Nanopore Methylation Calling from Limited Training Data

Brian Yao
Jennifer Listgarten

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-15
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-15.html

May 1, 2022

Acknowledgement

First and foremost, I would like to thank Professor Listgarten for her guidance and mentorship throughout my time working in her group. The last two years have been a deeply rewarding and educational experience. I would also like to thank Professor Holmes for providing valuable commentary on this research project and Chloe Hsu for her extremely helpful advice and direction. Lastly, thank you to my family and friends for their support throughout the unprecedented circumstances of this past year.

Nanopore Methylation Calling from Limited Training Data

by Brian Yao

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Jennifer Listgarten
Research Advisor

5/11/2021

(Date)

* * * * * * *

Ian Holmes

Professor Ian Holmes
Second Reader

5/12/2021

(Date)

Nanopore methylation calling from limited training data

Brian Yao*[1], Chloe Hsu[1], Gal Goldner[2], Yael Michaeli[2], Yuval Ebenstein[2], and Jennifer Listgarten*[1]

[1]Dept. of Electrical Engineering & Computer Sciences, University of California Berkeley
[2]Dept. of Chemical Physics, Tel Aviv University
*Correspondence to: brian.yao@berkeley.edu, jennl@berkeley.edu

Abstract

Nanopore sequencing platforms combined with machine learning models have been shown to be effective for detecting base modifications in DNA such as 5mC and 6mA. However, a challenge in building machine learning-based callers is access to labelled training data that span all modifications on all possible DNA k-mer backgrounds—a *complete* training dataset. Nanopore calling has historically been done with Hidden Markov Models (HMMs); these HMMs cannot make successful calls in k-mer contexts not seen during training because of their independent emission distributions. However, deep neural networks (DNNs) are increasingly being used to make base and modification calls, often outperforming their HMM cousins in the complete data setting. Moreover, it stands to reason that the DNN approach should be able to better generalize to unseen examples because its parameters are more fully shared across all training examples. Herein, we demonstrate that indeed a common DNN approach (DeepSignal) outperforms a common HMM approach (Nanopolish) in the incomplete data setting. Furthermore, we propose a novel hybrid approach, *AmortizedHMM*, demonstrating that it outperforms both the pure HMM and DNN approaches on methylation calling when the training data are incomplete.

Nanopore sequencing for epigenetics

Nanopore sequencing is a third-generation technology for sequencing DNA and RNA that provides advantages over other technologies, such as long read lengths, inexpensive sample preparation, real-time sequencing [1, 2], and owing to its small size, mobile sequencing [3]. Additionally nanopores are increasingly being used to detect epigenetic modifications to DNA, particularly methylation marks [4]. The nanopore device works by running an ionic current through nanometer-wide pores. As a DNA molecule passes through the pore, the current across the pore changes in a manner that is characteristic of the molecules in the pore, namely the RNA/DNA sequence and its modifications. From measuring the current from known sequences and modifications, one can build up a supervised training dataset suitable for machine learning (ML) methods that are then able to transform future, unlabelled current signals to their corresponding sequence of bases and modifications [5].

Early studies demonstrated that nanopore sequencing could be used for the detection of epigenetic modifications in DNA, showing that distinct current levels are produced when a modified base is present in the pore [6, 7]. These successes sparked the development of supervised machine learning methods for methylation calling on nanopore data [8, 9, 10, 11]. The first methylation marks to be tackled by nanopore technology were those of 5-methylcytosine (5mC), which has been particularly well-studied due to its abundance in the human genome [12], with previous studies linking 5mC content to a number of key biological processes such as aging and cancer [13, 14]. Another base modification of interest is 5-hydroxymethylcytosine (5hmC), which is common in mammalian brain tissue, accounting for 40% of modified cytosine in the central nervous system [15]. Moreover, 5hmC content in brain cells increases with age, suggesting that it is linked to neurodevelopment [16]. Early results suggested that nanopore devices may also be able to pick up on 5hmC-specific current signal [6], although calling these marks accurately in the presence of other marks has not yet been conclusively achieved.

Generalization in nanopore callers

Although supervised ML methods are developed specifically for their ability to generalize to unseen examples, the notion of generalization for nanopore sequencing is nuanced. For example, one form of generalization for base calling is from the current observations for one k-mer, to slightly different current observations, arising from stochastic noise in the system, for that same k-mer. We call this *sensor generalization*, because the generalization is required owing to sensor noise. Another form of generalization relevant to nanopore sequencing is *k-mer generalization*, wherein an ML-based caller must make accurate calls for k-mers that it has never seen current observations for. Note that in the case of epigenetics, in addition to k-mers comprised of the standard nucleotides, we also consider *modified k-mers*, which include methylated bases.

When constructing a training dataset for *base* callers, it is relatively easy to generate a *k-mer complete* dataset—one in which current observations associated with all possible k-mers are present. This can be achieved by taking, for example, a sample of human DNA, amplifying the DNA and running it through the nanopore. Labels for training can be obtained using alternative sequencing platforms. Consequently, typically it suffices to require only sensor generalization for base calling.

When it comes to constructing a training dataset for a particular methylation mark, it can be more difficult to obtain a similarly comprehensive dataset, largely due to the burden of obtaining high-confidence reference labels for these modifications. In the case of detecting 5mC modifications, previous studies have used the gold standard assay of bisulfite sequencing to obtain supervisory calling labels [17, 18], achieving a 6-mer complete dataset. However, as we move to other modifications, such as 5hmC, achieving similarly complete training data becomes increasingly difficult. TET-assisted bisulfite sequencing (TAB-seq) and oxidative bisulfite sequencing (oxBS) are currently the standard methods for reading 5hmC at single-base resolution [19]. However, both methods are expensive and low-throughput [20]; they also require high coverage to make high-confidence 5hmC calls (particularly oxBS) [19]. Additionally, beyond these sequencing challenges, rarity of certain epigenetic marks may also present a problem, as it may be the case that not all k-mers containing a given modification are represented in a specific genome. As the field progresses to simultaneous calling of multiple types of epigenetic marks, achieving a complete dataset with respect to all of the marks will become harder still. Consequently, as nanopore sequencing technology is used to call more and more epigenetic marks, we require ML-based callers that are accurate even with limited training data. In particular, the callers will require both k-mer and sensor generalization.

To further illustrate why it might be challenging to obtain high quality k-mer complete training data for a given methylation mark, consider that for base calling, the alphabet is of size four: `{A, C, G, T}`, whereas for a given methylation mark that can occur only on a cytosine, we expand the alphabet to size five: `{A, C, G, T, M}`. For current pore models where $k = 6$, we go from $4^6 = 4,096$ unique k-mers to $5^6 = 15,625$. In practice, only methylated sites in certain contexts are possible, such as detecting 5mC modifications which, in mammalian genomes, can occur only on the C in a CpG dinucleotide [21]. Additionally, even if the pore contains only, say, six bases at a time, ML callers may be able to make use of larger contexts still to improve calling, exacerbating the combinatorial explosion of possible k-mers. Indeed, recently developed pore models for the new

R10 Oxford Nanopore Technology (ONT) chemistry use $k = 9$ [22], which yields $5^9 = 1,953,125$ unique methylation contexts. Herein, we will restrict ourselves to $k = 6$, although the conclusions that emerge should be equally, if not more, applicable to larger values of k.

Next we describe the two main modelling paradigms currently used for nanopore-based methylation calling, discussing how they relate to sensor and k-mer generalization. Then we propose and demonstrate the utility of our new method, which is a hybrid between the two approaches. Note that in both of these modelling approaches, and later in our own, newly developed approach, methylation calling occurs after base calling.

Hidden Markov Model-based callers. Simpson et al. [10] developed the widely-used Nanopolish, a Hidden Markov Model (HMM)-based approach to detecting 5mC in CpG contexts. The Nanopolish HMM assumes a different current distribution for each unique k-mer, including distinct distributions for modified versions of a k-mer. For example, a k-mer, `CGAACG`, that has a 5mC in the fifth position, denoted `CGAAMG`, has its own mean and variance of current distribution in Nanopolish, and `MGAAMG` in turn has its own, and so forth. That is, every possible modification on top of any DNA background—a unique modified k-mer—has its current distribution modelled independently. The Markov transitions in the HMM ensure a coherence of calls as the DNA sequence moves through the pore. That is, if the HMM believes the last call in the sequence being pulled through the pore was a `CAMGAT`, then the next call in the sequence should be off-set by a shift of one, `AMGATX`, for wildcard `X`. Because of the independent current distributions—called emission distributions in HMM parlance—for each modified k-mer, the *HMM-based Nanopolish approach requires having seen all possible modified k-mers in the training data.*

Deep Neural Network-based callers. Recently, there has been a shift to using deep neural networks (DNN) for base [23, 24, 25] and methylation calling [17, 18]. In particular, for methylation calling, Ni et al. [17] combined a bidirectional recurrent neural network (RNN) with long short-term memory (LSTM) units that constructs sequence-based features and a convolutional neural network (CNN) that processes raw signal values in order to perform methylation calling. This approach is called DeepSignal [17]. Liu et al. [18] similarly used an LSTM-RNN architecture in their tool DeepMod, also adding a secondary network to account for correlation of methylation marks on nearby sites. These DNN-based methods have been shown to provide a performance improvement over the HMM-based Nanopolish for 5mC calling [26]. Importantly, because these DNN approaches do not have parameters that are *a priori* independent for each modified k-mers, it stands to reason that *they should perform better than HMM-based approaches in generalizing to new modified k-mers*—that is perform better k-mer generalization. Although it has not previously been shown, we will demonstrate herein that this is indeed the case.

A novel hybrid HMM-DNN approach to methylation calling. Although we show that the DNN has better k-mer-generalization than the HMM approach, we hypothesized that combining the two modelling approaches may provide better k-mer-generalization, and therefore better robustness to incomplete training datasets. Our approach, AmortizedHMM, first trains a Nanopolish-like HMM on the training data that is available. This yields a learned emission distribution for each modified k-mer in the training data. Next we train a feedforward deep neural network (FDNN) to estimate the emission distribution for a given modified k-mer. Finally, in our hybrid approach, we use the Nanopolish HMM where we impute any missing modified k-mer emission distributions with that predicted by the FDNN. Because we are sharing information between the emission distributions by way of the FDNN, we say that we are amortizing the emission distributions, hence the name, AmortizedHMM. In addition to developing this hybrid model, we also develop a new algorithm for choosing which k-mers to use for training in the k-mer incomplete setting.

Next we describe a series of experiments comparing and contrasting our proposed hybrid approach to pure DNN and HMM approaches, across a range k-mer incompleteness settings, showing that for complete training data, the DNN is best, but that as training data becomes incomplete, that our hybrid approach dominates in performance.

Results

We focused our empirical investigation on the problem of 5mC calling, for which several high quality datasets exist, and for which existing callers have been developed with the intent of having approximately k-mer complete training data. However, we consider this a proof-of-principle for harder tasks such as 5hmC calling, or joint calling of 5mC and 5hmC, and so forth.

We trained three types of methylation callers, described in the previous section, on k-mer incomplete datasets. The first two callers are existing approaches for which we used code provided by the authors: the HMM-based Nanopolish [10], and the DNN-based DeepSignal [17]. No model selection or architecture search was performed for these methods. The third approach is our newly proposed approach, AmortizedHMM, for which we performed architecture search in the k-mer complete setting, as was done for Nanopolish and DeepSignal. We did not change any of the model architectures when testing them in k-mer incomplete settings.

We focused on datasets representing naturally occurring 5mC marks in human genomes, as in Ni et al. [17] and Liu et al. [18], which include k-mers that are unmethylated, partially methylated, and fully methylated. In particular, we trained and evaluated our models using two primary nanopore datasets obtained from sequencing two different human genomes, HX1 [8] and NA12878 [2]. Additionally, we obtained gold standard bisulfite 5mC labels for NA12878 from ENCODE (ENCFF835NTC) [27] and for HX1 from the NCBI Sequence Read Archive (PRJNA301527) [8].

From these primary datasets, we constructed a range of k-mer incomplete training datasets. Briefly, these were created within 6-fold cross-validation. By default, each training fold contains k-mer complete data. Next, to create a, say, 10%-complete training dataset, we compute the number of modified k-mers that this corresponds to, say 250 k-mers. Although in principle we could then simply choose 250 of the training modified k-mers at random for our 10%-complete dataset, this would not correspond to a real physical situation owing to the fact that a single methylated site in a genome corresponds to six modified k-mers (for $k = 6$), all shifted from each other by one position. Random selection would not guarantee modified k-mers shifted from each other in this fashion. Thus, to actually order k-mers for training, we need to account for this physical reality. We use a slight modification of the random modified k-mers selection scheme, whereby we enforce that all six modified k-mers for that one modified site are simultaneously included in the training data. In practice, we use a linear integer program to compute these k-mer-incomplete training datasets. The algorithm we developed to perform this training data k-mer selection also accounts for the frequency with which any modified k-mer[1] occurs in the genome of choice (*e.g.*, human genome), so that priority is given to more commonly occurring modified k-mers. However, this can also be run simply with frequency of base composition if the modifications in the target genome are not known. In any case, given a set of k-mers, we then filter the k-mer complete training data such that the only remaining modified k-mers are those present in the set. Note that each individual modified

[1] This part of the algorithm actually looks at the frequency of the 11-mer (for 6-mers) sequence that contains the "central" modification.

k-mer will generally appear many times in the training data (with distinct sensor readings), but the total number of unique k-mers is limited. Additionally, for a given fold, test sets remained the same for every level of incompleteness and, as was the case with our training folds prior to k-mer filtering, were k-mer complete.

We will denote different levels of k-mer completeness by p. That is, p is the percentage of all possible modified k-mers that are present in the training data. A k-mer complete dataset has $p = 100$, while increasingly less complete data have $p < 100$. The smallest p we consider is five, which corresponds to fewer than 150 unique modified k-mers in the training data.

Accuracy of 5mC calling across a range of k-mer-incompleteness

For each of the two primary datasets and their various k-mer-incomplete versions, we ran each of the three ML-based callers (Figure 1). In this scenario, which most closely mimics a real-use case, the test set (which is k-mer complete) typically contains modified k-mers that also appeared in the training data, albeit with different (hold out) instantiations of the current observations. In the next section, we will evaluate again, testing only those modified k-mers not appearing in the training data, so as to assess k-mer-generalization specifically.

On both datasets, the performances of Nanopolish and AmortizedHMM were very similar for high values of p. This is to be expected, since when p is close to 100, AmortizedHMM does not need to impute many emission distributions; rather it can use the Nanopolish emission distributions directly. Note that even for the k-mer complete setting, AmortizedHMM and Nanopolish may diverge because Nanpolish requires a certain number of data points to learn each emission distribution, and otherwise sets this distribution to the default of being unmethylated. DeepSignal outperforms both other methods in this setting. This is consistent with earlier results in where it was shown that DeepSignal outperforms Nanopolish in the k-mer complete setting [17, 26]. We speculate that DeepSignal performs better than AmortizedHMM here because the amortization implicit therein applies to the raw data inputs, rather to the HMM parameters used as inputs to AmortizedHMM.

As the training data become increasingly incomplete, AmortizedHMM starts to systematically outperform Nanopolish because of its ability to impute missing emission probabilities corresponding to modified k-mers not in the training data. Although AmortizedHMM has fewer of the estimated HMM parameters to fit its FDNN with when p is smaller, AmortizedHMM nevertheless is able to generalize quite well by virtue of its amortization. Meanwhile, DeepSignal continued to hold an advantage over the other methods for $p \geq 20 - 30$, the cross-over point for where AmortizedHMM starts to outperform it. We hypothesize that the diminished performance of DeepSignal with increasingly k-mer incomplete data arises from insufficient data to train on. It is possible that if we had performed an architecture search for DeepSignal at each level of k-mer incompleteness, that its performance could have been boosted. Importantly, however, none of the methods, including AmortizedHMM, had their architecture selected other than on the basis of k-mer complete training data, so as to make the comparison fair.

Decomposition into sensor and k-mer generalization

In order to better understand the source of AmortizedHMM's comparative success in low k-mer coverage regimes, we divided each of the cross-validation tests into two: one corresponding to modified k-mers not appearing in the training data to assess k-mer generalization, and one corresponding to only those appearing in the training data to assess sensor generalization. Across both primary datasets, DeepSignal is the clear winner for pure sensor generalization, whereas Nanpolish and AmortizedHMM perform similarly to each other, and well below DeepSignal. On the other hand, for

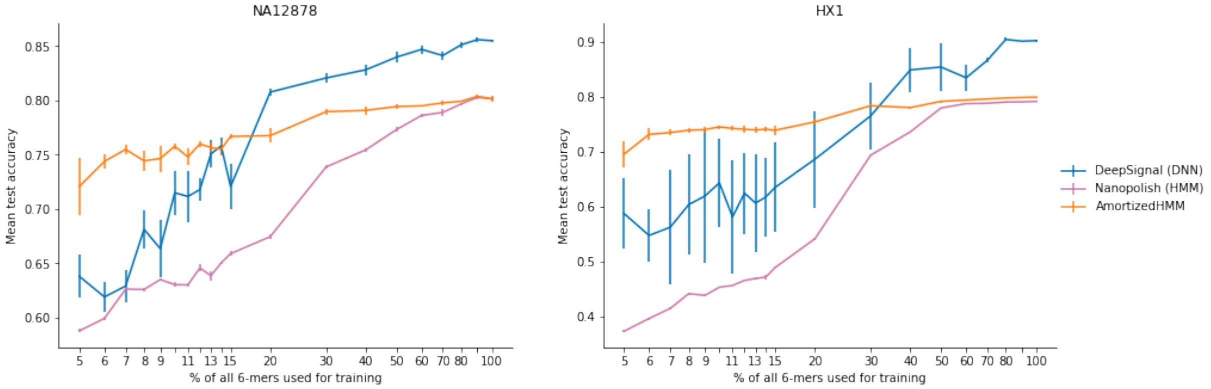

Figure 1: **Performance of 5mC calling across different k-mer incompleteness regimes.** Results averaged over 6-fold cross-validation, with length of error bars equal to one standard deviation across the folds. The k-mer complete case, $p = 100$, corresponds to a training dataset containing $2,669$ unique modified k-mers, whereas the case where $p = 5$ corresponds to a training dataset containing 133 unique modified k-mers.

k-mer-generalization, AmortizedHMM is the consistent winner, with DeepSignal coming in second, and Nanopolish, last.

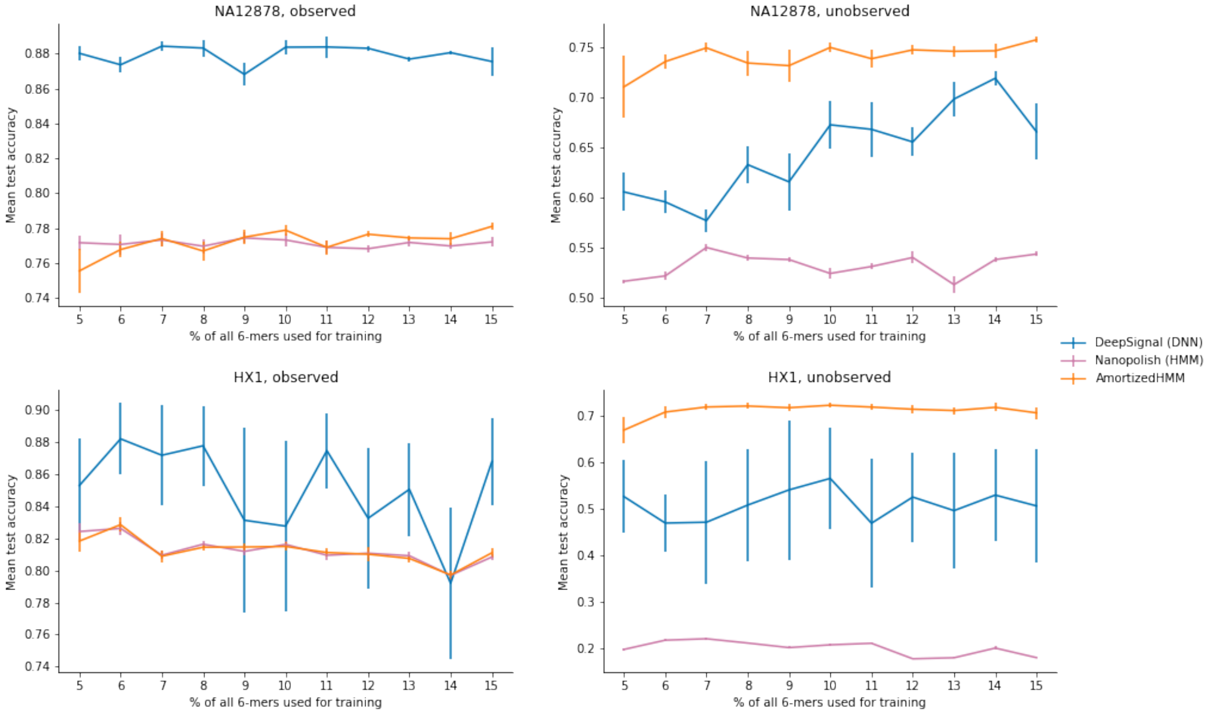

Figure 2: **Sensor and k-mer generalization.** Accuracy of methylation callers on k-mers previously appearing and not appearing in the training data. Results averaged over 6-fold cross-validation, with length of error bars equal to one standard deviation across the folds. The k-mer complete case, $p = 100$, corresponds to a training dataset containing $2,669$ unique modified k-mers, whereas the case where $p = 5$ corresponds to a training dataset containing 133 unique modified k-mers.

Investigation of low- and high-novelty in k-mer generalization. In the previous section, we treated all k-mer-generalization in the same way, whereas in reality, some modified k-mers not seen in the training data may be more similar to those in the training data than others, what we refer to as *low*- and *high*-novelty k-mers. To investigate if this issue affects performance, and how, we quantified distance to the training modified k-mers with the average Hamming distance to modified k-mers in the training data. Then we evaluated the calling accuracy for different distances. We performed this analysis for for all the values of p appearing in Figure 2, between five and fifteen, and averaged the results over these values of p (Figure 3). Although AmortizedHMM performed similarly to DeepSignal for low-novelty k-mers, it was far more accurate than DeepSignal for high-novelty k-mers. This difference in performance appears to underpin the success of AmortizedHMM over DeepSignal in k-mer generalization.

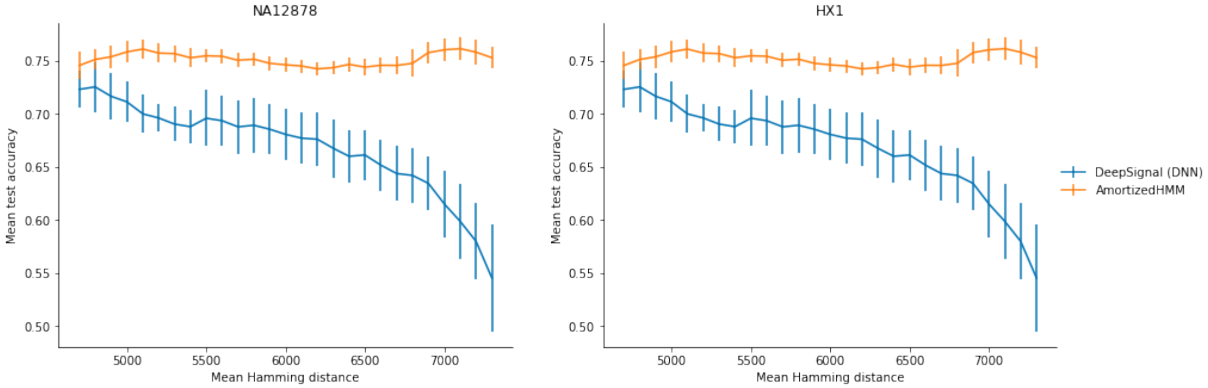

Figure 3: **Accuracy of methylation callers in detecting 5mC in previously unobserved k-mers, for different levels of k-mer novelty.** Accuracy of DeepSignal and AmortizedHMM on k-mers previously unobserved during training at varying levels of k-mernovelty. Results are averaged over model evaluations for values of p between 5 and 15, with length of error bars equal to one standard deviation.

Discussion

We investigated how several common modelling approaches, and our newly developed approach, for 5mC calling, are robust to generalizing to k-mers not seen at training time—what we refer to as the task of k-mer generalization. Although the DNN-based DeepSignal performed best with complete k-mer training data, as the training data became increasingly less and less complete, our newly proposed hybrid approach that combines HMMs and neural networks, AmortizedHMM, dominated in calling accuracy.

Although we focused our evaluation on 5mC detection, in practice, 5mC calling methods need not be specialized for the k-mer incomplete setting, as there already exist high k-mer coverage nanopore sequencing labelled datasets for 5mC. However, our method could be expanded to detect modifications for which obtaining such a dataset is not so straightforward, such as in cases where sequencing experiments for obtaining a ground truth reference are extremely costly, or in cases where a specific modification is especially rare in a given genome. In particular, we are working to improve calling for 5hmC.

Methods

Datasets

We trained and validated models for 5mC calling on two published Nanopore datasets. Jain et al. [28] sequenced the human genome NA12878 at 30x coverage using the ONT R9.4 pore chemistry, and Liu et al. [18] sequenced the HX1 genome, also at 30x coverage and using the R9.4 chemistry. In both cases, sequencing was performed on native DNA molecules containing native modifications (in contrast to other works where methylation was synthetically introduced in PCR-amplified samples using enzymes such as M.SssI methyltransferase [10, 11]). In addition to raw signal data, these datasets included base calling results obtained from running ONT-trained base callers: Guppy v2.3.8 for the NA12878 dataset and Albacore v2.3.1 for the HX1 dataset.

As a reference for 5mC modifications, we used bisulfite sequencing datasets obtained from sequencing NA12878 [27] and HX1 [18]. Following preprocessing steps using Bismark [29], we obtained a set of methylation calls for each CpG site corresponding to the bisulfite sequencing reads that cover that site. Similarly to [18] and [17], we then filtered for CpG sites that were (1) covered by a sufficient number of reads (in our case, at least 5) and (2) consistently called as methylated or unmethylated across every read covering the given site. The second step is necessary since methylation may not be consistent across the sequenced samples. This process provided us with a set of CpG sites for which we could confidently assign non-methylated or methylated labels for training and evaluation purposes.

k-mer selection

Recall that the practical application of our method is that one may not have k-mer complete data, possibly due to, for example, some limitation in the number of regions in a genome that can be sequenced because of experimental cost or the base modification of interest occurring infrequently in the genome. However, since the datasets considered in this work are k-mer complete, we must further filter our nanopore datasets so that only some subset of k-mers are represented in the data in order to simulate the k-mer incomplete setting. Here, we detail the process by which we select which k-mers to retain. Note that we continue to assume that $k = 6$.

We begin by defining which k-mers are candidates for removal. First, note that we are generally more concerned with how well-represented *methylated* k-mers are in the dataset. This is because in the HMM setting, the emission distribution parameters for unmethylated k-mers can be assumed to have already been learned. Moreover, in general, modified bases are more common than unmodified ones, so obtaining a dataset in which all unmethylated k-mers are represented is not very challenging. Thus, in practice, we only update the parameters for methylated k-mers. Second, we are only interested in calling 5mC in CpG contexts, so many methylated k-mers are not relevant for our analysis. For example, `AAAMAG` is not a valid k-mer, whereas `AAAAMG` and `AAAAAM` are. Simpson et al. [10] refer to this as the *CpG alphabet*.

We let n denote the number of valid k-mers that contain a methylated base. As described in the main text, we let p denote the percentage of k-mers that our reduced dataset ought to cover. For example, if $p = 100$, then we retain all k-mers. The obvious approach would then be to randomly select a set of $\lfloor \frac{pn}{100} \rfloor$ k-mers, which we denote as T, and then remove any appearances of k-mers that are not present in T. However, this approach faces immediate problems. Let us fix a position at which there is a methylated cytosine in a CpG dinucleotide (*i.e.*, an `MG`). We can consider the 11-mer centered on this methylated site, which we denote as S. As an example, S could be the 11-mer `GATTTMGCAAC`. This 11-mer may be viewed as a combination of six overlapping 6-mers $s_1, s_2, \ldots, s_6$, as described in Figure 4.

$$\begin{aligned}
S &= \texttt{GATTTMGCAAC}\\
s_1 &= \texttt{GATTTM}\\
s_2 &= \phantom{\texttt{G}}\texttt{ATTTMG}\\
s_3 &= \phantom{\texttt{GA}}\texttt{TTTMGC}\\
s_4 &= \phantom{\texttt{GAT}}\texttt{TTMGCA}\\
s_5 &= \phantom{\texttt{GATT}}\texttt{TMGCAA}\\
s_6 &= \phantom{\texttt{GATTT}}\texttt{MGCAAC}
\end{aligned}$$

Figure 4: Example decomposition of an 11-mer into 6-mers, each containing a modified base.

We note that our goal is to simulate a real, physical experiment producing a k-mer incomplete dataset. In this setting, when a methylated site passes through the nanopore, all six 6-mers which include that methylated site will necessarily pass through the pore together, or not at all. In other words, it is only sensible to keep this specific methylated site in the dataset if all of $s_1, s_2, \ldots, s_6$ are in T, since removing just one of these k-mers while keeping the rest would not be consistent with an actual sequencing experiment. Then, since $\mathbb{P}\left[s_i \in T\right] \approx \frac{p}{100}$ for $i = 1, \ldots, 6$, the probability that we keep this particular methylated site in our dataset is near $\left(\frac{p}{100}\right)^6$ (we assume that the events $s_i \in T$ are approximately independent). In our analysis, we will take p to be as small as 5, in which case virtually every methylated site will be removed from the data. Moreover, we have assumed that $k = 6$—if we were to consider longer k-mers, the probability of retaining a methylated site would decay even more quickly as a function of p. Consequently, in order to retain more methylated sites in the dataset, we must select the k-mers we keep more carefully. Intuitively speaking, the main issue with randomly selecting k-mers is that a random procedure is unlikely to select k-mers which are adjacent (*i.e.*, k-mers of the form $\texttt{XY}_\texttt{1}\texttt{Y}_\texttt{2}\texttt{Y}_\texttt{3}\texttt{Y}_\texttt{4}\texttt{Y}_\texttt{5}$ and $\texttt{Y}_\texttt{1}\texttt{Y}_\texttt{2}\texttt{Y}_\texttt{3}\texttt{Y}_\texttt{4}\texttt{Y}_\texttt{5}\texttt{Z}$, where $\texttt{X}, \texttt{Y}_\texttt{i}, \texttt{Z} \in \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}, \texttt{M}\}$) and can combine to form a longer sequence centered on a methylated site (such as in Figure 4).

To encourage the selection of a more coherent set of k-mers, we formulate our k-mer selection problem as an integer linear program (ILP). Informally, we select a set T of methylated k-mers with maximum size $B = \lfloor \frac{pn}{100} \rfloor$, and aim to maximize the frequency-weighted count of possible $(2k-1)$-mers (in this case, 11-mers) that (1) are centered on a methylated base in a CpG site, such as S in Figure 4 and (2) can be generated as a combination of overlapping k-mers in the set T.

$$\max \sum_{i=1}^{m} w_i y_i \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{n} x_i \leq B, \tag{2}$$

$$0 \leq -k y_i + \sum_{j=1}^{k} x_{i_j} \leq k - 1 \quad \forall i \in \{1, \ldots, m\} \tag{3}$$

Here, the constant n still denotes the number of methylated k-mers, whereas m denotes the total number of possible $(2k-1)$-mers centered on a methylated CpG. The integer variables $x_i \in \{0, 1\}$ denote whether the i^{th} k-mer is included in the set T. Thus, Equation (2) represents a constraint on the size of T. For each methylation-centered $(2k-1)$-mer S_i (with $1 \leq i \leq m$), $x_{i_1}, \ldots x_{i_k}$ correspond to the k-mers of which S_i is comprised. For example, using the example from Figure 4 where we

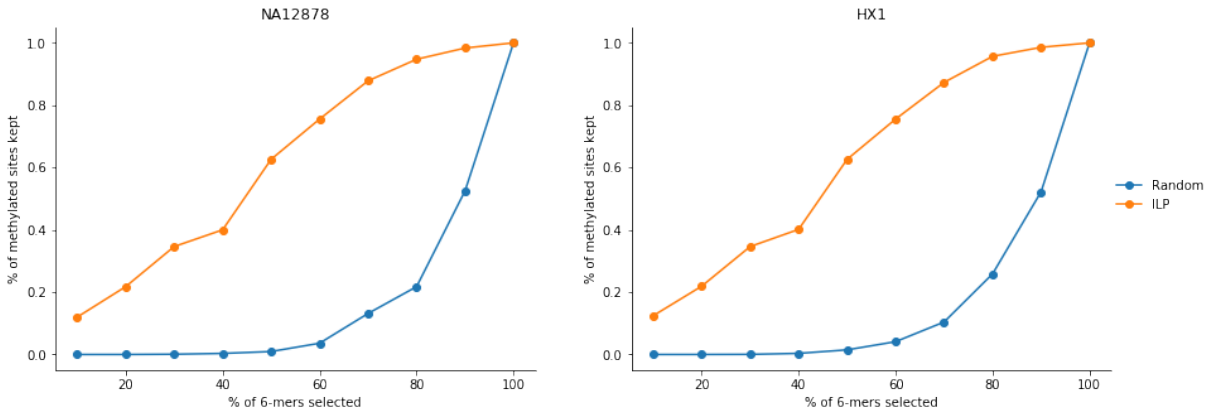

Figure 5: **Change in percentage of methylated sites retained in data with respect to percentage of k-mers kept, compared between randomly selected k-mer sets and ILP-selected k-mer sets.**

take S_i = `GATTTMGCAAC`, $x_{i_1}, \ldots, x_{i_k}$ denote whether the k-mers `GATTTM`, `ATTTMG`, `TTTMGC`, `TTMGCA`, `TMGCAA`, and `MGCAAC` are in T. Note that Equation (3) is equivalent to the k-way AND constraint

$$y_i = \bigwedge_{j=1}^{k} x_{i_j} \tag{4}$$

Thus, the integer variables $y_i \in \{0, 1\}$ denote whether the i^{th} $(2k-1)$-mer can be written as a combination of overlapping k-mers which are present in T. Moreover, for each $1 \leq i \leq m$, the constant w_i represents the proportion of the i^{th} $(2k-1)$-mer among all methylation-centered $(2k-1)$-mers in the reference genome; therefore, the objective function (1) counts the number of methylation-centered $(2k-1)$-mers that can be generated using the selected set of k-mers, weighted by each $(2k-1)$-mer's relative frequency. Thus, maximizing it is equivalent to maximizing the number of methylated CpG sites that we retain in the data upon removing all occurrences of k-mers not included in T.

Using k-mer sets produced by solving the ILP reduces the number of methylated sites removed from the data significantly compared to random selection (Figure 5). Continuing to let p denote the percentage of modified k-mers that we retain in the data, we consider the following values of p throughout this study: $p \in \{5, 6, \ldots, 15, 20, 30, \ldots, 90, 100\}$. Note that, in total, we consider 20 values of p.

We emphasize once more that, in practice, this rather involved method for selecting k-mers would not be necessary, and it serves only to simulate a setting where sequencing experiments originally produce k-mer incomplete data. One potential scenario would be to perform high-coverage nanopore sequencing experiments on a small region of the genome, which would then be less costly to generate a methylation reference over using methods such as bisulfite sequencing or TAB-seq.

k-mer incomplete training of pre-existing methods

Equipped with sets of varying size which describe k-mers that we retained in our dataset, we now discuss our process for training two pre-existing methods for 5mC modification calling: Nanopolish, which is HMM-based [10], and DeepSignal, which is DNN-based [17]. Note that for both methods, we split the set of reads into 6 folds and ran the described training procedure 6 separate times, each time leaving out a different fold for validation purposes. Moreover, the entire procedure was run independently for the NA12878 and HX1 datasets.

Nanopolish. Training emission distributions for the Nanopolish HMM is based on event alignment, as described in [10]. First, the signal time series in each Nanopore read is segmented into events, which are then aligned to a reference genome. Then, each k-mer is associated with a list of events. Second, these lists of events are then used to update the emission distribution parameters for each k-mer. This process is repeated from the alignment step for five iterations, following the work of Simpson et al. [10].

Thus, in order to train parameters for modified k-mers, the primary pre-processing step is to edit the reference genome by changing `CG` dinucleotides that have been consistently called as methylated in our bisulfite sequencing reference data into `MG`. For a given value of p, we filtered out any methylated site that is covered by a k-mer that does not appear in the corresponding ILP-produced k-mer set. This process produced 20 modified reference genomes, each with a varying number of methylated sites. We then executed the standard Nanopolish training procedure for each of these 20 modified genomes.

DeepSignal. Next, we describe the process for training the DeepSignal DNN methylation caller on k-mer incomplete data. For a given CpG site in the reference genome and a read covering the site, DeepSignal extracts a feature vector containing nucleotide sequence information, signal summary statistics, and raw current values corresponding to a window centered on the CpG dinucleotide. We then annotated the feature vector with a binary methylation label obtained from bisulfite sequencing reference data. In cases where the CpG represents a methylated site, we then determined whether to filter out the example for each of the 20 values of p in the exact same fashion as we did for our k-mer incomplete HMM training procedure: checking if any k-mer covering the methylated site does not appear in the current k-mer set, and removing the example if so. Via this procedure, we obtained 20 filtered datasets, one for each of the different values of p.

There are two notable side effects of our filtering approach. First is that, since we are only concerned with removing methylated sites from the data, this procedure naturally introduces significant class imbalance in the training data, since unmethylated sites are untouched during k-mer filtering. To remedy this, we downsampled the negative (unmethylated) class. Second is that we remove significantly more methylated sites when we keep, for example, 5% of all methylated k-mers compared to 90%. Consequently, the training datasets produced by this method were much larger for higher values of p. To control for this, we uniformly limited training dataset size by downsampling all of the datasets to match the size of the dataset produced for the lowest value of p. For both the NA12878 and HX1 datasets, we ultimately obtained training datasets with approximately 1 million positive examples and 1 million negative examples for every value of p.

AmortizedHMM

AmortizedHMM extends the HMM method for methylation calling through use of a DNN. Following our HMM training procedure detailed in the previous section, we obtained sets of emission distribution parameters, with each set corresponding to a different value of p. In each set, some number of k-mers have had their parameters updated from the default values, with the count of such k-mers being larger the larger p is. We then let the set U_p be comprised of all of the triples (k_i, μ_i, σ_i), where k_i denotes a methylated k-mer whose emission distribution parameters μ_i, σ_i were updated when training on p-filtered data.

k-mer featurization. In our method, we train a feedforward deep neural network (FDNN) that can estimate the emission distribution parameters for previously unobserved k-mers. First, we describe the string featurization method we apply for each k-mer given as input to the FDNN. Given a k-mer, we extract a feature vector comprised of the following binary features:

1. At each position in the k-mer, a one-hot encoding of the nucleotide at that position. Again, $k = 6$, and we use a modified nucleotide alphabet `{A, C, G, T, M}`, so this represents 30 values.

2. For each of the first $k-1$ positions in the k-mer, a one-hot encoding of the dinucleotide starting at that position. With 5 choices of starting positions and $5^2 - 4 = 21$ possible dinucleotides (since `MA`, `MC`, `MT`, `MM` are not valid pairings), this corresponds to 105 features.

3. At each position in the k-mer, a boolean value representing whether a C or an M is present at that position. The motivation for including these features is that M is closely tied to C by way of being a modified cytosine, so we hypothesized that they may have similar effects on the nanopore current. This accounts for another 6 features, giving a feature vector of total length 141.

We additionally experimented with one-hot encoding trinucleotides in the k-mer, but this did not improve performance.

Model training. The AmortizedHMM takes the string featurization for a k-mer as input and outputs estimates of the emission distribution parameters (a mean and a standard deviation) for that k-mer. When training the AmortizedHMM, instead of using a mean-squared error loss as would be typical for regression problems such as ours, we minimize a symmetrized Kullback-Leibler (KL) divergence loss. This is because our goal is ultimately to emulate k-mer emission distributions, and simply using the numerical difference between parameter estimates may not accurately reflect how different our estimated distributions are from the emission distributions that would have been learned when training on k-mer complete data. In particular, we let $P \sim N(\mu, \sigma^2)$ be a Gaussian random variable with parameters estimated via the training procedure described in the previous section, and $\hat{P} \sim N(\hat{\mu}, \hat{\sigma}^2)$ be a Gaussian variable with parameters estimated by the AmortizedHMM. We define the symmetrized KL-divergence as

$$f(P, \hat{P}) = D_{KL}(P \parallel \hat{P}) + D_{KL}(\hat{P} \parallel P) \tag{5}$$

Since P and $\hat{P}$ are Gaussian, there exist closed-form solutions for the D_{KL} terms in Equation (5). Specifically,

$$D_{KL}(P \parallel \hat{P}) = \log \frac{\hat{\sigma}}{\sigma} + \frac{\sigma^2 + (\mu - \hat{\mu})^2}{2\hat{\sigma}^2} - \frac{1}{2} \tag{6}$$

$$D_{KL}(\hat{P} \parallel P) = \log \frac{\sigma}{\hat{\sigma}} + \frac{\hat{\sigma}^2 + (\hat{\mu} - \mu)^2}{2\sigma^2} - \frac{1}{2} \tag{7}$$

Hyperparameter search. Now, we detail the process by which we choose the architecture of the AmortizedHMM. We determine the number of hidden layers (denoted as d) in this network and the size of each hidden unit (denoted as h) via cross-validation. In particular, we divide the set U_{100} (methylated k-mers with parameters updated from training on k-mer complete data) into a training and a validation set according to an 80%/20% split. Then, we perform grid search over the hyperparameter values $d \in \{3, 4, 5, 6\}$ and $h \in \{16, 32, 64, 128\}$ (these candidate values were selected based on a preliminary analysis using the Nanopolish-provided emission distributions).

Then, for each value of p, we initialize a copy of the best-performing network architecture from this cross-validation procedure, and train it on U_p. The resulting network is then used to estimate the emission distribution parameters for k-mers that were not updated during the initial rounds of training (that is, the k-mers that do not appear in U_p). This provides a “complete" set of emission distribution parameters, some of which were learned during the standard HMM training step, while the rest were estimated using the AmortizedHMM. Given this set of parameters, we can then perform methylation calling in the usual fashion using Nanopolish.

Note that we use the same network architecture chosen by training and evaluating on distribution parameters from the $p = 100$ case for every value of p. This is because we wish to produce a comparison between our hybrid method and a pre-existing DNN methylation caller (DeepSignal). Since the network architecture for this caller was chosen based on its performance when $p = 100$ (the k-mer complete setting), our methods are comparable only if we uniformly use an architecture optimized for the $p = 100$ case as well. However, we note that since the size of the set of parameters U_p decreases as p decreases, it stands to reason that networks of lower capacity compared to the one selected via this cross-validation procedure could potentially be more suitable for small values of p.

Acknowledgments

This research used the Savio computational cluster resource provided by the Berkeley Research Computing program at the University of California, Berkeley (supported by the UC Berkeley Chancellor, Vice Chancellor for Research, and Chief Information Officer).

References

[1] Daniel Branton et al. “The potential and challenges of nanopore sequencing”. eng. In: *Nature biotechnology* 26.10 (Oct. 2008). nbt.1495[PII], pp. 1146–1153.

[2] Miten Jain et al. “The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community”. In: *Genome Biology* 17.1 (Nov. 2016), p. 239.

[3] Joshua Quick et al. “Real-time, portable genome sequencing for Ebola surveillance”. eng. In: *Nature* 530.7589 (Feb. 2016). PMC4817224[pmcid], pp. 228–232.

[4] Q. Gouil and A. Keniry. “Latest techniques to study DNA methylation”. In: *Essays Biochem* 63.6 (Dec. 2019), pp. 639–648.

[5] Franka J. Rang, Wigard P. Kloosterman, and Jeroen de Ridder. “From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy”. In: *Genome Biology* 19.1 (July 2018), p. 90.

[6] Andrew H. Laszlo et al. “Detection and mapping of 5-methylcytosine and 5-hydroxymethylcytosine with nanopore MspA”. In: *Proceedings of the National Academy of Sciences* 110.47 (2013), pp. 18904–18909.

[7] Jacob Schreiber et al. “Error rates for nanopore discrimination among cytosine, methylcytosine, and hydroxymethylcytosine along individual DNA strands”. In: *Proceedings of the National Academy of Sciences* 110.47 (2013), pp. 18910–18915.

[8] Qian Liu et al. “NanoMod: a computational tool to detect DNA modifications using Nanopore long-read sequencing data”. In: *BMC Genomics* 20.1 (Feb. 2019), p. 78.

[9] Arthur C. Rand et al. “Mapping DNA methylation with high-throughput nanopore sequencing”. In: *Nature Methods* 14.4 (Apr. 2017), pp. 411–413.

[10] Jared T. Simpson et al. “Detecting DNA cytosine methylation using nanopore sequencing”. In: *Nature Methods* 14.4 (Apr. 2017), pp. 407–410.

[11] Marcus Stoiber et al. “De novo Identification of DNA Modifications Enabled by Genome-Guided Nanopore Signal Processing”. In: *bioRxiv* (2017).

[12] Achim Breiling and Frank Lyko. “Epigenetic regulatory functions of DNA modifications: 5-methylcytosine and beyond”. In: *Epigenetics & Chromatin* 8.1 (July 2015), p. 24.

[13] S. Gonzalo. “Epigenetic alterations in aging”. In: *J Appl Physiol (1985)* 109.2 (Aug. 2010), pp. 586–597.

[14] Steve Horvath and Kenneth Raj. “DNA methylation-based biomarkers and the epigenetic clock theory of ageing”. In: *Nature Reviews Genetics* 19.6 (June 2018), pp. 371–384.

[15] Yong-Hwee Eddie Loh et al. “Comprehensive mapping of 5-hydroxymethylcytosine epigenetic dynamics in axon regeneration”. eng. In: *Epigenetics* 12.2 (Feb. 2017). PMC5330438[pmcid], pp. 77–92.

[16] Keith E. Szulwach et al. “5-hmC-mediated epigenetic dynamics during postnatal neurodevelopment and aging”. eng. In: *Nature Neuroscience* 14.12 (Oct. 2011). nn.2959[PII], pp. 1607–1616.

[17] Peng Ni et al. “DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning”. In: *Bioinformatics* 35.22 (Apr. 2019), pp. 4586–4595.

[18] Qian Liu et al. “Detection of DNA base modifications by deep recurrent neural network on Oxford Nanopore sequencing data”. In: *Nature Communications* 10.1 (June 2019), p. 2449.

[19] Yu He et al. “DeepH&M: Estimating single-CpG hydroxymethylation and methylation levels from enrichment and restriction enzyme sequencing methods”. In: *Science Advances* 6.27 (2020).

[20] Tomasz P. Jurkowski. “Chapter Thirteen - Technologies and applications for the assessment of 5-hydroxymethylcytosine”. In: *Epigenetics Methods*. Ed. by Trygve Tollefsbol. Vol. 18. Translational Epigenetics. Academic Press, 2020, pp. 261–278.

[21] Chongyuan Luo, Petra Hajkova, and Joseph R. Ecker. “Dynamic DNA methylation: In the right place at the right time”. eng. In: *Science (New York, N.Y.)* 361.6409 (Sept. 2018). 361/6409/1336[PII], pp. 1336–1340.

[22] Jared T. Simpson. *Nanopolish*. `https://github.com/jts/nanopolish/tree/r10`. 2019.

[23] Vladimír Boža, Broňa Brejová, and Tomáš Vinař. “DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads”. In: *PLOS ONE* 12.6 (June 2017), pp. 1–13.

[24] Neng Huang et al. “An attention-based neural network basecaller for Oxford Nanopore sequencing data”. In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2019, pp. 390–394. DOI: `10.1109/BIBM47256.2019.8983231`.

[25] Ryan R. Wick, Louise M. Judd, and Kathryn E. Holt. “Performance of neural network basecalling tools for Oxford Nanopore sequencing”. In: *Genome Biology* 20.1 (June 2019), p. 129.

[26] Zaka Wing-Sze Yuen et al. “Systematic benchmarking of tools for CpG methylation detection from Nanopore sequencing”. In: *bioRxiv* (2021).

[27] E.P. Consortium et al. “An integrated encyclopedia of DNA elements in the human genome”. In: *Nature* 489.7414 (Sept. 2012), pp. 57–74.

[28] Miten Jain et al. “Nanopore sequencing and assembly of a human genome with ultra-long reads”. In: *Nature Biotechnology* 36.4 (Apr. 2018), pp. 338–345.

[29] F. Krueger and S. R. Andrews. “Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications”. In: *Bioinformatics* 27.11 (June 2011), pp. 1571–1572.