - read-level base modification container

A single PromethION flow-cell can produce up to 290 billion basepairs of sequencing data [15]. Given that nearly 1% of the human genome consists of CpG sites [16], this would result in up to 2.7 billion CpG-methylation calls per flow-cell. As each of the reads of the flow cell represent different molecules it is crucial that storage and analysis of these base modification calls can be performed efficiently without compromising these unique advantages provided by ONT sequencing. The various DNA base modification callers for ONT data implement different output formats. Nanopolish and DeepSignal output unsorted tab-separated text files, with one line per read-mapping. HTSLib [17] has recently implemented two new tags (MM and ML) which store base modifications together with the read-alignments in the SAM format. Support for these tags, however, is currently still limited and subsetting base modification calls to a specific region requires cross-referencing the MM tag with the reference sequence, since base modifications are stored without explicit genomic coordinates.

We therefore define a file format specifically designed to allow for efficient downstream analysis, while keeping the ONT specific information. Implemented as an Hierarchical Data Format (HDF) version 5[18] container, we denote the format MetH5 (**Figure 1**). We also implement a python API in the package meth5 in order to abstract away the architecture and provide a developer-friendly interface. In designing the MetH5 format, we consider the following design principles. *Read-level storage*: all base modification calls are stored together with the read they originated from, in order to allow read-level and read-group-level analyses. *Minimal precision loss:* instead of binary calls, we can store the actual confidence values output by the base modification caller, in the current implementation log-likelihood ratios from Nanopolish. *Rapid random access*: base modification calls are stored in order of their genomic coordinate and indexed such that they can be retrieved with minimum disk IO. *Designed for parallel processing*: Chunked storage and accessor methods allow for even load distribution when used in parallel systems, even when copy number differs and coordinate-based parallel processing would result in uneven load distribution if processed on genomic windows. *Efficient use of storage*: using the correct data-types, data compression, and avoiding data duplication (such as read names or chromosome names). *Flexible annotations:* reads can be annotated with arbitrary read-group qualifiers (e.g. sample, haplotype group, haplotype id).

In addition to a python API, the package meth5 also implements a command line user interface (CLI) with the following sub-commands: `create_m5` reads one or more Nanopolish output files to create a single MetH5 container, `merge_m5` reads multiple MetH5 containers and merges them into a single one while retaining read-group annotations, `annotate_reads` stores read-group annotation from a tab-delimited file in an existing MetH5 container. MetH5 offers favorable random and sequential access times compared to previous solutions (**Figure 1D**).

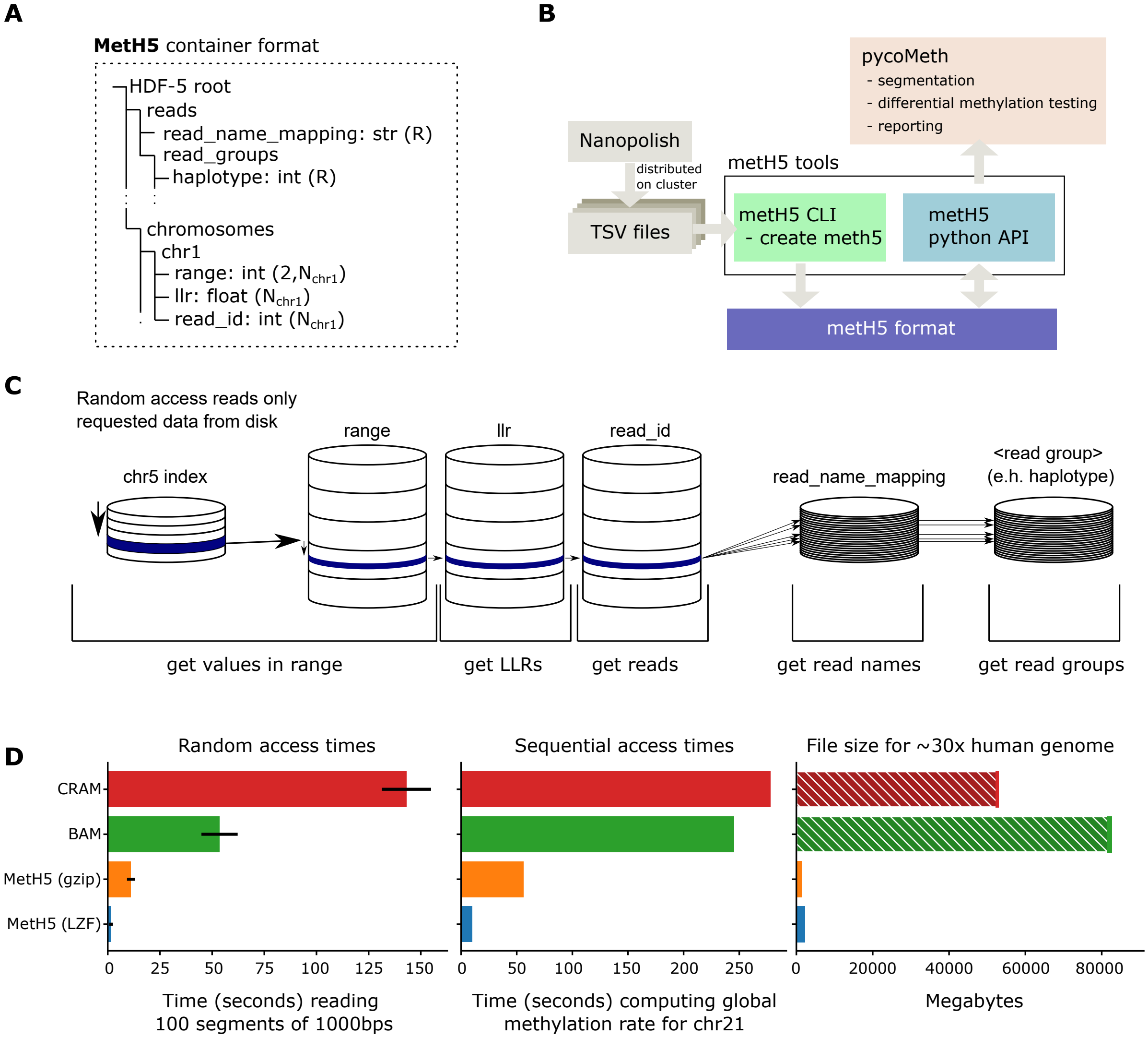


Figure 1: MetH5 file format. **A)** Structure of the HDF5 container including dataset types and shapes. N_x refers to the number of methylation calls per chromosome x. R refers to the total number of reads in the entire container. Methylation calls are stored together with their genomic coordinate on the chromosome (range), the log-likelihood ratio (LLR) of methylation, and a numeric read-id unique to this container. Read names are stored optionally, mapping each read-id to their globally unique read name. An arbitrary number of read groupings can be stored, assigning each read to exactly one read group per grouping. **B)** Process of converting Nanopolish result files from multiple Nanopolish runs (e.g. runs parallelized on a compute cluster) to a single MetH5 file which can then be accessed using the meth5 API by other tools such as PycoMeth. **C)** Schematic representation of random access in the MetH5 format. An index per chromosome allows direct access to the required chunk. The range data-set can then be searched for the start and end-index. Once these indices have been acquired, LLRs and read-ids can be read directly and optionally. If globally unique read-names are required, they can be looked up directly using the read-id, equally with read groups such as haplotype assignments. **D)** Performance comparison between MetH5 and BAM/CRAM format with MM tag (**Methods**). In the file-size comparison, the hatched area is the size of the regular BAM/CRAM file without modification scores.

Bayesian methylome segmentation

As part of the PycoMeth suite we provide PycoMeth Meth_Seg, a Bayesian changepoint detection algorithm (**Figure 2A**) designed for the segmentation of CpG-methylation profiles. Briefly, our approach is based on modeling methylation states as Bernoulli trials, with the respective methylation rate parameter depending on both the read-group as well as the segment. A hidden Markov model (HMM) then models methylation calls as uncertain observations of these methylation states. The segmentation is then optimized to determine the segmentation that maximizes the likelihood of observations. Default hyperparameters are optimized to maximize sensitivity such that typically an oversegmentation is achieved. In a second pass model parameters are then compared between neighboring segments, whereby segments with concordant estimates of methylation rate parameter in all read-groups are merged (default threshold is a minimum of 0.25 methylation rate difference in one read-group).

Note that unlike segmentations of differential methylation profiles, this segmentation is unbiased towards change-points which will result in differentially methylated sites. Furthermore, it allows consideration of more than two read-groups, such as when creating a joint segmentation of $N > 2$ samples or when considering haplotypes in a multiple sample analysis.

Segmentation can be performed either via a python API, or using a CLI which takes one or more MetH5 files as the input. The CLI also supports chunked operations, taking advantage of the chunked data storage, in order to allow efficient load distribution on parallel systems.

Sequence based segmentation

Next to the CpG-methylation based segmentation, PycoMeth Meth_Seg also implements a sequence-based segmentation method in its CGI-Finder algorithm. This is aimed towards finding the CpG-rich regions known as CGIs. Segments of at least 200 basepairs length with a minimum of 0.5 CG content are classified as CGIs if an enrichment of CpGs (compared to the expected CpG distribution given the observed CG content) is found. The parameters for this test (minimum length, minimum CG content, and minimum enrichment) can be user specified. This segmentation can be launched from PycoMeth's CLI and requires only the reference genome as an input and uses no sample-specific information.

Differential methylation testing

PycoMeth's Meth_Comp subcommand performs differential methylation testing on CpG-methylation calls between two or more samples. CpG-methylation calls can either be provided as one MetH5 file per sample, or a single MetH5 file containing read-group annotations. Furthermore, PycoMeth Meth_Comp requires an annotation file, in bed-format, defining the regions to be tested (**Figure 2B**). Either of the two segmentation methods provided by PycoMeth above provide reasonable candidate regions that can be used by PycoMeth Meth_Comp for differential methylation testing, but any user provided annotation will also work. If haplotype information is stored as read-groups in the MetH5 file, differential methylation testing between haplotypes can be used to determine allele specific methylation (ASM) patterns within a single sample. Statistical testing performed by PycoMeth can be parameterized with a variety of options. Comparison of methylation rates can be performed in two modes: The parameter `-hypothesis llr_diff` performs an unpaired ranked test between all methylation call LLRs of the two samples. This mode assumes all LLRs are independent and draws statistical power from both segment size and read-depth. More conservatively, the parameter `-hypothesis bs_diff` first computes a methylation rate (β-score), and draws statistical power only from read-depth (**Figure S1**). For multiple testing correction, PycoMeth optionally implements independent hypothesis weighing [19] and a number of options for p-value adjustment. Differentially methylated regions (DMRs) are reported as a tab-separated file, and the Comp_Report subcommand provides a method to generate easily digestible HTML reports visualizing DMRs in a functional context.

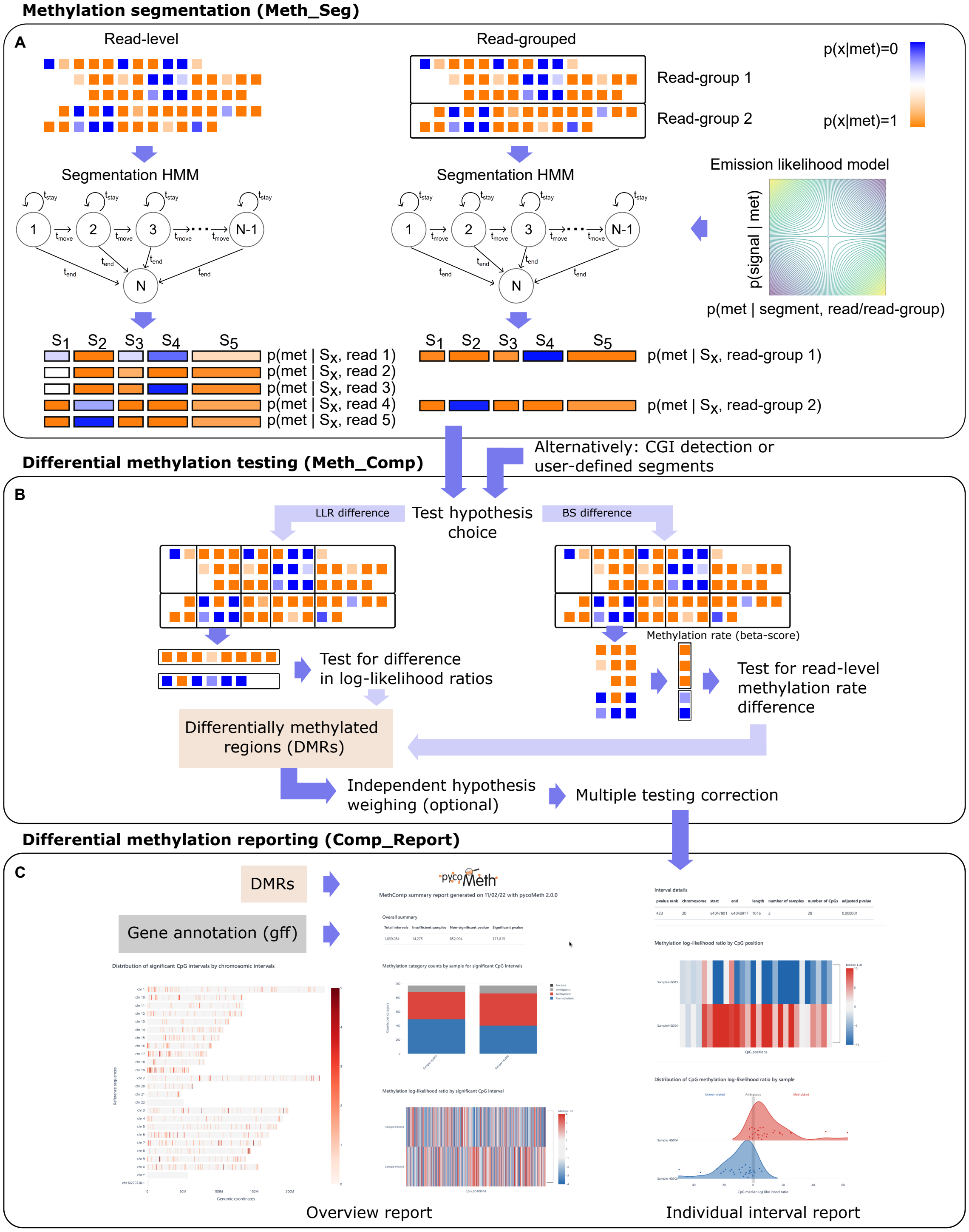

Methylation segmentation (Meth_Seg)
A
Read-level
Read-grouped
Read-group 1
Read-group 2
p(x|met)=0
p(x|met)=1
Segmentation HMM
Segmentation HMM
Emission likelihood model
p(signal | met)
p(met | segment, read/read-group)
1
2
3
N-1
N
t_{stay}
t_{move}
t_{end}
S_1 S_2 S_3 S_4 S_5
p(met | S_X, read 1)
p(met | S_X, read 2)
p(met | S_X, read 3)
p(met | S_X, read 4)
p(met | S_X, read 5)
p(met | S_X, read-group 1)
p(met | S_X, read-group 2)
Alternatively: CGI detection or user-defined segments
Differential methylation testing (Meth_Comp)
B
LLR difference
Test hypothesis choice
BS difference
Methylation rate (beta-score)
Test for difference in log-likelihood ratios
Test for read-level methylation rate difference
Differentially methylated regions (DMRs)
Independent hypothesis weighing (optional)
Multiple testing correction
Differential methylation reporting (Comp_Report)
C
DMRs
Gene annotation (gff)
pycoMeth
MethComp summary report generated on 11/02/22 with pycoMeth 2.0.0
Overall summary
Distribution of significant CpG intervals by chromosomic intervals
Methylation category counts by sample for significant CpG intervals
Methylation log-likelihood ratio by significant CpG interval
Interval details
Methylation log-likelihood ratio by CpG position
Distribution of CpG methylation log-likelihood ratio by sample
Overview report
Individual interval report

Figure 2: PycoMeth workflow. **A)** Methylome segmentation using a Bayesian changepoint detection HMM. Segmentation can be computed on a read-level or on a read-group (e.g. haplotype) level. Emission likelihood in the HMM models methylation call uncertainties as well as an optional methylation rate prior. **B)** Differential methylation testing either by comparing overall distribution of log-likelihood ratios in a segment, or by comparing read-level methylation rates, followed by multiple testing correction. **C)** The reporting module generates an overview HTML report, as well as individual interval reports.

Application to whole genome ONT sequencing data

We apply PycoMeth to ONT sequencing data from a father-mother-son trio sequenced by the Genome in a Bottle (GIAB) consortium [20]. Methylation calls on the samples HG002 (son), HG003 (father), and HG004 (mother) have been collected from Nanopolish, both as BAM file with MM tag as well as stored in MetH5 format (**Methods**). We use PycoMeth Meth_Seg to generate a methylome segmentation and test for DMRs between the parental samples. Using haplotype assigned reads, we further analyze HG003 for ASM.

To show the benefits of the MetH5 container format, we compare our format with the current definition and implementation of modification scores in HTSLib (MM and ML tag). Assessing the advantages of the MetH5 file format, **Figure 1D** compares random access and sequential access times of methylation scores stored in MetH5 versus BAM and CRAM (compressed BAM) format, as well as storage space required. While BAM/CRAM files make slightly more efficient (between 11 to 65 percent depending on compression) use of storage, they are significantly more expensive to read (about 5 to 90 times slower depending on compression).

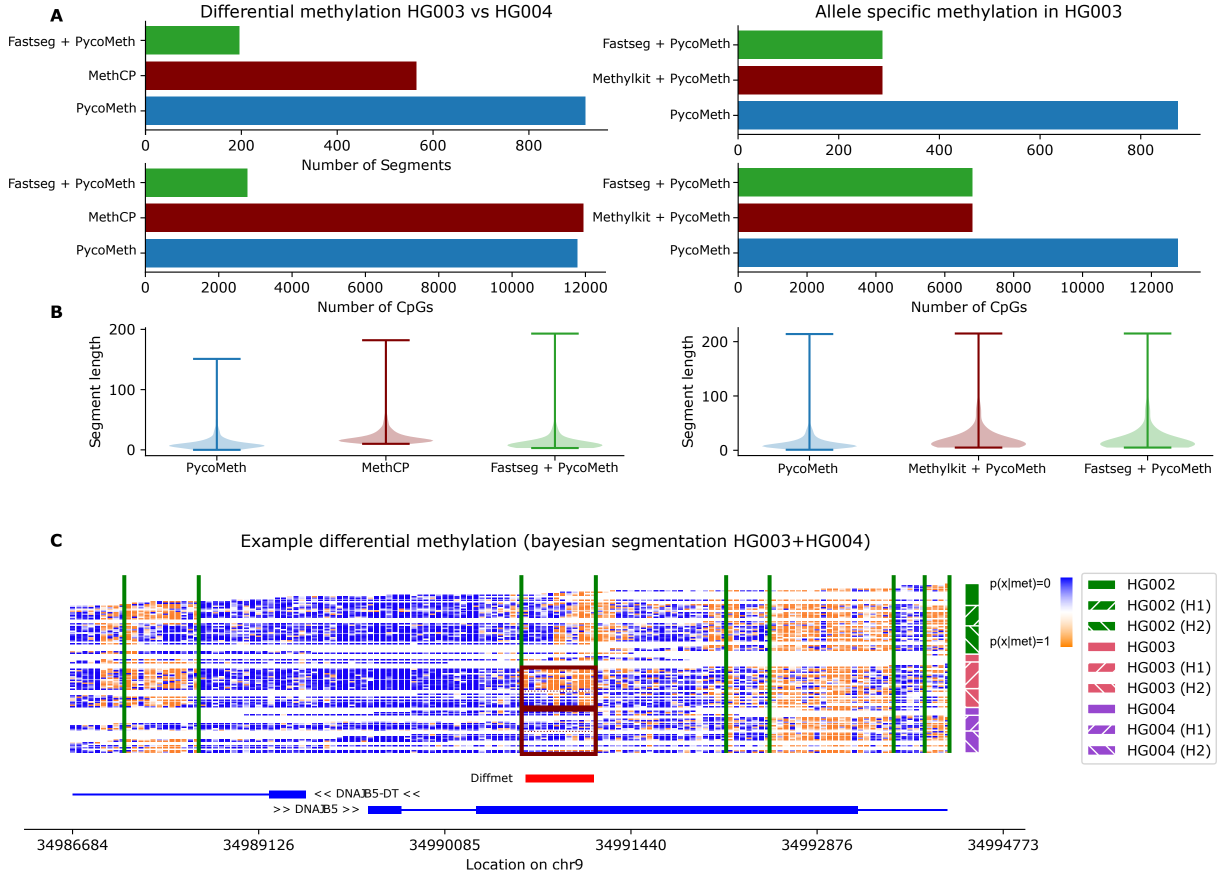


Figure 3: Benchmark of findings on HG003 and HG004. **A)** Number of segments and number of CpG-sites within segments that were identified as differentially methylated between different segmentations and differential methylation testing tools. **B)** Length distribution of differentially methylated segments in basepairs. **C)** Example differentially methylated segment identified by PycoMeth with Bayesian segmentation method based on HG003 and HG004 including haplotype information. Green lines represent the segmentation. Differentially methylated region is proximal to transcription start sites of various transcripts of gene DNAHB5. We also observe that differential methylation between the two samples is induced by methylation in only a single haplotype of HG003, whereas the other haplotype and both haplotypes of HG004 are unmethylated.

A combined methylome segmentation based on HG003 and HG004 with haplotype information identifies 1,039,084 segments to be tested for differential methylation. PycoMeth identifies 924 of these segments as significantly differentially methylated, with a minimum absolute methylation rate difference of 0.5 and FDR less than 0.05 **Figure 3A**). A sample-specific haplotype-informed segmentation of HG003 identified 1,046,252 segments, of which 874 were found to be allele specifically methylated, covering a total of 12,775 ASM CpG-sites **Figure 3A**).

Due to the lack of tools designed for segmentation and differential methylation analysis in single-molecule sequencing data, we instead compare our segmentation with three commonly used methods designed for segmentation of methylation rates from bisulfite sequencing data, methylKit [21], which implements single-sample methylation rate segmentation, fastseg [22], which is called by methylKit and can be used in a multi-sample setting, and MethCP [23]. When comparing with the methylKit and fastseg, we then apply PycoMeth Meth_Comp for differential methylation testing, when comparing to MethCP we leverage the MethCP segmentation and differential methylation test implementation.

Figure 3A shows the number of segments and CpG sites within segments found as differentially methylated in both the sample comparison as well as the ASM setting. With the PycoMeth Meth_Seg segmentation, the differential methylation test was able to identify about four times as many differentially methylated CpG sites in the sample comparison compared to MethylKit and about three times as many in the ASM scenario. **Figure 3C** shows an example segment called as differentially methylated and containing ASM, which was called by PycoMeth and missed by other calling methods. The MethCP segmentation and testing performed well and identified a slightly larger number of CpG sites as differentially methylated, albeit distributed over fewer segments. This is expected, seeing how MethCP also draws statistical power from the number of CpGs in a segment, whereas PycoMeth in the read-level β-score mode draws statistical power only from read-depth (**Figure S1**). We also note that MethCP finds largely different DMRs from PycoMeth, with only a small overlap in DMRs identified between the two methods (only 497 from 1489 total DMRs were called by both methods).

Conclusion

We present a toolkit and efficient file-format for epigenetic analyses on ONT reads. Due to the novelty of single-molecule methylation calling, few standards for storage and analysis exist. With the MetH5 format, we attempt to provide an efficient method of storing reference-anchored methylation calls without compromising on read-level information or methylation call uncertainty information. The PycoMeth Bayesian segmentation method and differential methylation testing take advantage of read-level or read-group level information, which tools designed for bisulfite sequencing typically do not consider. This new approach performs comparable or better than previous tools in terms of number of differentially methylated CpG sites identified, while integrating sample and haplotype information during segmentation and drawing power from added read depth. We also find that the type of segmentation greatly affects the kind of DMR identified and at this point would recommend to combine multiple segmentations, as the Bayesian segmentation by PycoMeth and the segmentation by MethCP are highly complementary.

Discussion

All tools have been developed and tested with methylation calls from Nanopolish[10] in mind, but are fully applicable to methylation calls from any methylation caller. More recent methods have show better methylation calling accuracy and thus given a preview of what is to come, but at this time, we found Nanopolish to be the most stable and user-friendly of the published tools. Once a new gold-standard methylation caller can be determined, we aim to adapt our toolbox to provide support. Specifically, with methylation calling capabilities currently being integrated with ONT basecaller guppy[24] and the development of bonito[25], we intend to expand the meth5 library and PycoMeth to support methylation calls from these callers as they get a stable release. While mainly developed for evaluation of CpG-methylation, all methods (aside from the CGI-finder) are also applicable to other types of epigenetic marks, such as adenine methylation, or cytosine methylation in GpC context, another avenue we are pushing for future versions of the software.

We have shown that the segmentation method used to determine segments for DMR testing has a great impact on DMRs found. We intend to further study possible improvements to methylome segmentation without biasing the segmentation towards differential methylation testing.

While using MetH5 as a file storage shows the markedly better performance to current SAM based file formats, we are aware of the advantages of having methylation calls stored in a well supported file format. In our experiments, we still observe compatibility issues between BAM files generated by Nanopolish, both with read-anchored as well as reference-anchored calls, and modbampy, and pysam support for modification calls is still limited. Still, we intend to implement support for BAM and CRAM files with modification calls in future versions of PycoMeth, as well as a conversion tool to and from meth5 format, as part of the meth5 library.

Materials and Methods

Data preparation

Raw fast5 files were downloaded from the Human Pangenome Project's S3 bucket. In order to reach approximately 20x to 30x coverage, we use 4 flow-cells from HG002 and 3 flowcells from HG003 and HG004 respectively (**Table S1**). Phased SVs were downloaded from the NCBI ftp server (**Table S1**). Reads have been re-basecalled using guppy version 5.0.11 with the high-accuracy model with modbases. Alignment to reference genome GRCH38 was performed using minimap2 [26] with the map-ont preset and otherwise default settings. Reads were haplotagged using whatshap[8] version 1.1. To produce Nanopolish[10] methylation calls in MetH5 format, we run Nanopolish call-methylation with Nanopolish version 0.13.3 and then use the python meth5 API to convert the Nanopolish output to the MetH5 format. In order to generate BAM files with MM tags, the "methylation_bam" branch (commit 9B01ad7) of Nanopolish has been used. BAM files were compressed to create CRAM files using `samtools view -C` and both BAM and CRAM were indexed using "samtools index". Performance comparisons between MetH5 and BAM/CRAM files were performed using the meth5 version 0.8.0 and modbampy version 0.4.1 [27]

Methylome segmentation

PycoMeth Meth_Seg was called with a window size of 300 CpG-calls (grouped calls are counted as one call) with a maximum of 20 segments per window. No methylation rate prior was provided and haplotype information was provided as read-groups in the MetH5 format. For the segmentation using MethylKit, fastseg, and MethCP, pseudo-bisulfite sequencing data has been created from the Nanopore methylation calls. Log-likelihood ratios were thresholded (level 2.0), binarized, and methylation rates were computed. The MethylKit segmentations for ASM calling were created based on total methylation rate per sample. For fastseg and MethCP, methylation rates per haplotype were computed. MethylKit segmentations were created using the function methSeg with parameters maxInt=100 and minSeg=10 as suggested in the MethylKit documentation. Fastseg segmentations were created using the function fastseg and the parameter cyberWeight=100 in order to get a comparable number of segments to PycoMeth. MethCP was run with default parameters.

Differential methylation testing

In all comparisons, PycoMeth Meth_Comp was run with the "bs_diff" hypothesis option for a more conservative evaluation. This which bases the test's power on read-depth rather than segment length. P-value adjustment was performed using the Benjamini-Hochberg method[28] on p-values weighted using independent hypothesis weighting . MethCP differential methylation testing was run with Fisher's combined probability test. P-values reported by MethCP are already reported as adjusted by MethylKit's implementation of SLIM. All significantly different intervals were then subset to those with a minimum of 0.5 methylation rate difference between group (methylation rate computed per CpG site and then averaged over the segment). Intervals called by MethCP which were based on a single call were removed, since these obtained false significance from grouped Nanopolish calls being duplicated in the pseudo-bulk generation (**Figure S1**)

Acknowledgments

The present contribution is supported by the Helmholtz Association under the joint research school "HIDSS4Health - Helmholtz Information and Data Science School for Health".

We acknowledge Adrien Leger for developing the first version of PycoMeth, which included the sequence based (CGI) segmentation, LLR distribution based differential methylation testing, and reporting.

Abbreviations

- API: application programming interface
- ASM: allele specific methylation
- CGI: CpG-Island
- CLI: command line interface
- CpG: 5'-cytosine-phosphate-guanine-3'
- DMR: differentially methylated region

- HDF5: Hierarchical Data Format (HDF) version 5
- HMM: Hidden Markov Model
- LLR: log-likelihood ratio
- NCBI: National Center for Biotechnology Information
- ONT: Oxford Nanopore Technologies

Availability of data and materials

Code availability

Software:

- PycoMeth: `https://github.com/snajder-r/pycometh`
- meth5: `https://github.com/snajder-r/MetH5Format`

Benchmark scripts: `https://github.com/snajder-r/benchmark_meth5`

Data availability

Sequencing raw data and variant calls were downloaded from the Genome in a Bottle (GIAB) consortium [20]. Download links are provided in **Table S1**.

Ethics approval and consent to participate

Not applicable

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable

Authors' contributions

R.S. with guidance by M.B. and O.S. designed and developed the MetH5 format and Bayesian methylation segmentation method, developed version 2.0 of the differential methylation testing in PycoMeth. R.S. prepared the figures and wrote the manuscript with input from M.B. and O.S.

References

[1] Lisa D Moore, Thuc Le, and Guoping Fan. DNA methylation and its basic function. *Neuropsychopharmacology*, 38(1):23–38, January 2013.

[2] En Li and Yi Zhang. DNA methylation in mammals. *Cold Spring Harb. Perspect. Biol.*, 6(5):a019133, May 2014.

[3] Suresh Kumar, Viswanathan Chinnusamy, and Trilochan Mohapatra. Epigenetics of modified DNA bases: 5-methylcytosine and beyond. *Front. Genet.*, 9:640, December 2018.

[4] Sergey Kurdyukov and Martyn Bullock. DNA methylation analysis: Choosing the right method. *Biology*, 5(1), January 2016.

[5] Fumihito Miura, Yusuke Enomoto, Ryo Dairiki, and Takashi Ito. Amplification-free whole-genome bisulfite sequencing by post-bisulfite adaptor tagging. *Nucleic Acids Res.*, 40(17):e136, September 2012.

[6] Suhua Feng, Zhenhui Zhong, Ming Wang, and Steven E Jacobsen. Efficient and accurate determination of genome-wide DNA methylation patterns in arabidopsis thaliana with enzymatic methyl sequencing. *Epigenetics Chromatin*, 13(1):42, October 2020.

[7] Medhat Mahmoud, Nastassia Gobet, Diana Ivette Cruz-Dávalos, Ninon Mounier, Christophe Dessimoz, and Fritz J Sedlazeck. Structural variant calling: the long and the short of it. *Genome Biol.*, 20(1):246, November 2019.

[8] Murray Patterson, Tobias Marschall, Nadia Pisanti, Leo van Iersel, Leen Stougie, Gunnar W Klau, and Alexander Schönhuth. WhatsHap: Weighted haplotype assembly for Future-Generation sequencing reads. *J. Comput. Biol.*, 22(6):498–509, June 2015.

[9] Sergey Nurk, Sergey Koren, Arang Rhie, Mikko Rautiainen, Andrey V Bzikadze, Alla Mikheenko, Mitchell R Vollger, Nicolas Altemose, Lev Uralsky, Ariel Gershman, Sergey Aganezov, Savannah J Hoyt, Mark Diekhans, Glennis A Logsdon, Michael Alonge, Stylianos E Antonarakis, Matthew Borchers, Gerard G Bouffard, Shelise Y Brooks, Gina V Caldas, Haoyu Cheng, Chen-Shan Chin, William Chow, Leonardo G de Lima, Philip C Dishuck, Richard Durbin, Tatiana Dvorkina, Ian T Fiddes, Giulio Formenti, Robert S Fulton, Arkarachai Fungtammasan, Erik Garrison, Patrick G S Grady, Tina A Graves-Lindsay, Ira M Hall, Nancy F Hansen, Gabrielle A Hartley, Marina Haukness, Kerstin Howe, Michael W Hunkapiller, Chirag Jain, Miten Jain, Erich D Jarvis, Peter Kerpedjiev, Melanie Kirsche, Mikhail Kolmogorov, Jonas Korlach, Milinn Kremitzki, Heng Li, Valerie V Maduro, Tobias Marschall, Ann M McCartney, Jennifer McDaniel, Danny E Miller, James C Mullikin, Eugene W Myers, Nathan D Olson, Benedict Paten, Paul Peluso, Pavel A Pevzner, David Porubsky, Tamara Potapova, Evgeny I Rogaev, Jeffrey A Rosenfeld, Steven L Salzberg, Valerie A Schneider, Fritz J Sedlazeck, Kishwar Shafin, Colin J Shew, Alaina Shumate, Yumi Sims, Arian F A Smit, Daniela C Soto, Ivan Sović, Jessica M Storer, Aaron Streets, Beth A Sullivan, Françoise Thibaud-Nissen, James Torrance, Justin Wagner, Brian P Walenz, Aaron Wenger, Jonathan M D Wood, Chunlin Xiao, Stephanie M Yan, Alice C Young, Samantha Zarate, Urvashi Surti, Rajiv C McCoy, Megan Y Dennis, Ivan A Alexandrov, Jennifer L Gerton, Rachel J O'Neill, Winston Timp, Justin M Zook, Michael C Schatz, Evan E Eichler, Karen H Miga, and Adam M Phillippy. The complete sequence of a human genome. May 2021.

[10] Jared T Simpson, Rachael E Workman, P C Zuzarte, Matei David, L J Dursi, and Winston Timp. Detecting DNA cytosine methylation using nanopore sequencing. *Nat. Methods*, 14(4):407–410, April 2017.

[11] Shangqian Xie, Amy Wing-Sze Leung, Zhenxian Zheng, Dake Zhang, Chuanle Xiao, Ruibang Luo, Ming Luo, and Shoudong Zhang. Applications and potentials of nanopore sequencing in the (epi)genome and (epi)transcriptome era. *Innovation (N Y)*, 2(4):100153, November 2021.

[12] Peng Ni, Neng Huang, Zhi Zhang, De-Peng Wang, Fan Liang, Yu Miao, Chuan-Le Xiao, Feng Luo, and Jianxin Wang. DeepSignal: detecting DNA methylation state from nanopore sequencing reads using deep-learning. *Bioinformatics*, April 2019.

[13] megalodon: Megalodon is a research command line tool to extract high accuracy modified base and sequence variant calls from raw nanopore reads by anchoring the information rich basecalling neural network output to a reference genome/transriptome, .

[14] Zaka Wing-Sze Yuen, Akanksha Srivastava, Runa Daniel, Dennis McNevin, Cameron Jack, and Eduardo Eyras. Systematic benchmarking of tools for CpG methylation detection from nanopore sequencing. *Nat. Commun.*, 12 (1):3438, June 2021.

[15] Product comparison. `https://nanoporetech.com/products/comparison`, . Accessed: 2022-2-8.

[16] Vladimir N Babenko, Irina V Chadaeva, and Yuriy L Orlov. Genomic landscape of CpG rich elements in human. *BMC Evol. Biol.*, 17(Suppl 1):19, February 2017.

[17] Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A McCarthy, Robert M Davies, and Heng Li. Twelve years of SAMtools and BCFtools. *Gigascience*, 10(2), February 2021.

[18] Quincey Koziol and Dana Robinson. HDF5. [Computer Software] https://doi.org/10.11578/dc.20180330.1, March 2018.

[19] Nikolaos Ignatiadis, Bernd Klaus, Judith B Zaugg, and Wolfgang Huber. Data-driven hypothesis weighting increases detection power in genome-scale multiple testing. *Nat. Methods*, 13(7):577–580, July 2016.

[20] Justin M Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E Mason, Noah Alexander, Elizabeth Henaff, Alexa B R McIntyre, Dhruva Chandramohan, Feng Chen, Erich Jaeger, Ali Moshrefi, Khoa Pham, William Stedman, Tiffany Liang, Michael Saghbini, Zeljko Dzakula, Alex Hastie, Han Cao, Gintaras Deikus, Eric Schadt, Robert Sebra, Ali Bashir, Rebecca M Truty, Christopher C Chang, Natali Gulbahce, Keyan Zhao, Srinka Ghosh, Fiona Hyland, Yutao Fu, Mark Chaisson, Chunlin Xiao, Jonathan Trow, Stephen T Sherry, Alexander W Zaranek, Madeleine Ball, Jason Bobe, Preston Estep, George M Church, Patrick Marks, Sofia Kyriazopoulou-Panagiotopoulou, Grace X Y Zheng, Michael Schnall-Levin, Heather S Ordonez, Patrice A Mudivarti, Kristina Giorda, Ying Sheng, Karoline Bjarnesdatter Rypdal, and Marc Salit. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci Data*, 3:160025, June 2016.

[21] Altuna Akalin, Matthias Kormaksson, Sheng Li, Francine E Garrett-Bakelman, Maria E Figueroa, Ari Melnick, and Christopher E Mason. methylkit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles. *Genome Biol.*, 13(10):R87, October 2012.

[22] P Baldi and A D Long. A bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. *Bioinformatics*, 17(6):509–519, June 2001.

[23] Boying Gong and Elizabeth Purdom. MethCP: Differentially methylated region detection with change point models. *J. Comput. Biol.*, 27(4):458–471, April 2020.

[24] Nanopore community. `https://nanoporetech.com/community`, . Accessed: 2022-2-16.

[25] bonito: A PyTorch basecaller for oxford nanopore reads, .

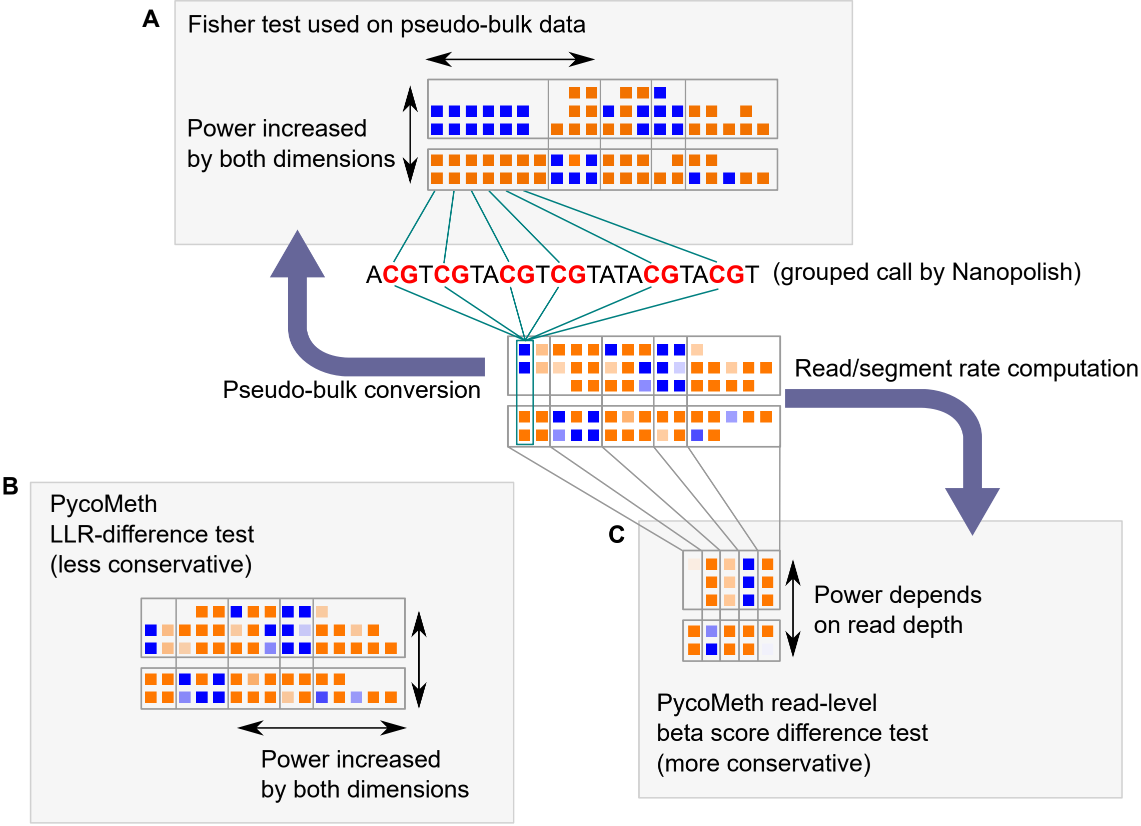
[26] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, September 2018.

[27] modbampy. `https://pypi.org/project/modbampy/`, . Accessed: 2022-2-13.

[28] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. R. Stat. Soc. Series B Stat. Methodol.*, 57(1):289–300, 1995.

Additional Files

- **Table S1**: `s1_giab_data.xlsx` - Benchmark data including download links



Supplementary Figure 1: Illustration of where different differential methylation testing methods draw their power. **A**) In attempting to analyze Nanopolish methylation calls with tools developed for bulk bisulfite sequencing data, we create pseudo-bulk data. Tests generated for pseudo-bulk comparison (such as MethCP which we evaluated in this work) test based on CpG-level methylation rates and coverage across all reads and therefore draw power from the segment size and read depth. Furthermore, since Nanopolish generates grouped calls for nearby CpG-sites, some calls are therefore not independent and thus artificially generate more testing power. **B**) PycoMeth with the parameter "–hypothesis llr_diff" performs the less conservative test, implemented in the PycoMeth package, where each individual methylation call is treated as independent and samples are compared based on their LLR distribution. Here discovery power is determined also by a combination of segment length and sequencing depth. **C**) PycoMeth with the parameter "–hypothesis bs_diff" instead computes a methylation rate per read per segment and draws power only from the independent information (sequencing depth).