University of California, Santa Cruz

Master of Science Capstone

Methods for Analysis of Nanopore Reread Data

Author:
John Vivian

Supervisors:
Dr. Mark Akeson
Dr. Kevin Karplus

A capstone submitted in fulfillment of the requirements for the degree of Master of Science in Bioinformatics

Biomolecular Engineering Department
Jack Baskin School of Engineering

September 1, 2015

Declaration of Authorship

I, John Vivian, declare that this capstone and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this capstone has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the capstone is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________________

Date: ______________________________________

“For me, it is far better to grasp the Universe as it really is than to persist in delusion, however satisfying and reassuring.”

Carl Sagan

Acknowledgements

I would like to acknowledge and thank my colleagues: Arthur Rand, Miten Jain, Stas Friedland, Logan Mulroney, Robin Abu-Shumays, Zachary Wescoe, Doug Marks, Joey Dahl and Andrew Smith.

I would also like to thank my Principal Investigator Dr. Mark Akeson who has supported my desire to be involved in research since I was a fledgling undergraduate and whose continued support has allowed me to develop my abilities as a bioinformatician. I would also like to thank Dr. Kevin Karplus for acting as an advisor for our group's statistical and computational questions.

Finally, I would like to thank my parents for all their love and support for the endeavors I have chosen to pursue in my life.

Abstract

Error rates for long-read sequencing are significantly higher than other leading next generation sequencing platforms. Although still useful, improving the accuracy of these platforms is critical to moving the world of genomics forward. In this study, we have engineered a system based on enzymatically controlled DNA translocation through a protein nanopore that attempts to improve the accuracy of epigenetic classification by 'rereading' a single molecule of DNA more than one time. Initial results appear to demonstrate that events containing more than one read are classified with higher accuracy than events that only have one read.

1 Introduction

Nanopore Sequencing has quickly risen to the forefront of the sequencing world for its potential to sequence the genome for less than \$1,000 [1] and with read-lengths comparable to PacBio's sequencing technique [2], which can average read-lengths between 10-20,000 bases. Illumina still maintains its position as the leader in the sequencing field for lowest per-base cost, highest throughput, and a low error rate [3]. Error rates for the Illumina platform typically fall below 0.4%, whereas PacBio averages error rates between 13-15% [4]. Long reads can be valuable depending on the application and this has allowed PacBio to retain a position in the market because the length of the reads generated are suitable for completing genome assemblies [3]. Oxford Nanopore Technologies (ONT) has developed a small and inexpensive sequencer (The MinION) that it hopes can compete in the market by offering similar read-lengths at a fractional cost [1]. Reports from those involved in the initial trial-runs of the MinION report error rates comparable or exceeding those of the PacBio platform [5]. Rereading a single DNA strand several times to reduce the error rate has been suggested as a possible solution to this problem [6]. Here, a single-molecule rereading mechanism for the nanopore system is presented in which the nanopore is used to classify strands with epigenetic modifications.

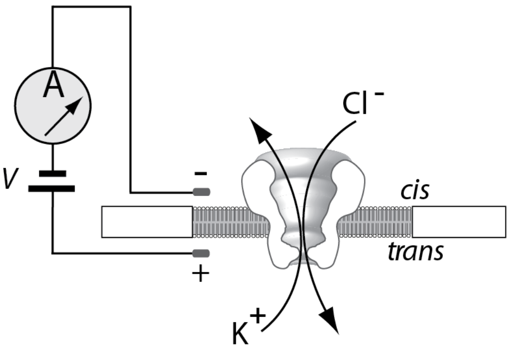


Figure 1: A classic nanopore diagram featuring a Mycobacterium Smegmatis Porin A (MspA) embedded into a lipid bilayer. A voltage-patch amplifier applies a constant voltage (180mV) across the membrane/pore so that as DNA translocates through the pore and impedes the flow of ions, a change in current is directly observed which corresponds to the nucleotide bases that are inside the limiting vestibule at that time.

2 Nanopore Background

Nanopore sequencing (Figure 1) was first demonstrated as a proof-of-concept in the late 1990's when single-stranded DNA/RNA was translocated through a biological pore embedded into a lipid membrane [7]. The next challenge was to overcome the problem that ssDNA/RNA passed through the nanopore too quickly (translocation rate of ˜300,000 bases per second) [8] to acquire any information about the DNA besides how the length and concentration of DNA affected the density of translocations observed. Methods were proposed to reduce the speed of translocation down to 100-1,000 bases per second [9], although substantial progress wasn't made until a decade later. UC Santa Cruz helped pioneer the now widely used technique of pairing an enzymatic motor with a DNA substrate in such a way as to control translocation and reduce the rate of translocation down to a median rate of 40 bases per second when using a ϕ29 DNA polymerase and an α-hemolysin nanopore [10]. The power of the nanopore system was further demonstrated when classifications were reported between epigenetic modifications (methylcytosine and hydroxymethylcytosine) on DNA substrates that had otherwise identical nucleotide bases [6]. That study concluded with a suggestion that given the ability to 'reread' a single molecule, the error rates associated with distinguishing between epigenetic modifications could be reduced (perhaps significantly in some cases).

The 'Break-Away' system (BA system) was devised by Arthur Rand at UC Santa Cruz and is depicted in Figure 2. This system allows for a single molecule of DNA to be translocated through a pore more than once without ever leaving the pore, guaranteeing that all 'rereads' will originate from the same molecule. The goal of constructing this system was to test the hypothesis that rereading single molecules could increase the accuracy of classifying epigenetic modifications. This required oligomer design that would be conducive to computational analysis as well as being capable of rereading consistently in order to generate enough data to make a statistically significant conclusion.

Break-Away Nanopore Reread System

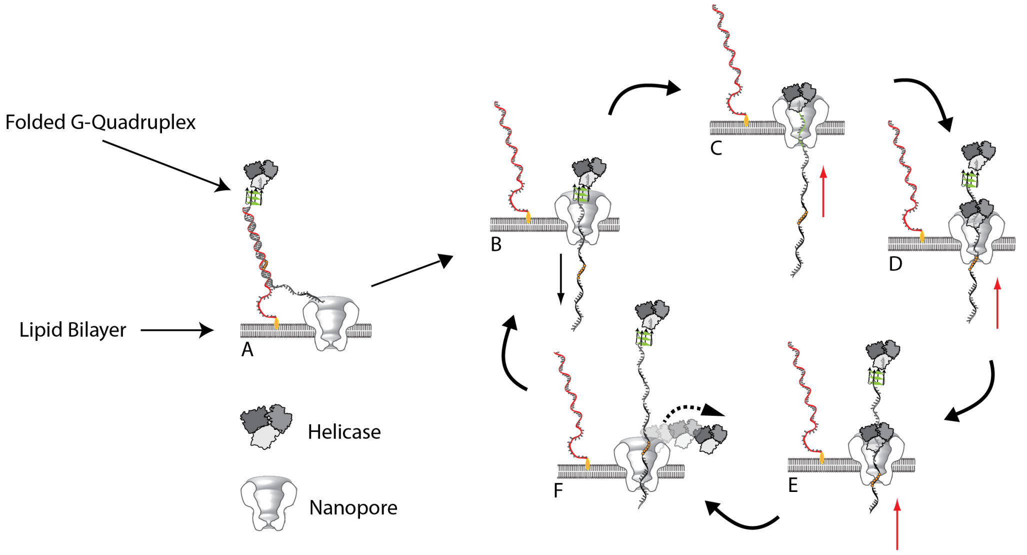


Figure 2: A: Substrate bound to the lipid bilayer via a cholesterol tag with a helicase (Hel308) bound and inactive at the other end. B: Electric potential causes the DNA duplex to unwind, leaving behind the cholesterol tether bound to the complement strand. C: Electric potential and the steady-state nature of the G-Quadruplex (GQ) causes it to unfold allowing the helicase to become enzymatically active in the 3' → 5' direction. D: As Hel308 translocates the DNA back up through the pore, the GQ will refold allowing an additional enzyme to become bound. E: Continued translocation. F: More than 3-4 abasic residues causes Hel308 to dissociate from the substrate (9 are used in the substrate). The DNA strand is then pulled back down and the cycle is able to repeat itself once more.

3 Substrate Design

Figure 3 shows the general design scheme required for the BA system to function. An important design consideration was an oligmer that had a (mostly) non-repeating set of 4-mers so

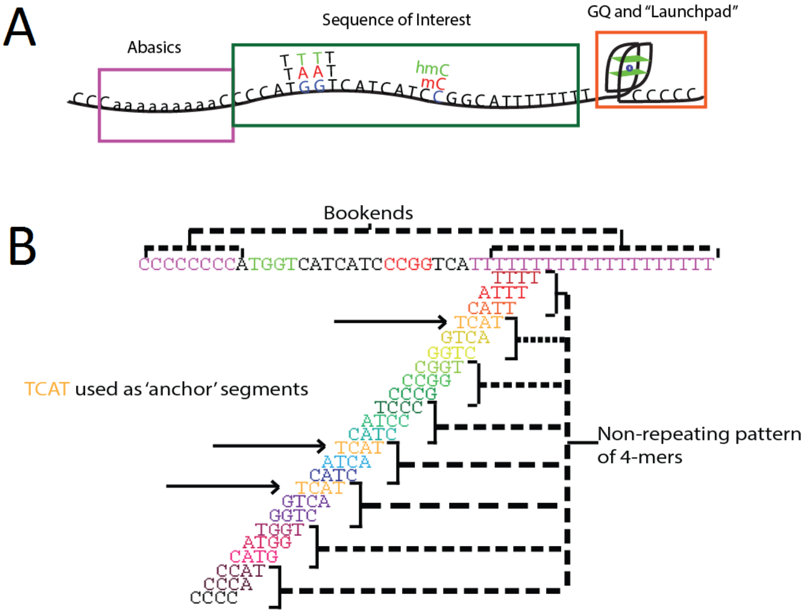


Figure 3: A: **Abasics**: abasic residues cause dissociation of Hel308 from the substrate and also act as a marker in the current trace. **Sequence of Interest**: the 'read' that contains the label and context region. **GQ and Launchpad**: the GQ sequence and 4-C launchpad that allow Hel308 to bind but not translocate through the GQ. B: 4-mers were picked such that each amplitude could be mapped to the strand by generating a unique current trace (Figure 4). The TCAT 'anchor' acts as a recognizable landmark, as it is the only repetition in the current trace, making it easier to map every 4-mer in the oligomer to its amplitude in the current trace.

the resultant current trace would be unique at every position. This feature allows every 4-mer in the oligomer to be mapped to its associated current in the trace as seen in Figure 4.

The other features in the oligomer are what allow the BA system to work (Figure 2). The 3' terminus of the reading strand contains 4 cytosines that act as the binding site for Hel308 [13]. The adjacent G-Quadruplex (GQ) folds into a tetrad coordinated by a potassium ion [14] and prevents Hel308 from translocating through the strand until the GQ has been unfolded. Finally, the abasic residues located at the opposite (5') end of the strand cause Hel308 to dissociate from the substrate which 'resets' the system given that another helicase has bound to the 3' end behind a refolded GQ. These abasic residues also serve as a useful tool when doing data analysis as the abasic creates a characteristic spike in the current trace when the region passes through the limiting vestibule of the pore. An example of an event with 2 rereads (for a total of 3 reads) is shown in Figure 5, demonstrating the capability of the BA system to generate multiple reads from a single molecular substrate.

4 Data Analysis Methods

Nanopore data used for this analysis was collected as per the Experimental Methods (Section 4).

4.1 Designing the Hidden Markov Model

Hidden Markov Models (HMMs) have been used in many computational applications including statistical modeling, database searching, and multiple sequence alignment of protein families

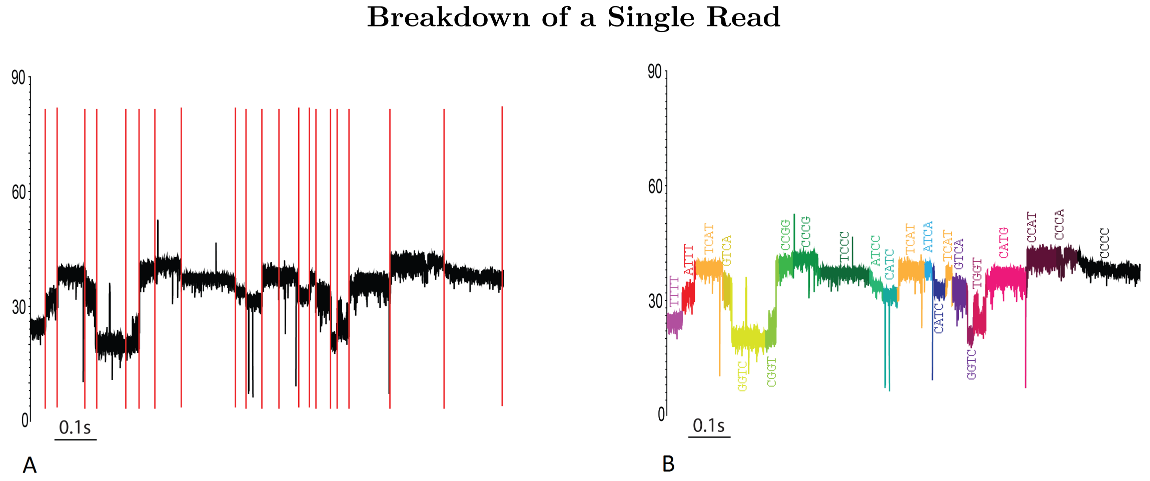


Figure 4: A: Current trace of a complete ‘read’ of the substrate. The red lines indicate demarcation of distinct amplitudes associated with a different grouping of 4 nucleotide bases. B: The same event has been color-coded and each associated 4-mer is paired alongside the appropriate segment.

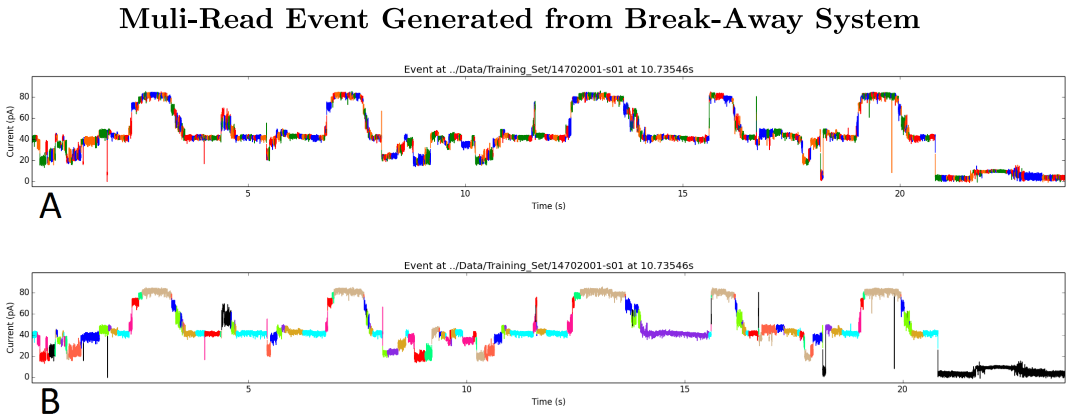


Figure 5: A: A current trace of a single event with three distinct reads. The coloring scheme is a 4-color cycle that indicates how the segmenter [11] divided up the current trace. The high amplitude states (80pA) are the result of abasic residues passing through the pore. B: The same trace as A, but colored by HMM state [12] (black segments represent insert states [noise/off-pathway segments]). This figure helps give a visual representation to how a set of observations are aligned to the HMM.

and domains [15]. A standard profile HMM [16] was used as a reference point, with an added modular structure (conducive to construction of the complete HMM) and a backslip pathway meant to model molecular phenomena observed by Hel308 during translocation. Hel308 can backslip one or more nucleotides [17] causing a repeat of segments to be observed. Hel308 can also dissociate from the substrate at any time, and if another helicase happens to be bound directly behind it (a common occurrence), the DNA will ‘fall back’ the length of one helicase (∼12 nts, 6 segments).

In order to account for the two distinct ways the system can jump back to a prior state (backslips and helicase dissociations), probabilities were selected such that backslips of 1-5 nucleotides carry an exponential decay in transition probability and yet the probability of a backslip of 1 is equivalent to a backslip of 6. This way, helicase dissociations are not modeled as just an extension of the backslip pathway. Let P_0 be the edge probability representing $\mathbf{M} \rightarrow B_1$ ($\mathbf{S}_3$ in Figure 6), and a transition B_i to B_{i+1} is P_i, then

1. Probability of 1 backslip = Pr(Length=1) = $P_0(1 - P_1)$
2. Probability of more than 1 backslip = Pr(Length > 1) = $P_0 * P_1$

These statements can then form the following equality:

$$\frac{\Pr(\text{Length=1})}{\Pr(\text{Length} > 1)} = \frac{1 - P_i}{P_i}$$

$$\text{Solving for} P_i :$$

$$P_i = \frac{(1 - P_i)\Pr(\text{Length} > i)}{\Pr(\text{Length} = i)}$$

$$P_i = \frac{\Pr(\text{Length} > i)}{\Pr(\text{Length} > i - 1)}$$

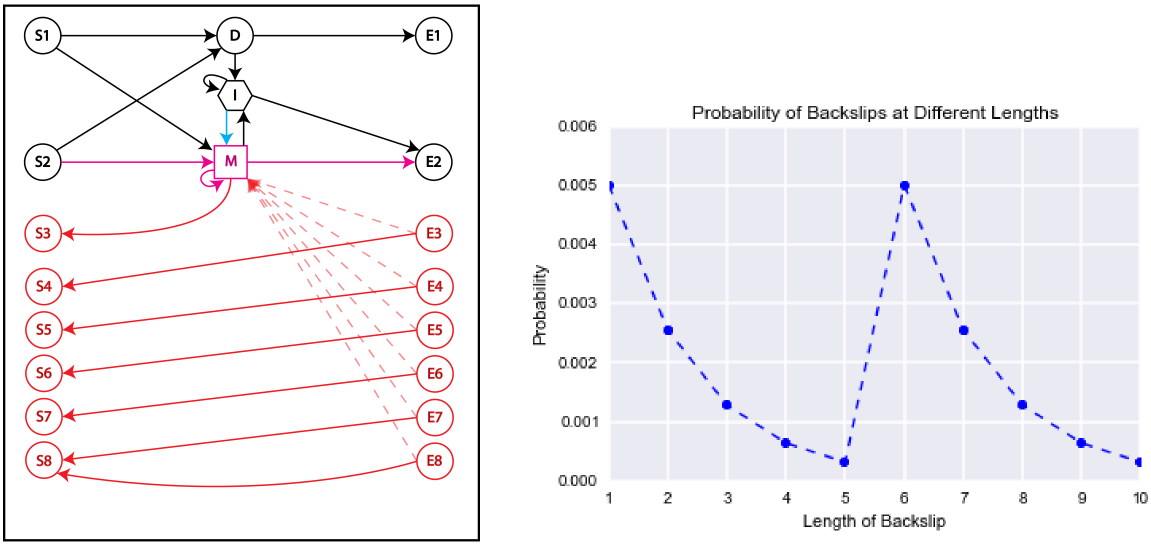


Figure 6: [Left] A modular 'board' in the HMM representative of a specific current mean (segment) in a nanopore trace. Circular nodes represent silent states (non-emitting states), **D** is the delete state (missing segment), **I** is the insert state (off-pathway segment/noise spikes), **M** is the match state (aligning to a segment of the same mean), and the red states represent the backslip pathway. [Right] A graph showing the probability of a backslip at differing lengths given transition probabilities chosen in Table 1.

The HMM had to be designed to allow for classification and the 'forks' built into the structure of the model allowed the differences that exist between the different strands to align properly. Figure 7 highlights the differences in amplitude that exist between the three epigenetic modifications in the CCpGG context of the substrate. Taking these differences into account led to the complete model seen in Figure 8, which allow the different substrates to align to their respective track in the fork. The model was written using the YAHMM [18] repository and the code that generates the fork structure was taken from the author's repository on automation [19].

Transition Edge	Transition Probability	Calculated Probability Through the Model	
P_0 (Start)	0.02		
P_1	0.75	Pr(Len=1)	**0.005**
P_2	0.83	Pr(Len=2)	0.00255
P_3	0.897	Pr(Len=3)	0.00128
P_4	0.943	Pr(Len=4)	0.0006365
P_5	0.970	Pr(Len=5)	0.0003159
P_6	0.500	Pr(Len=6)	**0.005**

Table 1: This table contains the edge transition probabilities that appear in the backslip pathway (graphically modeled in Figure 6[R]) and the probabilities of different backslip lengths as generated by the model. P_0 is the first edge connecting **M** to **S3** (Backslip$_1$), and P_i is the transition edge from backslip state i to $i+1$.

Context Region and Heatmap of Epigenetic Modifications

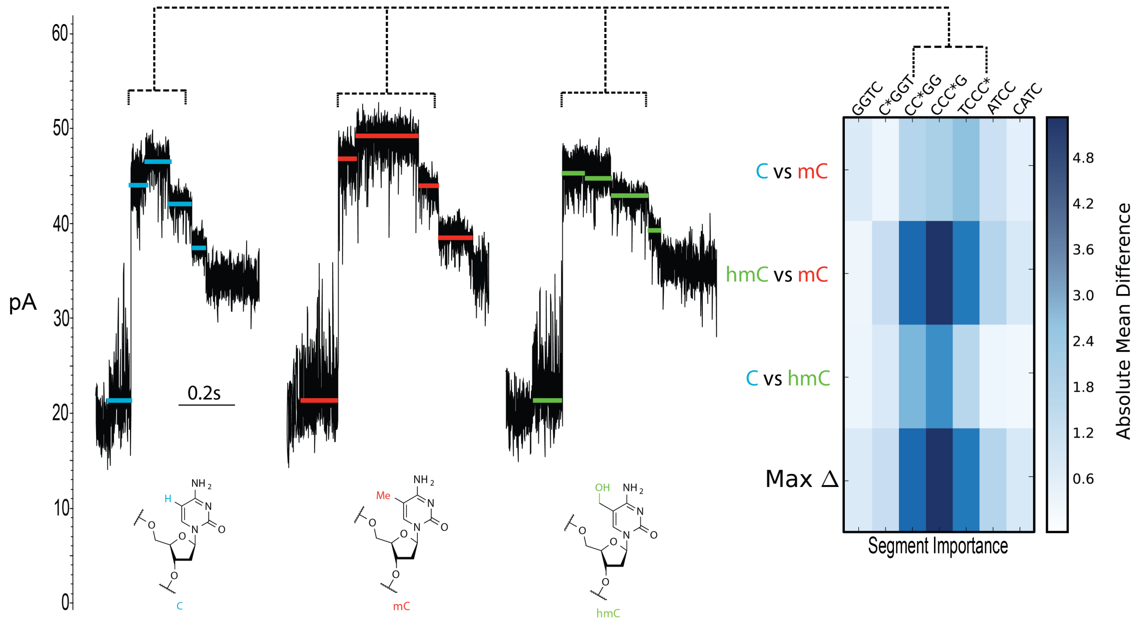


Figure 7: A subsection of 3 current traces in the context region highlight how the different epigenetic modifications affect the amplitude of the current. A heatmap was constructed to highlight the differences (pA) that exist between the 4-mers in the modified context region.

4.2 The CHUNK Method

4.2.1 Partitioning Regions of Interest from an Event

The forward-backward algorithm returns an $m \times n$ emission (ems) matrix, composed of observations (segment means)$[m]$ by the emission states $[n]$ contained within the model [20]. Partitioning of an event into discrete groupings of context and label regions requires the retrieval of probabilities associated with the match states that exist within the context and label 'fork' of the model (see Figure 9). Code snippets for the steps listed below are located in Section 5.2.1.

1. Use the forward-backward algorithm to return an emissions matrix.

2. Find the column indices associated with the match states.

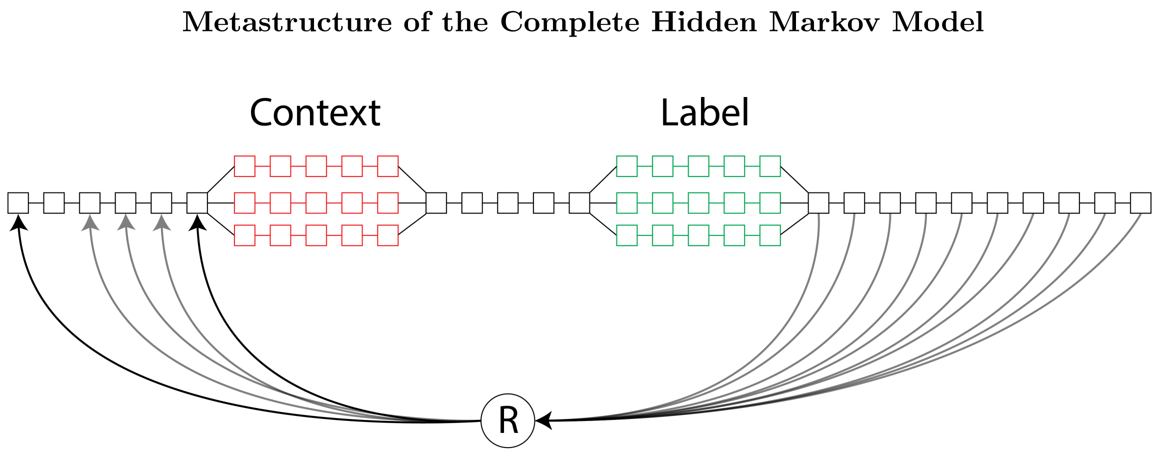
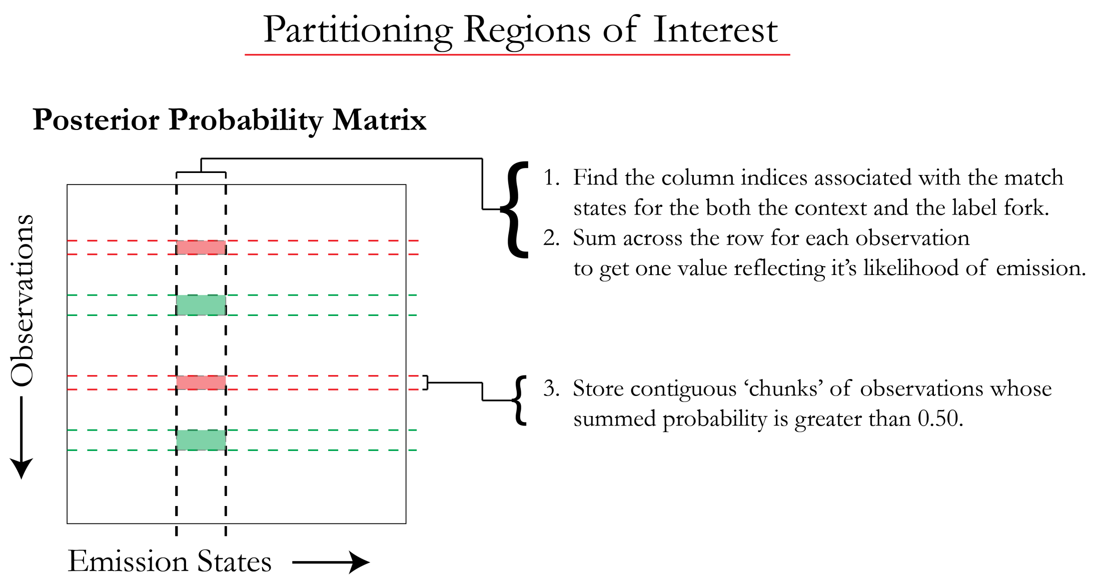


Figure 8: The metastructure of the complete profile HMM used in this analysis. Each box corresponds to the modular board depicted in Figure 6. The forks for the context and the label region allow each substrate to be aligned to their own track in the fork based on their respective amplitudes (Figure 7). The linear portions of the structure represent the consensus in amplitude that exists between the strands omitting the 10 states perturbed by the context and label region. In order to handle alignment of the molecules that are reread, transition edges exist after the label that transition to one of the beginning states in the model.


Partitioning Regions of Interest
Posterior Probability Matrix
Observations
Emission States

1. Find the column indices associated with the match states for the both the context and the label fork.
2. Sum across the row for each observation to get one value reflecting it's likelihood of emission.
3. Store contiguous 'chunks' of observations whose summed probability is greater than 0.50.

Figure 9: The posterior probability matrix, or emissions (ems) matrix, contains a set of all observed segment means and their probability of emission for every state in the model (i.e. each row sums to 1 across the entire matrix). The match states associated with the context or label fork are selected from the matrix, then summed by row for each observation. Contiguous blocks (chunks) of observations are stored if their respective emission by the match states is greater than 0.50.

3. Create a flattened array by summing across the row for each observation. This gives each observation a single value reflecting the likelihood that it was emitted from one of the match states in the fork.

4. Iterate through every observation, storing the mean and summed emission probability if the probability is greater than 0.50 (a reasonable estimate). Any time the probability falls below 0.50 for an observation it is used as a delimiter to separate the previous 'chunk' of means into a group.

5. Repeat with the label region to obtain label chunks.

4.2.2 Obtaining a Score for each Chunk

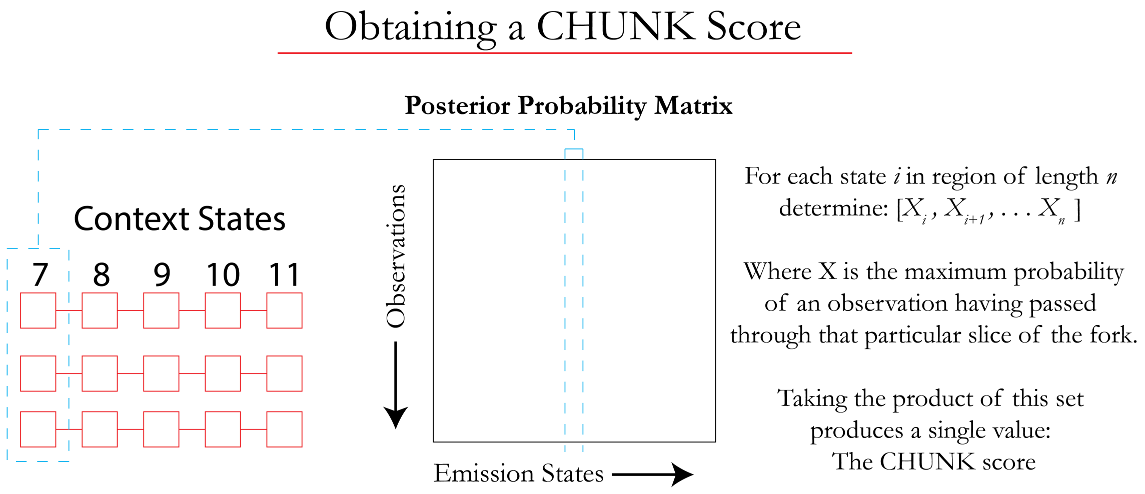

Obtaining a CHUNK Score
Posterior Probability Matrix
Context States
7
8
9
10
11
Observations
Emission States
For each state i in region of length n determine: $[X_i, X_{i+1}, \dots X_n]$
Where X is the maximum probability of an observation having passed through that particular slice of the fork.
Taking the product of this set produces a single value: The CHUNK score

Figure 10: Each 'slice' of the fork is used to guarantee that the chunk is representative of the complete context region. Taking the maximum probability of each slice appearing in the chunk produces a vector of 5 values that are then combined into one value known as the 'chunk score'. These scores are used to rate the level of confidence associated with the chunk.

Once one or more chunks have been obtained from the data set, there needs to be a step to confirm that the chunk is of good quality. For every group of observations, $[X_i, X_{i+1}, \cdots X_n]$ is determined, where X is the maximum probability of an observation having passed through a slice of the fork representing one of the five states in the context region. This ensures that every part of the fork is represented and selects against groups that only happened to have matched to some portion of the fork. Once the list $[X_i, X_{i+1}, \cdots X_n]$ has been formed, the 5 probabilities are merged into one value by taking the product of the list $(\Pi_{i=1}^{5} X_i)$. Code snippets for the steps listed below are located in Section 5.2.2.

1. Use the forward-backward algorithm to return an emissions matrix.

2. Iterate through each column of match states in the fork.

3. Take the maximum probability of that slice's emission probability within in the observations that make up the chunk.

4. Obtain one number for each column in the fork (5 total numbers).

5. Reduce this vector down to a single value by taking the product of the vector.

4.3 Results

The results section of this paper used data acquired up until 12/15/2014 and will serve solely as cross-training data for the publication submission of this project.

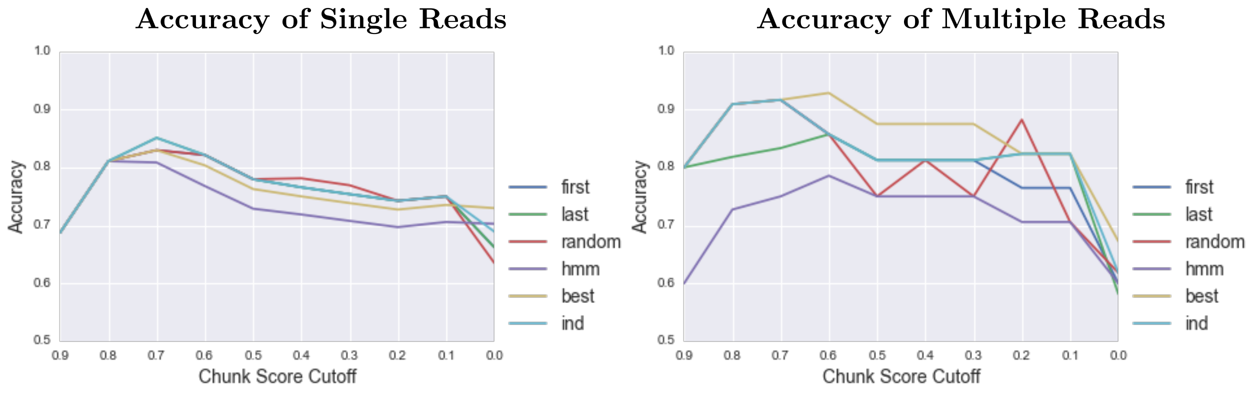


Figure 11: Events with only one context chunk (single reads) were compared with those that had more than one. As predicted, **best** and **ind** outperform the other methods given multiple reads to work with. The sample size for this set was small (16 multi-read events, 59 single-read events at chunk cutoff of 0.5), which means these results should be considered tentative until a larger repository of multi-read events can be analyzed.

The data set was divided up into test data and a cross-training set that was used to train the HMM via Baum-Welch training [21] and adjustments to the initial profiles. Three separate linear HMMs were constructed (one for each substrate) to train the context and label regions, whose values were used to improve the complete model. Each event in the data set was assigned the correct label by a human (Barcode) which was used to confirm the model was aligning correctly.

The Chunk Method was utilized in several different ways to call accuracy to ensure systematic bias was avoided. In an event with one or more chunks, the chunk chosen to make the accuracy call is either the **first, last, best** (by chunk score), or a **random** selection. Given long multi-read events, one would assume the **best** method would outperform the others as the highest confidence chunk is used to make the accuracy call. For single read events they will all perform the same since the chunk chosen to make the call will be the same one. The other two methods used were standard posterior decoding (**hmm** [22]), which uses sum-over-paths to determine the correct call, and independent consensus (**ind**), which is a method for refining accuracy given multiple chunk scores and assuming independence of the reads. A derivation of the independent consensus method is found in Section 5.1. Figure S-12 (Supplement, Section 5.3) shows the accuracy of the label as confirmed by the Barcode, which mean the label is almost never miscalled by the HMM. Accuracies for the context are shown in the adjacent figure (S-13). A confusion matrix was constructed to show how each context was being called / mis-called by the HMM (Figure S-14). C is miscalled as mC and hmC equally, whereas both mC and hmC have a bias of being called as cytosine when they are not.

Finally, events were separated into two groups: those with only one chunk for the context and those with more than one (multi-read events). The tentative results for this comparison can be seen in Figure 11. As one would expect, the methods perform nearly identically for single reads and then begin to diverge when the accuracy is judged on events with more than one chunk. As longer and more multi-read events are added to the dataset, the differences should become even more pronounced. Given more data to work, reads could possibly be shown to be independent of one another. This means given the appropriate reread system, high quality scores may be associated with classification and base-calling. This system could also potentially be adopted to remove indel and SNP errors by consensus coverage of a single molecule.

5 Experimental Methods

5.1 Basic Preparation

5.1.1 Creating an Aperture

1. Affix a tungsten needle and immerse one end in a bath of sodium hydroxide
2. Apply a voltage of 20V across the bath which will begin to chemically etch the tungsten needle. Turn off once the end of the needle is molecularly thin
3. Cut a small length (∼8-10cm) of heat resistant tubing and place on the end of a smaller piece of shrink-wrap tubing
4. Have the needle protrude into the shrink-wrap tubing and slowly pull through a heating element (similar to a blow dryer) to affix the shrink wrap tubing around the point of the needle
5. Pull out the needle in one motion and examine under a microscope, using a 15-25 micron thick wire to gauge how large the opening is
6. Insert into manufactured teflon piece using a bent paperclip and cut to appropriate length

5.1.2 Making Lipid Solution

1. Remove ampule from -20C freezer
2. Use 5ml pipette to add 2.5ml of chloroform to ampule
3. Prime vials by pouring in chloroform and dumping out waste
4. Parafilm the top of the unused portion of the ampule to store
5. Shake ampule, avoid touching parafilm to liquid
6. Parafilm aliquots and then store in -4C freezer.

5.1.3 'Coat Tubes'

1. Add 20 μl of lipid solution to a standard glass test tube
2. Place each vial in a vacuum-desiccator for a minimum of 20 minutes.
3. When ready to use, add 100μl of hexane to lipid coating and cap to prevent evaporation.

5.1.4 'Lipid Ball' Slides

1. Wipe down and clean a culture dish
2. Affix 4-5 slides to the interior of the culture dish
3. Place ∼4 drops (1-5 μl) of solution per slide, reserving one slide for later use (the slide will hold hexadecene).

5.2 Preparation for Nanopore Experiments

5.2.1 Sterilization

1. Place teflon/aperture piece inside a beaker and add 50 ml of 10% nitric acid
2. Heat for 8-10 minutes on a hotplate such that it boils
3. After boiling, pour acid into sodium bicarbonate bath.
4. Rinse twice with 50ml of H_2O
5. Block *cis*-well of the teflon piece and attach a syringe with tubing to the *trans* well.
6. Pull plunger of syringe to draw liquid through aperture, washing once with H_2O and once with ethanol
7. Cover in kimwipe and place in desiccator

5.2.2 Buffer

1. 0.3M KCl in 50mL
2. 10 μM HEPES in 50ml
3. 1M ATP in 50ml
4. Add 2M KOH ($\sim$260μl) to bring pH to 8

5.2.3 Station Setup

1. Cool temperature of peltier heater/cooler (199 KΩ)
2. Once cooled, insert aperture into peltier block and seat firmly.
3. Set temperature of peltier to 10.3 KΩ (23°C)
4. Rinse electrodes with dH_2O and insert into both the cis and trans side
5. Attach air syringe to trans well
6. Add 1-2 μl of lipid coating to aperture and pull through with an air syringe

In between applications of the lipid coating:

1. (Between 1-2): Rinse perfusion syringes fill with buffer
2. (Between 2-3): Get icebox and grab samples from -20°C freezer.
3. (Between 3-4): Roll lipid ball (described below).
   (a) Add hexadecene (1μl) to clean slide
   (b) Brush hexadecene over lipid drop with small paintbrush (cut off all but one bristle)
   (c) Carefully push lipid into a line, dipping the brush in hexadecene when necessary
   (d) Flatten, push, and reform until lipid ball is clear and homogenous.

5.3 Acquisition of a Single Nanopore Channel

1. Attach buffer syringe to trans well and slowly fill the U-tube with buffer (50μl)
2. Check for overload on Axon patch to ensure current is freely passing through the two wells
3. 'Paint' lipid ball around the aperture (avoid clogging; watch if overload indicator turns off)
4. Use a pipette tip ($\sim$5μl) to form an air bubble on the surface of the aperture by depressing plunger and dragging it over the aperture, then releasing pluger slowly to draw the bubble back into the pipette
5. Apply 180mV across the *cis* and *trans* well via the Axon patch
6. Add pore protein to *cis*-well and mix
7. Slide perfusion tubing close to the aperture and wait until a single nanopore spontaneously inserts itself into the membrane then use Clampex to record a control file
   (a) MspA porin will have $\sim$110-115 pA positive voltage and $\sim$-135 negative voltage.
   (b) The channel will 'gate' (drop/fluctuate near zero) when the voltage is reversed
   (c) RMS (noise): $\sim$0.6 (with faraday cage lowered)
8. Once a single channel is inserted, immediately perfuse the *cis*-well by lowering the perfusion tubing until the ends enter the meniscus. While maintaining the *cis*-well's volume, proceed to simultaneously depress and extend both the empty air syringe and the syringe filled with buffer. Cycle the entire 50mls of buffer through the *cis*-well to avoid additional nanopore insertions during the course of the experiment.
   (a) If more than one channel is inserted, reform bilayer and try again. Conversely, add pore/mix and reform bilayer as often as necessary until nanopore insertions are observed (optimally at a rate of 1 per minute to avoid multiple insertions)
9. To start an experiment:
   (a) Start and stop recording a control file (file 0) and set voltage to 0.
   (b) Add 1mM EDTA, 1mM DTT, 10mM $MgCl_2$, 4nM substrate, 300nM Hel308; mix well
   (c) Secure cover over aperture and screw in place (do not let cover touch meniscus or the bilayer will break)
   (d) Start recording (file 1) and flip the voltage back to positive to allow for translocation events to begin

6 Supplement

6.1 Independent Consensus Method

6.1.1 Derivation Assuming Independence

Independence can be defined several ways:

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(A)\Pr(B)}{\Pr(B)} = \Pr(A)$$

$$\text{or: } \Pr(A \cap B) = \Pr(A)\Pr(B)$$

To simplify, independence states that two events' probabilities have no association with one another, i.e. the outcome of one event will not affect the other. An example would be two fair coins flipped consecutively; whatever the result of the first coin, it has no bearing on the second coin's probabilities (which remain 50/50).

By claiming individual rereads in a nanopore system are independent, their respective likelihoods can be combined by taking the product of their errors. Given two accuracies, 95% and 99%, the product of their complements ($1-p$, the error) yields: $(1-.95)(1-.99) = 0.005$ which is the combined error. Taking the complement of *that* value produces the combined accuracy: $1 - 0.005 = 0.995$.

The equation for independent consensus used in our analysis is a generalization of the above principle:

$$\Pr(IC) = \Big[1 - \Pi_{i=1}^{n}(1-p_i)\Big]$$

Given two probability vectors, they would be combined as follows:

1. {'C': 0.05, 'mC': 0.1, 'hmC': 0.85 }
2. {'C': 0.01, 'mC': 0.19, 'hmC': 0.80}

$$\begin{aligned}
\Pr(IC_C) &= \Big[1-(1-0.05)(1-0.01)\Big] = \Big[1-(0.95)(0.99)\Big] = 0.0595\\
\Pr(IC_{mC}) &= \Big[1-(1-0.1)(1-0.19)\Big] = \Big[1-(0.9)(0.81)\Big] = 0.271\\
\Pr(IC_{hmC}) &= \Big[1-(1-0.85)(1-0.80)\Big] = \Big[1-(0.15)(0.2)\Big] = 0.97\\
\text{new vector } &= \{\text{'C': 0.0595, 'mC': 0.271, 'hmC': 0.97}\}\\
\text{normalized } &= \{\text{'C': 0.046, 'mC': 0.208, 'hmC': 0.746}\}
\end{aligned}$$

6.1.2 Code Implementation (Python)

```
#
# contexts is a dictionary that contains the chunk vector (sum over paths for the
    context)
#

contexts = [ x for x in contexts if x[0] >= cscore ] # Selects for all chunks above a
    certain cutoff

C_prod, mC_prod, hmC_prod = 1, 1, 1 # Rolling product

for chunk_vector in contexts: # Probability vector
        C_prod *= (1 - chunk_vector[’C’])
        mC_prod *= (1 - chunk_vector[’mC’])
        hmC_prod *= (1 - chunk_vector[’hmC’])

combined_vector = [ 1-C_prod, 1-mC_prod, 1-hmC_prod ]

normalized_vector = [ c/sum(combined_vector) for c in combined_vector ] # Sums to 1
```

```
context_call = normalized_vector.index( max( normalized_vector ) )

# Index position defines call. 0=’C’, 1=’mC’, 2=’hmC’
```

The **context_call** is then compared to the Barcode (human call) to determine if correctly called.

6.2 Code Snippets

All code, models, pictures, and data used in this project are available at: https://github.com/jvivian/Helicase-Reread-Project

6.2.1 Partitioning Regions of Interest

1. Find the column indices in the ems matrix associated with the match states in both the context and label fork.

```
indices = {state.name: i for i, state in enumerate(model.states)}
C_temp = [x for x in indices.keys() if ’(C)’ in x and ’I’ not in x and ’b’ not
    in x and ’D’ not in x]
C_fork = [x for x in C_temp if int(x.split(’:’)[1]) in xrange(7,12)]
C_tag = [x for x in C_temp if int(x.split(’:’)[1]) in xrange(17, 22)]
## Repeat for each context (not shown for brevity), then:
all_contexts = C_fork + mC_fork + hmC_fork
all_labels = C_tag + mC_tag + hmC_tag
## Store Indices
C_all = np.array(map(indices.__getitem__, all_contexts))
L_all = np.array(map(indices.__getitem__, all_labels))
```

2. Create a flattened array by summing across the row for each observation. This gives each observation a single value reflecting the likelihood that it was emitted from one of match states in the fork.

```
pC_all = np.exp( ems[ :, C_all ] ).sum( axis=1 ) # Context
pL_all = np.exp( ems[:, L_all ] ).sum( axis=1 ) # Label
```

3. Iterate through every observation, storing the mean and summed emission probability if the probability is greater than 0.50 (a reasonable estimate). Anytime the probability falls below 0.50 for an observation, it is used as a delimiter to separate the previous ‘chunk’ of means into a group.

```
## Partition Events given emission matrix
    contexts, labels = [], []
    temp_c, temp_l = [], []

    for i in xrange(len(pC_all)):
        if pC_all[i] > 0.5:
            temp_c.append( (round(means[i],4), i) )
        if pC_all[i] <= 0.5:
            if temp_c:
                temp = zip( *temp_c )
                contexts.append( [temp[0], temp[1]] )
                temp_c = []

        if pL_all[i] > 0.5:
            temp_l.append( (round(means[i],4), i) )
        if pL_all[i] <= 0.5:
            if temp_l:
                temp = zip( *temp_l )
                labels.append( [temp[0], temp[1]] )
                temp_l = []
```

6.2.2 CHUNK Score

1. Get the index values for each section of the context.

```
c_dict, l_dict = OrderedDict(), OrderedDictfor i in xrange(7,12):
  c_dict[i] = np.array(map(indices.__getitem__, [x for x in all_contexts if
      ':'+str(i) in x]))
for i in xrange(17, 22):
  l_dict[i] = np.array(map( indices.__getitem__, [x for x in all_labels if
      ':'+str(i) in x]))
```

2. Obtain a probability score for each section of the fork, i.e. $[X_i, X_{i+1}, \cdots X_n]$

```
p_dict = OrderedDict()
pscore = []
context_final = []
weights = [ 1.0/9, 2.0/9, 1.0/3, 2.0/9, 1.0/9 ] # Context Weights
for c in contexts:
  temp_ems = ems[ c[1], : ]
  for i in xrange(7,12):
    p_dict[i] = np.max( np.exp( temp_ems[:, c_dict[i] ]).sum( axis=1 ) )
```

3. Combine $[X_i, X_{i+1}, \cdots X_n]$ into a single value $\rho \in$[0,1]

```
## Combine P_scores into a single score
pscore = [ p_dict[x] for x in p_dict ]
pscore = [ a*b for a,b in izip(pscore, weights) ]
pscore = sum(pscore)
```

4. This process is then repeated for the label region.

6.3 Additional Figures

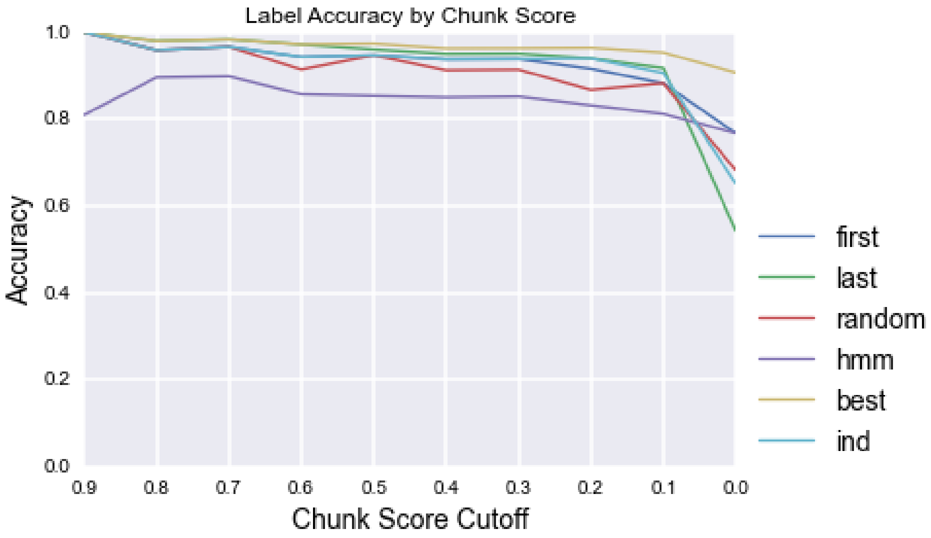


Figure 12: Accuracy of the label as compared to the human-called Barcode. This demonstrates that the labels were picked effectively as the HMM has almost no difficulty correctly calling the label.

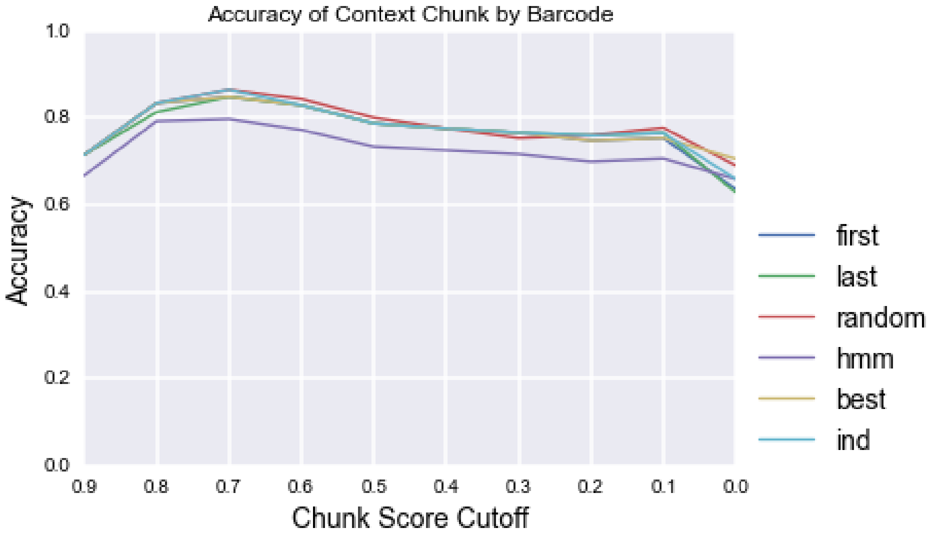


Figure 13: Accuracy of the contexts as compared to the Barcode. At a chunk cutoff of 0.7, there are 59 events that are called with an accuracy of about 85%.

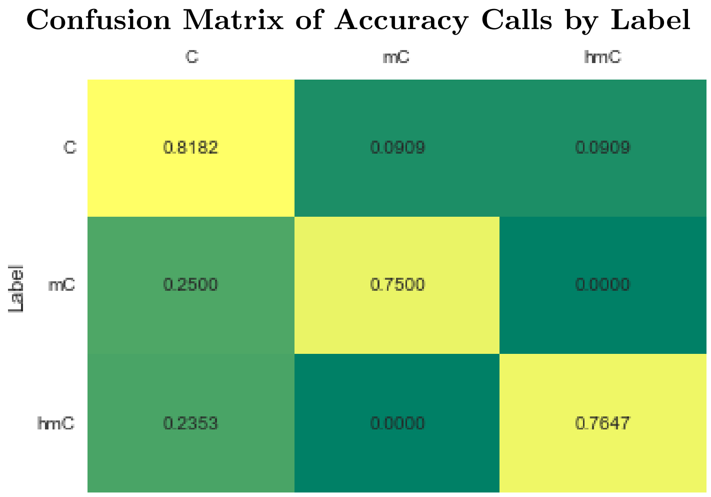


Figure 14: A confusion matrix of the context calls for most of the data set (which is why the diagonal accuracy is between 0.75-0.81). There is a bias for mC and hmC to be classified as C.

References

[1] Michael Eisenstein. Oxford Nanopore announcement sets sequencing sector abuzz. *Nature biotechnology*, 30(4):295–6, April 2012.

[2] Kevin J Travers, Chen-Shan Chin, David R Rank, John S Eid, and Stephen W Turner. A flexible and efficient template format for circular consensus sequencing and SNP detection. *Nucleic acids research*, 38(15):e159, August 2010.

[3] Erwin L. van Dijk, Hélène Auger, Yan Jaszczyszyn, and Claude Thermes. Ten years of next-generation sequencing technology. *Trends in Genetics*, 30(9):418–426, August 2014.

[4] Michael A Quail, Miriam Smith, Paul Coupland, Thomas D Otto, Simon R Harris, Thomas R Connor, Anna Bertoni, Harold P Swerdlow, and Yong Gu. A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC genomics*, 13(1):341, January 2012.

[5] Mick Watson. Thoughts on oxford nanopore's minion mobile dna sequencer. `https://biomickwatson.wordpress.com/2014/09/07/thoughts-on-oxford-nanopores-minion-mobile-dna-sequencer/`, 2014.

[6] Jacob Schreiber, Zachary L Wescoe, Robin Abu-Shumays, John T Vivian, Baldandorj Baatar, Kevin Karplus, and Mark Akeson. Error rates for nanopore discrimination among cytosine, methylcytosine, and hydroxymethylcytosine along individual DNA strands. *Proceedings of the National Academy of Sciences of the United States of America*, October 2013.

[7] J. J. Kasianowicz, E. Brandin, D. Branton, and D. W. Deamer. Characterization of individual polynucleotide molecules using a membrane channel. *Proceedings of the National Academy of Sciences*, 93(24):13770–13773, November 1996.

[8] A. Meller, L. Nivon, E. Brandin, J. Golovchenko, and D. Branton. Rapid nanopore discrimination between single polynucleotide molecules. *Proceedings of the National Academy of Sciences*, 97(3):1079–1084, February 2000.

[9] John Kasianowicz George Church, David Deamer, Daniel Branton, Richard Baldarelli. Measuring physical properties, August 1998.

[10] Gerald M Cherf, Kate R Lieberman, Hytham Rashid, Christopher E Lam, Kevin Karplus, and Mark Akeson. Automated forward and reverse ratcheting of DNA in a nanopore at 5-Åprecision. *Nature biotechnology*, 30(4):344–8, April 2012.

[11] J Schrieber and K Karplus. Segmentation of noisy signals generated from a nanopore. *Bioinformatics*, 2014.

[12] Jacob Schreiber. Pypore. `https://github.com/jmschrei/PyPore`, 2014.

[13] Colin P Guy and Edward L Bolt. Archaeal Hel308 helicase targets replication forks in vivo and in vitro and unwinds lagging strands. *Nucleic acids research*, 33(11):3678–90, January 2005.

[14] Nancy H Campbell and Stephen Neidle. G-quadruplexes and metal ions. *Metal ions in life sciences*, 10:119–34, January 2012.

[15] A Krogh, M Brown, I S Mian, K Sjölander, and D Haussler. Hidden Markov models in computational biology. Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–31, February 1994.

[16] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, October 1998.

[17] Rudolf K F Beran, Michael M Bruno, Heath a Bowers, Eckhard Jankowsky, and Anna Marie Pyle. Robust translocation along a molecular monorail: the NS3 helicase from hepatitis C virus traverses unusually large disruptions in its track. *Journal of molecular biology*, 358(4):974–82, May 2006.

[18] Jacob Schreiber. Yet another hidden markov model. `https://github.com/jmschrei/yahmm`, 2014.

[19] Jacob Schreiber. Automation. `https://github.com/UCSCNanopore/Data/tree/master/Automation`, 2014.

[20] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[21] Leonard E. Baum and Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, December 1966.

[22] Sean R Eddy. What is a hidden Markov model? *Nature biotechnology*, 22(10):1315–6, October 2004.