

©Copyright 2018  
Matthew T. Noakes

# Improving the Accuracy and Application of Nanopore DNA Sequencing

Matthew T. Noakes

A dissertation  
submitted in partial fulfillment of the  
requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Jens H. Gundlach, Chair

Paul A. Wiggins

Sreeram Kannan

Program Authorized to Offer Degree:  
Physics

University of Washington

## **Abstract**

### Improving the Accuracy and Application of Nanopore DNA Sequencing

Matthew T. Noakes

Chair of the Supervisory Committee:  
Dr. Jens H. Gundlach  
Physics

DNA contains the code of life, forming the molecular basis for all of life's diversity. The past several decades have witnessed remarkable progress in our ability to read and understand life's code through DNA sequencing. While fast and cheap DNA sequencing technologies are revolutionizing both science and healthcare, a new generation of technologies capable of single-molecule sequencing<sup>1</sup> promise to further revolutionize the field of DNA sequencing by addressing many of limitations of the previous methods. Nanopore DNA sequencing is one such emerging single-molecule sequencing technology, capable of long reads and direct detection of epigenetically-relevant modified bases.

The basic nanopore sequencing devices consists of two wells filled with a conductive electrolyte solution separated by an impermeable membrane containing a single nanometer-size hole, or nanopore. A voltage applied across the membrane drives an ionic current through the nanopore. DNA is negatively charged in solution and so will be drawn through the pore by the voltage, blocking some of the ionic current. As the different nucleotides along the DNA block the ionic current to different extents, the series of current fluctuations in the recorded time series can be used to decode the sequence of the DNA molecule moving through the pore. DNA motion through the pore is controlled using a DNA-processing motor

---

<sup>1</sup>Single-molecule sequencing means to read the sequence from a single copy of a target DNA molecule. Previous sequencing technologies required making many copies of the target DNA prior to sequencing.

enzyme, which steps the DNA through in discrete steps slow enough to allow resolution of the sequence-dependent fluctuations in the ionic current.

Commercial nanopore sequencing devices have recently become available, making good on the decades-long promise of this technology. However, despite considerable early success and fanfare accompanying these first nanopore sequencers, technology development is not complete. Particularly, the single-read *de novo* sequencing accuracy must be improved for this technology to reach its full potential <sup>2</sup>. In order to fully realize its promise, we must both improve the accuracy of nanopore sequencing and devise better methods of handling error-prone sequencing data.

In this dissertation, I discuss my work in the Gundlach nanopore lab at the University of Washington towards the goals of improved nanopore sequencing accuracy and improved application of existing error-prone sequencing data. In chapter 1, I introduce the broad field of DNA sequencing. I cover the history of scientific interest in DNA and DNA sequencing and provide motivation for DNA sequencing as a worthwhile pursuit both for its scientific and medical merits. I also discuss previous and existing DNA sequencing technologies, as well as the limitations of these technologies that motivate the development of new methods such as nanopore sequencing. In chapter 2 I describe and introduce nanopore sequencing. I summarize the development of nanopore sequencing technology, how various challenges were overcome, and how currently available nanopore sequencing devices work, setting the stage for understanding the primary error modes limiting the sequencing accuracy of this technology. In chapter 3, I present my work on improving nanopore sequencing accuracy using a new method of DNA control for enzyme-actuated nanopore DNA sequencing. This new method, in which we use a time-varying voltage to control DNA motion through the pore in addition to a DNA-processing enzyme, is able to mitigate two of the primary error modes in nanopore sequencing and dramatically improve sequencing accuracy. I discuss the motivation

---

<sup>2</sup>*De novo* sequencing means sequencing without the aid of any known reference sequence: a completely unknown DNA molecule must be sequenced from scratch.



behind this new method, outline how we were able to realize nanopore sequencing using this method, and demonstrate the improved sequencing accuracy it affords. In chapter 4, I shift the discussion over to my work on improving the application of nanopore sequencing data. Specifically, I introduce a method of aligning nanopore data that enables highly sensitive and specific sequence alignment and species identification even for low accuracy reads. I go over the motivation for this method, and present our findings of its improved performance over alternative methods. Finally, I conclude in chapter 5 where I discuss the implications of the demonstrated advances in the accuracy and application of nanopore sequencing, as well as look out towards further progress that can be made in both arenas.



# TABLE OF CONTENTS

	Page
List of Figures . . . . .	vi
List of Tables . . . . .	ix
Glossary . . . . .	xi
Chapter 1: Introduction . . . . .	1
1.1 Foundations . . . . .	1
1.2 Early DNA Sequencing . . . . .	4
1.3 Second Generation Sequencing . . . . .	6
1.4 Applications of DNA Sequencing . . . . .	12
1.4.1 Molecular Biology . . . . .	12
1.4.2 Evolutionary Biology . . . . .	12
1.4.3 Metagenomics . . . . .	12
1.4.4 Clinical Diagnosis . . . . .	13
1.4.5 Personalized Medicine . . . . .	13
1.5 Limitations of Second Generation Sequencing . . . . .	14
1.5.1 Cost and Speed . . . . .	14
1.5.2 Epigenetic Modifications . . . . .	15
1.5.3 Read Length . . . . .	16
1.6 Third Generation Sequencing . . . . .	17
Chapter 2: Nanopore DNA Sequencing . . . . .	20
2.1 Basic Concept . . . . .	20
2.2 Choosing a Nanopore . . . . .	21
2.2.1 Solid State Nanopores . . . . .	23
2.2.2 Biological Protein Nanopores . . . . .	23

2.2.3	<i>Mycobacterium smegmatis</i> porin A . . . . .	25
2.3	Controlling DNA Translocation . . . . .	26
2.4	Relating Ionic Current with DNA Sequence . . . . .	28
2.5	Sequencing with 4-mers . . . . .	32
2.6	Commercial Nanopore Sequencing . . . . .	34
2.7	Sequencing Error Modes . . . . .	38
2.7.1	Indistinguishable Ionic Current States . . . . .	38
2.7.2	Enzyme Missteps . . . . .	41
2.8	A Foundation for Improvement . . . . .	42
Chapter 3:	Variable-Voltage Nanopore DNA Sequencing . . . . .	44
3.1	Parallel Pathways to High Accuracy . . . . .	44
3.2	Shortcomings in the Signal . . . . .	45
3.3	A New Enzyme Sheds New Light . . . . .	46
3.4	Voltage Shifts DNA Position . . . . .	47
3.5	Hybrid Control . . . . .	51
3.6	Variable-Voltage Data Reduction . . . . .	52
3.7	The Smooth Conductance Profile . . . . .	55
3.8	Error Correction with Variable-Voltage Data . . . . .	57
3.9	Sequencing with Variable-Voltage . . . . .	60
3.9.1	Feature Extraction . . . . .	62
3.9.2	Enzyme Misstep Correction . . . . .	63
3.9.3	Signal-to-Sequence Model . . . . .	64
3.9.4	Sequencing Algorithm . . . . .	66
3.10	Sequencing Results . . . . .	67
3.11	Discussion and Conclusions . . . . .	68
3.11.1	Context for Improvement . . . . .	68
3.11.2	Towards Higher Accuracies . . . . .	70
Chapter 4:	A High Sensitivity BLAST Algorithm for Nanopore Sequencing . . . . .	73
4.1	BLAST . . . . .	73
4.2	Nanopores and BLAST . . . . .	75
4.3	Beyond Sequence Alignment . . . . .	77

4.4	Current-to-Current BLAST . . . . .	78
4.4.1	BLAST Implementation . . . . .	78
4.4.2	Adapting the Algorithm . . . . .	82
4.4.3	Choice of Ionic Current . . . . .	84
4.5	Performance Evaluation . . . . .	85
4.6	Performance as a Function of Read Accuracy . . . . .	85
4.7	Alignment Significance . . . . .	87
4.8	Discussion and Conclusions . . . . .	89
4.8.1	Scaling to Larger Reference Databases . . . . .	89
4.8.2	Implications for Nanopore Sequencing Applications . . . . .	92
Chapter 5:	Conclusions . . . . .	94
Bibliography	. . . . .	96
Appendix A:	Sequencing Experiments . . . . .	107
A.1	Materials and Methods . . . . .	107
A.1.1	Proteins . . . . .	107
A.1.2	Nanopore Experiments . . . . .	107
A.1.3	Operating Buffers . . . . .	108
A.1.4	Data Acquisition and Analysis . . . . .	108
A.1.5	DNA Sequences and Constructs . . . . .	109
A.1.6	Constant-Voltage Sequencing Experiments . . . . .	110
A.1.7	Variable-Voltage Sequencing Experiments . . . . .	111
A.2	Using Hel308 as a Translocase . . . . .	112
A.3	Hel308 Processivity . . . . .	114
A.4	Hel308 Step Durations . . . . .	116
A.5	Sequencing Verification Experiment . . . . .	118
A.6	Random Sequencing Accuracy . . . . .	122
A.7	DNA Sequences . . . . .	124
A.8	Experimental Statistics . . . . .	126
A.9	Work Fuel . . . . .	126

Appendix B: Change Point Detection Algorithm . . . . .	128
B.1 Basic Description . . . . .	128
B.2 Mathematical Description . . . . .	129
Appendix C: Variable Voltage Data Reduction . . . . .	137
C.1 Capacitance Compensation . . . . .	137
C.2 Conductance Normalization . . . . .	142
C.3 Feature Extraction . . . . .	145
C.4 Elongation of DNA in MspA . . . . .	148
C.5 Position Shift Calculation . . . . .	152
Appendix D: Pre-Sequencing State Filtering . . . . .	153
D.1 Flicker Filter . . . . .	153
D.2 Variable-Voltage State Filtering . . . . .	154
D.2.1 Removal Filter . . . . .	155
D.2.2 Recombination Filter . . . . .	161
D.2.3 Reordering Filter . . . . .	167
Appendix E: Constructing the 6-mer Model . . . . .	171
E.1 General Considerations . . . . .	171
E.2 Initial Model . . . . .	172
E.3 Measuring Genomic DNA of Known Sequence . . . . .	174
E.3.1 $\Phi$ X174 . . . . .	174
E.3.2 $\lambda$ Phage . . . . .	176
E.4 Building the Empirical 6-mer Model from Genomic Reads . . . . .	180
E.5 Filling in Unmeasured 6-mers . . . . .	182
E.6 Constant-Voltage Model Extraction . . . . .	183
Appendix F: Sequencing Algorithm . . . . .	185
F.1 Match Scores . . . . .	185
F.2 Hel308 Backstep Kinetics . . . . .	186
F.3 Transition Probabilities . . . . .	187
F.4 Transition Matrix . . . . .	189
F.5 Markov Model . . . . .	190

F.6	Traceback and Sequence Construction . . . . .	192
Appendix G: Supplemental Information for BLAST . . . . .		193
G.1	Seed Scanning . . . . .	193
G.2	Binning . . . . .	195
G.3	Seed Extension . . . . .	198
G.4	Algorithm Parameters . . . . .	203
G.5	Seed Size Parameter for Variable Read Lengths . . . . .	206
G.6	True Positive Rate at Zero False Positive Rate . . . . .	208
G.7	Calculating Alignment P-Values . . . . .	211
G.8	Computation Time . . . . .	215

## LIST OF FIGURES

Figure Number	Page
1.1 Structure of DNA . . . . .	3
1.2 Biological information flow . . . . .	4
1.3 Method of Sanger sequencing . . . . .	8
1.4 Cost per genome . . . . .	9
1.5 Sequencing by synthesis . . . . .	11
1.6 Genome reconstruction . . . . .	18
2.1 Basic nanopore sequencing scheme . . . . .	22
2.2 $\alpha$ HL and MspA . . . . .	25
2.3 Enzyme-actuated nanopore sequencing . . . . .	29
2.4 MspA sensitivity . . . . .	31
2.5 Predictive power of the 4-mer model . . . . .	33
2.6 Sequencing using 4-mers . . . . .	36
2.7 4-mer Variation . . . . .	40
2.8 Sequencing error modes . . . . .	43
3.1 $\Phi$ 29 DNAP vs. Hel308 Helicase . . . . .	48
3.2 Voltage-induced DNA position shift . . . . .	51
3.3 Hybrid control sequencing scheme . . . . .	53
3.4 Variable-voltage data reduction . . . . .	57
3.5 Discrete vs. continuous conductance sampling . . . . .	58
3.6 Automatic error correction . . . . .	61
3.7 Sequencing confusion matrices . . . . .	69
3.8 Read accuracy distributions . . . . .	71
4.1 BLAST calculation efficiency . . . . .	76
4.2 Principle of current-to-current comparison . . . . .	80
4.3 Comparisons made by ssBLAST and iiBLAST . . . . .	81



4.4	Example ssBLAST scoring matrix . . . . .	83
4.5	Distribution of accuracies for full MinION reads of M13mp18 in test dataset. . . . .	86
4.6	True positive rate for ssBLAST and iiBLAST . . . . .	88
4.7	ssBLAST vs iiBLAST p-value comparison . . . . .	90
A.1	DNA constructs for $\Phi$ 29 and Hel308 . . . . .	110
A.2	Hel308-controlled DNA translocation . . . . .	113
A.3	Hel308 processivity . . . . .	115
A.4	Hel308 step durations . . . . .	117
A.5	Random sequencing accuracies . . . . .	123
B.1	Principal components for change point detection . . . . .	130
C.1	Capacitance compensation . . . . .	142
C.2	Fraying correction . . . . .	144
C.3	Principal component vectors for feature extraction . . . . .	147
C.4	Principal component description of conductance states . . . . .	147
C.5	DNA stretching in MspA . . . . .	151
D.1	Example of flicker states . . . . .	154
D.2	Removal filter confusion matrix . . . . .	158
D.3	Converting SVM outputs to $P_{bad}$ probabilities . . . . .	160
D.4	Self-alignment procedure for recombination filter . . . . .	166
D.5	Self-alignment transition penalties . . . . .	166
E.1	Theoretical 6-mer map generation . . . . .	173
E.2	$\Phi$ X174 for 6-mer map . . . . .	177
E.3	$\lambda$ for 6-mer map . . . . .	181
E.4	Iterative map construction method . . . . .	182
E.5	Fill-in strands for 6-mer map . . . . .	183
E.6	Constant-voltage model extraction . . . . .	184
G.1	Finite state machine for seed finding . . . . .	194
G.2	Fuzzy FSM . . . . .	197
G.3	Seed extension algorithm . . . . .	200
G.4	Acceptance threshold scheme . . . . .	208
G.5	Receiver operating characteristics for BLAST . . . . .	210

G.6	Length correction for alignment scores . . . . .	212
G.7	Converting alignment scores to p-values . . . . .	214

## LIST OF TABLES

Table Number	Page
A.1 pET-28a fragments . . . . .	120
A.2 DNA sequences I . . . . .	124
A.3 DNA sequences II . . . . .	125
A.4 Experimental statistics . . . . .	126
A.5 Tea usage . . . . .	127
G.1 BLAST parameters for validation experiment . . . . .	205
G.2 BLAST computation time . . . . .	215

## LIST OF ALGORITHMS

1	Simple 4-mer sequencer . . . . .	35
2	Sequence-to-sequence BLAST . . . . .	75
3	Current-to-current BLAST . . . . .	82
4	Change point detection . . . . .	130
5	Removal Filter . . . . .	156
6	Recombination filter . . . . .	164
7	Reordering filter . . . . .	170
8	Mealy FSM seed scan . . . . .	195
9	Extend seed to the right . . . . .	201

## GLOSSARY

ABASIC: Site on the DNA backbone where the nucleobase is absent.

ATP: Adenosine triphosphate. The substrate molecule used by Hel308 to generate the energy needed to move along DNA.

AMINO ACID: The basic monomer building blocks of proteins.

BASE CALLING: The process of assigning a DNA sequence to an the signal generated by a sequencing device.

CODON: A set of three bases that together code for an amino acid during protein synthesis.

*DE NOVO* SEQUENCING: The task of sequencing DNA without reference to any information other than that provided by the sequencing platform itself (no reference genome).

DNA: Deoxyribonucleic acid. The molecule forming the genetic basis of life.

DNAP: DNA polymerase. An enzyme (protein) that catalyzes the synthesis of a new, complementary DNA strand from a single-stranded template.

DNA SEQUENCING: The process of reading the order of bases along a DNA molecule.

dNTP: Deoxynucleoside triphosphate. These molecules are the building blocks of DNA. There are 4 types of dNTPs, one for each of the 4 canonical nucleobases. These are denoted dATP, dCTP, dGTP, and dTTP.

DSDNA: Double stranded DNA.

ENZYME: A molecule that catalyzes a biological reaction.

EPIGENETICS: Heritable traits that do not arise from mutations in the DNA sequence.

EXOME: The exome is the portion of the genome comprising the exons, which are the sequences which are ultimately transcribed and translated into a gene's final protein product.

FLUOROPHORE: A molecule that can re-emit light of a specific spectrum upon excitation by input light.

GENOME: The complete DNA sequence in a cell or organism.

HEL308: A DNA helicase used in our studies to step DNA through the nanopore in controlled increments.

HELICASE: A motor enzyme that catalyzes the unwinding of double stranded DNA.

LIPID: A class of biological molecules with a hydrophobic tail and hydrophilic head.

K-MER: A DNA sequence  $k$  bases long.

MESSENGER RNA: mRNA. RNA molecules transcribed from genes, ultimately translated into proteins.

MOTOR ENZYME: A class of enzymes that move along a nucleic acid track.

MSPA: *Mycobacterium smegmatis* porin A. A bacterial outer membrane protein with properties well suited for nanopore DNA sequencing.

NGS: Next generation sequencing. Will be referred to as second generation sequencing (SGS) throughout this work.

NUCLEOBASE: The bases along the DNA sugar-phosphate backbone that comprise the DNA sequence.

NUCLEOTIDE: The structural unit of the DNA polymer. Consists of a nucleobase bound to a sugar-phosphate backbone.

PCR: Polymerase Chain Reaction. A process used to exponentially amplify a DNA molecule.

$\Phi$ 29 DNAP: A DNA polymerase used in this work to step DNA through the pore

**PROTEIN:** A biological molecule comprised of amino acid building blocks, serving a wide variety of biological functions. Proteins are synthesized from mRNA during translation.

**PURINE:** A two-ring nucleobase. Adenine (A) and guanine (G) are purines.

**PYRIMIDINE:** A single-ring nucleobase. Cytosine (C) and thymine (T) are pyrimidines.

**RNA:** Ribonucleic acid. A biological molecule, similar to DNA. RNA serves many functions in the cell, but is primarily acts as an intermediary carrying the code from DNA to be translated into proteins.

**SANGER SEQUENCING:** An early method of DNA sequencing that used inextensible dNTPs along with gel electrophoresis to determine the sequence.

**SBS:** Sequencing by synthesis. A second generation sequencing technology that works by monitoring the synthesis of a new DNA strand from the target DNA molecule.

**SGS:** Second generation sequencing. Any of a broad class of DNA sequencing technologies that replaced Sanger sequencing in the mid 2000's.

**SSDNA:** Single stranded DNA.

**SVM:** Support vector machine. A simple machine learning classifier that differentiates between two categories by finding the plane best partitioning between labeled examples of the two categories.

**TRANSCRIPTION:** The process whereby RNA is synthesized from a DNA template.

**TRANSLATION:** The process whereby a protein or polypeptide is synthesized from an RNA template.

**TRANSLOCASE:** An enzyme that walks along a single stranded DNA track.

## ACKNOWLEDGMENTS

I want to thank my advisor Jens Gundlach for his enthusiasm, vision, and patience. His endless passion for scientific inquiry and discovery over the past 5 years has been an enormous motivation in my pursuit of the work presented here, as his scientific vision and guidance have helped me avoid or overcome the numerous challenges encountered along the way. Thank you also to Henry Brinkerhoff, with whom I've worked closely on the variable-voltage sequencing project. His statistical and algorithmic knowledge and innovation has been instrumental in taking the project from concept to reality. Thanks as well to previous lab members Ian Derrington and Kyle Langford, who laid the foundations of my work on sequencing, as well as to Brian Ross who wrote many of the initial versions of the algorithms used in this work. Thank you to Andrew Laszlo, who was a source of advice and guidance throughout my time in the lab, and whose foundational work on nanopore species identification paved the way for the work I present here on BLAST. Thanks also to Ian Nova, who provided steadfast friendship and helpful discussion on scientific goals in addition to elevating the stoke factor on both surfing and skiing. Thank you to Jon Craig who was instrumental in first getting me involved in the nanopore lab, and in keeping me there by promoting a fun and productive lab environment. Thank you to all of my coworkers and collaborators whose enthusiastic work and stimulating discussion (scientific and otherwise) has made working here a joy—Ben Tickman, Hugh Higinbotham, Jasmine Bowman, Jesse Huang, Jenny Mae Samson, Jon Mount, Josh Bartlett, Katie Baker, Kenji Doering, Noah DeLeeuw, Sinduja Marx, and Yihan Jiang. Thanks to the graduate program advisor Catherine Provost for keeping me on track. Thank you also to Tony She, who has helped keep me sane <sup>3</sup>, and Daniel Noakes who

---

<sup>3</sup>somewhat



has been a source of support throughout my PhD. Thank you to Sarah Whiteside who has supported, encouraged, and believed in me through the entire process.

This work was supported by the National Institutes of Health, National Human Genome Research Institute \$1,000 Genome Program Grant R01HG005115.

## **DEDICATION**

To my parents, Tim and Brenda, who taught me to work hard and pursue my dreams.

## Chapter 1

# INTRODUCTION

### **1.1 Foundations**

In 1952, the Hershey-Chase experiment [1] proved that deoxyribonucleic acid (DNA) carries the genetic code of life. This discovery that a single molecule is responsible for encoding, preserving, and propagating all of the information guiding the complexity of life is stunning. Particularly, the existence of a centralized genetic code contained within a single-molecule leads immediately to a tantalizing corollary: if we could determine the structure and composition of DNA, we could directly read and potentially understand the code of life. The next goals were clear: understand the structure of DNA, find out how the genetic information is encoded, then gain access to that information.

The first of these goals was achieved soon after. Only a year following the demonstration that DNA encodes the genetic code, Watson and Crick, along with Franklin and Wilkins [2] determined the structure of DNA (Fig 1.1). Specifically, DNA is a long polymer made up of two anti-parallel strands (termed “sense” and “antisense”) forming a double helix. Each of the two strands is composed of an alternating sugar-phosphate backbone. The strands have a directionality, with the 5’ end terminating in a phosphate and the the 3’ end having a terminal hydroxyl group. The two strands run antiparallel to one another, with the 5’ end of one corresponding with the 3’ end of the other. Each sugar binds to one of 4 canonical nucleobases: adenine (A), cytosine (C), guanine (G), or thymine (T). Hydrogen bonds between complementary nucleobases bond the two strands together, with A forming 2 hydrogen bonds with T and G forming 3 hydrogen bonds with C.

Solving the structure of DNA revealed how information is encoded in the molecule. The

sugar-phosphate backbone is homogenous over the long scales of the DNA molecule and so serves only a structural role. The genetic information “payload” must then be carried by the nucleobases. Namely, the code of life is “written” in the sequence of the nucleobases A, C, G, and T from 5’ to 3’ along the backbone. The two strands are completely complementary: an A, C, G, T in the sense strand is always accompanied by a T, G, C, or A in the antisense strand. Thus, the double-stranded DNA molecule encodes two identical copies of the same information. This apparent redundancy is critical to maintenance and replication of the genetic code, wherein each of the two strands is individually used as a template to generate a new copy of the DNA [3].

The understanding that genetic information is carried by the sequence of nucleobases (“DNA sequence”) was expanded upon less than a decade later by work demonstrating that sets of three bases (codons) correspond to specific amino acids—the building blocks of proteins [4]. We now understand information flow in biology as occurring through the hierarchical interactions of three key biopolymers: DNA, RNA, and protein [5]. This unifying biological theory—termed the central dogma—states that DNA templates the transcription of ribonucleic acid (RNA) molecules. RNA in turn guides the translation of the polypeptides which form proteins—a versatile class of biomolecules which ultimately perform the myriad functions crucial to life (Fig 1.2).

The goal of understanding how the genetic information in DNA is decoded into RNA and protein is far from complete. Both transcription (DNA into RNA) and translation (RNA into proteins) form the basis of rich and complex fields of research. With regards to reading the genetic code however, these early results and the basic formulation of the central dogma painted a clear picture of where to go next. If the code of life is written in the sequence of bases along DNA, then to read and ultimately understand the code, we must be able to determine the DNA sequence.

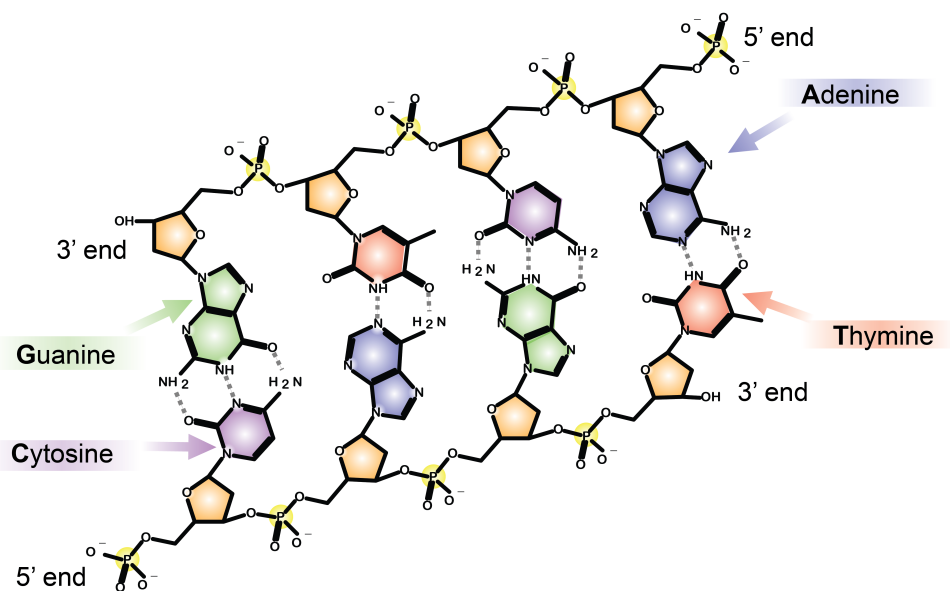


Figure 1.1: Structure of DNA. The basic unit of the DNA molecule is the nucleotide, composed of a phosphate (yellow), sugar (orange), and nucleobase (red, blue, purple, or green). Each of DNA's two complementary strands is composed of a chain of these nucleotides, with the alternating sugars and phosphates making up the polymer's backbone. DNA's two strands have a directionality. The 5' end terminates in a final phosphate group, and the 3' end terminates in a final hydroxyl group. The four nucleobases Adenine (A, blue), Thymine (T, red), Cytosine (C, purple), and Guanine (G, green) encode the biological information stored in the DNA. Base pairing between complementary nucleobases across the two strands holds the polymer together, with A pairing with T through 2 hydrogen bonds (gray dashed lines) and C pairing with G through 3 hydrogen bonds. In its double stranded form, DNA forms a double helix with a diameter of 2.4 nm and an inter-nucleotide spacing of 0.34 nm. Single stranded DNA is not a double helix and is much more flexible. It is half the diameter of dsDNA at 1.2 nm, and the nucleotides are spaced out to an inter-nucleotide spacing of 0.69 nm. This image is adapted from <https://biologydictionary.net/dna/>

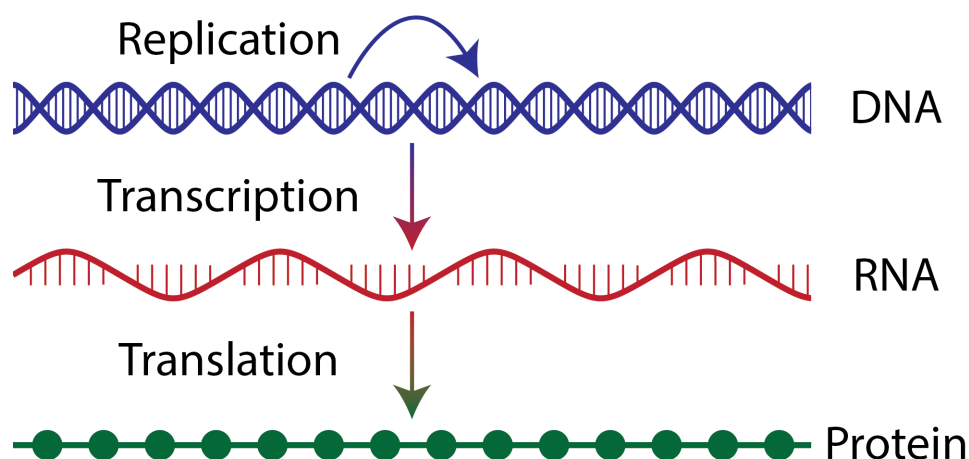


Figure 1.2: Biological information flow. DNA acts as the base repository of biological information. DNA templates the synthesis of new DNA during the process of replication. Additionally, DNA templates the synthesis of RNA in the process of transcription. The intermediary RNA then templates the synthesis of proteins in the process of translation.

## 1.2 Early DNA Sequencing

The goal of reading the sequence of nucleobases along a DNA molecule (DNA sequencing) is a conceptually simple problem whose practical complexities and tremendous promise have led to it occupying an enormous fraction of the scientific consciousness of the past several decades. The obvious challenge in DNA sequencing lies in the scale of the molecule. The bases along single stranded DNA (ssDNA) are spaced by only 0.69 nm along the backbone. Furthermore, the chemical differences between the 4 bases are quite subtle. Particularly, the single-ring pyrimidine bases C and T differ by only a few atoms from each other. The same is true for the two-ring purine bases A and G.

In juxtaposition to the minuscule scale of the DNA bases is the enormous scale of the entire polymer. Whole genomes can run from thousands (viruses) to millions (prokaryotes) even to billions (eukaryotes) of bases. The human genome comprises over 3 billion base pairs (Gbp), and would measure over a meter in length if stretched out end-to-end (double

stranded; over twice that if single stranded). The confluence of the small scale of the bases making up the DNA sequence and the vast length of the entire sequence makes sequencing an organism's entire genome a daunting task.

It took nearly 25 years following the discovery of DNA's structure for the first genome to be sequenced [6]. the first completed genome was that of the  $\Phi X174$  bacteriophage. The  $\Phi X$ -174 genome is modest in size (only 5386 bases), but its completion opened the door to more ambitious sequencing projects. Notably, this first genome was sequenced using the first broadly successful DNA sequencing technique, commonly known now as Sanger sequencing after its inventor.

Sanger sequencing works by first making many copies of the DNA to be sequenced, all starting at the same location in the genome (Fig 1.3). Each of these copies is then replicated *in vitro*. During replication, a new copy of the initial DNA strand (template) is synthesized by the successive incorporation of deoxynucleoside triphosphates (dNTPs, the building blocks of DNA) by a DNA polymerase <sup>1</sup> (DNAP, an enzyme that catalyzes the synthesis of DNA). A fraction of the dNTPs are chemically modified to be inextensible; that is, the DNAP is unable to incorporate more dNTPs into the nascent copy strand following the incorporation one of these modified dNTPs. These same inextensible dNTPs are additionally labeled with a colored fluorescent molecule (fluorophore), with a separate color labeling each of dATP, dCTP, dGTP, and dTTP. This process of *in vitro* replication with occasional random incorporation of fluorescently-labeled inextensible dNTPs results in a population of variable-length nascent copies of the target DNA sequence. Each is labeled at its 3' end with the fluorophore corresponding to the final base incorporated. This DNA is run on an agarose gel, which separates the strands by their length <sup>2</sup>. The target DNA can finally be sequenced

---

<sup>1</sup>The use of DNA-processing enzymes, such as DNA polymerases, has underpinned not only Sanger sequencing, but many more modern approaches as well. Both sequencing-by-synthesis and enzyme-actuated nanopore sequencing use DNA-processing enzymes, as will be discussed later

<sup>2</sup>The above-described method is an amalgam of the several different implementations of Sanger sequencing that have been used over the past decades. Older methods used four parallel reactions, one each for A, C, G, and T termination and no fluorescent labels. Newer methods use capillary electrophoresis in place of gels to allow more automated throughput.

by reading off the order of colors in the length-ordered DNA bands in the gel.

Sanger sequencing remained the state-of-the-art in DNA sequencing for nearly 40 years, and is still used in certain applications. The Human Genome Project (HGP), begun in 1990 and completed in 2003 [7, 8], relied heavily on Sanger sequencing to read the entire 3 Gbp human genome. Lasting over a decade and costing nearly \$3 billion, the HGP was an enormous undertaking and a resounding success. However, sequencing a single human genome proved not to be the culmination of the goal of reading the genetic code, but rather the starting point. The success of the HGP was a window into the enormous potential of DNA sequencing as a tool for both clinicians and researchers. A new wave of ambitious applications (section 1.4) no longer seemed far-fetched.

However, despite its broad success, Sanger sequencing would not be sufficient to meet the ambitious new goals of the DNA sequencing movement. High reagent costs, large requirements for input DNA necessitating *in vivo* DNA amplification, and the tedious difficulty of running countless gels made Sanger sequencing costly and slow. While innovation driven by the HGP had reduced Sanger sequencing costs more than 10-fold over the course of the decade, estimates still placed the cost of sequencing a full human genome around \$100 million in 2001. An orders-of-magnitudes further cost reduction would be critical for DNA sequencing to be clinically feasible for individuals and accessible to individual researchers.

### **1.3 Second Generation Sequencing**

Following the completion of the Human Genome Project, the cost of DNA sequencing has plummeted from nearly \$100 million to just above \$1000 over the course of about 15 years (Fig 1.4). This precipitous drop has far outpaced Moore's Law<sup>3</sup>—long the gold standard of technological innovation. The drop was driven by the advent of disruptive new technologies, a revolution commonly referred to as next generation sequencing (NGS). For clarity

---

<sup>3</sup>Moore's Law states that the number of transistors per area on a chip will double every 18 months. Commonly, this is interpreted and phrased as saying that computing costs will fall by half over this time period. In general, it serves as a benchmark for rapid progress in technological development.



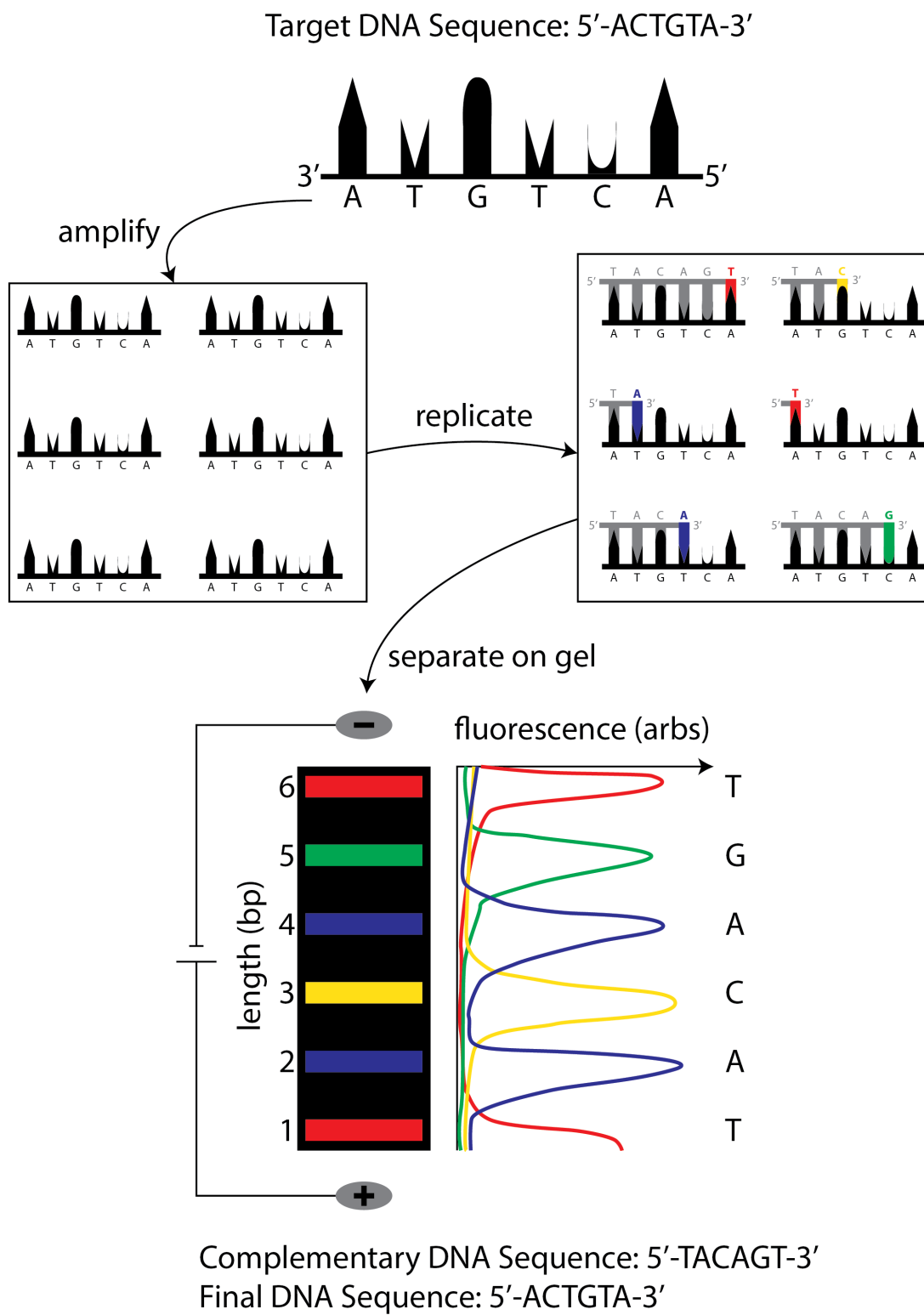


Figure 1.3: Method of Sanger sequencing. Sanger sequencing of a target DNA strand (5'-ACTGTA-3') begins by making many copies of the target strand (amplification). The duplicate copies are then replicated *in vitro* using a fraction of inextensible, fluorescently labeled dNTPs. Some fraction of all the replicates will be terminated at each position along the template. The terminated replicates are finally separated on a gel by their length. The fluorescence of the separated bands reveals the complementary sequence of the original target strand. The final sequence can ultimately be read off as the complement of the fluorescently decoded read.

going forward, I will refer to these technologies as second generation sequencing (SGS) to differentiate them from the more modern approaches discussed later. SGS technologies were able to massively parallelize the task of DNA sequencing, eliminate the need for gel-based (or capillary-based) readouts, and reduce <sup>4</sup> the sample size requirements on the input DNA. Together, these achievements substantially reduced the time, cost, and effort required to do DNA sequencing.

The method of sequencing-by-synthesis (SBS) underpins many of the most important SGS technologies. SBS works by monitoring the activity of a DNA polymerase as it synthesizes a copy of the target DNA strand. The target DNA molecule is amplified *in vitro* and many identical copies of the target are fixed to a location on a flow cell (Fig 1.5). DNA polymerase enzymes are then used to synthesize a complement to each copy of the target DNA in unison. Nascent strand synthesis uses specially modified dNTPs. The dNTPs used are reversibly-terminating (RT-dNTPs), meaning that their incorporation prevents further extension of the nascent strand until the terminating moiety is removed by another chemical process referred to as deblocking. Iteratively, one of the 4 types (A, C, G, or T) of RT-dNTPs is flushed into the flow cell. The DNA polymerase incorporates the correct nucleotide into the nascent strand, then is blocked from further incorporation. Incorporation is detected either optically using fluorescently-labeled RT-dNTPs (as in Illumina sequencing) [9] or electronically by

---

<sup>4</sup>Though the input DNA requirements for SGS technologies still require polymerase chain reaction (PCR) amplification, the scale of amplification is significantly reduced, and the need to amplify using bacterial vectors has been largely deprecated.

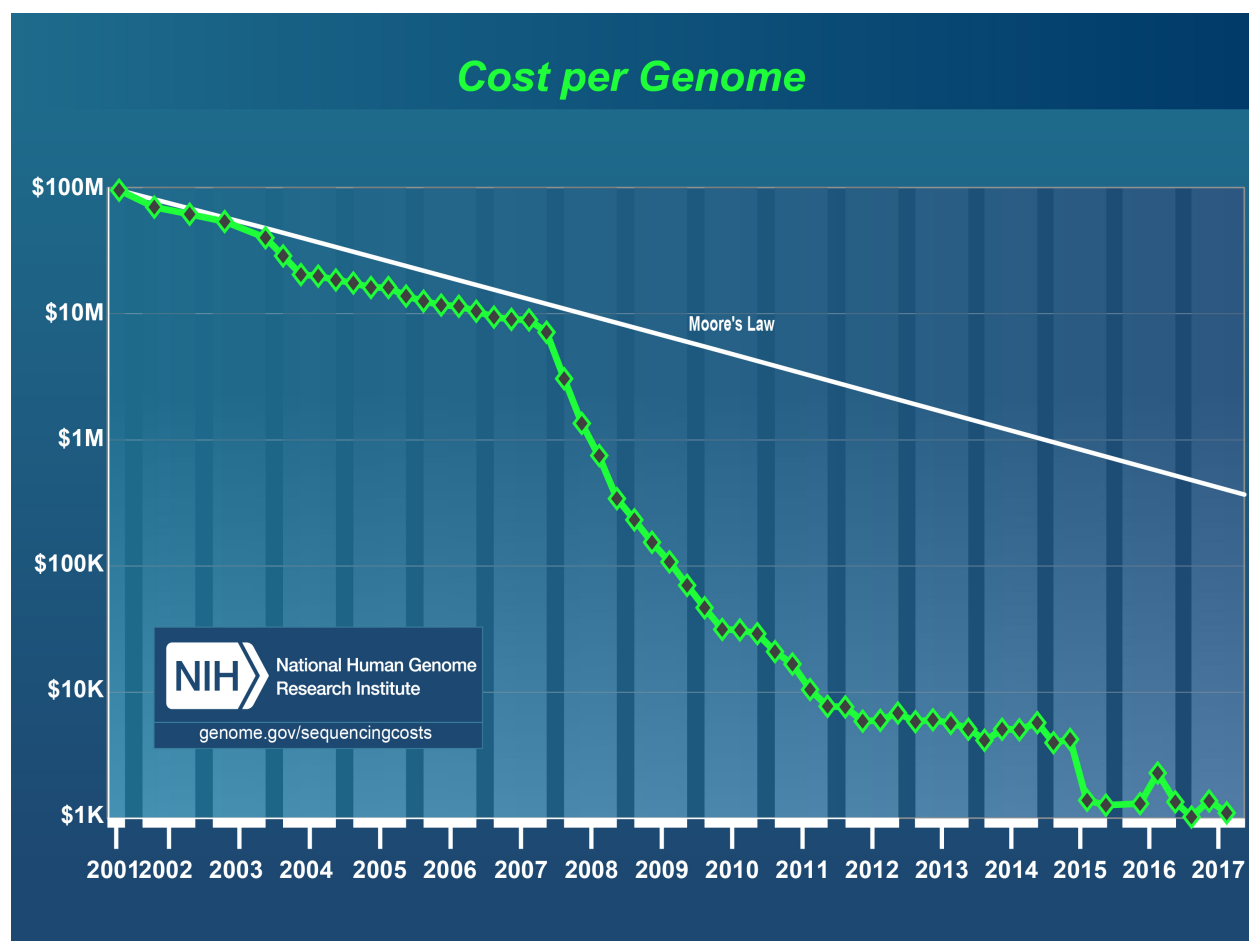


Figure 1.4: Cost per genome. The diamond data points and green line show how the cost to sequence a single human genome has fallen since the completion of the human genome project. The white line shows Moore's law over the same period—costs falling by half every 1.5 years. The cost per genome has dramatically outpaced Moore's law for over a decade, driven by disruptive advances in DNA sequencing technology. This image is adapted from the National Institutes of Health, National Human Genome Research Institute, <https://www.genome.gov/27541954/dna-sequencing-costs-data/>

detecting the release of  $H^+$  ions accompanying dNTP addition (as in Ion Torrent sequencing) [10]. Following incorporation, the deblocking agent is flowed through, allowing subsequent further extension of the nascent strand.

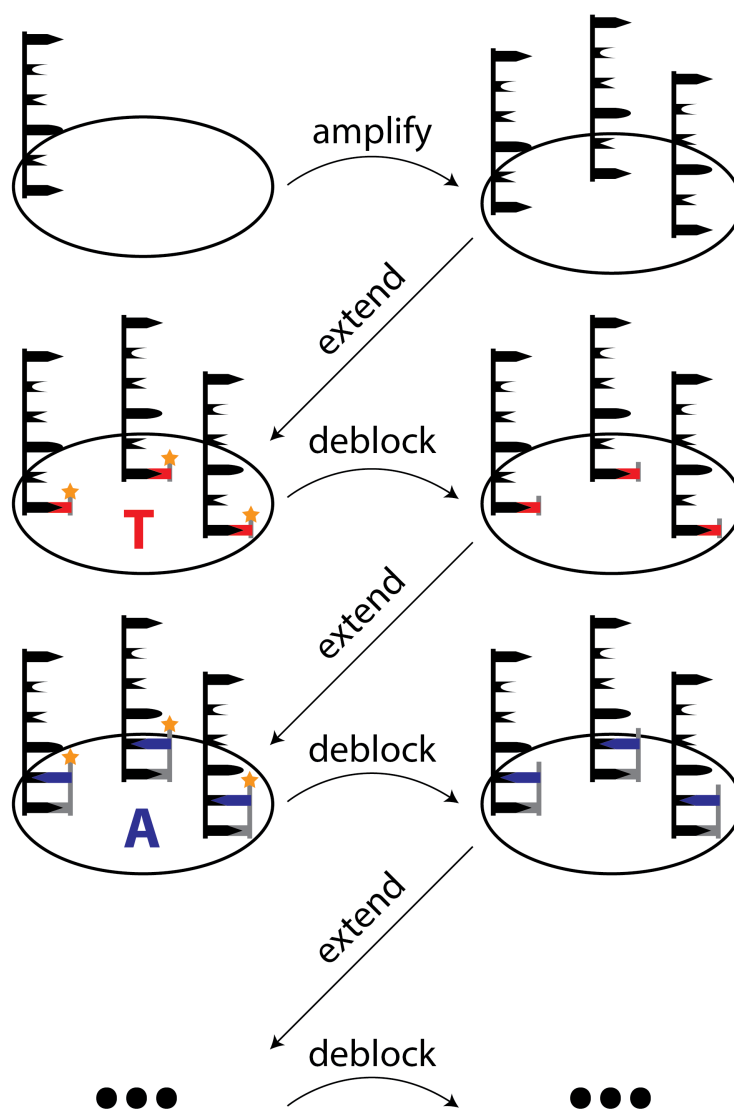


Figure 1.5: Sequencing by synthesis. Sequencing by synthesis starts by amplifying the target DNA strand to form a clonal sequencing colony. In the extension step, fluorescently labeled inextensible dNTPs are flowed in, and the template is extended via the incorporation of the correct complementary base. The deblocking step then removes the inextensible end of the incorporated nucleotide, allowing further extension. The extend/deblock cycle is repeated until the entire complement has been synthesized and the entire strand sequenced.

## **1.4 Applications of DNA Sequencing**

A few burgeoning scientific and medical applications leveraging the modern power of second generation DNA sequencing are discussed below.

### *1.4.1 Molecular Biology*

Our ability to sequence DNA has revolutionized our understanding of the molecular basis of life. We can now directly access the code governing cellular function. By sequencing DNA, researchers can now read the genes that are transcribed into messenger RNA and ultimately translated into proteins. Reading the sequence of genes allows researchers to understand how genes are expressed, and how mutations change the overall function of the cell.

### *1.4.2 Evolutionary Biology*

DNA sequencing has unsurprisingly proven to be a powerful tool for evolutionary biology—the study of how organisms change over time and are related to one another. Comparing the genomes of different species can reveal their shared evolutionary history and shed light on relational questions beyond the reach of macroscopic analysis. Within species, comparative sequencing of multiple individuals can be a tool to track evolution over faster time scales. One such application is in the tracking of viral or bacterial pathogens during an outbreak [11] [12]. Fast, cheap DNA sequencing of the outbreak vector can monitor the spread of the disease and track important evolutionary changes in the pathogen potentially leading to drug resistance or difficulties in treatment [13].

### *1.4.3 Metagenomics*

Metagenomics is a new field of scientific research enabled by the falling cost of DNA sequencing, focused on sequencing entire ecosystems rather than individuals or individual organisms. Rather than reading the genetic material from a single individual, metagenomics studies sequence all the genetic material contained in an environmental sample. A large sample of

material is collected from a given environment—for example, one study used 200 L of seawater [14]. DNA from the multitudinous organisms present in the sample is extracted and sequenced. The resulting sequencing data is then analyzed, providing profound insight into the biodiversity present in the ecosystem. Such large-scale sequencing projects have already revolutionized our understanding of ecosystems including the ocean [14] and the human microbiome [15]. These types of studies have driven critical advances in fields including biofuels research [16], environmental monitoring [17], and agriculture [18].

#### *1.4.4 Clinical Diagnosis*

DNA sequencing is making it easier for clinicians to diagnose patients. Perhaps the simplest case of clinical diagnosis using DNA sequencing is in the case of genetic diseases, where DNA sequencing allows direct detection of the mutation(s) responsible for the disease. In the future, it is possible that DNA editing technology will progress to the point where we can not only identify genetic disorders, but also correct them.

In addition to the genetic diseases application, a metagenomics-style approach has proved to be a powerful tool for pathogen detection and diagnosis of non-genetic illness. This type of application extracts the DNA present in a sample from the blood or gut of a patient. By sequencing the extracted DNA, we can determine the species present in the sample (viral, bacterial, etc.) and from this determine the source of the observed illness as well as the best way to treat it. This type of diagnosis could improve outcomes in cases such as sepsis. For septic patients, each hour that diagnosis and the administration of effective antibiotics or antimicrobials is delayed dramatically decreases the likelihood of survival [19] [20]. The fast identification of the infectious organism through sequencing and the corresponding informed administration of effective drugs would reduce sepsis mortality rates.

#### *1.4.5 Personalized Medicine*

The basic concept of personalized medicine—tailoring treatment to the unique circumstances and needs of the patient, rather than following a one-size-fits-all standard of care procedure—

is not new in and of itself. However, the advent of fast and affordable DNA sequencing has revolutionized the scope and precision of personalized medicine. The presence or absence of specific mutations can influence the range of likely outcomes for a given treatment plan. For instance, a patient with mutation  $X_1$  may be 90% likely to be cured by drug  $A$ , with a 5% chance of adverse side effects. Conversely, a patient with a different mutation  $X_2$  may only be 30% likely to be cured by the same drug, with a 50% chance of adverse side effects. By sequencing in advance of treatment, the clinician would be able to know to prescribe drug  $A$  to the first patient, but perhaps look instead for an alternative treatment plan for the second patient. This type of approach has already proven effective in treatment of pancreatic [21], promyelocytic leukemia [22], gastric [23], and non-small cell lung [24] cancers, amongst others [25]. In general, by sequencing the specific gene(s) of interest of the patient—or even entire exome or genome—clinicians can better predict the efficacy of various treatment plans prior to beginning treatment and better tailor their approach on a patient-by-patient basis.

## ***1.5 Limitations of Second Generation Sequencing***

SGS technologies have brought us a long way toward understanding the code of life by increasing the speed and decreasing the cost of DNA sequencing. The progress catalyzed by SGS over the past decade continues to revolutionize science and health care. However despite the substantial achievements of these technologies, they do not completely solve the myriad challenges of DNA sequencing.

### *1.5.1 Cost and Speed*

While SGS has made exponential progress in reducing the time and cost to sequence DNA, progress in this arena has begun to plateau over the past several years (Fig 1.4) as we approach some fundamental limitations of these technologies.

In particular, the speed limitations of SGS technologies present a challenge to various sequencing applications relying on fast time lines from taking a sample to getting an answer



(sample-to-answer). The extend/deblock cycle is inherently slow <sup>5</sup>, so it takes a long time to sequence an individual clonal sequencing colony. As each individual read is slow, SGS gains its speed through massive parallelization: taking many (millions or tens of millions) reads simultaneously. This parallelization improves the overall average speed of these technologies in terms of base pairs sequenced per hour of run time. However, it is only a partial workaround in terms of speedy sample-to-answer. The researcher or clinician must still wait until the parallelized reading is complete before the sequencing data is available. Further parallelization does not address the issue that a full SGS run takes a day or more. A new technology seeking to improve sample-to-answer time lines would need to reduce the per-base time for single reads.

### 1.5.2 *Epigenetic Modifications*

There is more information encoded in genomic DNA than just the order of the bases. Epigenetic factors are heritable changes outside of mutations in the DNA sequence that affect biological function. A broad class of the epigenetic factors take the form of modifications to the DNA bases. The presence or absence of modified bases such as 5-methylcytosine (a methylated cytosine) at specific genome locations can influence gene regulation and expression, affecting the rate and manner in which certain genes are transcribed and ultimately translated into proteins [26]. As such, the pattern of these epigenetically modified bases in the genome has implications for cell differentiation and life cycle [27], as well as cancer and other diseases [28] [29] [30].

Despite their importance, these epigenetically modified bases can not be directly detected by SGS technologies. The process of amplifying the target DNA molecule does not preserve modified bases present in the original DNA—all copies will be entirely unmodified. There exist a suite of methods to indirectly detect modified bases using SGS, but all of these

---

<sup>5</sup>Even ignoring the hours required for colony formation, each base incorporation cycle takes ~10 minutes in an Illumina sequencing device, meaning a 200 base read requires over a full day to run. A detailed breakdown of Illumina runtimes can be found at <https://support.illumina.com/bulletins/2017/02/run-time-estimates-for-each-sequencing-step-on-illumina-sequenci.html>

methods have drawbacks in cost, speed, and/or accuracy [31] [32] [33]. A technology capable of reading native DNA—without the need for amplification—would potentially be able to directly detect these epigenetically-relevant modified bases and open a new dimension of sequencing information.

### 1.5.3 Read Length

In addition to destroying epigenetic information, the DNA amplification requirement of all SGS methods limits the read lengths possible using these technologies. To generate a good signal, sequencing by synthesis relies on the clonal population being “in-phase”: the same (correct) dNTP being incorporated into all members of the colony during each cycle of extension/deblocking. However, the DNA polymerase incorporating the dNTPs is error prone and occasionally fails to incorporate a nucleotide when it should, or incorporates an extra nucleotide when it should not. During each extension cycle, some fraction of the clonal population will experience an error, becoming “de-phased”. Each successive cycle will see more and more members of the colony become de-phased and the signal will deteriorate. Eventually, there will be too few in-phase incorporations to generate a useful sequencing signal.

If the DNAP makes errors at some rate <sup>6</sup>  $\epsilon$ , after  $n$  cycles only  $(1 - \epsilon)^n$  of the colony members will still be in-phase. If the sequencing device requires some fraction  $\mathcal{F}$  of the colony members to be in-phase in order to generate accurate results, the maximum possible read length  $\mathcal{L}$  is given by

$$\mathcal{L} = \log_{1-\epsilon} \mathcal{F} \tag{1.1}$$

Even for generous estimates of  $\mathcal{F} = 0.25$  (only requiring 25% of clones to be in-phase) and  $\epsilon = 0.005$  (DNAP only making 1 error in 200 tries), the maximum possible read length  $\mathcal{L}$  is only 276 bases. Typical SGS reads are limited to  $\sim 200$  bases <sup>7</sup>.

---

<sup>6</sup>In this dissertation, I use the term error rate to refer to the per-base probability of a sequencing error.

<sup>7</sup>A summary of read length capabilities and other statistics for various Illumina sequencing devices can

Short read lengths make the task of whole genome sequencing difficult or even impossible. In order to reconstruct a DNA sequence (i.e. an entire genome) longer than a sequencer's read length, separate shorter reads of sections of the sequence must be stitched together. Reads can be stitched together based on their overlapping stretches [34] but this can be computationally difficult. Take the case of reconstructing a human genome using 200 base SGS reads: the 3 billion base human genome would require tens of millions of these short reads to cover its entire length. Finding the correct way to overlap and stitch together these tens of millions of reads to reconstruct the entire genome is difficult, slow, and potentially error prone.

Particularly, short reads are ill-suited for correctly sequencing repetitive sections of the genome. Large stretches of the human genome (and those of other eukaryotes) are composed of stretches of adjacent or interleaved copies of some DNA sequence [35]. These repetitive sequence motifs cannot be correctly reconstructed unless the read length is longer than the repeated sequence (Fig 1.6). Consequently, there are sections of the human genome (and other genomes) that are impossible to correctly reconstruct using short read sequencing technologies. A technology capable of long read lengths would reduce the computational cost of sequence reconstruction and allow us access the true sequence of difficult repetitive sections. Such a technology must not rely on the in-phase signal of many copies of the target molecule.

## **1.6 Third Generation Sequencing**

A new generation of sequencing technologies is now emerging, seeking to address the various limitations of SGS. This “third generation” of sequencers is moving away from the massively parallelized sequencing-by-synthesis approaches used in SGS. Instead, these new technologies aim to achieve single-molecule sequencing: reading the sequence of a target DNA strand using only a single unamplified copy. A single-molecule sequencing technology would not require

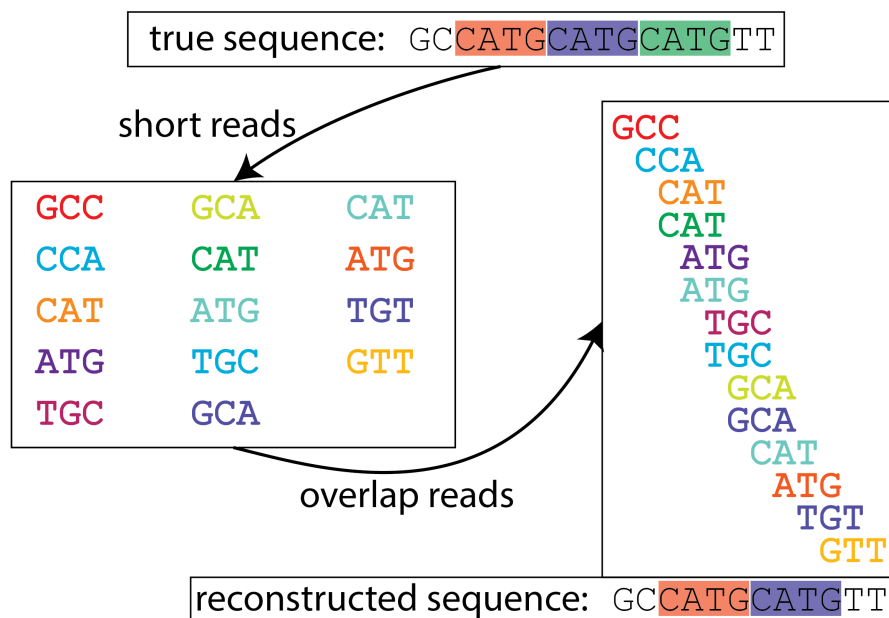


Figure 1.6: Genome reconstruction. In this example, the true DNA sequence has 3 repeats of a 4 base motif (red, blue, green highlights). If this DNA were sequenced using a sequencer with read length of 3, we would get a set of short reads covering the sequence. Stitching together these short reads based on their overlap would yield a final reconstructed sequence with only two copies of the repeated 4 base motif (red, blue highlights). The reconstructed sequence has lost a repeat relative to the true sequence.

DNA amplification and would therefore avoid many of the issues limiting SGS methods. First, avoiding amplification would improve cost and speed by eliminating the time and reagents required to amplify DNA. Furthermore, sequencing single molecules of native DNA could allow us to detect the epigenetic modifications otherwise lost during the amplification process. Finally, a single-molecule sequencing method would not rely on the in-phase signal of a clonal cluster and so would have no intrinsic limit on the read length.

My work in the Gundlach lab has focused on developing, improving, and applying one particularly promising third generation, single-molecule sequencing technology: nanopore DNA sequencing. In chapter 2, I introduce and describe nanopore sequencing technology.

## Chapter 2

# NANOPORE DNA SEQUENCING

Chapter 1 discussed second generation DNA sequencing technologies, and outlined their primary limitations. It also suggested that a single-molecule sequencing technology could address many of these limitations. In this chapter, I will go into depth on nanopore DNA sequencing, an emerging single-molecule sequencing technology with the potential to address many of the problems with existing sequencing methods. First proposed in 1996 [36], nanopore sequencing has overcome numerous hurdles over the past two decades en route to becoming a fully realized DNA sequencing technology. I will review the progress of nanopore sequencing as it has progressed from idea to reality, outline the challenges already overcome, and discuss the remaining challenges I hope to address in this work.

### 2.1 *Basic Concept*

The basic nanopore sequencing device consists of two wells separated by an impermeable membrane (Fig 2.1a). The two wells, termed *cis* and *trans*, are each filled with a buffered, conductive electrolyte solution (e.g. potassium chloride, KCl). A single nanometer-scale pore in the membrane—called the nanopore—provides the sole conductive pathway between the two wells. When a voltage is applied across the membrane, an ionic current flows through the nanopore. DNA molecules, which are poly-negatively charged in solution, are drawn into and through the pore by the voltage, moving from *cis* to *trans*. While passing through the pore, the DNA partially blocks the ionic current flow. Specifically, the extent of the ionic current blockage is primarily influenced by the nucleotides present within the pore’s narrowest region, or constriction. The fundamental idea underpinning nanopore sequencing is that the different chemical characteristics of the nucleotides would cause different nucleotides

to block different amounts of the ionic current (Fig 2.1b). By measuring the fluctuations in the ionic current during DNA translocation through the pore, we could read off the sequence of nucleotides as the DNA moves through.

A nanopore sequencing device as described above would have the potential to address several of the limitations of SGS [37]. Such a device would sequence DNA by measuring a single copy of the target DNA strand, and so would have no intrinsic limitation to its read length. Additionally, this device would sequence DNA by sensing the chemical differences between the 4 nucleobases; a device sensitive to these differences should also be able to detect the chemical differences characterizing epigenetically-relevant modified bases such as 5-methylcytosine. Finally, DNA translocation through the nanopore could proceed much faster than the extend/deblock cycle of SBS, allowing much faster speeds for individual reads and enabling dramatically improved sample-to-answer time lines.

## **2.2 *Choosing a Nanopore***

The first key ingredient in a functioning nanopore sequencing device is the nanopore itself. In order to actually sequence the DNA, the nanopore in the sequencing device must have certain crucial features. The first requirement is that the nanopore have the correct dimensions to sense the subtle chemical differences between the nucleotides along DNA. Ideally, the nanopore would have a single narrow region approximately the dimensions of a single nucleotide of ssDNA: just over 1 nm in diameter, and less than 1 nm in height. The nanopore must also be atomically reproducible. A robust sequencing platform must measure the same signal for the same DNA sequence on different devices on different days. Even subtle differences between the nanopores in different devices would lead to each device producing a different signal for the same DNA sequence, rendering consistent DNA sequencing difficult. Various different nanopores have been explored as potential candidates for sequencing. The candidates can be easily categorized into two main types: solid state nanopores and biological protein nanopores.

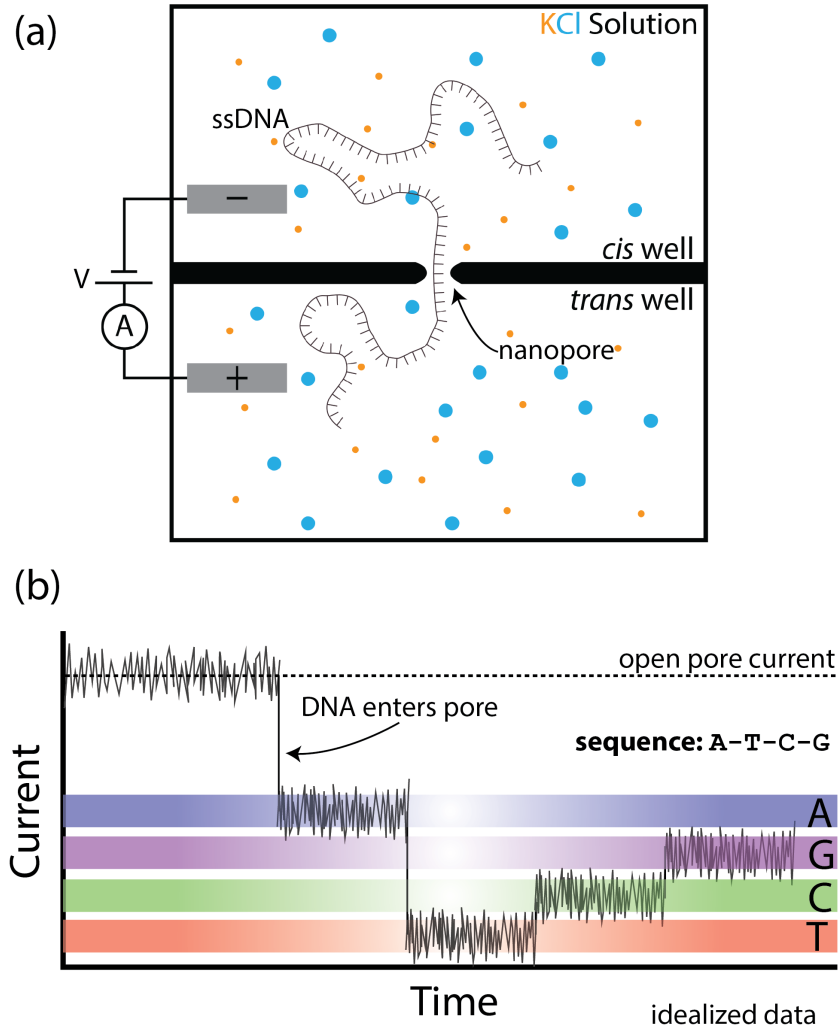


Figure 2.1: Basic nanopore sequencing scheme **(a)** Cross-sectional view of a simplified nanopore sequencing device. An impermeable membrane separates the *cis* and *trans* wells, with a single nanopore providing a conductive pathway across the membrane. We apply a voltage across the membrane and measure the ionic current through the pore as DNA moves through from *cis* to *trans*. **(b)** Idealized nanopore sequencing data. In a functioning nanopore sequencing device, the observed signal may be as follows. The open pore (no translocating DNA) displays a high characteristic open pore current. Once DNA enters the pore, the ionic current drops. The nucleotide-by-nucleotide translocation of the DNA through the pore results in a series of different ionic current values as the different nucleotides block the ionic current to different extents. In the simplest case, we may observe exactly 4 distinct current values—one for each of the 4 nucleotides. The DNA sequence could then be decoded from the series of observed current values, in this case A-T-C-G.



### 2.2.1 *Solid State Nanopores*

Solid state nanopores are man-made pores formed by drilling a small hole through a thin material such as silicon nitride, molybdenum disulfide, or graphene [38, 39, 40, 41, 42, 43]. This type of nanopore offers several advantages. The membranes in which solid state pores are drilled are quite strong, and so can be long-lasting under the constant application of voltage. Additionally, their fabrication lends itself well to massive parallelization and integration into a commercial sequencing device. However, modern nanofabrication techniques are not yet capable of achieving the scale and consistency required to produce good solid state pores for DNA sequencing. Without the ability to reliably fabricate nanopores small enough to probe DNA and with atomically consistent features, DNA sequencing using solid state nanopores is not yet realistic. However, their considerable long term advantages in robustness and ease of large-scale fabrication and parallelization make them an exciting avenue for continued research.

### 2.2.2 *Biological Protein Nanopores*

Another type of nanopore harnesses nature’s fabrication prowess in lieu of relying on man-made pores. There exist a broad class of biological protein pores: naturally occurring biomolecules that form small pores in cell membranes. Many of these protein pores biologically function as ion channels, allowing and regulating the flow of ions into and out of a cell. The immediate advantage of using a protein nanopore to sequence DNA is their atomic reproducibility. Nature requires that each copy of a protein be atomically identical to each other copy—without this property, life could not exist! By using protein pores to sequence DNA, we can make use of billions of years of advances in precision fabrication and quality control that we can not yet replicate in our own nanofabrication methods.

Biological protein nanopores are not without their drawbacks. These pores form channels within lipid bilayer membranes<sup>1</sup>, which are far less stable over long time periods and under

---

<sup>1</sup>Lipid bilayer membranes are synthetic cell membranes. They are composed of two oppositely-oriented

sustained voltage than solid state membranes. The process of establishing a single protein pore in a lipid bilayer membrane <sup>2</sup> is not as easily parallelized as the fabrication of solid state nanopores. Finally, protein pores do not afford the customizability available in solid state pores. As we are borrowing from nature’s design by using biological protein pores, we can’t design the pore structure from scratch and must instead hope to find a suitable pore for DNA sequencing within the catalog of existing proteins. The burgeoning field of *de novo* protein design <sup>3</sup> could to some day allow us to make our own entries in the catalog of available protein pores and design from scratch our ideal protein nanopore for sequencing [44]. For now however, both the nanofabrication techniques needed for sequencing-capable solid state pores and the *de novo* protein design techniques needed for usable synthetic protein pores are a ways away. Without the ability to create our own custom nanopores for DNA sequencing, we must hope to discover a suitable nanopore already existing in nature.

Much of the pioneering work in developing nanopore sequencing was done using the  $\alpha$ -hemolysin nanopore ( $\alpha$ HL, Fig 2.2a), a pore-forming protein toxin secreted by the bacteria *Staphylococcus aureus*. Early nanopore sequencing work using  $\alpha$ HL showed that DNA could indeed translocate through the pore when driven by a voltage, and that these translocations could be detected [36]. Later work in which DNA was held statically within the  $\alpha$ HL pore showed that the DNA sequence had a measurable effect on the on the current through the pore [45] and even showed that cytosine and 5-methylcytosine could be distinguished from each other [46]. However, nanopore sequencing using  $\alpha$ HL would ultimately prove difficult as its long constriction (Fig 2.2a) is simultaneously sensitive to many bases along

---

monolayers of lipid molecules. Lipids are composed of a hydrophilic head group and a hydrophobic tail. In aqueous solution, the lipid molecules will arrange themselves into a bilayer to as to bring the head groups in contact with water while shielding the tails.

<sup>2</sup>Protein pores will spontaneously insert themselves into a lipid bilayer. However, insertion is a random process. To isolate a single channel, the user must recognize the characteristic jump in conductance associated with a single pore insertion, then flush out all additional pores present in solution prior to a second insertion. In a massively parallelized nanopore sequencing device, the fraction of wells with a working single channel insertion will be poisson limited.

<sup>3</sup>In *de novo* protein design, researchers “write” a protein from scratch, customizing the sequence of amino acids in order to achieve an objective functionality.

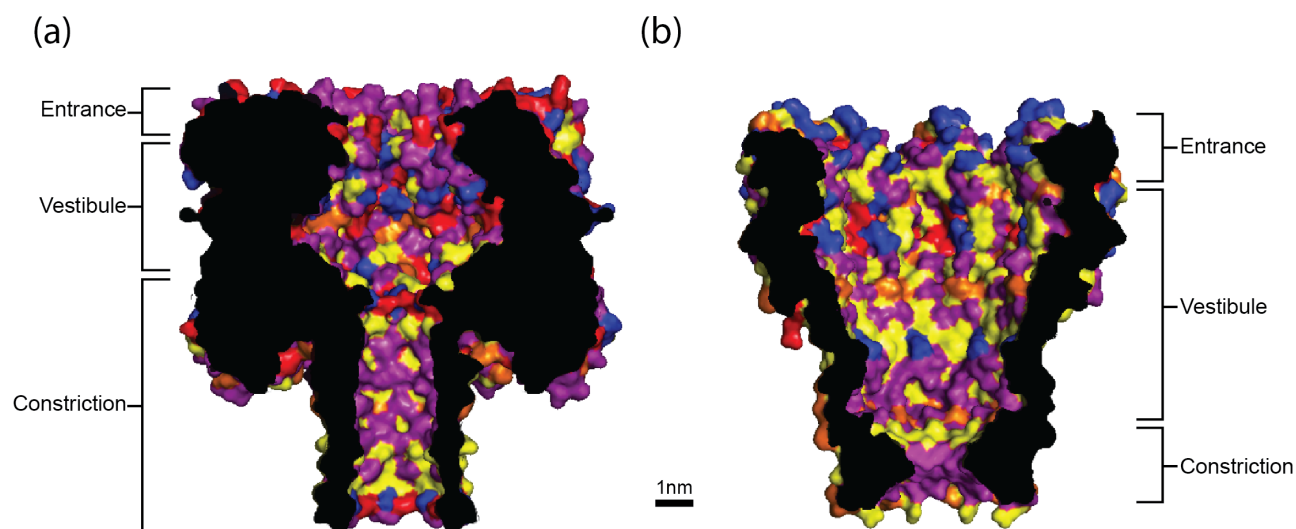


Figure 2.2:  $\alpha$ HL and MspA. Both (a) the  $\alpha$ HL pore from *Staphylococcus aureus* and (b) the MspA pore from *Mycobacterium smegmatis* have both been used for nanopore sequencing. The main advantage of MspA over  $\alpha$ HL is its single short constriction, in contrast to the long uniform constriction of  $\alpha$ HL. The space-filling representations of the two pores' crystal structures are shown here. This figure has been modified from [48]. Colors correspond to amino-acid classes: negatively charged amino acids are shown in blue, positively charged shown in red, polar are shown in purple, non-polar aromatic are in orange, and non-polar aliphatic are in yellow.

the translocating DNA [47]. A pore with a shorter constriction would need to be found to make nanopore sequencing a reality.

### 2.2.3 *Mycobacterium smegmatis* porin A

In 2008, my research group pioneered the use of a new protein pore for nanopore sequencing: *Mycobacterium smegmatis* porin A (MspA, Fig 2.2b) [49, 48]. MspA is an octomeric membrane protein from the *Mycobacterium smegmatis* bacterium. Compared to  $\alpha$ HL, MspA has a much shorter constriction (Fig 2.2). Particularly, MspA has a single narrow constriction only  $\sim 1.2$  nm in diameter at its narrowest point and only  $\sim 0.6$  nm in height, making it

ideally suited to probe single nucleotides of ssDNA <sup>4</sup>. Despite this ideal geometry, naturally-occurring (“wild type”) MspA was not immediately suited for to DNA sequencing. The presence of negatively-charged aspartic acid residues in the constriction prevented ssDNA from translocating through MspA. However, by mutating the negatively charged residues in the constriction to neutral asparagines, DNA translocation through mutant MspA was achieved [48]. Further mutagenesis continued to optimize MspA for DNA sequencing applications by increasing the capture rate of ssDNA into the pore <sup>5</sup>. Now armed with an atomically reproducible nanopore capable of translocating ssDNA and with the correct geometry to probe single nucleotides, the stage was set to begin characterizing the ionic current signal of ssDNA moving through MspA.

### **2.3 Controlling DNA Translocation**

Free ssDNA moves through MspA at around 2-10 nucleotides per  $\mu s$  [48]. At such a high rate of translocation, it is not possible to resolve the sequence-dependent ionic current fluctuations caused by the various nucleotides and actually sequence the DNA <sup>6</sup>. A method to slow down DNA motion through the nanopore is required if we are to sequence DNA.

Once again, nature provided a solution. In collaborative work between the nanopore group here at the University of Washington and researchers at UC Santa Cruz, a DNA-processing enzyme was used to control DNA motion through a nanopore [51] [52]. This first demonstration of enzyme-actuated nanopore DNA sequencing used the  $\Phi 29$  DNA polymerase enzyme ( $\Phi 29$  DNAP) as a molecular motor to move DNA through the pore in stochastic,

---

<sup>4</sup>ssDNA (section 1.1) is 1.2 nm in diameter, with an inter-nucleotide spacing of 0.69 nm

<sup>5</sup>The final mutant MspA used for DNA sequencing in our lab has a total of 6 point mutations relative to wild type MspA: D90N/D91N/D93N/D118R/E139K/D134R. D denotes aspartic acid (negatively charged), N denotes asparagine (neutral), R denotes arginine (positively charged), E denotes glutamic acid (negatively charged), and K denotes lysine (positively charged).

<sup>6</sup>Some researchers are working on developing advanced electronics with sufficiently high bandwidth and low noise to sequence freely translocating DNA [50]. However, substantial further research is required before such an approach will be feasible.

discrete, single-nucleotide steps <sup>7</sup> (Fig 2.3a). With DNA controlled by  $\Phi$ 29 DNAP, the ionic current signal displays a series of distinct states each characterized by a well-defined mean (Fig 2.3b). Transitions between states are caused by single-nucleotide steps by the  $\Phi$ 29 DNAP. The state durations are a product of the stochastic stepping behavior of the motor protein and are not indicative of the translocating DNA sequence <sup>8</sup>. The ionic current time series data is optimally partitioned into distinct states using a maximum-likelihood-based change point detection algorithm (appendix B) and the duration information is discarded. This reduces the nanopore signal to a series of sequential mean ionic current values which will ultimately be used to sequence the DNA.

Repeated measurements of the same DNA sequence translocating through MspA under the control of  $\Phi$ 29 DNAP show a reproducible pattern of ionic current states (Fig 2.3c). As the motor enzyme takes one step per nucleotide, we observe one ionic current state per nucleotide in the target DNA. Aligning the known DNA sequence with the observed states, we see that certain sequence motifs correlate with specific patterns in the ionic current. Notably, thymine tends to cause lower currents, adenine higher currents, and the abasic site results in high currents not observed for any of the 4 nucleobases.

With the demonstration of enzyme-controlled motion of DNA through MspA, all the components necessary for functional nanopore sequencing were present. The MspA nanopore provides the proper geometry to sense the chemical differences of single nucleotides, and the atomistic consistence for reproducible measurements of the same DNA sequence. Concurrently, the motor protein controls DNA translocation to proceed slow enough to resolve the

---

<sup>7</sup>The  $\Phi$ 29 DNAP was used in two different modes in this work. In the first mode, termed “stripping”,  $\Phi$ 29 DNAP was used as a physical brake, slowly lowering ssDNA through the pore as it stripped through the upstream duplexed DNA from 5’ to 3’. The second mode, termed “synthesis”, had  $\Phi$ 29 DNAP working as a polymerase, pulling ssDNA up out of the pore as it polymerased a complement to the template DNA strand being sequenced.

<sup>8</sup>Although not useful for DNA sequencing, there is rich scientific information in the state durations as to the chemical kinetics of the motor protein. The detailed study of this information has been the subject of extensive parallel research within the Gundlach nanopore lab, and is the foundation for the burgeoning single-molecule biophysics tool “Single-molecule Picometer Resolution Nanopore Tweezers”, or SPRNT [53, 54, 55].

nucleotide-by-nucleotide fluctuations in the ionic current caused by the chemical differences between the bases. Together, the motor enzyme and nanopore form a system that generates reproducible ionic current signals with the single-nucleotide resolution necessary to sequence DNA. The final remaining hurdle to realizing the goal of nanopore sequencing was to determine the relationship between the observed ionic current states and the translocating DNA sequence <sup>9</sup>.

## 2.4 *Relating Ionic Current with DNA Sequence*

To sequence DNA using a nanopore, we must have a way of decoding an observed ionic current signal into an inferred DNA sequence. In the idealized version of nanopore sequencing data (Fig 2.1b), this decoding was simple: each observed ionic current state was influenced by only one base, and each base had a unique and well-differentiated <sup>10</sup> mean ionic current value. Measurement of enzyme-controlled DNA translocation would result in a series of ionic current states, each with one of four distinct values corresponding to the 4 bases A, C, G, and T. DNA sequencing would simply amount to reading off the order of the 4 different current states as the 4 different bases.

As it turns out, decoding ionic current into DNA sequence is not as simple as for the idealized model. The ionic current signal of translocating DNA exhibits far more than 4 distinct mean values (Fig 2.3c). The abundance of distinct ionic current values is caused by more than one base influencing each state. In order to determine the sensitivity of each ionic current state to the translocating DNA sequence, my predecessors in the Gundlach nanopore lab conducted the following experiment. They measured the ionic current states of a DNA sequence consisting of consecutive tri-nucleotide ‘CAT’ repeats (5’-CATCAT...CATCAT3’)

---

<sup>9</sup>There exist other proposed methods of nanopore DNA sequencing than enzyme-actuated nanopore sequencing (sometimes termed nanopore strand sequencing). Methods such as nanopore sequencing by synthesis [56] are actively under development. However, enzyme-actuated nanopore sequencing is the most fully realized approach and the only technology that is commercially available. It will be the sole focus of the remainder of this dissertation.

<sup>10</sup>In this case, “well-differentiated” means that the mean ionic current state for each base was significantly removed from the means of the other bases when accounting for noise.

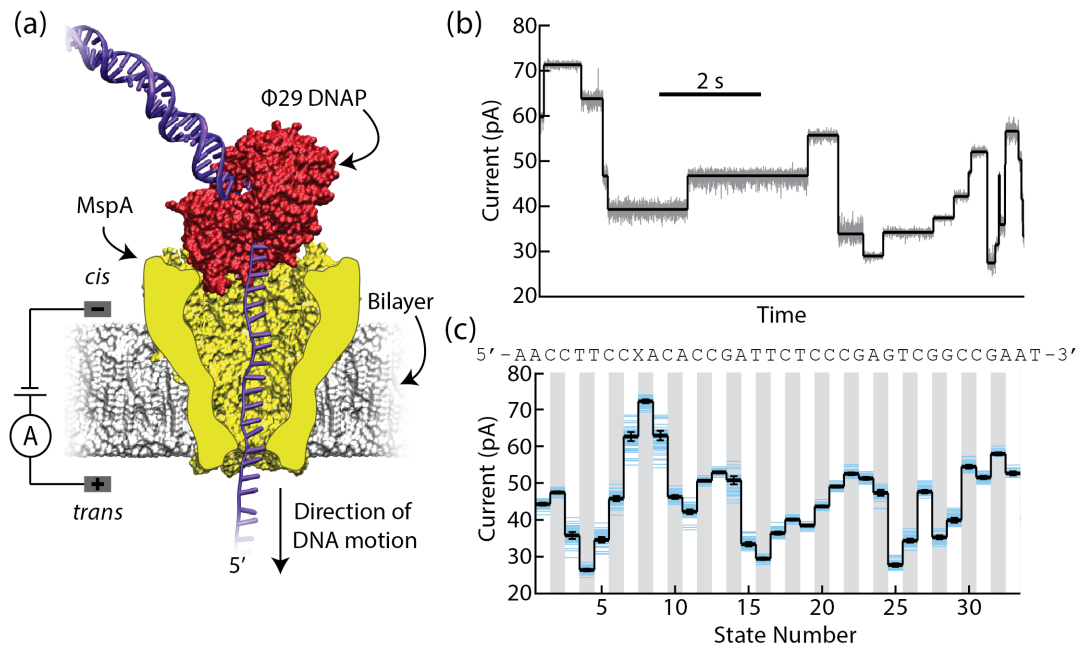


Figure 2.3: Enzyme-actuated nanopore sequencing. **(a)** In an enzyme-actuated nanopore sequencing experiment, a  $\Phi 29$  DNAP enzyme (red) controls the motion of DNA (purple) through MspA (yellow). Here the enzyme acts as a physical brake, slowly stripping from 5' to 3' through the upstream dsDNA as the voltage pulls the ssDNA through the pore. **(b)** In the raw current time series data, we see a series of distinct ion current states separated by abrupt transitions caused by enzyme steps. The raw data (downsampled to 5 kHz) is shown in gray. The black lines show the mean currents for the states found by the change point detection algorithm. **(c)** Repeated enzyme-controlled DNA translocation events of the same DNA sequence show a reproducible pattern of ionic current states. Blue lines the stacked current states for  $N = 86$  different measurements of the same DNA sequence. The generating DNA sequence is aligned above; the “X” represents an abasic site along the strand. The black line shows the overall mean ionic current at each state, with error bars showing the standard deviation of the various measurements of that state.

with a single T→G substitution in the middle of the sequence. The goal was to observe how a single base substitution in a repeating background would influence the observed pattern of ionic currents.

The repeating tri-nucleotide sequence generated a repeating series of three ionic current values (Fig 2.4). This pattern was interrupted at the location of the substitution, where the 4 states nearest the substitution all diverged from repeating pattern. Particularly, the two states directly adjacent to the substitution showed the largest change relative to the repeating background, with smaller yet still significant changes observed for the two more distant states. This observation indicated that each base influences 4 separate ionic current states. Each base has its strongest effect on the ionic current for the two states during which it is nearest the pore’s constriction, then continues to influence the ionic current less strongly at positions farther away from the constriction (either upstream towards *cis* or downstream towards *trans*). We can consider that if each base influences 4 states, each state is influenced by 4 bases. Consequently, each observed ionic current state is representative of the 4 base combination—termed 4-mer<sup>11</sup>—centered in MspA’s constriction for the duration of the state.

MspA’s multi-base sensitivity is not a product of its geometry but rather of the thermal nature of the nanopore sequencing system. Indeed, MspA’s constriction is sufficiently narrow so that only a single base will reside within the pore’s high sensitivity region at a given instant. However, DNA is not static within the nanopore throughout the duration of each state. Rather, the elastic DNA molecule is in constant thermal motion, rapidly repositioning itself relative to the pore’s constriction [57]. This thermal motion occurs on a time scale of nanoseconds—orders of magnitude faster than the discrete stepping behavior of the motor enzyme<sup>12</sup>. Consequently, the mean ionic current observed for each state is not characteristic of a single point along the DNA molecule, but is instead a time-averaging of the effects of the

---

<sup>11</sup>Throughout this dissertation, strings of k bases are referred to as k-mers.

<sup>12</sup>The motor protein stepping rate depends on various factors, including the choice of enzyme to use as the motor protein and the concentration of necessary substrate molecules in the electrolyte solution. Typical stepping rates however are on the scale of ~10-20 Hz. For a more detailed discussion of our experimental operating conditions and of enzyme behavior in these conditions, see appendix A.



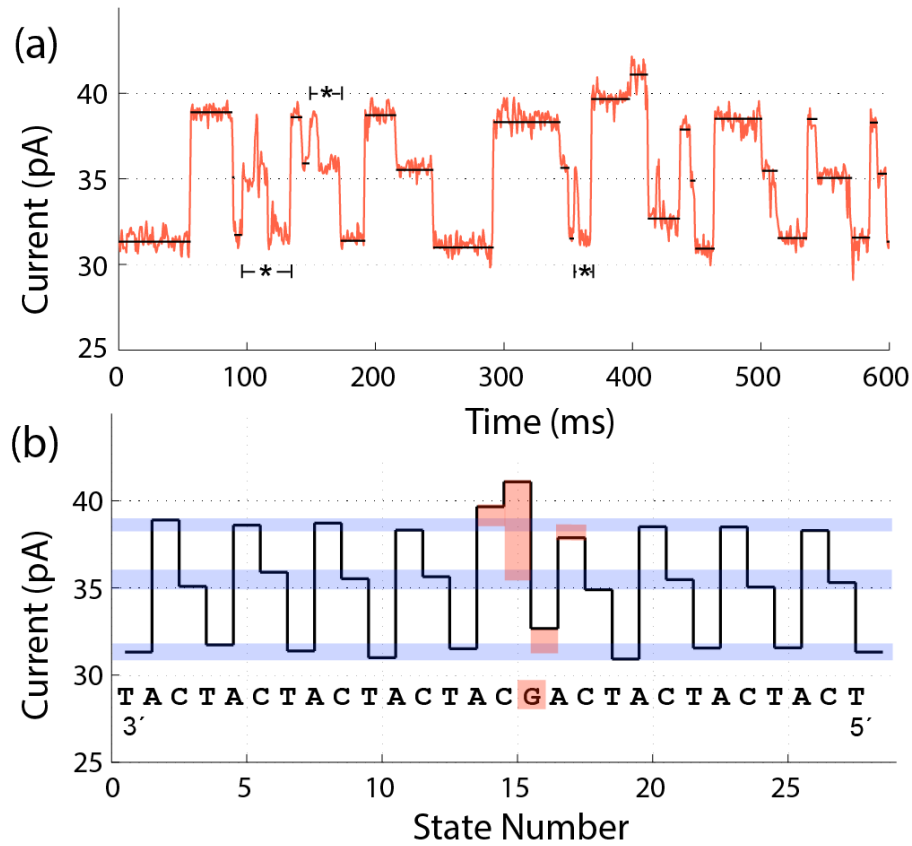


Figure 2.4: MspA sensitivity. **(a)** An example  $\Phi 29$  DNAP-controlled DNA translocation event of a repeating 5'-CAT-3' DNA sequence, interrupted by a single T $\rightarrow$ G substitution midway through the sequence. The measured ionic current (downsampled to 500 Hz) is shown in red, with the states found by the change point algorithm shown in black lines at their mean ionic current values. Asterisks mark enzyme missteps (section 2.7); states caused by missteps are not included in the extracted data below. **(b)** The extracted mean ionic current states from (a) are plotted with the associated DNA sequence (from 3' to 5') aligned below. The repeating 3 base DNA sequence generates a repeating pattern of 3 ionic current states (blue bars), which is interrupted at the site of the T $\rightarrow$ G substitution. At the substitution site, 4 consecutive states have significantly different means relative to the repeating pattern (highlighted in red). The 2 states (15 and 16) nearest the substitution show the largest change, and the two further states (14 and 17) show a smaller change. This figure has been modified from [51].

various positions along the DNA that transiently occupy the pore’s constriction as a result of the rapid thermal motion. The mean ionic current for each state is thus influenced by the several bases nearest the pore’s constriction for the duration of the state, each of which transiently occupy the pore’s high sensitivity region. The bases nearest the constriction will spend the most time centered in the sensing region and thus have the largest influence on the observed ionic current, with the more distant bases spending less time there and having a more modest effect on the measurement.

## 2.5 Sequencing with 4-mers

In 2014, my research group successfully demonstrated enzyme-actuated nanopore DNA sequencing using the MspA nanopore[58]. The multi-base sensitivity of this system does not mean that enzyme-actuated nanopore sequencing using MspA cannot sequence DNA with single-nucleotide resolution. Indeed, the same substitution experiment demonstrating MspA’s multi-base sensitivity equivalently shows this system’s exquisite sensitivity to individual bases: a single base substitution generates a significant signal (Fig 2.4).

The task of sequencing DNA with this system is to decode the DNA sequence most likely to have generated an observed series of ionic current states. This decoding requires some model relating DNA sequence to ionic current. Given with the understanding that each observed ionic current state is influenced by the 4 bases nearest the pore’s constriction, a 4-mer model mapping ionic currents to DNA sequence is a natural choice for the ionic current-to-DNA sequence model. In this 4-mer model, each 4 base combination will map to a characteristic mean ionic current. The model is described by a map of the  $4^4 = 256$  possible 4-mers<sup>13</sup> to the ionic currents typically observed when they are in the pore.

Laszlo *et al.* generated a 4-mer model by measuring the typical ionic currents for all 256 4-mers, using reads of known DNA sequence. They found that the 4-mer model was predictive for new sequences. Given an unmeasured DNA sequence, the model could be used to predict

---

<sup>13</sup>In general, a k-mer model will comprise  $4^k$  possible k-mers, as any of the 4 bases A, C, G, or T can occupy each of the k positions in the k-length “word”.

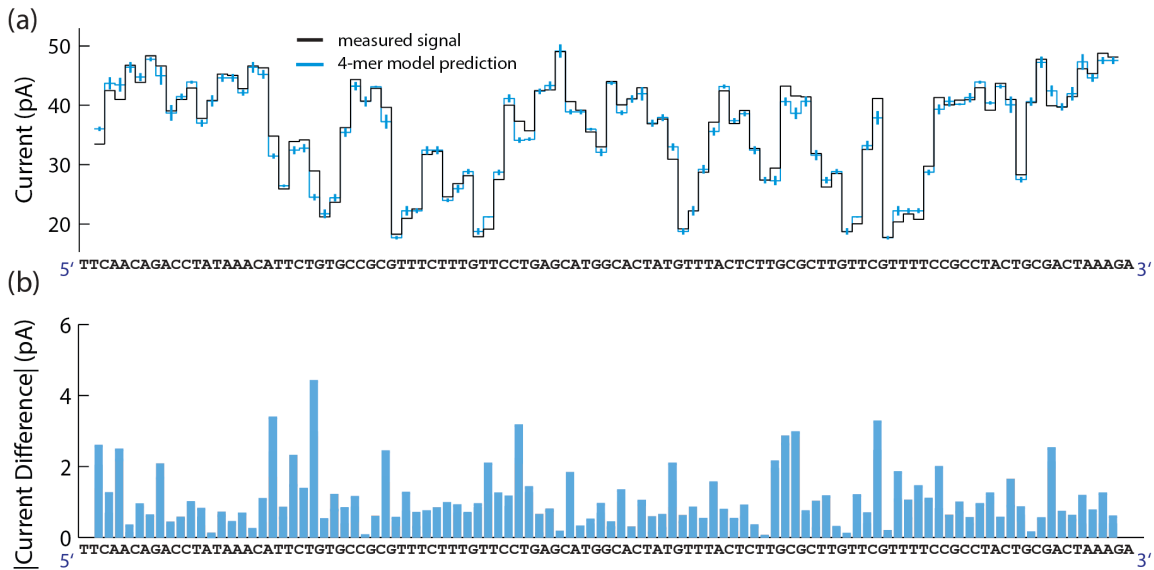


Figure 2.5: Predictive power of the 4-mer model. **(a)** The 4-mer model is used here to predict the ionic current states (blue) that would be observed for a previously unmeasured DNA sequence (error bars show standard deviation, DNA sequence is aligned below). The ionic currents measured for this DNA sequence (black) compare well with the 4-mer model prediction. **(b)** The predicted and measured ionic currents are in good agreement throughout the sequence, and in most cases differ by less than 2 pA. Figure has been modified from [58].

the ionic current states that would be observed for enzyme-controlled translocation of that DNA through MspA. The model predictions were found to match well with the observed ionic current states once the DNA was measured (Fig 2.5).

Using this model, they were able to decode the observed ionic current states into the generating DNA sequence using a hidden Markov model (HMM) [59]—a process called “base calling” [58] [60]. A detailed description of the sequencing algorithm used to decode signal into sequence in this work can be found in appendix F. The HMM sequencing algorithm works by decoding an optimal set of “hidden” states (here, the series of 4-mers) from a set of observed states (here, the measured mean ionic currents), subject to a set of allowed

transitions <sup>14</sup>. Put more concretely, a simplified HMM base calling process could proceed as follows (Fig 2.6, algorithm 1). Each measured ionic current state is compared against the 256 ionic currents in the 4-mer map to determine how well the observed state is modeled by the various 4-mers. Then, model 4-mer states are matched to each of the observed states so as to satisfy 2 conditions:

1. Only certain transitions are allowed from one state to the next. Namely, we can only step by one nucleotide at a time, so subsequent 4-mers must overlap in 3 of their 4 bases. For example, *ACCT* can only transition to one of *CCTN* where  $N \in \{A, C, G, T\}$ .
2. The model ionic currents for the 4-mers matched to each measured ionic current state should match as well as possible, subject to the above condition.

This sequencing algorithm will determine the DNA sequence most likely to have generated the observed ionic currents <sup>15</sup>. With this demonstration of enzyme-actuated nanopore sequencing, all the pieces were in place for the first nanopore sequencing devices to make their way to market.

## 2.6 Commercial Nanopore Sequencing

In 2014, Oxford Nanopore Technologies (ONT) made the MinION available to early access users [61], marking the beginning of commercial nanopore sequencing. These first commercial nanopore sequencing devices work using the same principles described above <sup>16</sup>. Specifically,

---

<sup>14</sup>The Markov property states that the transitions allowed out of a given state depend solely on the present state itself, and not on which states were visited in the past.

<sup>15</sup>This claim is exactly true in the case that the 4-mer model exactly models the relation between sequence and ionic current. This is not the case. While the 4-mer model describes the observed data well, it is an incomplete description. Measured ionic currents are influenced by more than exactly 4 bases. Although the 4 central bases are most important, bases further from the constriction have a small effect as well, rendering the 4-mer model incomplete.

<sup>16</sup>The ONT sequencers are based on the same principles as the device described above but differ in some details. Certainly, the MinION uses a different motor protein and possibly a different pore. While ONT initially used a k-mer model as described above to relate ionic current to sequence, recent advances (and the availability of large data sets) have seen k-mer models replaced by recurrent neural networks (RNNs) which “learn” the relationship between ionic current and sequence [62] [63] [64].

---

**Algorithm 1** Simple DNA sequencing using 4-mers. This algorithm gives a formalized presentation of a HMM sequencing algorithm using a 4-mer model. Here, we only allow transitions corresponding to single nucleotide steps. For instance, *ACCT* can only transition to *CCTN*, with  $N \in \{A, C, G, T\}$

---

```

1:  $\exists N$  observed ionic current states  $\{I_i\}$ ,  $i \in 1 : n$ 
2: For  $\forall$  4-mers  $\{k_j\}$ ,  $j \in 1 : 256$ ,  $\exists$  an associated ionic current  $\{\mathcal{I}_j\}$ 
3: Compute the  $n \times 256$  score matrix  $\mathbb{S}$ , where  $\mathbb{S}_{i,j} = \text{score}(I_i, \mathcal{I}_j)$   $\triangleright$  The score function
   assigns a log likelihood that a measured ionic current  $I$  matches a model ionic current  $\mathcal{I}$ 
4:  $\exists$  a  $256 \times 256$  transition matrix  $\mathbb{T}$ 
5: if  $k_i = N_1N_2N_3N_4$  and  $k_j = M_1M_2M_3M_4$  are such that  $N_2N_3N_4 = M_1M_2M_3$  then  $\triangleright$ 
    $N, M \in \{A, C, G, T\}$  denote the bases in the 4-mer
6:    $\mathbb{T}_{(i,j)} \leftarrow 1$ , meaning  $k_i$  can transition to  $k_j$ 
7: else
8:    $\mathbb{T}_{(i,j)} \leftarrow 0$ , meaning  $k_i$  cannot transition to  $k_j$ 
9: end if
10: Initialize the  $n \times 256$  alignment matrix  $\mathbb{A}$  to zeros
11: Initialize the  $n \times 256$  traceback matrix  $\mathbb{B}$  to zeros
12:  $\mathbb{A}_{(1,1:256)} \leftarrow \mathbb{S}_{(1,1:256)}$ 
13: for  $i \in 2 : n$  do  $\triangleright$  Filling the alignment and traceback matrices
14:   for  $j \in 1 : 256$  do
15:      $\text{allowed} \leftarrow \{l\}$  such that  $\mathbb{T}_{(i-1,l)} = 1$   $\triangleright$  Determine which previous states can
      transition into the present state
16:      $\mathbb{A}_{(i,j)} \leftarrow \mathbb{S}_{(i,j)} + \max(\mathbb{A}_{(i-1,\text{allowed})})$   $\triangleright$  The alignment matrix fills using the best
      scoring of the possible transitions in
17:      $\mathbb{B}_{(i,j)} \leftarrow l$  with  $l$  such that  $\mathbb{A}_{(i-1,l)} = \max(\mathbb{A}_{(i-1,\text{allowed})})$   $\triangleright$  The traceback matrix
      tracks which transition gave the best incoming score
18:   end for
19: end for
20: Find  $\max(\mathbb{A}_{(n,:)})$ 
21:  $l$  is such that  $\mathbb{A}_{(n,l)} = \max(\mathbb{A}_{(n,:)})$ 
22: Initialize output sequence  $\text{seq} \leftarrow k_l$   $\triangleright k_l = N_1N_2N_3N_4$  with  $N \in \{A, C, G, T\}$ 
23: for  $i \in n : -1 : 2$  do  $\triangleright$  Conduct traceback along the best scoring pathway
24:    $l \leftarrow \mathbb{B}_{i,l}$ 
25:    $\text{seq} \leftarrow \{N_1, \text{seq}\}$  where  $k_l = N_1N_2N_3N_4$   $\triangleright$  At each step in the traceback, append the
      correct base to the start of  $\text{seq}$ 
26: end for
   return  $\text{seq}$ 

```

---

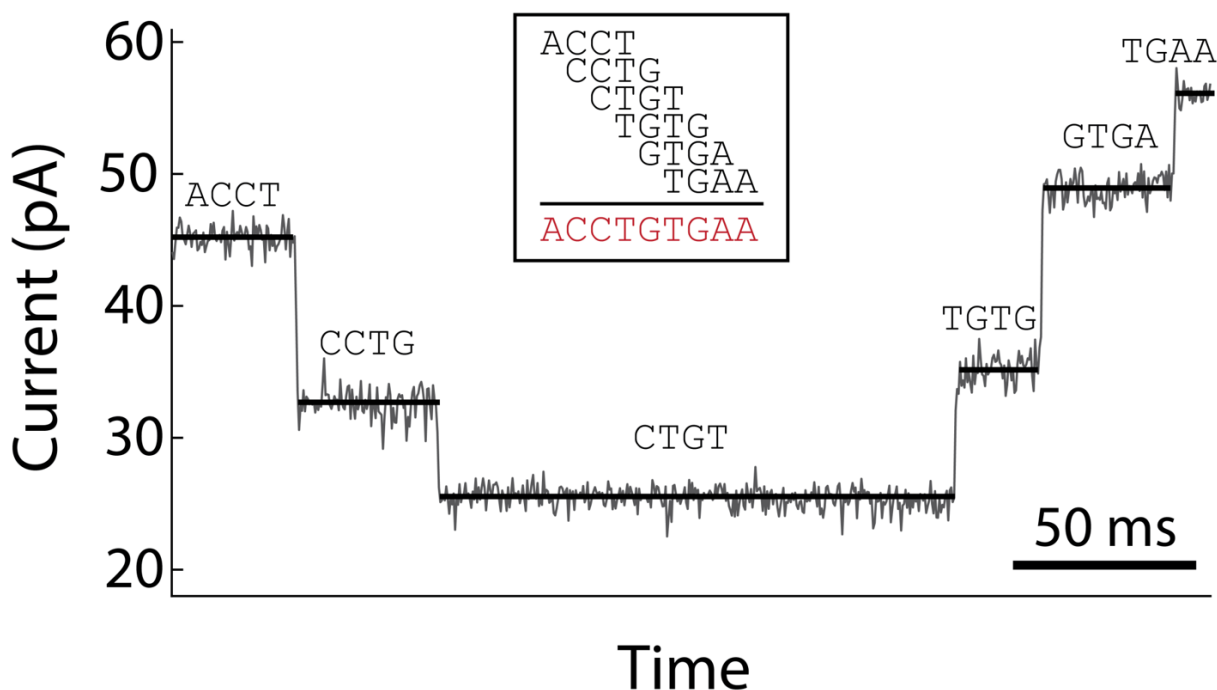


Figure 2.6: Sequencing using 4-mers. Example raw nanopore sequencing data is shown (gray, downsampled to 500 Hz) with the mean ionic current values of the distinct states overlaid (black lines). The DNA sequence generating the observed series of states is decoded by matching each state with the 4-mer known to generate a matching ionic current. In this case, the first (left-most) state may correspond to the 4-mer *ACCT* (written 5' to 3'). At each subsequent state, the enzyme has stepped the DNA forwards by 1 nucleotide, so each subsequent 4-mer must overlap with the preceding 4-mer in 3 of its 4 bases. For example, the state following *ACCT* can only correspond to one of *CCTN*, where *N* is one of *A*, *C*, *G*, or *T* (in this case, *G*). By stitching together the overlapping 4-mers of the several states, we reconstruct the generating DNA sequence *ACCTGTGAA*.

a motor protein controls the step-wise translocation of ssDNA through a nanopore under the application of a constant voltage across the membrane. The enzyme-controlled translocation yields ionic current time series data that is partitioned into distinct states, each characterized by its mean ionic current. This series of ionic current states is then decoded into the generating DNA sequence.

The milestone of the first working nanopore sequencing devices would not mark the end of nanopore technology development. Enzyme-actuated nanopore sequencing is still limited by its low single-read *de novo* sequencing accuracy. The single-read *de novo* sequencing accuracy is the accuracy of a single sequencing read of a completely unknown DNA sequence. It is important to note that a technology’s single-read sequencing accuracy does not represent the accuracy level to which a DNA sequence can ultimately be determined using that technology. In many cases—including that of nanopore sequencing—many of the single-read errors are random. Such errors drop out when multiple low accuracy reads with different random errors are combined into a consensus sequence. Through consensus sequencing, a technology with a low single-read accuracy can generate a much higher accuracy sequence. Nevertheless, single-read accuracy is a crucial benchmark for performance. Even when high accuracy can be achieved through consensus sequencing, such an approach comes at a cost of throughput. Given lower accuracy single reads, more will need to be combined to generate a satisfactory consensus sequence, increasing the time and cost of sequencing. Conversely, improvements in single-read accuracy will decrease the time and cost of sequencing as a satisfactory consensus sequence can be derived from fewer individual reads.

Initial results on commercial nanopore sequencing devices showed single-read accuracies in the low 60 percents [65]. Since these early results, various improvements have driven the single-read accuracy up into the 70 percents [65]. However, there is still a significant ways to go before nanopore sequencing accuracy is commensurate (or even comparable) with the accuracy of more established second generation sequencing technologies, which are well above 99%<sup>17</sup> [9]. The long-term path forward to fully-realized nanopore sequencers requires

---

<sup>17</sup>“Well above” refers to the error rate (100% - sequencing accuracy) rather than the accuracy. For example,

substantial improvement in the baseline single-read accuracy.

## 2.7 Sequencing Error Modes

Many of the errors hampering enzyme-actuated nanopore sequencing’s single-read accuracy are caused by two primary error modes. These two error modes are indistinguishable ionic current states and enzyme missteps. Each of these error modes is discussed individually in more detail below.

### 2.7.1 Indistinguishable Ionic Current States

As discussed above (sections 2.4, 2.5), the base calling algorithm used to decode DNA sequence from an observed ionic current signal requires a model that maps observed ionic current values to the likely generating DNA sequence. As the observed signals are a complicated function of several bases near the pore’s constriction [51], this ionic current-to-sequence model must comprise many multi-nucleotide ionic current-generating sequence “states”. For example, the 4-mer model discussed above entails 256 states for the 256 possible 4 base combinations.

For any such empirical ionic current-to-sequence model, there will be higher-order effects that are not accounted for, such as the combined influence of the other bases in the pore more distant from the constriction. These higher-order effects, along with the electrical noise intrinsic to the nanopore signal and experiment-to-experiment variations in electrolyte concentrations and temperature all conspire to introduce instance-to-instance variability in the observed signal for different measurements of the same DNA sequence state. In our 4-mer model<sup>18</sup>, the average standard deviation of the instance-to-instance ionic current values of the 4-mers is 0.95 pA (Fig 2.7a). Additionally, the ionic current values of the lowest

---

the difference between 99% and 99.9% accuracy is the difference of 1% to 0.1% in terms of error rate—an order-of-magnitude improvement.

<sup>18</sup>Specifically, the 4-mer model for the experimental conditions used for  $\Phi$ 29 DNAP-controlled DNA translocation. Experimental conditions are discussed in appendix A



(*CGTC*, 23 pA) and highest (*AGAA*, 60 pA) 4-mers in the model are separated by only 37 pA (Fig 2.7b). This 37 pA represents the entire parameter space within which the sequencer must distinguish between the 256 different 4-mer states. Inevitably, given only 37 pA of parameter space, 256 4-mers, and nearly 1 pA of variation for each 4-mer, the ionic currents of many different 4-mers will be indistinguishable within noise. Functionally, this means that many different sequences can generate states with statistically identical ionic currents (Fig 2.8, green bars). These indistinguishable signals force the base calling algorithm into under-determined decisions where it must choose between multiple possible generating sequences for an observed set of states, only one of which is correct. These under-determined decisions ultimately lead to errors in sequencing. Fundamentally, a state's mean ionic current alone does not provide enough information to unambiguously decode the signal into sequence.

Beyond forcing the base caller into under-determined and error-prone decisions, indistinguishable ionic current states can also cause errors during step finding. In some cases, two consecutive 4-mer states ( $k_1 = N_1N_2N_3N_4 \rightarrow k_2 = N_2N_3N_4N_5$ <sup>19</sup>) may both generate similar ionic currents making the transition between them difficult to find (Fig 2.8, orange diamond). Typical measurement error on a single instance of a state (distinct from the instance-to-instance variation) is  $\sim 1$  pA; transitions smaller than this typical error will be difficult to find for the change point detection algorithm. For our 4-mer model, 114 out of the 1020 (11.2%) possible step transitions<sup>20</sup> lie below the 1 pA threshold for confident transition finding (Fig 2.7c). Transitions missed during change point detection ultimately lead to missing states in the final signal passed to the base caller, potentially causing deletions (too few called bases) in the final sequence.

---

<sup>19</sup>In the context of DNA sequences,  $N$  denotes any one of the 4 bases:  $N \in \{A, C, G, T\}$ .

<sup>20</sup>The 1020 possible transitions represent the 256 4-mers with 4 transitions each, less the 4 homopolymer transitions (i.e.  $AAAA \rightarrow AAAA$ ).

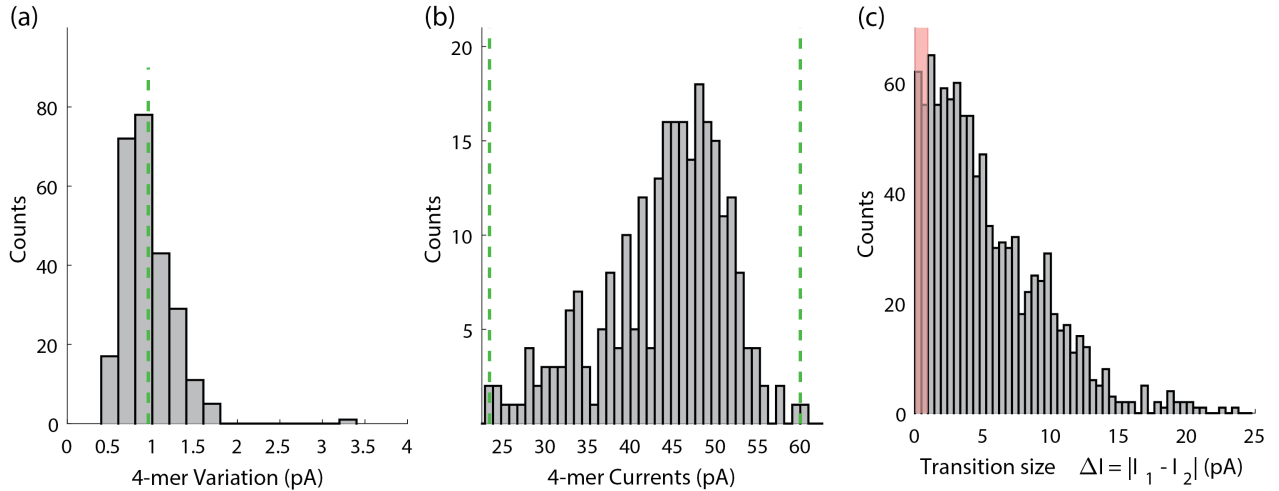


Figure 2.7: Variation between 4-mer instances. **(a)** The histogram shows the average instance-to-instance variation of the 4-mer states in our model. The green dashed line shows the mean 4-mer variation of 0.95 pA. **(b)** The histogram of 4-mer ionic current values in the 4-mer model shows that many different sequences produce nearly identical ionic currents. Bins are 0.95 pA wide—the average uncertainty in each 4-mer value. 4-mers residing within the same bin will be difficult to distinguish. Dashed green lines mark the ionic currents of the lowest 4-mer (*CGTC*) and highest 4-mer (*AGAA*). **(c)** The histogram shows the current differences between possible consecutive 4-mers ( $N_1N_2N_3N_4 \rightarrow N_2N_3N_4N_5$ ). Transitions separated by less than the typical measurement error ( $\sim 1$  pA, shaded red box) will be difficult to correctly identify and are likely to be missed.

### 2.7.2 Enzyme Missteps

Irregular stepping by the DNA-controlling motor enzyme are the second primary error mode for nanopore sequencing. Ideally, the enzyme would move DNA unidirectionally through the pore in discrete steps of uniform length. However, the stochastic stepping of real enzymes frequently diverges from this ideal behavior [66, 58, 55]. In addition to uniform forward steps, “backsteps” can occur when the enzyme backtracks to a previously-observed position along the DNA. Backsteps introduce extra states into the observed signal as we read the same DNA position multiple times (Fig 2.8, red stars). Additionally, “skips” can occur when multiple forward steps take place in quick succession too fast to resolve the intermediate step (or steps). Skips lead to missing states in the observed signal.

The existence of these irregular enzyme steps means that the observed time order of the ionic current states does not necessarily match the sequence order of the DNA generating them. This mismatch complicates the decoding process, as we now must consider many more possible transitions than in the case of uniform forward stepping. As discussed previously, a single forward step from some 4-mer can only take us to one of 4 possible new 4-mer states. For example, a forward step from  $k_1 = ACGT$  can only take us to  $k_2 = CGTN$ . However, if we are to consider the possibility of single base skips as well as steps, we now have 20 possible new 4-mer states. Returning to the previous example, we must still consider the step transitions from  $k_1 = ACGT$  to  $k_2 = CGTN$  (4 possibilities) and also consider the single skip transitions to  $k_2 = GTN_1N_2$  (16 possibilities). Further accounting for backsteps and larger (multi-base) skips continues to expand the list of possible transitions.

By expanding the list of allowed transitions, enzyme missteps expand the list of possible sequences that could have generated the observed ionic current states. The larger the list of possible sequences, the harder it becomes for the base caller to accurately identify the correct sequence. Thus, enzyme missteps reduce sequencing accuracy by adding and removing states from the signal, forcing the base caller into a difficult expanded decision space. The fundamental issue at the core of this error mode is that transition types cannot be easily inferred

from the nanopore signal. Without a direct indicator in the signal to label transition types, the base caller must account for all possible transitions, increasing the rate of erroneous base calls.

## **2.8 A Foundation for Improvement**

Through the work of my research group and others, operational enzyme-actuated nanopore sequencing has been realized, with the first commercial nanopore sequencing devices now available. The next hurdle on the path towards nanopore sequencing realizing its full potential is its low single-read *de novo* sequencing accuracy. The low accuracy is primarily caused by two distinct error modes: indistinguishable ionic current states and enzyme missteps. My work in the Gundlach nanopore lab has focused on improving nanopore sequencing by raising its sequencing accuracy as well as designing better ways of answering important sequencing questions with low accuracy reads. Chapter 3 will describe my work towards the first of these two goals: raising the single-read *de novo* sequencing accuracy possible for enzyme-actuated nanopore sequencing.

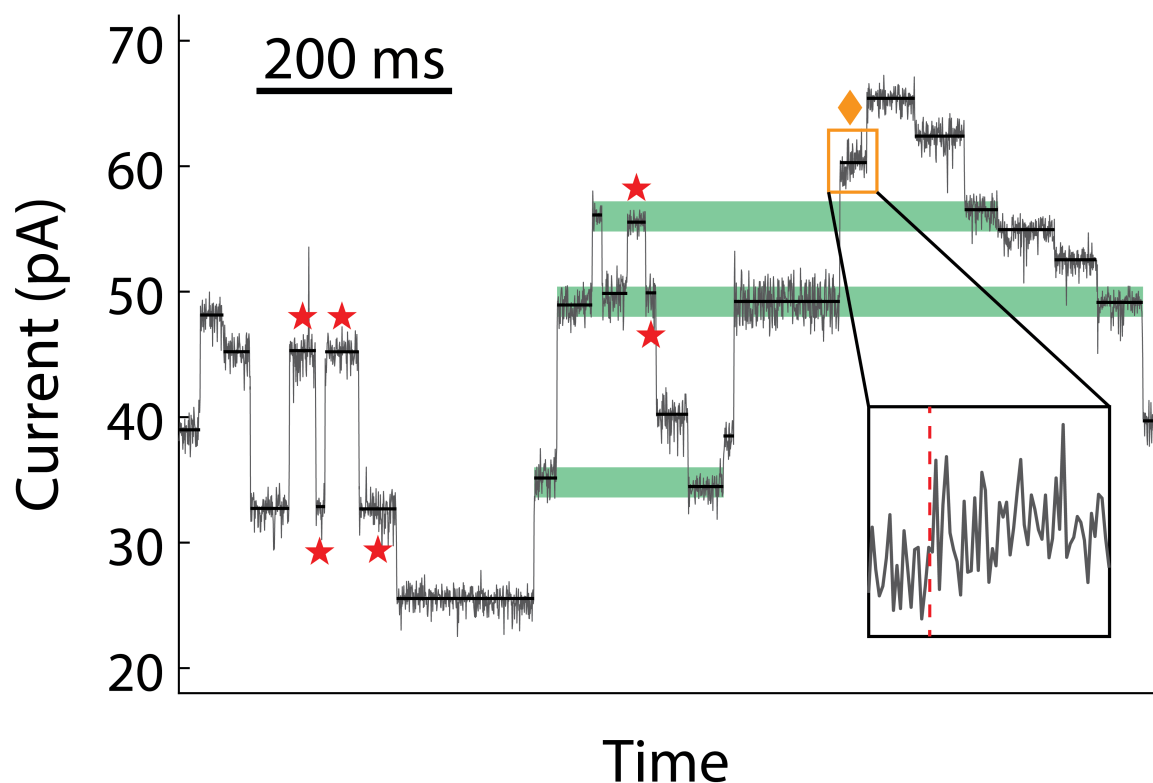


Figure 2.8: Sequencing error modes. The primary error modes limiting nanopore sequencing accuracy can be found in this example read. Raw data (downsampled to 500 Hz) is in gray, with black lines showing the mean ionic current of the found states. Green bars highlight indistinguishable states: states that were generated by different DNA sequences but with mean ionic current values that are indistinguishable from one another within noise. The orange diamond and zoomed-in inset show a location where the change point detection algorithm missed a transition. Here, two consecutive states have similar mean ionic currents, so no change point is detected at the transition between the states (red dashed line in inset). Red stars mark states caused by enzyme missteps. On multiple occasions in this read, the enzyme steps backwards, returning to a previously observed state before proceeding with processing translocation, generating extra states in the observed signal.

## Chapter 3

# VARIABLE-VOLTAGE NANOPORE DNA SEQUENCING

In chapter 2, we saw that enzyme-actuated nanopore sequencing works by using a motor protein to incrementally step DNA through a nanopore, generating a sequential series of ionic current states that will ultimately be decoded into the generating DNA sequence. We also discussed the primary outstanding limitation of this technology: its low single-read *de novo* sequencing accuracy. Many of the sequencing errors limiting the accuracy can be attributed to two primary error modes: indistinguishable ionic current states (section 2.7.1) and enzyme missteps (section 2.7.2). In this chapter, I will present my work on developing a new method of enzyme-actuated nanopore sequencing designed to directly address both of these error modes and thereby improve sequencing accuracy.

### 3.1 *Parallel Pathways to High Accuracy*

Since the advent of enzyme-actuated nanopore sequencing, researchers have pursued several parallel pathways towards improving the technique’s sequencing accuracy. Information extraction from the signal has been improved by the introduction of new base calling methods based on recurrent neural networks (RNNs) trained on the massive datasets generated by commercial sequencers [62] [63]. The quality of the signal itself has also improved through the use of new nanopores and motor enzymes with better behavior for DNA sequencing [67] [68], as well as biochemical methods that allow reading of both sense and antisense strands of the target DNA molecule in a single read [69].

Together, these two complementary approaches—clean up the signal, and improve information extraction from that signal—have led to meaningful progress in sequencing accuracy [65] [69] [68]. However, there is still significant room (and need) for improvement. A third

complementary pathway towards improvement is to reexamine the fundamental character of the nanopore signal itself, which has not yet been considered for redesign. In this work, we modify the method of control over DNA motion in the pore in order to measure a more information-rich signal that allows us to directly address the two primary nanopore sequencing error modes of indistinguishable ionic current states and enzyme missteps.

### **3.2 Shortcomings in the Signal**

The two major error modes of indistinguishable ionic current states and enzyme missteps (Fig 2.8) can both be thought of as arising from a lack of information in the ionic current signal. In the case of indistinguishable ionic current states, we have insufficient information characterizing each state. The mean ionic current alone does not adequately differentiate between all of the possible ionic current-generating sequence states (Fig 2.7). To reliably assign an observed state unambiguously to the correct signal-generating DNA sequence, we need more information to characterize each state than its mean ionic current alone. If each state had more identifying information, the sequencer’s task of decoding the DNA sequence from the observed states would become easier and fewer errors would occur.

The error mode of enzyme missteps could also be addressed by a more information-rich signal. Specifically, enzyme missteps hurt the sequencing accuracy because it is difficult to tell from the signal alone (without the aid of comparison against the known DNA sequence) which type of enzyme steps occurred where. If the type of enzyme step leading into each observed state could be inferred directly from the signal, enzyme missteps could be identified and corrected prior to sequencing and would no longer cause sequencing errors.

This understanding of not only *what* the nanopore sequencing error modes are, but also exactly *why* they are so detrimental points a way forwards to improving the sequencing accuracy. We seek a way to generate a more information-rich signal from the nanopore device. Precisely, we want each state to have more characterizing information than just its mean ionic current, and we want enzyme missteps to be directly detectable based on the signal.

### 3.3 A New Enzyme Sheds New Light

To understand how to redesign the nanopore sequencing device to generate a more information-rich signal, we need a better understanding of the signal itself. Changing the motor enzyme used to control DNA motion through MspA provided a crucial insight into the fundamental nature of the nanopore signal.

The foundational work in enzyme-actuated nanopore sequencing primarily used the  $\Phi 29$  DNAP enzyme to control DNA motion through the pore, which takes a single step per nucleotide <sup>1</sup>. The resulting signal is a series of states, each characterized by its mean ionic current, with one state corresponding to each nucleotide translocated [51] [52].

In exploring alternative enzyme options to  $\Phi 29$  DNAP for controlling DNA in nanopore sequencing, our lab discovered that the Hel308 helicase enzyme <sup>2</sup> deviates from this single-nucleotide stepping behavior [53]. The ionic current states for the same DNA sequence were measured for  $\Phi 29$ -controlled and Hel308-controlled DNA translocation through MspA (Fig 3.1). The ionic current states observed in both cases exhibited the same overall pattern of peaks and troughs, but the signal generated by Hel308-controlled translocation resulted in twice as many states marking out this overall pattern than the signal from  $\Phi 29$ -controlled translocation.

The integer ratio between the number of states observed for the two enzymes and the similarity in the qualitative structure of the ionic currents indicate that the two enzymes are taking different size steps along the DNA. Indeed, further kinetic analysis confirmed that the Hel308 helicase takes two distinct steps per nucleotide, with each step approximately half a nucleotide in length [53] [55]. This discovery pointed toward an immediate, simple path to improving enzyme-actuated nanopore sequencing: replace  $\Phi 29$  with Hel308. Sequencing

---

<sup>1</sup>Commercial nanopore sequencing devices make use of different motor enzymes, but to our knowledge all the enzymes in use take single-nucleotide steps.

<sup>2</sup>Specifically, the enzyme used here is the Hel308 helicase from *Thermococcus gammatolerans*, a thermophilic and radiation-tolerant archaea found in deep ocean vents. Where applicable, we abbreviate this as “tga”.



data generated using Hel308 has two measurements per nucleotide, rather than one. This amounts to additional information in the signal characterizing each base, as we now have two mean ionic currents per base, rather than just one.

Simply changing to a better motor enzyme is not the “crucial insight” alluded to earlier. In addition to pointing us to a better enzyme, the discovery of Hel308 half-steps hinted at the fundamental nature of the nanopore signal. The fact that the half-nucleotide steps of Hel308 interpolate smoothly between the full-nucleotide steps of  $\Phi 29$  indicates that the step-wise ionic current signal is in fact a discrete sampling of some smooth underlying profile. If this smooth profile indeed exists, and if we could find a way to access it during a nanopore sequencing experiment, it could provide the additional information necessary to dramatically improve sequencing accuracy.

### 3.4 Voltage Shifts DNA Position

In an effort to confirm the hypothesis that the ionic current signal of the DNA in the pore is a smooth function of DNA position, we sought a way to further sample the inter-nucleotide signal.

DNA’s elasticity in response to an applied force [70] [71] [72] [73] provides a convenient method to probe the DNA’s ionic current profile at sub-nucleotide intervals. In an enzyme-actuated nanopore sequencing experiment, the applied voltage exerts a force on the DNA threaded through the pore, pulling it towards the *trans* well. Meanwhile, the DNA is held static at its other end by the motor enzyme. Thus, the applied voltage stretches the segment of DNA between the anchor point at the enzyme and the pore’s constriction (Fig 3.4a) <sup>3</sup>.

As the DNA stretches in response to the voltage, the number of nucleotides spanning the distance between the anchor point at the enzyme and the pore’s constriction changes. Consequently, the DNA will be centered slightly differently within the constriction at different voltages (Fig 3.4). By changing the voltage, we can change the DNA’s registration

---

<sup>3</sup>Appendices C.4 and C.5 give a complete accounting and analysis of DNA stretching within MspA in response to voltage.

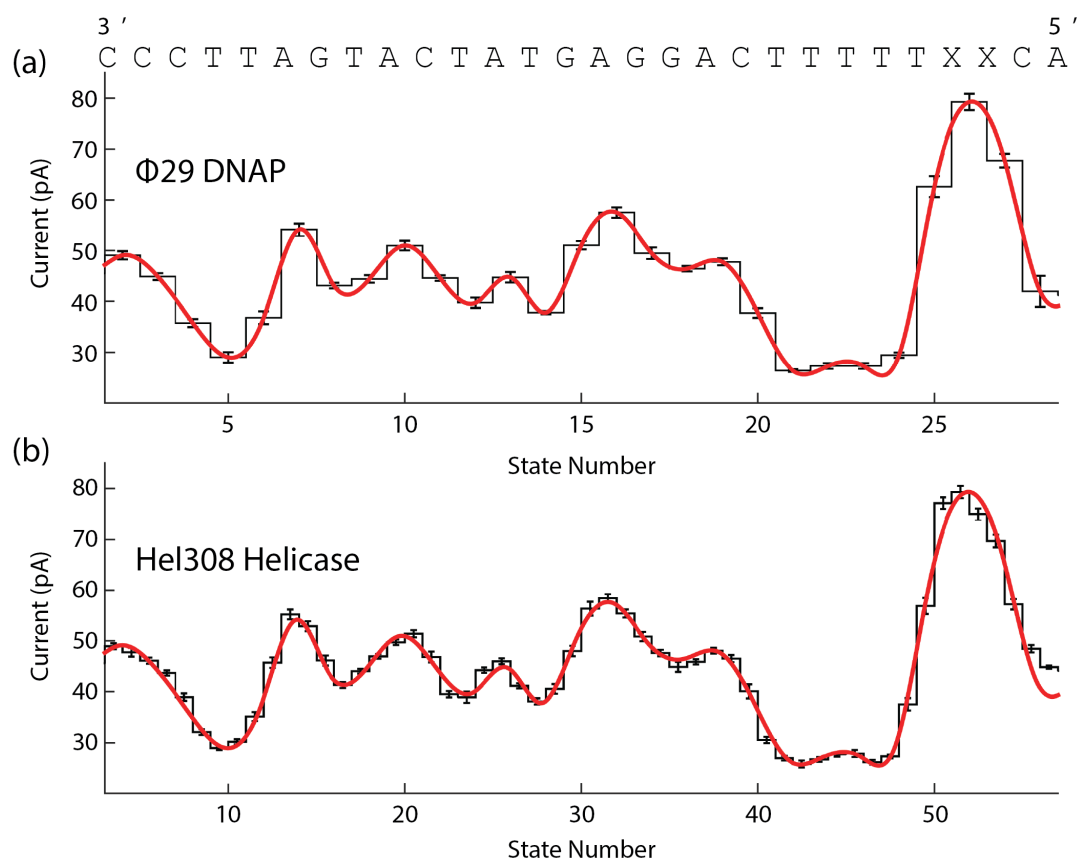


Figure 3.1:  $\Phi 29$  DNAP vs. Hel308 Helicase. **(a)** The consensus pattern of ionic current states measured using  $\Phi 29$  DNAP to control the DNA for the above DNA sequence are shown in black, with the red spline showing the cubic spline fit to the mean ionic current values. **(b)** The same DNA sequence was measured, but using the Hel308 helicase enzyme. The consensus measured states exhibit the same pattern of peaks and troughs, and are well fit by the same spline (red) as the  $\Phi 29$  DNAP states. However, we observe twice as many total states. This figure has been adapted from [53].

relative to the constriction. Thus, by conducting multiple enzyme-mediated DNA translocation experiments of the same DNA sequence at different voltages, we can take an ensemble measurement of the inter-nucleotide ionic current profile of the nanopore signal.

We measured the ionic current states for  $\Phi 29$ -controlled reads of the same DNA sequence at a variety of voltages between 100 and 200 mV. At each distinct voltage, we observed the same qualitative set of features (i.e. peaks and troughs), but with the locations of the maxima and minima shifted relative to the other voltages, in addition to a general increase in the observed ionic currents at higher voltages (Fig 3.4c). Once the overall increase of the ionic currents at higher voltages was removed by converting from current to conductance<sup>4</sup>, the shift in the features between the different voltages is easier to see (Fig 3.4d). The various voltages can be shifted horizontally to align all of the myriad sets of conductance measurements along a single smooth curve (Fig 3.4e).

This confirms the hypothesis that the ionic current (or conductance<sup>5</sup>) signal for the DNA in the pore changes smoothly as a function of the DNA position. Additionally, the results of the above experiment also point a way to measuring this underlying smooth curve in a nanopore sequencing experiment. Changing the voltage from 100 to 200 mV shifts the DNA position within the pore by over a full nucleotide (Fig 3.4f). The voltage therefore affords fine control over the sub-nucleotide positioning of the DNA in MspA, while the motor enzyme provides long-range coarse control as it walks the DNA through the pore. We will use the voltage in conjunction with the enzyme to precisely control DNA motion through the pore, allowing us to probe the entire conductance profile.

---

<sup>4</sup>Actually, to “normalized” conductance. The normalized conductance is the conductance with the confounding non-DNA-sequence-dependent contributions removed, as discussed in detail in appendix C.2.

<sup>5</sup>From this point forward, I will largely refer to conductance or normalized conductance in lieu of ionic current, as we will be comparing data collected at a variety of different voltages. Conductance makes the comparison of these data more convenient.

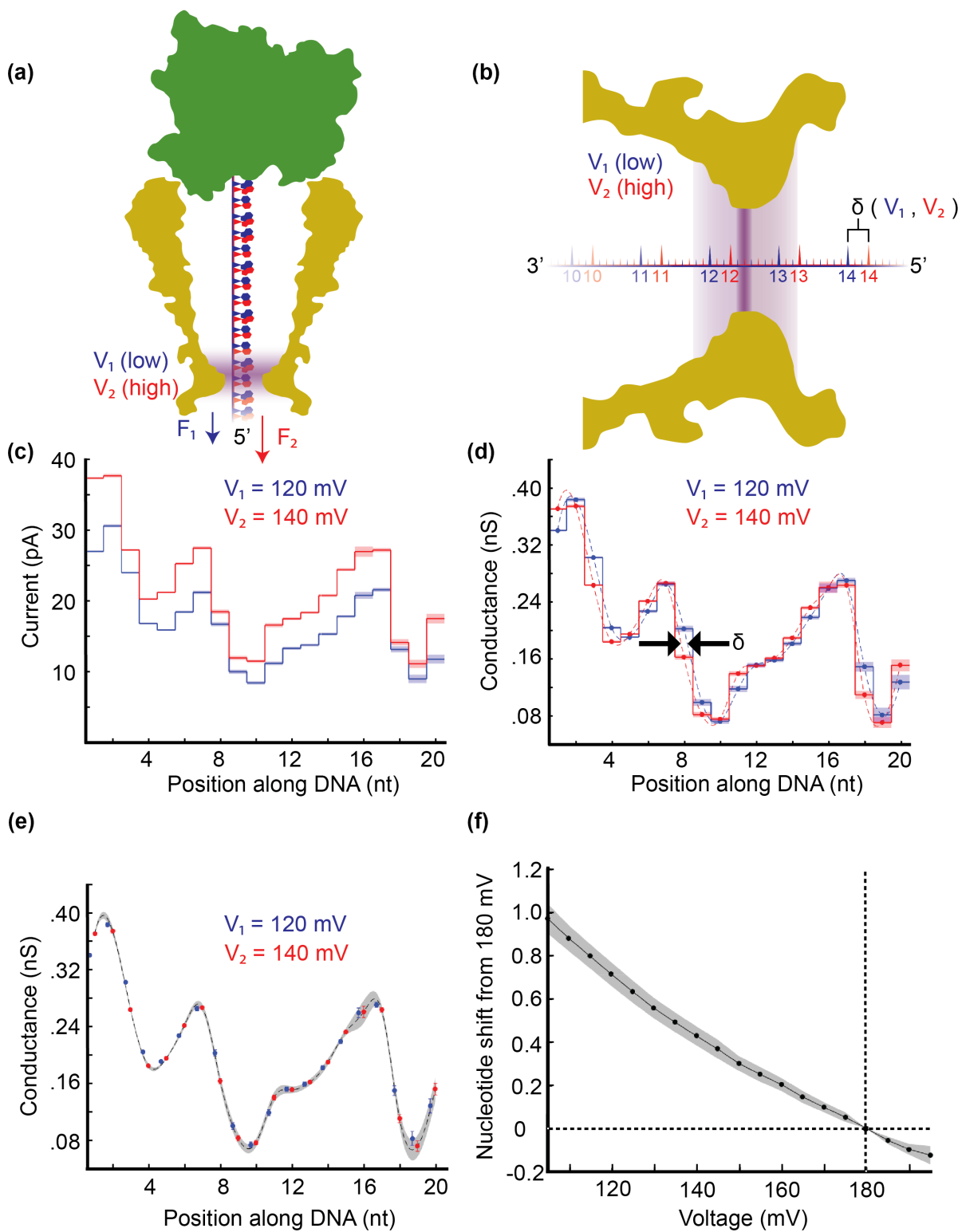


Figure 3.2: Voltage-induced DNA position shift. **(a)** The nucleotide positions at high voltage (red) are shifted down from the positions at low voltage (blue) as the higher voltage applies a larger force to the thread end of the DNA. Consequently, there are fewer nucleotides between the enzyme (green) and the pore’s constriction (shaded purple) at higher voltages. **(b)** Increasing the voltage from the smaller  $V_1$  to the larger  $V_2$  changes the time-averaged number of nucleotides between the enzyme and the constriction, positioning a different part of the DNA within the center of the constriction. **(c)** The ionic current values are extracted for 140 mV (red) and 120 mV (blue) for every step of a multi-read averaged set of consensus  $I - V$  curves. Shaded errors are S.D. **(d)** Converting current to conductance removes the scaling difference between the two measurements. A cubic spline interpolant (dashed line) to each set of states shows the same overall features, shifted by a fixed distance  $\delta$ . Shaded errors are S.D. **(e)** Shifting the 120 mV states along the x-axis places both sets of measurements on the same interpolating curve (dashed line). The shift from 120 mV to 140 mV was found to be  $0.29 \pm 0.03$  nt. Gray shading shows S.D. **(f)** The complete position shift vs. voltage curve is shown in black, with the shaded gray errors showing the one sigma confidence interval of the calculated voltage-to-position mapping. All shifts are given relative to the DNA position at 180 mV.

### 3.5 Hybrid Control

Our proposed sequencing method (Fig 3.3) will measure the continuous conductance profile as a function of DNA position in the pore through the use of a time-varying, rather than constant, applied voltage. The time-varying voltage will provide fine control over the DNA position by variably stretching the DNA, with small (1 mV) changes in the voltage precisely repositioning the DNA by  $\sim 0.01$  nucleotides (Fig 3.4f). The fine control over DNA position provided by the time-varying voltage complements the motor enzyme’s discrete stepping. The enzyme’s discrete stepping provides directed translocation over the entire length of the DNA, while the continuous repositioning of the DNA by the voltage gives us precise position control within each enzyme step.

For our variable-voltage sequencing experiments, we use the Hel308 helicase as the motor enzyme, as its half-stepping behavior provides a denser sampling of the DNA’s conductance

profile than is provided by the single-stepping  $\Phi 29$  DNAP. The voltage was applied as a 200 Hz, 100 mV peak-to-peak symmetric triangle waveform voltage, biased to an average voltage of 150 mV. The overall positive bias is necessary to keep the DNA anchored in the pore at all times. The 100 to 200 mV voltage range provides just over 1 nucleotide of total stretch (Fig 3.4).

The combination of the Hel308's half-steps and the voltage's 1 nt shift gives a complete (in fact, overlapping) sampling of the DNA's conductance profile. At each enzyme registration, the voltage shifts the DNA forwards and back several times, as the 200 Hz voltage cycling frequency is much faster than the typical enzyme stepping rate (10-20 Hz, appendix A.4). When the enzyme progresses by a half-nucleotide step, we again probe the conductance profile along a full nucleotide distance on the DNA's contour, resulting in an overlapping, complete measurement of the smooth conductance profile.

### **3.6 Variable-Voltage Data Reduction**

The data reduction is more complicated for the variable-voltage sequence data than for constant voltage sequencing. The entire data reduction process is discussed in more detail in appendices B and C, but the basic process will be covered here.

One chief difference between handling the variable-voltage data versus the constant-voltage data is that the bilayer separating the *cis* and *trans* wells acts as a capacitor, resulting in a charging and discharging current in response to the variable applied voltage. Consequently, the observed time-series ionic current data exhibits large swings, masking the DNA-sequence-dependent signal we wish to measure (Fig 3.4a).

To access the interesting, sequence-dependent signal, we must determine and remove the capacitive contribution to the measured ionic current. However, the capacitive current response is influenced not only by the voltage swing and the bilayer capacitance, but also by the resistance in the system. Inconveniently, this resistance changes with each enzyme step as the DNA moves through the pore—in fact, this variable resistance is exactly the fundamental signal we wish to measure in order to decode the DNA sequence. So, prior to calculating and

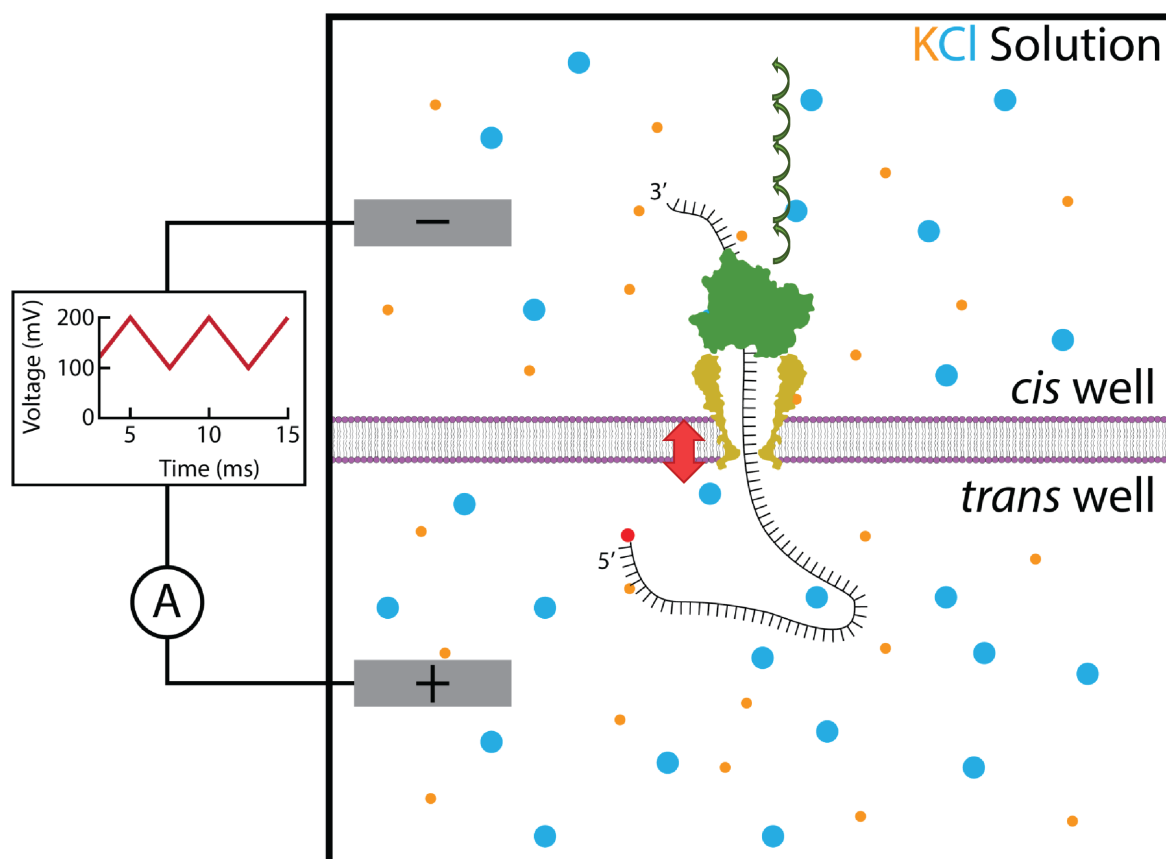


Figure 3.3: Hybrid control sequencing scheme. The variable-voltage sequencing method uses a time-varying applied voltage sweeping from 100 to 200 mV in a symmetric triangle wave at 200 Hz (left). The Hel308 helicase enzyme (green) is used to control DNA translocation through the pore in discrete steps (green arrows). The voltage simultaneously oscillates the DNA up and down in the pore, providing a fine control to complement the enzyme's coarse control (red arrow).

removing the capacitive signal, we first partition the time-series data into separate enzyme steps using our change point detection algorithm (appendix B).

Once the time series ionic current data is partitioned into distinct enzyme states, we individually calculate and remove the capacitive contribution to the observed ionic current using our capacitance compensation procedure (appendix C.1). The resulting ionic current signal is free of the large swings caused by the bilayer’s charging and discharging and reveals an oscillating pattern in phase with the applied triangle wave voltage (Fig 3.4b). The capacitance compensated data is consolidated into a set of states, each representing one enzyme step and characterized by its average current-vs-voltage ( $I - V$ ) curve (Fig 3.4c). The several complete voltage cycles completed within each enzyme step are treated as distinct measurements of the  $I - V$  response for that particular DNA registration within the pore and are averaged together to yield the state’s  $I - V$  curve.

To reconstruct the desired continuous conductance profile of the DNA, we must now account for the effects of the variable applied voltage. Changing the applied voltage changes both the overall magnitude of the observed ionic currents (higher voltage causes larger ionic current) as well as the exact position of the DNA within the constriction. The voltage dependence of the overall ionic current magnitude is removed by converting ionic currents to “normalized” conductance, as described in appendix C.2. The applied voltage can be mapped directly to a DNA position within the pore <sup>6</sup>. This mapping is given by the measured DNA shift vs. voltage curve (Fig 3.4f) and its corresponding interpolating fit (appendix C.4). By mapping the voltage to DNA position and the ionic current and voltage together to conductance, we have transformed the  $I - V$  curves for each state into segments of the conductance vs. DNA position profile (Fig 3.4d).

The final step in reconstructing the desired conductance profile is to recombine the state-by-state conductance vs. position measurements onto a single axis. We accomplish this by accounting for the Hel308 enzyme’s half-nucleotide steps. Each subsequent state’s conduc-

---

<sup>6</sup>This voltage-dependent DNA position is measured relative to the DNA position at 180 mV, the operating voltage for our constant-voltage sequencing experiments.



tance curve segment is shifted a half nucleotide right (towards the 5' end of the DNA) relative to the previous state's segment. With this accounting, the measured conductance segments for all of the measured states can be plotted together on a single position axis, revealing the DNA's smooth conductance profile (Fig 3.4e).

### ***3.7 The Smooth Conductance Profile***

The conductance profile recovered from the variable-voltage data reduction is a significantly richer signal than is measured in constant-voltage sequencing. As constant-voltage sequencing relies entirely on the motor enzyme as the sole method of control over DNA position in the pore, its signal is a sparse sampling of the conductance profile of the DNA at half-nucleotide intervals (Fig 3.5a). In this data, each state is only characterized by its mean conductance value. Furthermore, the states do not contain any intrinsic information as to their ordering. Nothing about each state indicates which state should precede or follow it in the correct ordering. The correct ordering is the ordering that reflects the order of bases along the DNA, rather than the order that reflects how the DNA was moved through the pore, which can be marred by enzyme missteps.

The variable-voltage sequencing method gives a dense sampling of the conductance profile (Fig 3.5b). Each enzyme state is now characterized by a conductance curve segment, rather than simply by a mean conductance. These curves have identifying “shape” information (slope, curvature, etc.) that is not present in the constant-voltage data. These additional features can aid in distinguishing between states that would have been indistinguishable based solely on their means.

The variable-voltage states also contain information as to their correct ordering. At each enzyme step, we measure a set of positions along the DNA that overlaps with both the previously- and subsequently-measured sets of positions. As subsequent enzyme steps each sample overlapping portions of the DNA's conductance profile, states separated by a single Hel308 half-nucleotide steps should have overlapping conductance curves. Correctly ordered states should overlap with both their predecessor and successor states—if they don't, it means

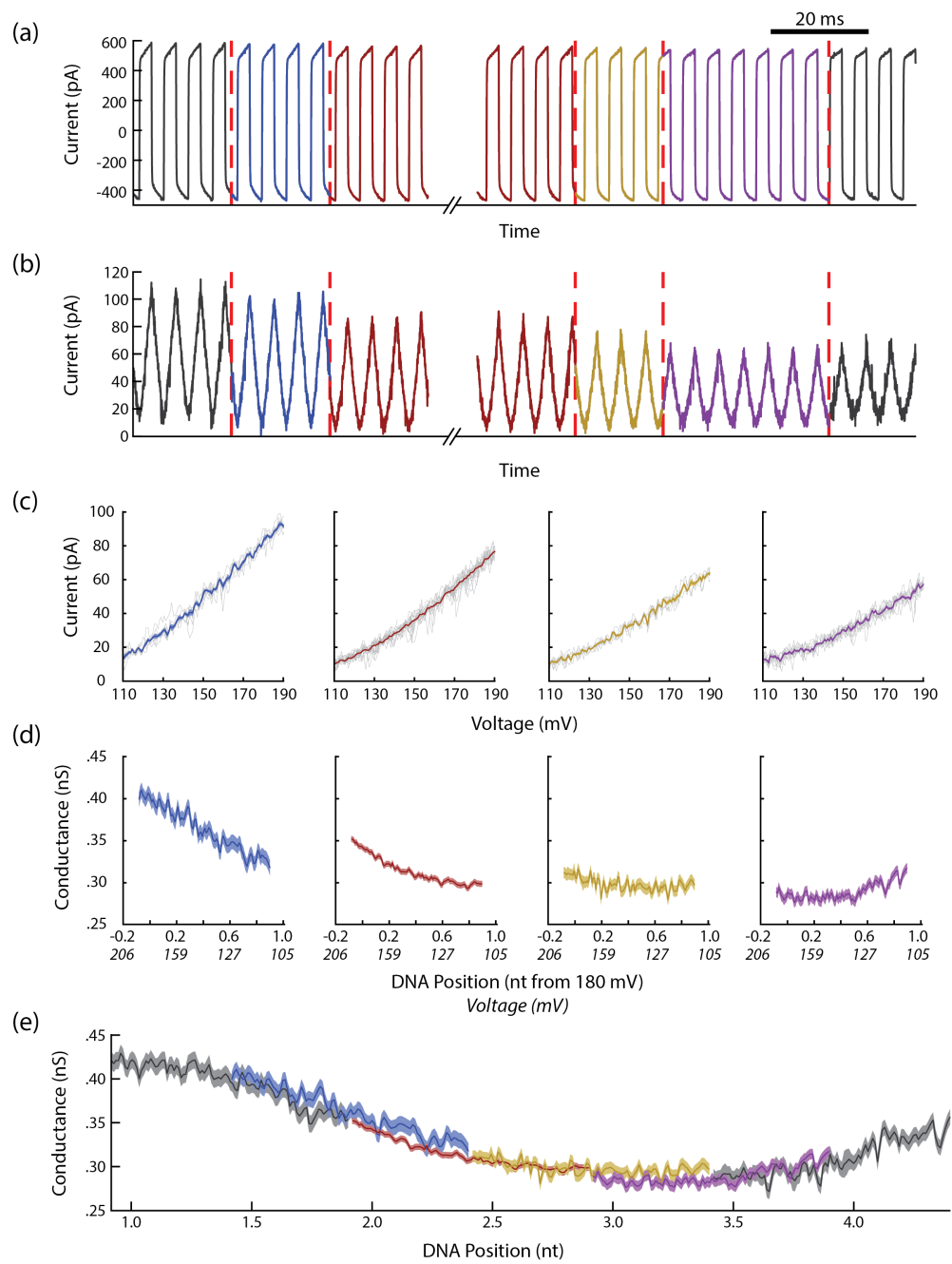


Figure 3.4: Variable-voltage data reduction. **(a)** The raw time-series data in variable-voltage sequencing exhibits large swings due to the capacitive charging and discharging of the bilayer separating the *cis* and *trans* wells. The first step of data reduction is partition the time-series data by finding enzyme steps (red dashed lines mark partitions, separate steps marked by color). **(b)** After the data is partitioned into enzyme steps, the capacitive effect is removed by capacitance compensation. **(c)** Each enzyme step is characterized by a current-vs-voltage ( $I - V$ ) curve, with the several voltage cycles within the step averaged together to give the final  $I - V$  curve. **(d)** Each step's  $I - V$  curve is transformed into conductance-vs-DNA position, with the DNA position calculated based on the DNA extension curve calculated earlier (Fig 3.4) and the conductance calculated as in appendix C.2. **(e)** The segments of the conductance profile probed at the separate enzyme steps can finally be plotted together on the same conductance-vs-position plot, revealing the measurement of the smooth conductance profile.

that the enzyme did not take a half-nucleotide step. Thus, variable-voltage signal addresses both of the major shortcomings of the constant-voltage signal (section 3.2). Individual states now have more characterizing features in addition to their mean conductance, and enzyme missteps can be directly detected based on whether or not successive states overlap with one another.

### 3.8 Error Correction with Variable-Voltage Data

We can use the additional information present in the variable-voltage sequencing signal to automatically correct the two major sequencing error modes of indistinguishable states and enzyme missteps. In Fig 3.6a and b, we see the variable-voltage states observed for two separate measurements of the same DNA sequence (lower panels). The upper panels show the extracted conductance values from a single voltage value in the variable-voltage data, representing the signal that would be available in a constant-voltage experiment.

These example events reveal the ability of the variable-voltage signal to correct error modes present in the constant-voltage signal. First off, in several cases, states that are indistinguishable in the constant-voltage data by their mean conductance alone can be easily

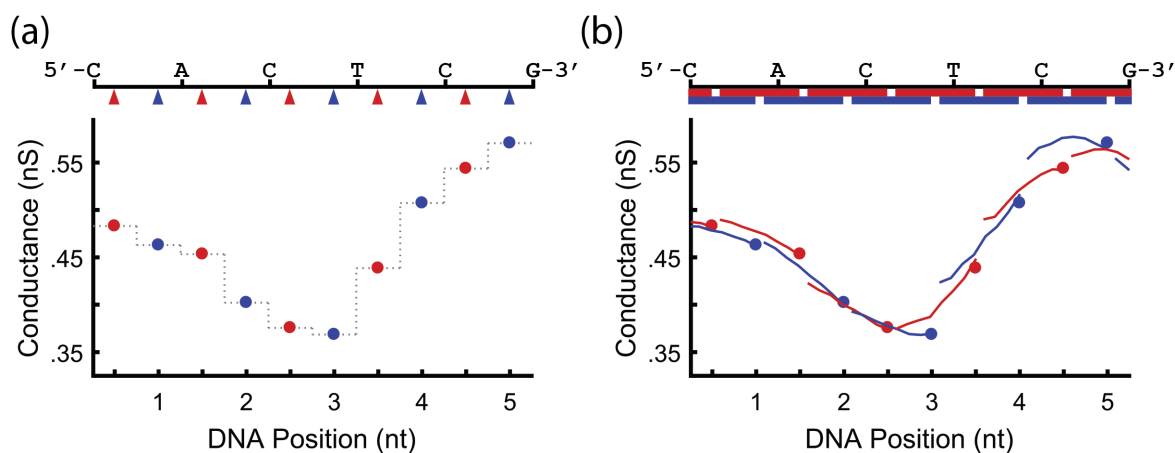


Figure 3.5: Discrete vs. continuous conductance sampling. **(a)** Constant-voltage sequencing provides a discrete sampling of the DNA's conductance as a function of position. The translocating sequence (top) is sampled at half-nucleotide intervals by the two step motion of the Hel308 helicase (odd steps in red, even steps in blue). The discrete sampling locations (triangle pointers, top) result in a disconnected set of conductance values (points, dashed line) for the sequencer to decode. **(b)** Variable-voltage sequencing provides a continuous sampling of the DNA's conductance profile. Red and blue bars (top) show the ranges along the DNA molecule probed during the voltage swing at odd (red) and even (blue) enzyme steps. Red and blue curves show the corresponding segments of the conductance profile explored at each state. Blue and red points show the information that was available in constant-voltage sequencing.

distinguished in the variable-voltage data. For example, in (a), four consecutive states all have nearly identical mean conductance values (steps 16-19, upper panel). It would be difficult to confidently assign these measurements to the correct generating DNA sequence states. Furthermore, these steps would be difficult for the change point algorithm to detect. In the variable-voltage signal below, these same four states are easily distinguishable by their overall shape. Two of the states (16, 18) are have little slope and notably positive curvature, while the other two (17, 19) have decidedly positive slopes and little curvature. We can thus determine that states 16 and 18 are two repeated measurements of the same DNA sequence state, as are states 17 and 19.

This repeated measurement is caused by an enzyme backstep, leading into the second type of error mode that can be corrected using the variable-voltage signal. As discussed in section 3.7, consecutive states separated by a single half-nucleotide step should exhibit overlapping conductance profiles. This overlap requirement allows us to identify locations where the enzyme took a non-standard step by looking for locations where consecutive segments fail to overlap <sup>7</sup>. In the previous example of states 16-19 in (a), we observe a large discontinuity between states 17 and 18, indicating that the enzyme took a non-standard step at this transition. The discontinuity information, combined with our above observation that states 16 and 18 seem to match, as do states 17 and 19, reveals that the enzyme took a backstep at this transition, moving backwards along the DNA and causing the pore to re-measure a previously observed sequence state.

We can use the overlap information throughout the read to determine what type of step the enzyme took at each transition. This step type determination is conducted automatically by custom support vector machines (appendix D.2) trained to recognize different enzyme steps based on the variable-voltage data. Automatic step identification results in the colored state labels between the upper and lower panels. Single half-nucleotide steps (green arrows),

---

<sup>7</sup>In (a) and (b), states are spaced by full nucleotide steps (rather than the half nucleotide steps taken by the Hel308 enzyme) to more clearly show the individual states, so consecutive steps do not overlap in this visualization.

skips <sup>8</sup> (blue double arrows), backsteps <sup>9</sup> (red back arrows), and holds <sup>10</sup> (gold pause symbols) can all be accurately and automatically labeled and corrected.

Using the transition labels, we can account for the enzyme step type separating each of the measured states, and reconstruct the error-free conductance profile for the target DNA strand. The corrected signal for the read in (a) is shown in (c); the corrected signal for the read in (b) is shown in (d). The corrected signals for the two reads are nearly identical, despite the qualitative dissimilarity of the two reads prior to error correction. The read-to-read consistency of the corrected variable-voltage signal indicates that this method should improve sequencing accuracy. The constant-voltage signals (top) are difficult to correct and vary substantially read-to-read, likely resulting in a different base calling result for what should be the same DNA sequence. In comparison, the two reads' similar corrected variable-voltage signals will likely both decode to the same (correct) DNA sequence.

### 3.9 Sequencing with Variable-Voltage

Now armed with a method of measuring a more information-rich nanopore signal capable of addressing the major sequencing error modes, we sought to test the performance of variable-voltage nanopore sequencing against the constant-voltage method. Full realization of variable-voltage sequencing required substantial re-engineering and addition to the constant-voltage sequencing procedure. A brief accounting of the major new or modified components required to sequence the variable-voltage data includes

1. A change point detection procedure capable of identifying transitions in the raw variable-voltage data (section 3.6, appendix B).

---

<sup>8</sup>Skips occur when a transition is longer than a half nucleotide and are typically caused by the enzyme taking one or more steps that are too fast to be detected.

<sup>9</sup>Backsteps occur when the enzyme moves backwards along the DNA and are a natural product of Hel308's kinetics (and in fact the kinetics of all the motor proteins we've studied for nanopore sequencing).

<sup>10</sup>Holds are consecutive measured states representing the same DNA sequence. They are typically caused by over-calling by the change point detection algorithm, or by aberrant transient electrical noise causing a false partitioning of a single state.

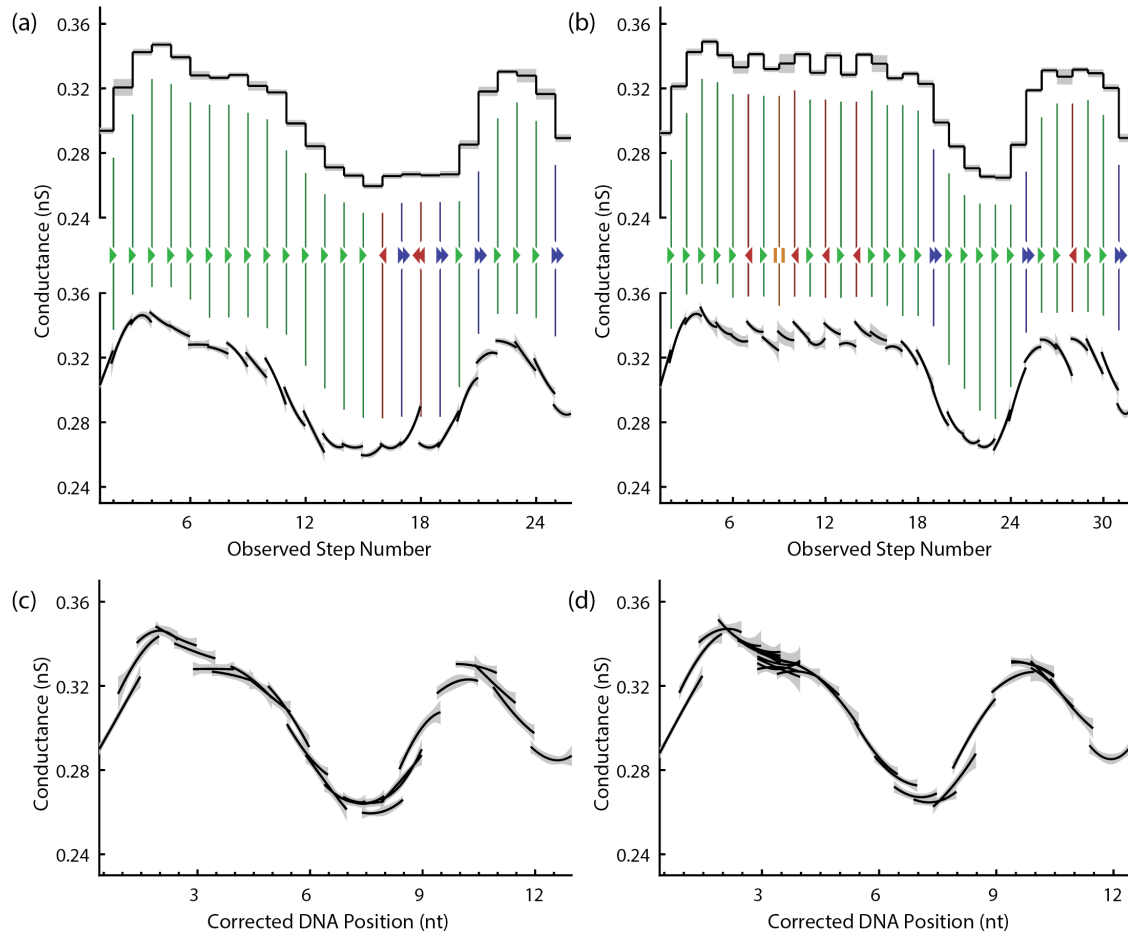


Figure 3.6: Automatic error correction. **(a)** and **(b)** compare the information available from the constant-voltage measurement of DNA-translocation events (above) against the information available from the variable-voltage measurement of the same DNA-translocation events (below). The constant-voltage data is extracted from the variable-voltage data. The type of enzyme step taken at each transition is determined automatically based on the variable-voltage data and labeled with the appropriate marker. Green arrows are steps, red arrows are backsteps, blue double arrows are skips, orange pause symbols are holds. **(c)** and **(d)** show the signals recovered from **(a)** and **(b)** respectively, after the enzyme missteps have been automatically corrected. We see that although the states in the events above look quite different, after correction the signal recovered from each of the two events is nearly identical. In reality, both events were measurements of the same DNA sequence, so this event-to-event reproducibility in the reconstructed signal indicates that the variable-voltage data should be much more robust for sequencing.

2. A capacitance compensation procedure to remove the capacitive current from the raw data (section 3.6, appendix C.1).
3. A choice of “features” to describe the conductance signal of each enzyme state (appendix C.3).
4. A method of automatically identifying and correcting enzyme missteps (appendix D.2).
5. A new signal-to-sequence k-mer model mapping between observed conductance states and the generating DNA sequence (appendix E).
6. A sequencing algorithm that harnesses all the information available in the variable-voltage signal to optimally decode the DNA sequence (appendix F).

The adapted change point detection algorithm and the capacitance compensation procedure have already been discussed briefly in section 3.6, and their full implementation is covered in depth in appendices B and C.1.

The remaining new and modified methods are discussed briefly below, with the full descriptions provided in the appendices.

### *3.9.1 Feature Extraction*

We chose to use the coefficients of the top three principal components to describe each state’s conductance curve (appendix C.3). We determined the principal components using principal component analysis of a large dataset of states’ conductance curves, and found that the first three principal components accounted for over 98% of the observed variance between states. Linearly combinations of these three principal component vectors can describe the conductance signal generated by all of the possible sequence states. This three-feature description of the conductance states provides a simple, low noise parameterization of our observed signal.



### 3.9.2 *Enzyme Misstep Correction*

We use a three stage “state filtering” process to identify and correct enzyme missteps in the variable-voltage signal (appendix D.2). Together, this filtering pipeline converts the time-ordered observed conductance states into the error-corrected, sequence-ordered conductance states that will be passed to the sequencing algorithm.

#### *Removal Filter*

The first stage is termed the “removal” filter and is responsible for removing conductance states that are not informative of the underlying DNA sequence (appendix D.2.1). A number of sources (discussed in the appendix) can produce conductance states that are recognized by the change point detection algorithm but are not actually representative of the DNA sequence in the pore. A support vector machine (SVM) has been trained on a hand-labeled dataset containing both “good” states (those informative of the DNA sequence in the pore) and “bad” states (those uninformative of the DNA sequence in the pore). The SVM is used to assign a “bad” probability to each observed state. States whose bad probability exceeds a set threshold are discarded for downstream analysis.

#### *Recombination Filter*

The second stage in the filtering pipeline, termed the “recombination” filter, looks for instances where enzyme missteps caused duplicate measurements of the DNA position (appendix D.2.2). The conductance states for these duplicate measurements are then averaged and recombined. Duplicate measurements be consecutive due to over-called transitions (holds) or be separated by several interleaving states due to enzyme backsteps. The recombination filter finds duplicate states by aligning the measured signal against itself. States will match to their duplicates nearly as well as they match to themselves. A “self-alignment” penalty biases against states simply aligning to themselves, allowing the self alignment to match up nearly identical states, identifying duplicate measurements.

The self-alignment is calculated using a Neeleman-Wunsch alignment procedure [74]. The transition penalties in the alignment are calculated on a state-by-state basis based on the overlap information in the variable-voltage signal. An ensemble of SVMs is used to assign a probability that each transition is either a single step, a backstep, a skip, or a hold. These step type probabilities are converted into appropriate alignment transition penalties, and help the self-alignment to find the correct path through the observed states.

### *Reordering Filter*

The final filtering step is the “reordering” filter which accounts for enzyme missteps not treated by the removal and recombination filters (appendix D.2.3). In some cases, enzyme missteps result in out-of-order states, without any single state being measured multiple times. One such case is that of a skip followed by a backstep, followed by another skip. This enzyme behavior would result in us measuring first state 1, then state 3 (a skip), then state 2 (a backstep), then state 4 (another skip). No state is measured more than once, but states 3 and 2 were measured in the wrong order. These compound error modes are less common than those addressed by the previous filters, but can still diminish sequencing accuracy.

In the reordering filter, we use the same set of SVMs as were used in the recombination filter to assign a probability that each transition was a step, skip, or backstep. Once we’ve determined the step type probabilities for each transition, we use a dynamic programming method to find the most likely set of transitions linking the states based on the calculated probabilities.

### *3.9.3 Signal-to-Sequence Model*

As discussed in section 2.4, to decode the DNA sequence that generated an observed signal, we need a model relative signal to sequence. Previous work on constant-voltage nanopore sequencing showed that a 4-mer model in which every possible 4 base sequence was associated with a specific signal modeled the data well [51] [58].

The existing 4-mer model mapping constant-voltage signal to DNA sequence is insufficient for variable-voltage sequencing for two reasons. First, the variable-voltage signal is fundamentally different from the constant-voltage signal—this is why variable-voltage sequencing is worth doing in the first place! Rather than associating each 4 base sequencing with a mean ionic current (or equivalently, conductance), the variable-voltage signal-to-sequence model must map each sequence to an associated conductance curve segment.

Second, the 4-mer model must be expanded due to the wider base sensitivity of the variable-voltage method. In the case of constant-voltage sequencing, a 4-mer model was chosen as the 4 bases nearest the constriction at a given enzyme step had the largest effect on the resulting signal. However, a 4-mer model is simply a specific case of the more general  $k$ -mer model, in which combinations of  $k$  bases are associated with specific signals. In general, the larger  $k$  is, the more predictive the model will be, as it will be able to account for the small but non-negligible effects of bases more distant from the constriction (appendix E.1)

<sup>11</sup>.

As the variable applied voltage shifts the DNA back and forth within each enzyme step, the bases to both the 3' and 5' of the central 4-mer have a larger effect on the observed signal than when applying a constant voltage. The contribution of these additional bases means that more than 4 bases contribute meaningfully to the variable-voltage signal, so we expanded our signal-to-sequence model to 6-mers. Additionally, Hel308 has two steps per nucleotide rather than one, so two distinct conductance states are measured for each 6-mer. All together, this means that our variable-voltage 6-mer model for Hel308-controlled DNA translocation entails 8192 ( $= 2 \times 4^6$ ) total conductance states. We measured the two conductance states of the variable-voltage signal for all possible 6-base combinations multiple times and in various sequence contexts to construct the variable-voltage 6-mer model (appendix E).

---

<sup>11</sup>The drawback of larger  $k$ -mer models is that they are more difficult to construct empirically. To construct a model with a given  $k$ , the signal generated by all  $4^k$  possible  $k$ -base combinations must be measured—ideally several times and in various surrounding sequence contexts to get a good estimate of the signal variance. Thus, the experimental task of measuring the  $k$ -mers grows exponentially with  $k$ .

### 3.9.4 Sequencing Algorithm

Given a  $k$ -mer model, section 2.5 discussed how a measured series of signal states can be decoded into the generating DNA sequencing using a hidden Markov model (HMM). As both variable-voltage and constant-voltage sequencing use a  $k$ -mer model, the same fundamental HMM-solving algorithm works in both cases, with a few modifications (appendix F). First, we adapted the sequencing algorithm to work for the half-nucleotide steps of Hel308, rather than the full nucleotide steps of  $\Phi$ 29 DNAP (appendix F.4). This modification is relevant to sequencing Hel308-controlled data in both the constant-voltage and variable-voltage cases.

Second, we use our understanding of Hel308's kinetics to improve base calling of both constant-voltage and variable-voltage data (appendix F.2). Hel308 hydrolyzes ATP to power its translocation along DNA. Kinetic analysis of the Hel308 enzyme revealed that the two steps the enzyme takes per nucleotide represent two distinct mechanical substates of the enzyme's ATP hydrolysis cycle. The durations of one of the two steps (the "dependent" step) depend upon the available concentration of ATP, while the durations of the second step (the "independent" step) do not [53]. Further analysis showed that most of the observed enzyme backsteps occur starting from the ATP-independent state. the sequencer can use this property to narrow the range of possible 6-mer model states an observed signal state can match to. Simply, if an observed state exhibits a backstep, it is far more likely to match to an ATP-independent state in the 6-mer model than to an ATP-dependent state, narrowing the match probabilities by half. This improvement is implemented for both constant-voltage and variable-voltage sequencing.

Finally, we modified the variable-voltage sequencing algorithm to make full use of the step-type identification abilities of the variable-voltage signal. The one type of enzyme misstep error mode that cannot be fully corrected using the variable-voltage scheme is the skip. As a skip causes a conductance state to be completely missed during measurement, the resulting gap cannot be filled in. However, being able to at least label the locations in the data where a skip occurred can help tremendously during sequencing. At each state-to-state

transition, the sequencer must decide the relative likelihood of different length steps in order to determine which 6-mers can be transitioned into. During constant-voltage sequencing, the various length steps (single step, skip 1, skip2, ...) are assigned constant likelihoods at every transition. Conversely, in variable-voltage sequencing, although we cannot know exactly what the skipped conductance state looked like, we can know that a skip occurred based on segment-to-segment continuity, and even infer how long the skip was. Using this information, we can assign state-by-state step size likelihoods that help the sequencer determine what length step occurred in between each pair of observed states (appendix F.3).

### 3.10 Sequencing Results

With the full variable-voltage sequencing pipeline in place, we tested the performance of variable-voltage against constant-voltage sequencing. We conducted this test by taking both constant-voltage and variable-voltage reads of the same DNA sequence. We used the pET-28a vector sequence (appendix A.5) for these experiments, as it provided a testing ground of genomic DNA that was not used in constructing the 6-mer signal-to-sequence model <sup>12</sup>.

To isolate the relative performance of the two applied voltage strategies, we held constant all other aspects of the sequencing experiment. Specifically, both experiments were conducted using Hel308 as the motor enzyme, in identical buffer conditions (appendix A.1.6, A.1.7). Additionally, base calling in both methods used a 6-mer model, with the constant-voltage 6-mer model extracted from the variable-voltage 6-mer model (appendix E.6). As both the variable-voltage and constant-voltage 6-mer models originate from the same underlying model, any errors present in one will be present in both, providing a level playing field for method-to-method comparison. Thus, any errors present in one model should be present in both, providing a level playing field for method-to-method comparison.

In total, we sequenced 73 variable-voltage reads totaling 12873 bases and 31 constant-

---

<sup>12</sup>It was important to conduct the validation experiment using a DNA sequence not used in model construction to avoid over-training. Over-training is less of an issue for pre-determined models (like the  $k$ -mer model) than for learned models (as are constructed by recurrent neural networks), but can still arise and should be avoided if possible for an accurate representation of sequencing performance.

voltage reads totaling 9496 bases. For each read, we determined the true generating sequence by aligning the called sequence to the pET-28a reference sequence <sup>13</sup>. The accuracy was calculated from the alignment as

$$accuracy = \frac{N_{match}}{N_{match} + N_{mismatch} + N_{insertion} + N_{deletion}} \quad (3.1)$$

$N_{match}$  is the number of alignment locations where the called base and true base match,  $N_{mismatch}$  is the number of locations where they don't match,  $N_{insertion}$  is the number of locations where an additional base is called relative to the true sequence (an insertion), and  $N_{deletion}$  is the number of locations where no base was called where one should have been. Overall, we obtained 62.5% sequencing accuracy over the constant-voltage reads. The variable-voltage reads yielded a 79.1% accuracy over the variable-voltage reads, representing nearly a 2-fold improvement in the error rate <sup>14</sup>. The 37.5% error rate over the constant-voltage reads broke down into a 13.3% miscall rate, 16.7% insertion rate, and 7.5% deletion rate. The variable-voltage reads' 20.9% error rate was consisted of a 7.7% miscall rate, 6.1% insertion rate, and 7.0% deletion rate. Fig 3.7 provides a full accounting of the per-base accuracies.

### 3.11 Discussion and Conclusions

#### 3.11.1 Context for Improvement

The jump in sequencing accuracy from 62.5% to 79.1% provided by the change from constant-voltage to variable-voltage is significant, even taken at face value. However, the jump takes on greater meaning when the performance of the two methods is compared against the performance of a random sequencer. Random sequencing accuracy represents the baseline performance of a hypothetical sequencing method that extracts no information about the

---

<sup>13</sup>Alignment was conducted using a local-to-global Smith-Waterman-style gapped alignment, in which we aligned the entire called sequence to the best matching section of the longer reference sequence

<sup>14</sup>Error rate = 1 - sequencing accuracy

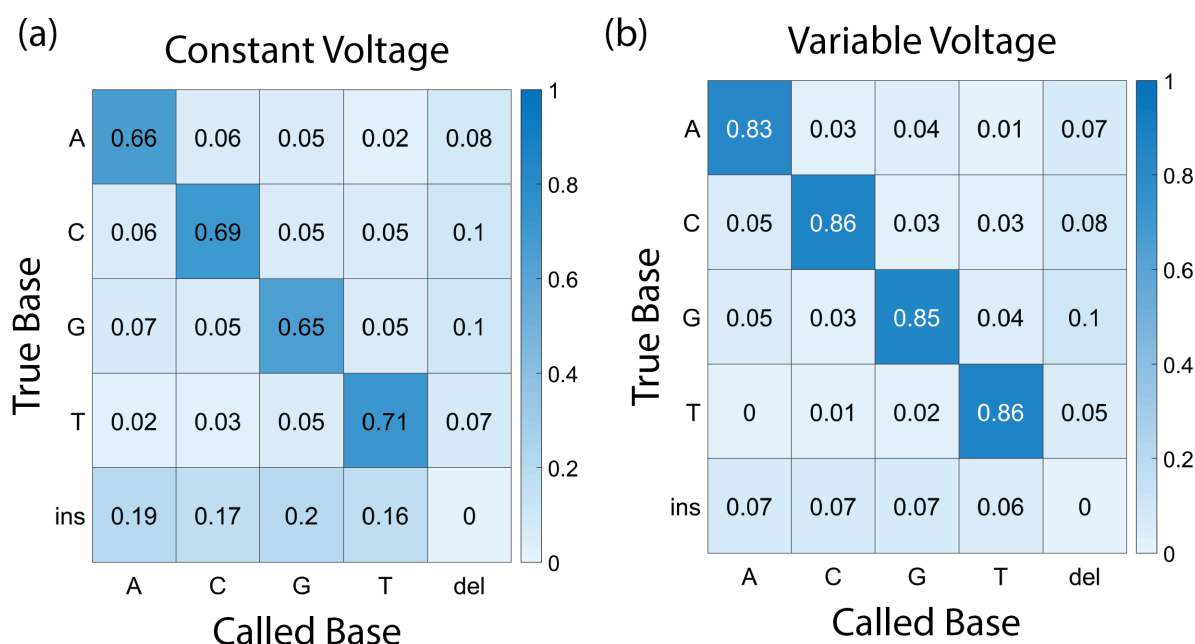


Figure 3.7: Sequencing confusion matrices. The confusion matrices for the constant-voltage **(a)** and variable-voltage **(b)** sequencing methods show the various error rates of the two methods. The rate at which a given true base (y-axis) is called as a given called base (x-axis) is shown in the corresponding cell. Diagonal entries represent correctly called bases, off-diagonal entries represent errors. The bottom row shows the base-by-base insertion rate, where an extra base has been called relative to the true sequence. The right-most column shows the base-by-base deletion rate, where too few bases have been called relative to the true sequence. Matrices are normalized to sum to 1 along the columns. The variable-voltage method exhibits across-the-board improvement in all calling rates.

DNA sequence in question. Comparing the performance of the constant-voltage and variable-voltage methods against this baseline contextualizes how far above this information-less baseline each method operates.

Counterintuitively, a randomly generated DNA sequence will not have 25% accuracy. The sequence-to-sequence alignment procedure used to determine the sequencing accuracy inserts gaps into both the called and true sequences to generate the best match between the two. As a consequence of this gapped alignment, aligning two random sequences will yield an accuracy well above 25%. In our case of local-to-global alignment of a short (called) sequence to a section of a longer (true) sequence, the random accuracy will depend on the lengths of both the called sequence and the true sequence, with shorter called sequences and longer true sequences resulting in higher random accuracy <sup>15</sup>.

We compared the distribution of full-read sequencing accuracies for constant-voltage and variable-voltage reads against the distribution of random accuracies generated by aligning random sequences the same length as the collected reads against the pET-28a reference sequence (Fig 3.8). With this context, we see that although the constant-voltage method is producing meaningful sequencing results above random, it only barely outperforms this baseline (Fig 3.8a). Comparatively, the variable-voltage reads are consistently well above the random accuracy baseline (Fig 3.8b). Against the random accuracy baseline, the variable-voltage sequencing method dramatically outperforms the constant-voltage method.

### 3.11.2 Towards Higher Accuracies

The variable-voltage method’s 79.1% accuracy is competitive with the best reported single-read *de novo* obtained using a nanopore sequencing device <sup>16</sup>. Importantly however, this

---

<sup>15</sup>Global-to-global alignment of the entirety of two random sequences will yield  $\sim 54.3\%$  identity, with local-to-global alignments generating higher identity as discussed in appendix A.6

<sup>16</sup>It is difficult to pin down definitive numbers for commercial nanopore sequencing devices, as companies rarely publish on their most recent results. A recent review of nanopore sequencing progress [65] shows a highest single-read accuracy of around  $\sim 75\%$ . More recent published results have used either “2D” reads or the newer “1D<sup>2</sup>” reads, both of which read both the sense and antisense strand of the target DNA strand to generate higher accuracy and are thus not comparable to true single-read accuracies.



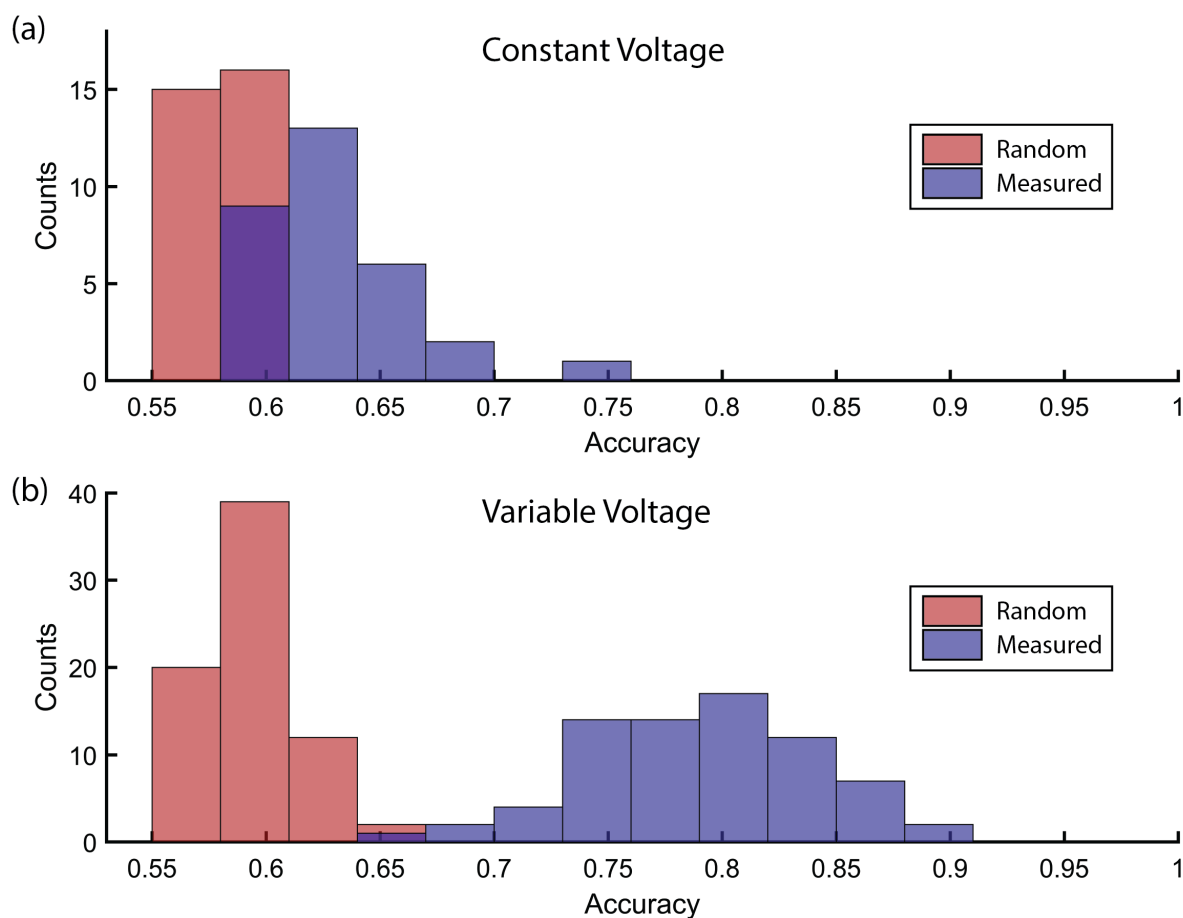


Figure 3.8: Read accuracy distributions. **(a)** The distribution of measured constant-voltage read accuracies (blue) is only marginally better than the distribution of random accuracies generated for reads of the same length (red). **(b)** The distribution of measured variable-voltage accuracies (blue) is consistently well above the distribution of random accuracies for equal length random sequences (red).

initial accuracy figure represents only the baseline starting point of the accuracies nanopore sequencing can ultimately achieve with the aid of the variable-voltage method. This method requires little re-engineering of the basic nanopore sequencing system beyond the application of a variable rather than constant voltage. Consequently, it is straightforward to combine this method with other techniques being developed to improve nanopore sequencing accuracy.

The improved techniques that have improved commercial constant-voltage sequencing accuracy over the past 5 years are easily applicable to variable-voltage sequencing as well. The constant-voltage versus variable-voltage comparison presented above tested both methods without the incorporation of any other new sequencing advances such as more processive and predictable motor enzymes, more sophisticated base calling algorithms, and reads measuring both the sense and antisense strand of the target DNA<sup>17</sup>. Just as the implementation of these several improvements have taken constant-voltage accuracy from the low 60%'s into the mid 70%'s, combining these techniques with variable-voltage sequencing should enable continued advances in sequencing accuracy well above the presented 79.1% mark. The merger of the variable-voltage method with these other advances offers a path forward toward realizing the ultimate goal of high accuracy nanopore sequencing.

---

<sup>17</sup>These sense and antisense reads are the “2D” and “1D<sup>2</sup>” reads referenced above. Both read methods are fundamentally the same. Combining the information from the two strand reads allows error correction where an error was made in only one of the two reads. In the case of completely random errors, the error rate will square (e.g. a 70% single read becomes a 91% 2D read). If errors are anti-correlated (as is the case in our sequencing, where certain  $k$ -mers are called more accurately than others), it is possible to do better than squared error rate.

## Chapter 4

# A HIGH SENSITIVITY BLAST ALGORITHM FOR NANOPORE SEQUENCING

In chapter 3, we focused on improving nanopore DNA sequencing by raising the single-read *de novo* sequencing accuracy. Here, we will shift our focus over to a parallel avenue towards improving the overall efficacy of nanopore sequencing technology. Rather than focusing on raising the accuracy of the next generation of nanopore sequencing devices, we will discuss how to make better use of the data produced by the devices that are already available. Specifically, I will present a new method of sequence alignment capable of generating strong and fast alignment results even for low accuracy nanopore reads.

## 4.1 BLAST

Many important clinical and in-field DNA sequencing questions can be answered without conducting whole genome sequencing. Applications including pathogen detection (section 1.4.4), outbreak tracking (section 1.4.2), and metagenomic studies (section 1.4.3) are more interested in coarse-grained information about the representation of different species or mutants in a sample, rather than in sequencing entire genomes. The common question asked in these studies is not “what is the DNA sequence of this organism (or organisms)?” but rather “what organism (or organisms) are represented by these measured DNA sequences?”.

These sorts of “what’s in my pot?” experiments rely on quickly searching for matches of the sequencing reads against a large database of previously-sequenced reference genomes. This type of large-scale sequence-to-sequence comparison problem is computationally difficult. The reference databases within which we want to look for matches can be dauntingly

large. The entire database of all known genomes <sup>1</sup> comprises over 2.9 Tb (nearly 3 trillion bases) <sup>2</sup>. Searches usually only use a subset of this vast database, but can still commonly run to several gigabases. <sup>3</sup>.

The computational issues posed by the large reference database sizes are compounded by the typical requirement that a search return inexact matches. Searching for exact sequence-to-sequence matches is computationally simple, even for long sequences. This sort of simple “word search” can run in  $\mathcal{O}(N)$  time, where  $N$  is the length of the sequence within which we are searching for matches (i.e. the reference database). However, given the possibility of mutations relative to the reference genome, and of errors in either the reference or the read, we are often interested in inexact matches. The optimal inexact match (“alignment”) of one smaller sequence to some subset of a larger sequence is efficiently found by the Smith-Waterman alignment algorithm <sup>4</sup> [75, 74, 76]. A full sequence-to-sequence Smith-Waterman alignment requires  $\mathcal{O}(N_1 * N_2)$  time, where  $N_1$  is the length of sequence 1 and  $N_2$  is the length of sequence 2. As the database of reference genomes becomes large, conducting a Smith-Waterman alignment of the sequencing reads against the entire reference becomes impractical.

The Basic Local Alignment Search Tool (“BLAST”) is a fast alignment algorithm commonly used to solve these computational difficulties [77]. Unlike the Smith-Waterman algorithm, BLAST is a heuristic algorithm and does not guarantee that it will find the optimal local alignment. However, it is able to reliably achieve near-optimal results while vastly reducing computation time relative to Smith-Waterman, making it a powerful tool for searching large sequence databases. BLAST achieves its speed advantage through a

---

<sup>1</sup>This database is available from the NCBI (National Center for Biotechnology Information).

<sup>2</sup>Statistics available at <https://www.ncbi.nlm.nih.gov/genbank/statistics/>

<sup>3</sup>The standard SI prefixes are typically used to denote genome size: a thousand bases is 1 kb, a million bases is 1Mb, etc.

<sup>4</sup>The Smith-Waterman algorithm is a special case of the more commonly known Needleman-Wunsch algorithm. Needleman-Wunsch finds the optimal global alignment between two sequences, aligning all of sequence 1 against all of sequence 2. Smith-Waterman instead finds the optimal local alignment between subsets of the two sequences.

seed-extend approach (algorithm 2). Given a “query” sequence (i.e. the sequencing read) and a reference database to search, BLAST first looks for seeds: short, exact matches of segments of the query within the database. Seed finding is a fast  $\mathcal{O}(N)$  word search problem, as described above. The found seeds are then extended using a Smith-Waterman-type local alignment algorithm, terminating when the quality of the nascent alignment starts to deteriorate <sup>5</sup>.

---

**Algorithm 2** Sequence-to-sequence BLAST

---

- 1: Start with a query sequence  $\mathcal{Q}$  and a reference database of known sequences  $\mathcal{R}$
  - 2: **List**: generate a list  $\mathcal{L}$  of all words of length  $k$  (k-mers) that exist within  $\mathcal{Q}$
  - 3: **Scan**: scan through  $\mathcal{R}$  for seeds: exact matches to k-mers in  $\mathcal{L}$
  - 4: **Extend**: extend seeds into candidate alignments using Smith-Waterman-type alignment algorithm
  - 5: **Evaluate**: evaluate the extended candidate alignments by their alignment scores to determine the confidence that each candidate represents a meaningful (non-random) alignment
- 

The seed-extend strategy saves time over full alignment by avoiding filling in the alignment matrix at locations unlikely to generate good matches. A Smith-Waterman alignment calculates a  $\text{length}(\text{query}) \times \text{length}(\text{reference})$  alignment matrix, representing all possible alignments of the two sequences (Fig 4.1, all boxes). BLAST instead only fills in the alignment matrix around the exactly-matching seeds (Fig 4.1, colored boxes). This selective calculation introduces some risk that the overall best alignment is missed, but this risk is low assuming the seeds are of a reasonable size. In return for this risk, we achieve a dramatic improvement in run time.

## 4.2 Nanopores and BLAST

Several of the advantages of nanopore sequencing make it ideally suited for BLAST-based sequencing applications. Already, nanopore sequencing devices have demonstrated fast

---

<sup>5</sup>Appendix G.3 contains a more detailed discussion of how the BLAST extension algorithm works.

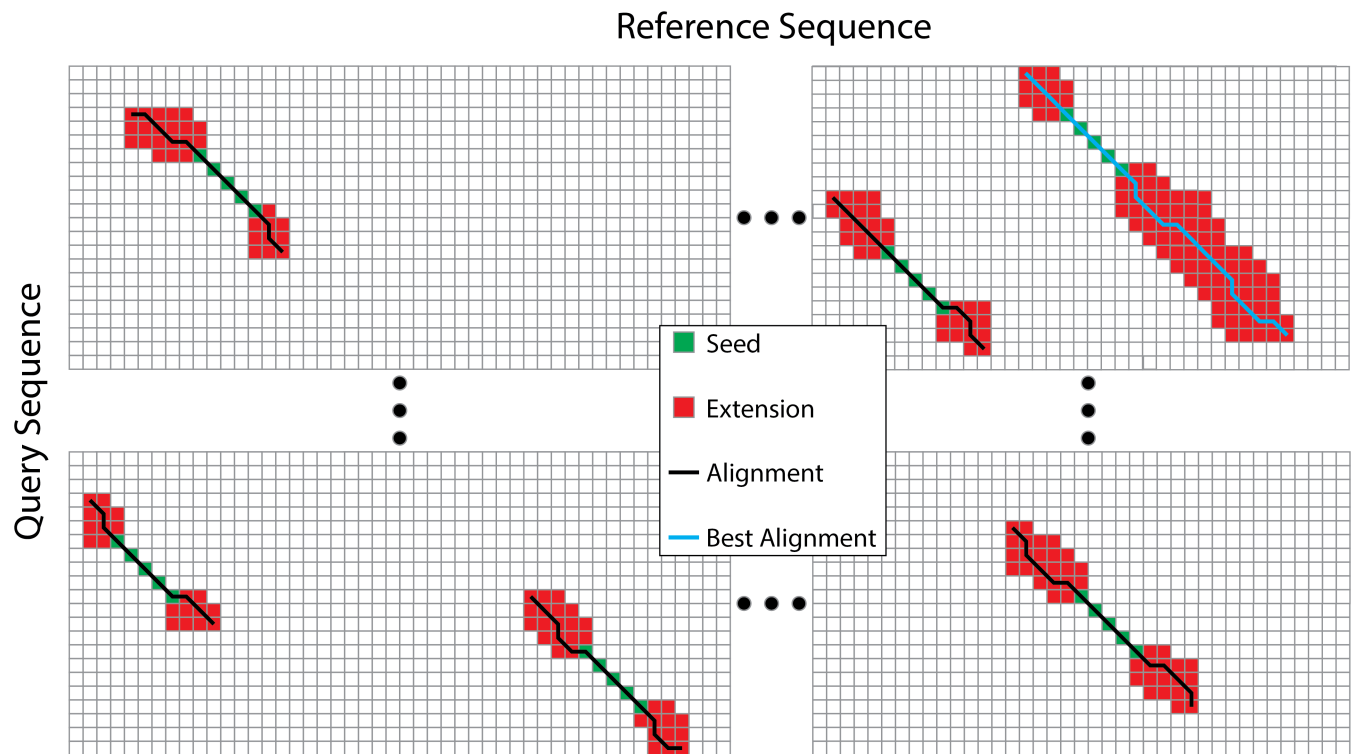


Figure 4.1: Calculation efficiency of BLAST vs. Smith-Waterman. In a full Smith-Waterman alignment, we must calculate the entire  $\text{length}(\text{query}) \times \text{length}(\text{reference})$  alignment matrix, shown here schematically as a grid of boxes. BLAST saves time by selectively only filling in regions of the alignment matrix likely to yield good results. It first finds exactly-matching seeds (green boxes), then extends the seeds (red boxes) into candidate alignments (black lines). The best local alignment is the best scoring of these candidates (blue line). Locations in the alignment matrix where no seeds are present are not filled in (blank boxes), saving on computational overhead.

sample-to-answer timelines, low input and sample preparation requirements, and portability [78, 79, 80]. These advantages indicate that nanopore sequencing technologies could be uniquely well-suited to these sorts of BLAST-based clinical and in-field sequencing applications where speed, ease-of-use, and portability are often of utmost importance. Indeed, researchers and clinicians have already used nanopore sequencing effectively in pathogen detection [81, 82, 83], outbreak tracking [80, 84, 85], species identification [86], and metagenomic sequencing [87, 88].

However, the low single-read sequencing accuracies typical of existing nanopore sequencing devices [68, 89, 90, 65] can hinder BLAST performance. Lower read accuracy means that a read will match less strongly to the correct on-target location in the reference database. Weaker on-target alignments mean that the correct alignment is less distinguishable from the crowd of off-target alignments. Thus, low accuracy reads can increase the rate of both false positives (alignments to incorrect reference locations being called as matches) and false negatives (alignments to the correct reference location not being called as matches). Widespread interest in BLAST-based nanopore sequencing applications has driven development of new algorithms designed to better handle the long, low-accuracy reads generated by nanopores<sup>6</sup> [91, 92, 93, 94]. These new algorithms help to mitigate the issues caused by low-accuracy reads, but analyses would be better and easier given stronger identity between the reads and the reference database.

### 4.3 *Beyond Sequence Alignment*

Previous work has shown that aligning the measured ionic current signal from the nanopore against the predicted signal for the reference sequence using a Smith-Waterman-style alignment can give strong alignments even for low accuracy reads [58]. Laszlo *et al.* showed that using the ionic current-to-sequence model (section 2.4) to predict the signal that would be observed for the reference sequence, then aligning the measured signal to that generated

---

<sup>6</sup>Long, low-accuracy reads are characteristic not only of nanopores, but of other single-molecule sequencing technologies as well. These new BLAST algorithms work generally for all sequencing data of this character.

strong read-to-reference alignment even for reads with low sequencing accuracy. Aligning current-to-current instead of sequence-to-sequence is effective due to the origin of many nanopore sequencing errors. As discussed in section 2.7.1, many errors are caused by incorrect decoding of ambiguous ionic current signals, rather than by fundamental errors in the signal itself. These base calling errors arise as many different DNA sequences can generate only slightly different electronic signals (Fig 4.2a, b). In the case of an ambiguous signal that could have been generated by multiple different underlying sequences, the base caller is forced to make a deterministic choice amongst the different sequence candidates. These deterministic choices in response to ambiguous information can freeze in errors and destroy information that was present in the original signal (Fig 4.2c). By aligning using the ionic current signal in lieu of the called bases, we preserve all the information available and avoid unnecessarily introducing errors into the reads (Fig 4.2d).

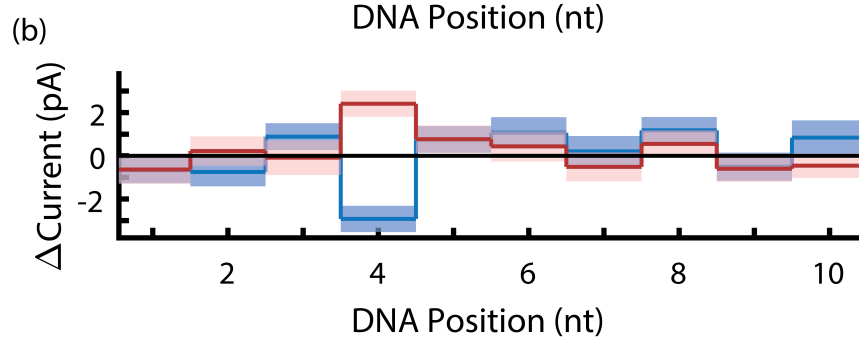
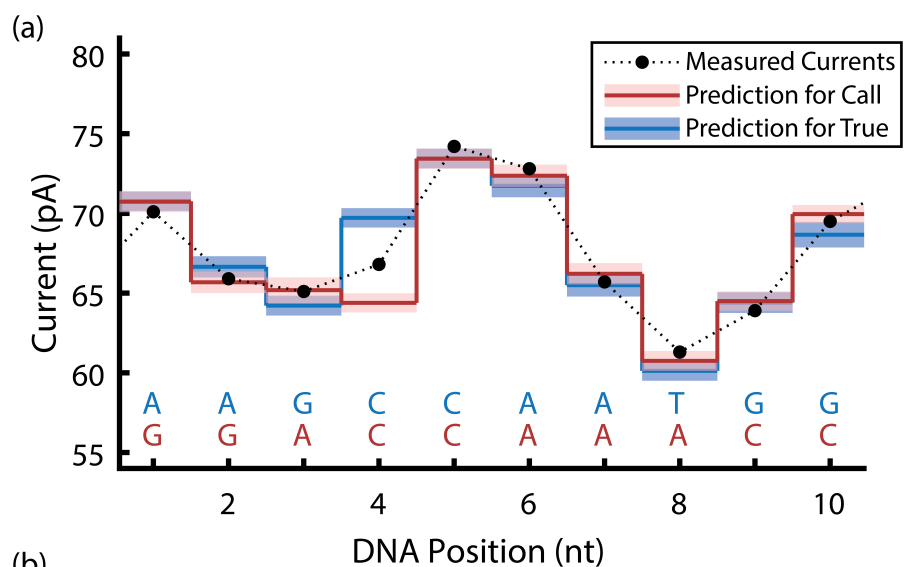
#### **4.4 *Current-to-Current BLAST***

In this work, we extend the principle of current-to-current comparison by adapting the BLAST algorithm to use the measured nanopore ionic currents instead of the called bases. The goal is to produce a fast heuristic sequence alignment method that optimally uses the data present in the nanopore signal. We will first implement an ionic current-based BLAST algorithm, then compare its performance against the standard sequence-based BLAST algorithm. We will test the two methods on data produced by Oxford Nanopore Technologies’s MinION device—that is, on data produced using an existing commercial nanopore sequencing device.

##### *4.4.1 BLAST Implementation*

To evaluate the relative performance of a heuristic alignment using ionic current in place of sequence, we implemented two versions of the BLAST algorithm in MATLAB—one (ssBLAST) comparing sequence-to-sequence, the other (iiBLAST) comparing current-to-current (Fig 4.3).





(c)

True Sequence:	A	A	G	C	C	A	A	T	G	G	
Call Sequence:	G	G	A	C	C	A	A	A	C	C	
Score:	-1	-1	-1	+1	+1	+1	+1	-1	-1	-1	-2

(d)

True Current:	71	67	64	70	73	72	66	60	64	69	
Call Current:	71	66	65	64	73	72	66	61	65	70	
Score:	+2.0	+1.5	+1.5	-1.1	+2.0	+1.7	+1.6	+1.6	+1.9	+1.3	+14.0

Figure 4.2: Principle of current-to-current comparison. **(a)** A hypothetical series of ionic current measurements (black) matches well to both the predicted ionic currents for the true DNA sequence (blue) and to the predicted ionic currents for an incorrect DNA sequence (red). Shaded regions above and below the prediction values show the prediction model's standard deviation for these states. The black dotted line linking the measured currents is to guide the eye. The true and called sequences (blue and red) are aligned below. **(b)** The residuals (measured ionic current minus predicted ionic current) of the data against the true sequence prediction (blue) and the incorrect sequence prediction (red) are both small. In this case, the data match slightly better to the incorrect sequence than to the true sequence, causing the base caller to call the incorrect sequence. The measured data diverges from the true prediction primarily due to a somewhat low measurement at DNA position 4. This single low measurement in the raw data causes 6 incorrect base calls. **(c)** If the base-called sequence for the example read were put into a BLAST search, it would yield a poor match to the true sequence, with only 4 out of 10 bases matching. Scores are assigned as +1 for match, -1 for mismatch. **(d)** The predicted ionic currents of the called (incorrect) sequence preserve the ambiguity of the measured data. Comparing the currents of the called sequence to those of the true sequence, we see a stronger match. Scores are assigned as 2 minus the z-score between the currents.

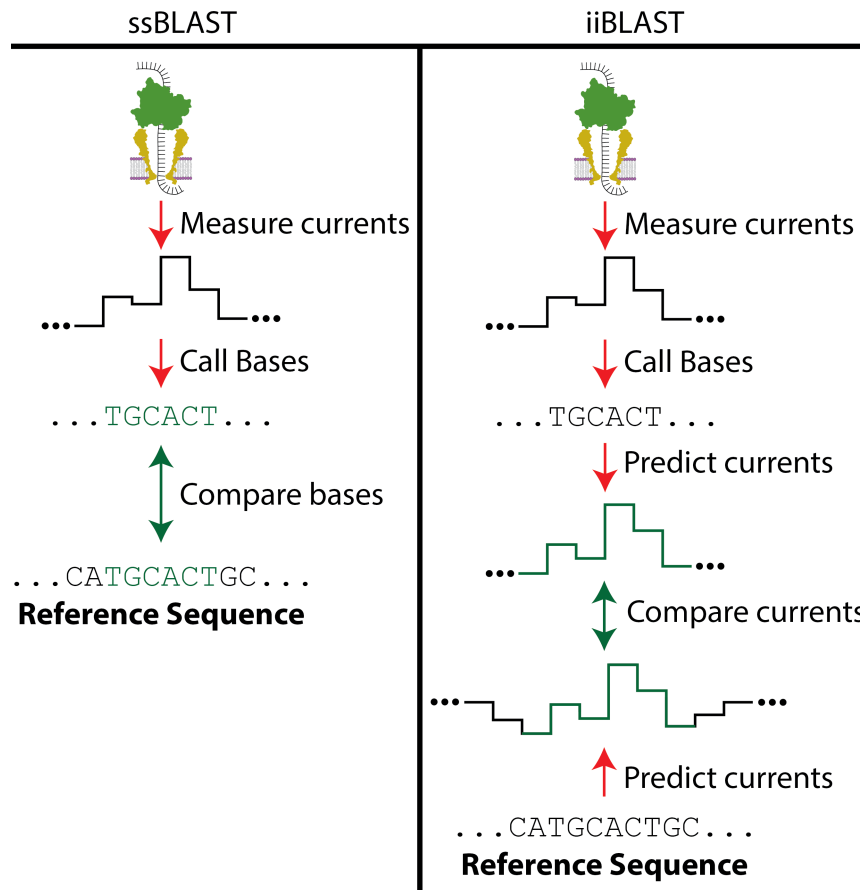


Figure 4.3: Comparisons made by ssBLAST and iiBLAST. ssBLAST uses the base calls based on the nanopore ionic current signal, then compares the bases against the reference sequence. iiBLAST takes the same base calls, then uses them to reconstruct the ionic current signal based on the ionic current-to-sequence model. The ionic currents for the reference sequence are reconstructed in the same way. Finally, the reconstructed ionic currents of the measurement and reference are compared.

The ssBLAST algorithm works as described by algorithm 2. The seed-finding (“scan”) phase is conducted using a Mealy-type finite state machine (appendix G.1) [95]. The seed-extending (“extension”) phase uses a windowed Smith-Waterman-style alignment algorithm allowing gaps (appendix G.3). Gapped alignment is critical to get strong alignments given the high frequency of insertions and deletions (indels) in typical nanopore reads. Ungapped alignments of nanopore data quickly encounter an indel and the query and reference get de-synchronized, leading to a bad alignment. The iiBLAST implementation (algorithm 3) uses the same scanning and extending machinery as ssBLAST.

---

**Algorithm 3** Current-to-current BLAST

---

- 1: Start with a query sequence  $\mathcal{Q}$  and a reference database of known sequences  $\mathcal{R}$
  - 2: **Convert**: reconstruct the ionic currents  $\mathcal{Q}_I$  and  $\mathcal{R}_I$  for the sequences  $\mathcal{Q}$  and  $\mathcal{R}$  using the ionic current-to-sequence model used in base calling
  - 3: **Bin**: place the continuously-valued ionic currents in  $\mathcal{Q}_I$  and  $\mathcal{R}_I$  into discrete bins
  - 4: **List**: generate a list  $\mathcal{L}$  of all k-mers in  $\mathcal{Q}_I$  ▷ Here, the “letters” composing the k-mers are the bin numbers of each ionic current measurement
  - 5: **Scan**: scan through  $\mathcal{R}_I$  for seeds—exact bin-to-bin matches to k-mers in  $\mathcal{L}$
  - 6: **Extend**: Extend seeds into candidate alignments ▷ Extension is conducted using the un-binned ionic current values of both  $\mathcal{Q}_I$  and  $\mathcal{R}_I$
  - 7: **Evaluate**: evaluate the extended candidate alignments by their alignment scores to determine the confidence that each candidate represents a meaningful alignment
- 

#### 4.4.2 Adapting the Algorithm

In order to adapt BLAST to current-to-current comparison, we had to work around some of the basic assumptions of the algorithm.

The standard BLAST algorithm (algorithm 2) is designed to handle discrete inputs: each entry in a DNA sequence will have exactly one of the four possible values  $\{A, C, G, T\}$ . Consequently, what is meant by an “exact match” during the seed phase is clear: all letters in the query word must be identical to those at the reference location for a seed to be reported. The discrete nature of the DNA sequence also makes the scoring of candidate matches during the extension phase straightforward. In the simplest case, a positive score

		Reference Base			
		A	C	G	T
Query Base	A	+1	-1	-1	-1
	C	-1	+1	-1	-1
	G	-1	-1	+1	-1
	T	-1	-1	-1	+1

Figure 4.4: Example sequence-to-sequence BLAST scoring matrix. In sequence-to-sequence comparison, the penalties (negative, red) or bonuses (positive, blue) for aligning any base to another can be compactly represented in a scoring matrix as shown here. This simple scoring matrix give +1 to matches and  $-1$  to mismatches, but in principle each individual entry could take any desired value.

(i.e. +1) can reward each pair of matched bases in the alignment and a negative score (i.e.  $-1$ ) can penalize each pair of mismatched bases. More generally, the match score between any pair of bases can be expressed concisely as a 4x4 scoring matrix (Fig 4.4).

To extend BLAST to handle current-to-current alignment we must adapt the algorithm for continuously-valued inputs. We adapted the seed-finding phase by discretizing the input ionic currents. The query signal is binned into a finite number of discrete values, as is the predicted signal of the reference database <sup>7</sup>. Exact bin-to-bin matches are required for a seed to be reported <sup>8</sup>. During seed extension, we revert to the continuous (not binned) ionic currents in both the query and the reference. Match scores are computed as a rescaling of the z-score between the query and reference currents, with smaller z-scores yielding more favorable alignment scores ( $z = \frac{i_Q - i_R}{\sqrt{\sigma_Q^2 + \sigma_R^2}}$ ,  $i_{Q,R}$  is the ionic current of the query or reference,

<sup>7</sup>Given our goal of not destroying information in the signal, binning may seem an unwise choice. We tested an alternative algorithm that avoided binning during seed-finding, but found that performance was both more robust and far faster using the binning method. For more discussion, see appendix G.2.

<sup>8</sup>Clearly, the number of bins used will be an important parameter. For a detailed discussion of algorithm parameters, see appendix G.4. For further depth on the choice of the number of bins, see appendix G.5

and  $\sigma_{Q,R}$  is the uncertainty for the query or reference ionic current).

#### 4.4.3 *Choice of Ionic Current*

There are two possible choices of which ionic currents to use in the current-to-current comparison. The first option is to use the raw ionic current states measured by the nanopore device. The second option is to take the base-called sequence and reconstruct it back into ionic currents using the ionic current-to-sequence model used in base calling. While the first choice (measured ionic currents) seems the more natural and obvious choice, we found the best results came using the second choice (reconstructed ionic currents). The reasons for this are two-fold.

The first and primary reason is that of calibration. The ionic currents observed during various nanopore reads of the same DNA sequence can vary in their overall magnitude (offset from zero) and the relative magnitudes of the different ionic current states (scale). Variation can occur day-to-day, experiment-to-experiment, and even pore-to-pore due to variation in electrode offsets. Prior to base calling, each individual read must be calibrated to the ionic current-to-sequence model used by the base caller. Further complicating matters, the correct calibration for a given read is not necessarily constant over the duration of the read. Particularly for long reads, the calibration required to match the measured ionic currents to the model can drift as the electrode offsets change. ONT’s sophisticated base calling software can account for the confounding effects of calibration, and the base-called results of each read represent the optimal decoding of the calibrated ionic current measurements. Thus, by reconstructing the base calls to ionic currents, rather than using the raw measured states, we circumvent the difficulties of calibration and are guaranteed a well-calibrated ionic current signal.

The second, less significant advantage to using the reconstructed ionic currents is also predicated on corrections made by ONT’s base caller. The measured nanopore ionic current signal can be marred by complex error modes. As discussed in section 2.7.2, the motor protein used to control DNA progression through the nanopore [51, 52] can take random, non-

single-nucleotide steps. Additionally, the change point detection algorithm used to partition the time-series ionic current data into discrete states can make errors, introducing erroneous partitions (extra states) or failing to call enough partitions (missed states). The combination of enzyme missteps and partitioning errors means that the reported ionic current states do not always faithfully represent the true sequence of the DNA strand being sequenced. The ONT base caller has some ability to handle and ameliorate these error modes. By using the ONT base calls and reconstructing the ionic current, we avoid the difficulties presented by these complex error modes while still harnessing the power of current-to-current alignment.

## 4.5 *Performance Evaluation*

With the two versions of BLAST implemented, we evaluated their relative performance by using both ssBLAST and iiBLAST to align the same set of nanopore reads against a database of reference genomes. For this study, we used nanopore reads of the M13mp18 bacteriophage genome from a previously published study using ONT’s MinION device<sup>9</sup>. The MinION reads were aligned using both ssBLAST and iiBLAST against a 30.2 Mb subset of the NCBI viral genome database containing the M13mp18 genome along with the sense and antisense sequences of 565 other viral genomes. To standardize run time, all alignments were run on a desktop computer with a 12 core Intel® Core™ i7-5820K CPU @ 3.30 GHz with 32 GB RAM. We ran a total of 1977 reads against the reference database using both algorithms, each with fixed run parameters (appendix G.4). Reads were truncated to a constant 100 nt length to give a wider variety of read accuracies (section 4.6) and to ensure parameter stability (appendix G.5).

## 4.6 *Performance as a Function of Read Accuracy*

We expect that the ssBLAST and iiBLAST methods will perform differently for reads with different accuracies. In the limiting case where read accuracy tends to 100%, the distinction

---

<sup>9</sup>Data were downloaded from the European Nucleotide Archive, accession number ERR739515.

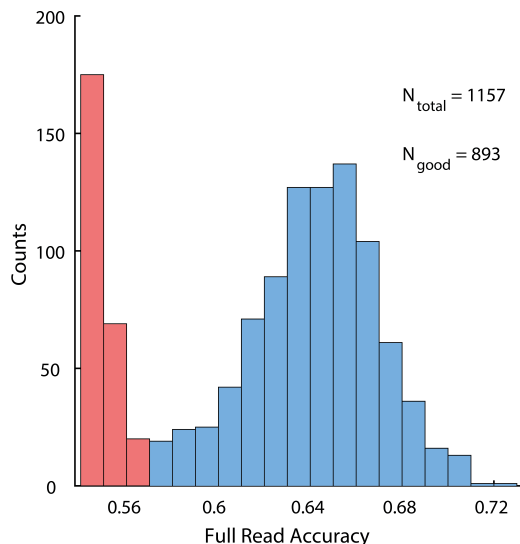


Figure 4.5: Distribution of accuracies for full MinION reads of M13mp18 in test dataset. A distinct population of reads ( $N = 264$ ) had poor accuracies (red) clearly apart from the typical distribution of accuracies (blue). Following the original authors of the dataset [96], we attribute these reads to off-target DNA, possibly from the *Escherichia coli* in which the M13mp18 was grown up. Of the remaining on-target reads ( $N = 893$ ), 90% fall between 59.1% and 68.5% accuracy, with a median accuracy of 64.5%.

between operating using bases or ionic currents becomes immaterial. The interesting case is for error-prone reads with less than 100% sequencing accuracy; here we expect iiBLAST to outperform ssBLAST as it avoids the artificial introduction of errors during base calling.

The full reads in the test dataset fell within a narrow range of accuracies (90% of reads were between 59.1% and 68.5% accurate, Fig 4.5). To explore a wider range of accuracies, we partitioned the longer reads into 100 base subsets which were then aligned against the reference genome database. These short subset reads allowed us to test the two BLAST methods for read accuracies ranging from 50% to 90%. In total, we ran  $N = 1977$  short read subsets against the reference genome database.

We binned the results of these alignments by read accuracy and evaluated the BLAST performance within each bin. To quantify performance, we used the maximum true positive



rate (TPR) provided zero false positive rate (FPR)<sup>10</sup>. This metric<sup>11</sup> tells us, given reads with a certain accuracy, what fraction of the reads we can expect to align unambiguously to the correct reference genome. The requirement that the FPR is identically zero is crucial to ensure that the BLAST results are useful. The number of off-target genomes ( $N_{off}$ ) can be large, so even a small FPR can result in the total number of false positives ( $= FPR * N_{off}$ ) vastly surpassing the number of true positives ( $= 1$ ).

As expected, both ssBLAST and iiBLAST perform similarly on high accuracy reads (unambiguously identifying all reads with accuracies 88% and above) and low accuracy reads (unambiguously identifying none of the reads with accuracies 56% and below, Fig 4.6). However, for intermediate accuracy reads, iiBLAST outperforms ssBLAST, achieving significantly higher true positive rates for reads between 58% and 86% accuracy.

#### 4.7 Alignment Significance

In addition to measuring the rate of on-target read alignment, we also evaluated the significance of the alignments produced by both ssBLAST and iiBLAST. We quantified alignment significance by calculating a p-value for each on-target alignment based on the alignment's score against the distribution of scores for all off-target alignments within the database, as described in appendix G.7.

The alignment p-value represents the probability that a BLAST alignment of the read in question to a random M13mp18-sized (7249 bp) genome of bases (in the case of ssBLAST) or ionic currents (for iiBLAST) would yield an alignment with a score as good or better than that of the alignment in question. The p-value can be used to estimate how large a database would be required before one would expect a random alignment to yield a BLAST alignment score as good as the given alignment. For example, for an on-target alignment with a p-value of  $10^{-4}$ , we expect an alignment to a random genome will match or exceed its score when

---

<sup>10</sup> $TPR = \frac{TP}{TP+FN}$ ,  $FPR = \frac{FP}{FP+TN}$ , where  $TP$ ,  $FP$ ,  $TN$ , and  $FN$  are the total numbers of true positives, false positives, true negatives, and false negatives, respectively.

<sup>11</sup>The calculation of this metric is described in more detail in appendix G.6.

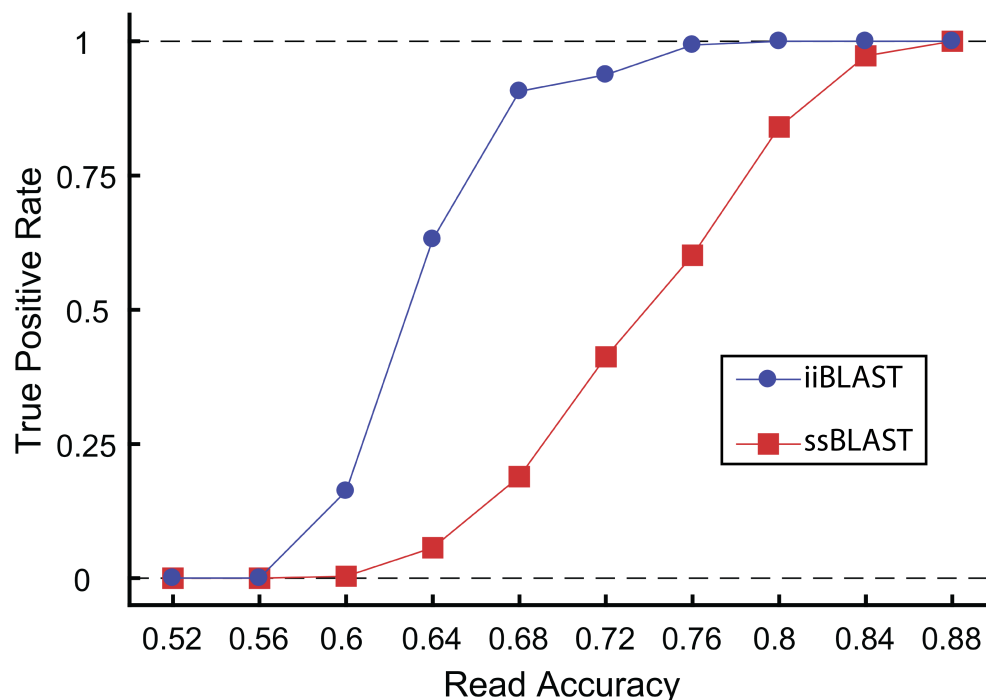


Figure 4.6: Comparison of true positive rate as a function of read accuracy for ssBLAST and iiBLAST algorithms. The maximum true positive rate provided a zero false positive rate is plotted as function of read accuracy for iiBLAST (blue, circles) and ssBLAST (red, squares). Black dashed lines show  $TPR = 0$  and  $TPR = 1$ , corresponding to no useful performance and perfect performance, respectively. Reads are binned by their accuracy into bins of width 0.04 (i.e. 62-66%). The two algorithms' performance converges at low and high read accuracies (56% bin, 88% bin), but iiBLAST performs better for intermediate read accuracies (60% bin to 84% bin), unambiguously aligning a greater portion of the available reads to the correct reference genome.

running against a reference database of 72.5 Mb (the size of the M13mp18 genome divided by our p-value, appendix G.7). If this read were aligned against references databases larger than 72.5 Mb, the on-target alignment would likely not be the highest scoring match found; an off-target alignment would likely exceed it. Alignments with p-values below  $10^{-10}$  are good enough that they should align unambiguously against the 2.9 Tb database of all known genomes in the NCBI database <sup>12</sup>.

Comparing the p-values of on-target alignments generated by ssBLAST and iiBLAST, we find that the alignment p-values are consistently several orders of magnitude smaller (better) using iiBLAST than ssBLAST (Fig 4.7). Furthermore, the iiBLAST method generates stronger alignments than ssBLAST across all read accuracies tested. Consequently, although both methods had a 100% true positive rate for the high accuracy reads (86-90%) in our test set against our trial database, iiBLAST would continue to outperform ssBLAST if the search were expanded to include a larger reference database.

## 4.8 Discussion and Conclusions

We've shown that by implementing a version of the BLAST algorithm that makes current-to-current, rather than sequence-to-sequence, comparisons between nanopore reads and reference genomes, we are able to dramatically improve the performance of the BLAST search for error-prone nanopore sequencing reads. The the next steps of scaling and integrating this technique bear discussion, as do the potential implications of this new method on the future of nanopore sequencing.

### 4.8.1 Scaling to Larger Reference Databases

We conducted our validation experiment using a relatively small reference database. Using a small reference allowed us to test many reads using both ssBLAST and iiBLAST using

---

<sup>12</sup>This is presently true, but the goalpost of maximum reference database size is always moving. DNA sequencing is growing rapidly, and with it is the database of known genomes. Statistics are available at <https://www.ncbi.nlm.nih.gov/genbank/statistics/>

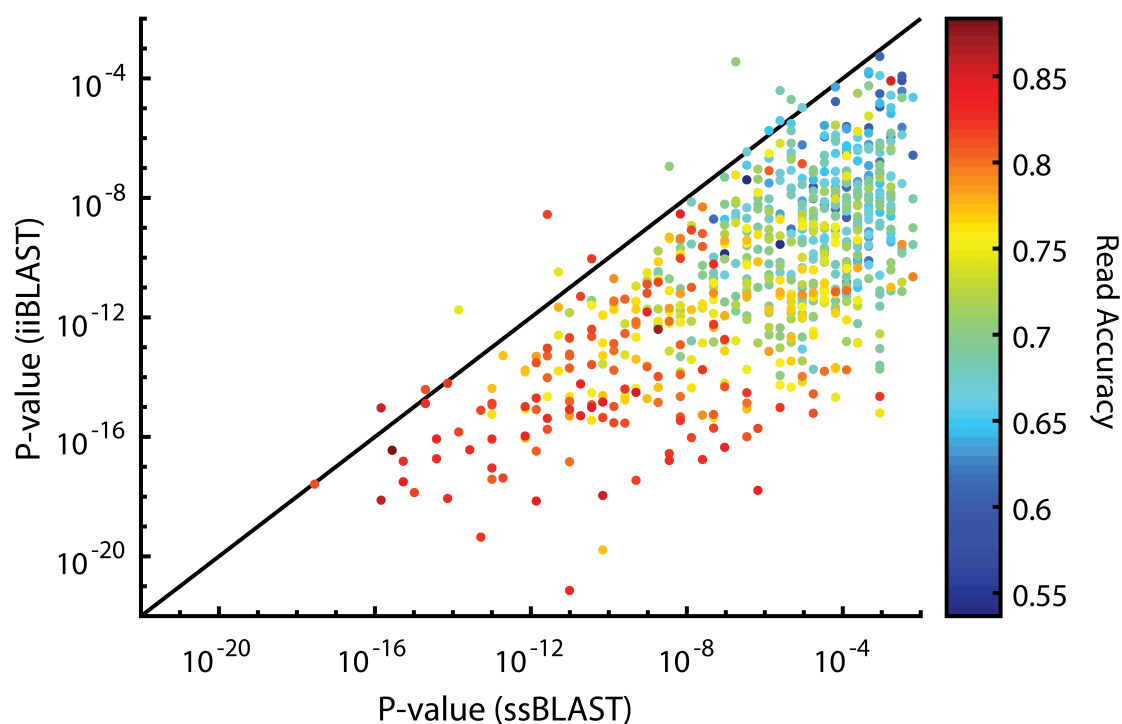


Figure 4.7: ssBLAST vs iiBLAST p-value comparison. Each on-target read in the validation experiment is plotted as a single point at the p-value of the read's alignment using ssBLAST ( $x$ -axis) and its p-value using iiBLAST ( $y$ -axis), color coded by the read's sequencing accuracy (color bar). The black line shows  $x = y$ , representing where the two algorithms yield equally strong alignments. Points below the line represent reads where iiBLAST produced a stronger alignment; points above the line represent reads where ssBLAST gave a stronger alignment.

limited computing power and without extensively optimizing our code for parallelization. However, the results from this experiment are extensible to applications using much larger references. The primary concerns with scaling to a larger reference are keeping the false positive rate low and managing the computation time.

### *False Positives in Large Reference Searches*

For larger reference databases, more high-scoring false positives will arise as there are more chances for random matches to occur. However, iiBLAST’s stronger alignment p-values indicate that the current-to-current method is better suited to large searches (Fig 4.7). Based on the p-value results, we expect iiBLAST will outperform ssBLAST on larger databases, even at higher read accuracies where the two methods both generated 100% true positive rates during the validation experiment (Fig 4.6). The orders-of-magnitude improvement in alignment significance indicates that iiBLAST queries will return useful, false-positive-free results against reference databases orders-of-magnitude larger than would be searchable using ssBLAST.

### *Computation Time for iiBLAST and ssBLAST*

It is important that current-to-current comparison not dramatically add to the computational burden of the BLAST algorithm. In terms of computational complexity, both the seed and extension phases of the iiBLAST implementation are identical to those of ssBLAST. As discussed in section 4.4.1, both approaches use a Mealy finite state machine (appendix G.1) to scan the reference database for seeds in  $\mathcal{O}(N)$  time ( $N$  is the size of the reference database). Likewise, both approaches use the same gapped alignment algorithm (appendix G.3) requiring  $\mathcal{O}(M * L)$  time ( $M$  is the number of extensions prior to terminating the alignment;  $L$  is the window lookout distance).

The iiBLAST method does require the additional computational step of predicting the signal for the reference sequences. However, the prediction only takes  $\mathcal{O}(N)$  time ( $N$  again the size of the reference database), and only needs to be done once for a given reference

genome. Once made, the prediction can be stored along with the sequence in the reference database for future use.

Overall in our validation experiment, iiBLAST aligned reads over 4 times faster than ssBLAST (appendix G.8 contains a more detailed breakdown of algorithm run times). The improved run time is due to better efficiency in finding good seeds rather than any difference in the intrinsic computational load of the two algorithms. The iiBLAST method found fewer bad seeds that led to uninteresting alignments and thus saved time during seed extension.

### *Future Integration*

Looking forward, the current-to-current method will be best used by integrating it into a more sophisticated heuristic alignment algorithm [91, 92, 93, 94] better suited to long, error-prone sequencing data than the standard BLAST algorithm. Such an implementation will harness the improved read-to-reference identity offered by current-to-current alignment within the architecture of an algorithm specifically designed to handle the unique aspects of nanopore sequencing data.

### *4.8.2 Implications for Nanopore Sequencing Applications*

The dramatic improvement in performance achieved by using current-to-current instead of sequence-to-sequence alignment has myriad implications for both the present and future applications of nanopore sequencing.

### *Immediate Implications*

Immediately, our ability to generate strong, unambiguous alignments with low accuracy reads will increase the effective throughput in BLAST-based nanopore sequencing experiments. Fewer reads will fail to align due to low accuracy, making a better fraction of the sequencing data useful to the researcher. It is particularly worth noting that iiBLAST consistently generates good results for reads in the 70-80% accuracy range—the typical accuracy range

for 1D nanopore reads [65] (in which only one strand of the DNA is read through the pore). In comparison, ssBLAST can only achieve similar results for reads well above 80%—above the accuracy of most 1D reads and typical only of 2D reads [65] (in which both strands are read one after the other, then combined into a single, higher accuracy sequence). The iiBLAST method’s better tolerance for low accuracy reads thus makes 1D reads substantially more useful and reduces the need for 2D reads which are necessarily only half as fast (as both strands must be read). By improving nanopore sequencing’s effective throughput by increasing the fraction of usable reads, iiBLAST can reduce sample-to-answer timelines, input sample requirements, and sequencing costs. Effectively, the iiBLAST method can increase the speed and decrease the cost of nanopore sequencing.

### *Further Applications*

The power of the current-to-current comparison method is not limited to improving only BLAST-based nanopore sequencing applications. The same fundamental method can improve nanopore sequencing’s ability to perform various other sequencing tasks, including variant detection [97] and epigenetic mapping [98, 99].

### *Technology Development Implications*

Looking forward, the ability to use low accuracy reads for BLAST-based sequencing experiments offers an alternative way forward for nanopore sequencing to make an impact on the larger sequencing community. To date, many efforts to improve nanopore sequencing have focused on improving raw single-read sequencing accuracy. This work shows that—while improved sequencing accuracy is certainly important—much more can be done with the currently-available low accuracy reads than was previously believed. A future nanopore sequencing device sacrificing sequencing accuracy for lower cost, faster turnaround time, improved portability, and better throughput could prove useful for clinicians and researchers interested in BLAST-based applications.

## Chapter 5

# CONCLUSIONS

The past two decades have witnessed nanopore DNA sequencing mature from a simple, promising, but unproven concept into a fully-realized commercial sequencing platform. Already, nanopore sequencing has demonstrated its ability to make good on many of the promises of single-molecule DNA sequencing, addressing several of the limitations of the previous generation of sequencing technologies. My work has aimed to further extend the frontiers of nanopore sequencing by reducing the limitations posed by its low single-read *de novo* sequencing accuracy. In chapter 2, I discussed the outstanding issues with this technology leading to its low single-read *de novo* sequencing accuracy. My work in chapter 3 showed that a simple re-engineering of the nanopore device—by replacing the constant applied voltage with a variable voltage—addresses these error modes and dramatically improves sequencing accuracy. In chapter 4, I presented a new method of sequence alignment that enables the use of nanopore sequencing devices in diverse applications including pathogen detection and metagenomics. This method is able to harness the information present in the raw nanopore signal to generate useful results even for low accuracy reads. With this method, currently available nanopore sequencing devices can already be used in a wide range of sequencing applications.

The progress presented in this dissertation marks a starting point for the next phase of nanopore technology development. Further improvement in nanopore sequencing accuracy is possible through the integration of the new variable-voltage method with other parallel advances that have also shown improved accuracy. The synthesis of these several techniques into a final nanopore sequencing device should be possible in the near future and result in a new generation of high accuracy nanopore sequencers. Likewise, the sequence alignment



method presented here can be incorporated into other new sequence alignment algorithms specially adapted to handle nanopore sequencing data.

Together, the advancements presented here in improving the accuracy and the application of nanopore sequencing paint a picture of an exciting future for this technology. The next several years should see the complete integration of the new techniques from this work into nanopore devices and workflow, enabling high accuracy nanopore sequencing with broad potential applications. This upcoming generation of nanopore sequencing devices will propagate the on-going genomics revolution through faster, cheaper, more accessible DNA sequencing capable of answering ever more questions, advancing both medicine and science.

## BIBLIOGRAPHY

- [1] A. D. Hershey and Martha Chase. Independent functions of viral protein and nucleic acid in growth of bacteriophage. *The Journal of General Physiology*, 36(1):39–56, 1952.
- [2] James D Watson and Francis HC Crick. The structure of dna. In *Cold Spring Harbor symposia on quantitative biology*, volume 18, pages 123–131. Cold Spring Harbor Laboratory Press, 1953.
- [3] James D Watson and Francis HC Crick. Genetical implications of the structure of deoxyribonucleic acid. *Nature*, 171(4361):964–967, 1953.
- [4] Marshall W Nirenberg and J Heinrich Matthaei. The dependence of cell-free protein synthesis in e. coli upon naturally occurring or synthetic polyribonucleotides. *Proceedings of the National Academy of Sciences*, 47(10):1588–1602, 1961.
- [5] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561, 1970.
- [6] Frederick Sanger, Gilian M Air, Bart G Barrell, Nigel L Brown, Alan R Coulson, John C Fiddes, Clyde A Hutchison III, Patrick M Slocombe, and Mo Smith. Nucleotide sequence of bacteriophage  $\varphi$ x174 dna. *nature*, 265(5596):687, 1977.
- [7] International Human Genome Sequencing Consortium et al. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931, 2004.
- [8] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.
- [9] David R Bentley, Shankar Balasubramanian, Harold P Swerdlow, Geoffrey P Smith, John Milton, Clive G Brown, Kevin P Hall, Dirk J Evers, Colin L Barnes, Helen R Bignell, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *nature*, 456(7218):53, 2008.
- [10] Nicole Rusk. Torrents of sequence. *Nature Methods*, 8(1):44, 2010.
- [11] Aleisha R Reimer, Gary Van Domselaar, Steven Stroika, Matthew Walker, Heather Kent, Cheryl Tarr, Deborah Talkington, Lori Rowe, Melissa Olsen-Rasmussen, Michael

- Frace, et al. Comparative genomics of vibrio cholerae from haiti, asia, and africa. *Emerging infectious diseases*, 17(11):2113, 2011.
- [12] Carol A Gilchrist, Stephen D Turner, Margaret F Riley, William A Petri, and Erik L Hewlett. Whole-genome sequencing in outbreak analysis. *Clinical microbiology reviews*, 28(3):541–563, 2015.
  - [13] Claudio U Köser, Matthew J Ellington, and Sharon J Peacock. Whole-genome sequencing to control antimicrobial resistance. *Trends in Genetics*, 30(9):401–407, 2014.
  - [14] J Craig Venter, Karin Remington, John F Heidelberg, Aaron L Halpern, Doug Rusch, Jonathan A Eisen, Dongying Wu, Ian Paulsen, Karen E Nelson, William Nelson, et al. Environmental genome shotgun sequencing of the sargasso sea. *science*, 304(5667):66–74, 2004.
  - [15] Junjie Qin, Ruiqiang Li, Jeroen Raes, Manimozhiyan Arumugam, Kristoffer Solvsten Burgdorf, Chaysavanh Manichanh, Trine Nielsen, Nicolas Pons, Florence Levenez, Takuji Yamada, et al. A human gut microbial gene catalogue established by metagenomic sequencing. *nature*, 464(7285):59, 2010.
  - [16] Luen-Luen Li, Sean R McCorkle, Sebastien Monchy, Safiyh Taghavi, and Daniel van der Lelie. Bioprospecting metagenomes: glycosyl hydrolases for converting biomass. *Biotechnology for biofuels*, 2(1):10, 2009.
  - [17] Isabelle George, Benoît Stenuit, Spirodon Agathos, and Diana Marco. Application of metagenomics to bioremediation. *Metagenomics: Theory, Methods and Applications*, 1:119–140, 2010.
  - [18] Timothy M Vogel, Pascal Simonet, Janet K Jansson, Penny R Hirsch, James M Tiedje, Jan Dirk Van Elsas, Mark J Bailey, Renaud Nalin, and Laurent Philippot. Terragenome: a consortium for the sequencing of a soil metagenome, 2009.
  - [19] Christopher W Seymour, Foster Gesten, Hallie C Prescott, Marcus E Friedrich, Theodore J Iwashyna, Gary S Phillips, Stanley Lemeshow, Tiffany Osborn, Kathleen M Terry, and Mitchell M Levy. Time to treatment and mortality during mandated emergency care for sepsis. *New England Journal of Medicine*, 376(23):2235–2244, 2017.
  - [20] Sarah A Sterling, W Ryan Miller, Jason Pryor, Michael A Puskarich, and Alan E Jones. The impact of timing of antibiotics on outcomes in severe sepsis and septic shock: a systematic review and meta-analysis. *Critical care medicine*, 43(9):1907, 2015.

- [21] Elaine R Mardis. Applying next-generation sequencing to pancreatic cancer treatment. *Nature reviews Gastroenterology & hepatology*, 9(8):477, 2012.
- [22] Guangwu Guo, Yaoting Gui, Shengjie Gao, Aifa Tang, Xueda Hu, Yi Huang, Wenlong Jia, Zesong Li, Minghui He, Liang Sun, et al. Frequent mutations of genes encoding ubiquitin-mediated proteolysis pathway components in clear cell renal cell carcinoma. *Nature genetics*, 44(1):17, 2012.
- [23] Joanna D Holbrook, Joel S Parker, Kathleen T Gallagher, Wendy S Halsey, Ashley M Hughes, Victor J Weigman, Peter F Lebowitz, and Rakesh Kumar. Deep sequencing of gastric carcinoma reveals somatic mutations relevant to personalized medicine. *Journal of translational medicine*, 9(1):119, 2011.
- [24] Antonio Marchetti, Maela Del Grammastro, Giampaolo Filice, Lara Felicioni, Giulio Rossi, Paolo Graziano, Giuliana Sartori, Alvaro Leone, Sara Malatesta, Michele Iacono, et al. Complex mutations & subpopulations of deletions at exon 19 of egfr in nslc revealed by next generation sequencing: potential clinical implications. *PLoS One*, 7(7):e42164, 2012.
- [25] Yan-Fang Guan, Gai-Rui Li, Rong-Jiao Wang, Yu-Ting Yi, Ling Yang, Dan Jiang, Xiao-Ping Zhang, and Yin Peng. Application of next-generation sequencing in clinical oncology to advance personalized treatment of cancer. *Chinese journal of cancer*, 31(10):463, 2012.
- [26] Adrian Bird. Perceptions of epigenetics. *Nature*, 447(7143):396, 2007.
- [27] Khursheed Iqbal, Seung-Gi Jin, Gerd P Pfeifer, and Pirooska E Szabó. Reprogramming of the paternal genome upon fertilization involves genome-wide oxidation of 5-methylcytosine. *Proceedings of the National Academy of Sciences*, 108(9):3642–3647, 2011.
- [28] Gal-Yam Einav Nili, Yoshimasa Saito, Gerda Egger, and Peter A Jones. Cancer epigenetics: modifications, screening, and therapy. *Annu. Rev. Med.*, 59:267–280, 2008.
- [29] Holger Heyn and Manel Esteller. Dna methylation profiling in the clinic: applications and challenges. *Nature Reviews Genetics*, 13(10):679, 2012.
- [30] Dvir Aran, Sivan Sabato, and Asaf Hellman. Dna methylation of distal regulatory sites characterizes dysregulation of cancer genes. *Genome biology*, 14(3):R21, 2013.
- [31] Peter W Laird. Principles and challenges of genome-wide dna methylation analysis. *Nature Reviews Genetics*, 11(3):191, 2010.

- [32] Michael J Booth, Miguel R Branco, Gabriella Ficz, David Oxley, Felix Krueger, Wolf Reik, and Shankar Balasubramanian. Quantitative sequencing of 5-methylcytosine and 5-hydroxymethylcytosine at single-base resolution. *Science*, 336(6083):934–937, 2012.
- [33] William A Pastor, Utz J Pape, Yun Huang, Hope R Henderson, Ryan Lister, Myunggon Ko, Erin M McLoughlin, Yevgeny Brudno, Sahasransu Mahapatra, Philipp Kapranov, et al. Genome-wide mapping of 5-hydroxymethylcytosine in embryonic stem cells. *Nature*, 473(7347):394, 2011.
- [34] Phillip EC Compeau, Pavel A Pevzner, and Glenn Tesler. How to apply de bruijn graphs to genome assembly. *Nature biotechnology*, 29(11):987, 2011.
- [35] AP Jason de Koning, Wanjun Gu, Todd A Castoe, Mark A Batzer, and David D Pollock. Repetitive elements may comprise over two-thirds of the human genome. *PLoS genetics*, 7(12):e1002384, 2011.
- [36] John J Kasianowicz, Eric Brandin, Daniel Branton, and David W Deamer. Characterization of individual polynucleotide molecules using a membrane channel. *Proceedings of the National Academy of Sciences*, 93(24):13770–13773, 1996.
- [37] Daniel Branton, David W Deamer, Andre Marziali, Hagan Bayley, Steven A Benner, Thomas Butler, Massimiliano Di Ventra, Slaven Garaj, Andrew Hibbs, Xiaohua Huang, et al. The potential and challenges of nanopore sequencing. In *Nanoscience And Technology: A Collection of Reviews from Nature Journals*, pages 261–268. World Scientific, 2010.
- [38] Jiali Li, Derek Stein, Ciaran McMullan, Daniel Branton, Michael J Aziz, and Jene A Golovchenko. Ion-beam sculpting at nanometre length scales. *Nature*, 412(6843):166, 2001.
- [39] J. L. Li, C. McMullan, D. Stein, D. Branton, and J. Golovchenko. Solid state nanopores for single molecule detection. *Biophysical Journal*, 80(1):1419, 2001.
- [40] J. Li, M. Gershow, D. Stein, E. Brandin, and J. A. Golovchenko. Dna molecules and configurations in a solid-state nanopore microscope. *Nature Materials*, 2(9):611–615, 2003.
- [41] Ben McNally, Meni Wanunu, and Amit Meller. Electromechanical unzipping of individual dna molecules using synthetic sub-2 nm pores. *Nano Letters*, 8(10):3418–3422, 2008.

- [42] Meni Wanunu, Jason Sutin, Ben McNally, Andrew Chow, and Amit Meller. Dna translocation governed by interactions with solid-state nanopores. *Biophysical Journal*, 95(10):4716–4725, 2008.
- [43] S. Garaj, W. Hubbard, A. Reina, J. Kong, D. Branton, and J. A. Golovchenko. Graphene as a subnanometre trans-electrode membrane. *Nature*, 467(7312):190–U73, 2010.
- [44] Po-Ssu Huang, Scott E Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320, 2016.
- [45] D. Stoddart, A. J. Heron, E. Mikhailova, G. Maglia, and H. Bayley. Single-nucleotide discrimination in immobilized dna oligonucleotides with a biological nanopore. *Proceedings of the National Academy of Sciences of the United States of America*, 106(19):7702–7707, 2009.
- [46] E. V. B. Wallace, D. Stoddart, A. J. Heron, E. Mikhailova, G. Maglia, T. J. Donohoe, and H. Bayley. Identification of epigenetic dna modifications with a protein nanopore. *Chem. Commun.*, 46:8195–8197, 2010.
- [47] D. Stoddart, A. J. Heron, J. Klingelhofer, E. Mikhailova, G. Maglia, and H. Bayley. Nucleobase recognition in ssdna at the central constriction of the alpha-hemolysin pore. *Nano Lett*, 10(9):3633–7, 2010.
- [48] Tom Z Butler, Mikhail Pavlenok, Ian M Derrington, Michael Niederweis, and Jens H Gundlach. Single-molecule dna detection with an engineered mspa protein nanopore. *Proceedings of the National Academy of Sciences*, 105(52):20647–20652, 2008.
- [49] Michael Faller, Michael Niederweis, and Georg E. Schulz. The structure of a mycobacterial outer-membrane channel. *Science*, 303(5661):1189–1192, 2004.
- [50] Siddharth Shekar, David J Niedzwiecki, Chen-Chi Chien, Peijie Ong, Daniel A Fleischer, Jianxun Lin, Jacob K Rosenstein, Marija Drndic, and Kenneth L Shepard. Measurement of dna translocation dynamics in a solid-state nanopore at 100 ns temporal resolution. *Nano letters*, 16(7):4483–4489, 2016.
- [51] Elizabeth A Manrao, Ian M Derrington, Andrew H Laszlo, Kyle W Langford, Matthew K Hopper, Nathaniel Gillgren, Mikhail Pavlenok, Michael Niederweis, and Jens H Gundlach. Reading dna at single-nucleotide resolution with a mutant mspa nanopore and phi29 dna polymerase. *Nature biotechnology*, 30(4):349, 2012.

- [52] 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-Å precision. *Nature biotechnology*, 30(4):344, 2012.
- [53] Ian M Derrington, Jonathan M Craig, Eric Stava, Andrew H Laszlo, Brian C Ross, Henry Brinkerhoff, Ian C Nova, Kenji Doering, Benjamin I Tickman, Mostafa Ronaghi, et al. Subangstrom single-molecule measurements of motor proteins using a nanopore. *Nature biotechnology*, 33(10):1073–1075, 2015.
- [54] Andrew H Laszlo, Ian M Derrington, and Jens H Gundlach. Mspa nanopore as a single-molecule tool: From sequencing to sprnt. *Methods*, 2016.
- [55] Jonathan M Craig, Andrew H Laszlo, Henry Brinkerhoff, Ian M Derrington, Matthew T Noakes, Ian C Nova, Benjamin I Tickman, Kenji Doering, Noah F de Leeuw, and Jens H Gundlach. Revealing dynamics of helicase translocation on single-stranded dna using high-resolution nanopore tweezers. *Proceedings of the National Academy of Sciences*, 114(45):11932–11937, 2017.
- [56] Carl W Fuller, Shiv Kumar, Mintu Porel, Minchen Chien, Arek Bibillo, P Benjamin Stranges, Michael Dorwart, Chuanjuan Tao, Zengmin Li, Wenjing Guo, et al. Real-time single-molecule electronic dna sequencing by synthesis using polymer-tagged nucleotides on a nanopore array. *Proceedings of the National Academy of Sciences*, 113(19):5233–5238, 2016.
- [57] Swati Bhattacharya, Ian M Derrington, Mikhail Pavlenok, Michael Niederweis, Jens H Gundlach, and Aleksei Aksimentiev. Molecular dynamics study of mspa arginine mutants predicts slow dna translocations and ion current blockades indicative of dna sequence. *ACS nano*, 6(8):6960–6968, 2012.
- [58] Andrew H Laszlo, Ian M Derrington, Brian C Ross, Henry Brinkerhoff, Andrew Adey, Ian C Nova, Jonathan M Craig, Kyle W Langford, Jenny Mae Samson, Riza Daza, et al. Decoding long nanopore sequencing reads of natural dna. *Nature biotechnology*, 32(8):829, 2014.
- [59] 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, 1966.
- [60] Jacob Schreiber and Kevin Karplus. Analysis of nanopore data using hidden markov models. *Bioinformatics*, 31(12):1897–1903, 2015.

- [61] Alexander S Mikheyev and Mandy MY Tin. A first look at the oxford nanopore minion sequencer. *Molecular ecology resources*, 14(6):1097–1102, 2014.
- [62] Vladimír Boža, Broňa Brejová, and Tomáš Vinař. Deepnano: deep recurrent neural networks for base calling in minion nanopore reads. *PloS one*, 12(6):e0178751, 2017.
- [63] Haotian Teng, Minh Duc Cao, Michael B Hall, Tania Duarte, Sheng Wang, and Lachlan JM Coin. Chiron: Translating nanopore raw signal directly into nucleotide sequence using deep learning. *GigaScience*, 7(5):giy037, 2018.
- [64] Matei David, Lewis Jonathan Dursi, Delia Yao, Paul C Boutros, and Jared T Simpson. Nanocall: an open source basecaller for oxford nanopore sequencing data. *Bioinformatics*, 33(1):49–55, 2016.
- [65] Franka J Rang, Wigard P Kloosterman, and Jeroen de Ridder. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome biology*, 19(1):90, 2018.
- [66] Kate R Lieberman, Joseph M Dahl, Ai H Mai, Ashley Cox, Mark Akeson, and Hongyun Wang. Kinetic mechanism of translocation and dntp binding in individual dna polymerase complexes. *Journal of the American Chemical Society*, 135(24):9149–9155, 2013.
- [67] Jean-Michel Carter and Shobbir Hussain. Robust long-read native dna sequencing using the ont csgg nanopore system. *Wellcome open research*, 2, 2017.
- [68] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature biotechnology*, 36(4):338, 2018.
- [69] Miten Jain, John R Tyson, Matthew Loose, Camilla LC Ip, David A Eccles, Justin O’Grady, Sunir Malla, Richard M Leggett, Ola Wallerman, Hans J Jansen, et al. Minion analysis and reference consortium: phase 2 data release and analysis of r9.0 chemistry. *F1000Research*, 6, 2017.
- [70] John F Marko and Eric D Siggia. Stretching dna. *Macromolecules*, 28(26):8759–8770, 1995.
- [71] Steven B Smith, Laura Finzi, and Carlos Bustamante. Direct mechanical measurements of the elasticity of single dna molecules by using magnetic beads. *Science*, 258(5085):1122–1126, 1992.



- [72] Steven B Smith, Yujia Cui, and Carlos Bustamante. Overstretching b-dna: the elastic response of individual double-stranded and single-stranded dna molecules. *Science*, 271(5250):795, 1996.
- [73] Alessandro Bosco, Joan Camunas-Soler, and Felix Ritort. Elastic properties and secondary structure formation of single-stranded dna at monovalent and divalent salt conditions. *Nucleic acids research*, 42(3):2064–2074, 2013.
- [74] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.
- [75] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 – 197, 1981.
- [76] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.
- [77] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.
- [78] Alexander L Greninger, Samia N Naccache, Scot Federman, Guixia Yu, Placide Mbala, Vanessa Bres, Doug Stryke, Jerome Bouquet, Sneha Somasekar, Jeffrey M Linnen, et al. Rapid metagenomic identification of viral pathogens in clinical samples by real-time nanopore sequencing analysis. *Genome medicine*, 7(1):99, 2015.
- [79] Sarah L Castro-Wallace, Charles Y Chiu, Kristen K John, Sarah E Stahl, Kathleen H Rubins, Alexa BR McIntyre, Jason P Dworkin, Mark L Lupisella, David J Smith, Douglas J Botkin, et al. Nanopore dna sequencing and genome assembly on the international space station. *Scientific reports*, 7(1):18022, 2017.
- [80] Joshua Quick, Nicholas J Loman, Sophie Duraffour, Jared T Simpson, Ettore Severi, Lauren Cowley, Joseph Akoi Bore, Raymond Koundouno, Gytis Dudas, Amy Mikhail, et al. Real-time, portable genome sequencing for ebola surveillance. *Nature*, 530(7589):228, 2016.
- [81] Sebastiaan Theuns, Bert Vanmechelen, Quinten Bernaert, Ward Deboutte, Marilou Vandenhoe, Leen Beller, Jelle Matthijssens, Piet Maes, and Hans J Nauwynck. Nanopore sequencing as a revolutionary diagnostic tool for porcine viral enteric disease complexes identifies porcine kobuvirus as an important enteric virus. *Scientific reports*, 8(1):9830, 2018.

- [82] Aline Bronzato Badial, Diana Sherman, Andrew Stone, Anagha Gopakumar, Victoria Wilson, William Schneider, and Jonas King. Nanopore sequencing as a surveillance tool for plant pathogens in plant and insect tissues. *Plant Disease*, pages PDIS-04, 2018.
- [83] Minh Duc Cao, Devika Ganesamoorthy, Alysha G Elliott, Huihui Zhang, Matthew A Cooper, and Lachlan JM Coin. Streaming algorithms for identification of pathogens and antibiotic resistance potential from real-time minion tm sequencing. *Gigascience*, 5(1):32, 2016.
- [84] Joshua Quick, Philip Ashton, Szymon Calus, Carole Chatt, Savita Gossain, Jeremy Hawker, Satheesh Nair, Keith Neal, Kathy Nye, Tansy Peters, et al. Rapid draft sequencing and real-time nanopore sequencing in a hospital outbreak of salmonella. *Genome Biology*, 16(1):114, 2015.
- [85] Nuno R Faria, Josh Quick, IM Claro, Julien Theze, Jacqueline G de Jesus, Marta Giovanetti, Moritz UG Kraemer, Sarah C Hill, Allison Black, Antonio C da Costa, et al. Establishment and cryptic transmission of zika virus in brazil and the americas. *Nature*, 546(7658):406, 2017.
- [86] Bert Vanmechelen, Mads Frost Bertelsen, Annabel Rector, Joost J Van den Oord, Lies Laenen, Valentijn Vergote, and Piet Maes. Identification of a novel species of papillomavirus in giraffe lesions using nanopore sequencing. *Veterinary microbiology*, 201:26–31, 2017.
- [87] Liana E Kafetzopoulou, Kyriakos Efthymiadis, Kuiama Lewandowski, Ant Crook, Dan Carter, Jane Osborne, Emma Aarons, Roger Hewson, Julian A Hiscox, Miles W Carroll, et al. Assessment of metagenomic minion and illumina sequencing as an approach for the recovery of whole genome sequences of chikungunya and dengue viruses directly from clinical samples. *bioRxiv*, page 355560, 2018.
- [88] Joanna Warwick-Dugdale, Natalie Solonenko, Karen Moore, Lauren Chittick, Ann C Gregory, Michael J Allen, Matthew B Sullivan, and Ben Temperton. Long-read metagenomics reveals cryptic and abundant marine viruses. *bioRxiv*, page 345041, 2018.
- [89] Camilla LC Ip, Matthew Loose, John R Tyson, Mariateresa de Cesare, Bonnie L Brown, Miten Jain, Richard M Leggett, David A Eccles, Vadim Zalunin, John M Urban, et al. Minion analysis and reference consortium: Phase 1 data release and analysis. *F1000Research*, 4, 2015.
- [90] Thomas Laver, J Harrison, PA O’neill, Karen Moore, Audrey Farbos, Konrad Paszkiewicz, and David J Studholme. Assessing the performance of the oxford nanopore technologies minion. *Biomolecular detection and quantification*, 3:1–8, 2015.

- [91] Mark J Chaisson and Glenn Tesler. Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): application and theory. *BMC bioinformatics*, 13(1):238, 2012.
- [92] Ivan Sović, Mile Šikić, Andreas Wilm, Shannon Nicole Fenlon, Swaine Chen, and Niranjana Nagarajan. Fast and sensitive mapping of nanopore sequencing reads with graphmap. *Nature communications*, 7:11307, 2016.
- [93] Mohammad Ruhul Amin, Steven Skiena, and Michael C Schatz. Nanoblaster: Fast alignment and characterization of oxford nanopore single molecule sequencing reads. In *Computational Advances in Bio and Medical Sciences (ICCABS), 2016 IEEE 6th International Conference on*, pages 1–6.
- [94] Szymon M Kielbasa, Raymond Wan, Kengo Sato, Paul Horton, and Martin Frith. Adaptive seeds tame genomic sequence comparison. *Genome research*, pages gr–113985, 2011.
- [95] George H Mealy. A method for synthesizing sequential circuits. *Bell System Technical Journal*, 34(5):1045–1079, 1955.
- [96] Miten Jain, Ian T Fiddes, Karen H Miga, Hugh E Olsen, Benedict Paten, and Mark Akeson. Improved data analysis for the minion nanopore sequencer. *Nature methods*, 12(4):351, 2015.
- [97] Laura Gómez-Romero, Kim Palacios-Flores, José Reyes, Delfino García, Margareta Boege, Guillermo Dávila, Margarita Flores, Michael C Schatz, and Rafael Palacios. Precise detection of de novo single nucleotide variants in human genomes. *Proceedings of the National Academy of Sciences*, 115(21):5516–5521, 2018.
- [98] Andrew H Laszlo, Ian M Derrington, Henry Brinkerhoff, Kyle W Langford, Ian C Nova, Jenny Mae Samson, Joshua J Bartlett, Mikhail Pavlenok, and Jens H Gundlach. Detection and mapping of 5-methylcytosine and 5-hydroxymethylcytosine with nanopore mspa. *Proceedings of the National Academy of Sciences*, 110(47):18904–18909, 2013.
- [99] Jared T Simpson, Rachael E Workman, PC Zuzarte, Matei David, LJ Dursi, and Winston Timp. Detecting dna cytosine methylation using nanopore sequencing. *Nature methods*, 14(4):407, 2017.
- [100] 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-a precision. *Nature biotechnology*, 30(4):344–348, 2012.

- [101] Colin H LaMont and Paul A Wiggins. The development of an information criterion for change-point analysis. *Neural computation*, 28(3):594–612, 2016.
- [102] Swati Bhattacharya, Jejoong Yoo, and Aleksei Aksimentiev. Water mediates recognition of dna sequence via ionic current blockade in a biological nanopore. *ACS nano*, 10(4):4644–4651, 2016.
- [103] Lalit Bahl, John Cocke, Frederick Jelinek, and Josef Raviv. Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *IEEE Transactions on information theory*, 20(2):284–287, 1974.
- [104] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

## Appendix A

### SEQUENCING EXPERIMENTS

#### A.1 *Materials and Methods*

##### A.1.1 *Proteins*

The same mutant MspA protein was used as the nanopore in all of the presented sequencing experiments. This mutant, M2-NNN-MspA, was custom ordered from GenScript. M2-NNN-MspA is engineered on the wild type MspA (accession number CAB56052.1) with the following mutations: D90N/D91N/D93N/D118R/E139K/D134R [48]. All sequencing experiments used the Hel308 helicase enzyme from *Thermococcus gammatolerans* EJ3 (accession number WP\_015858487.1). Hel308 was expressed using standard techniques by in-house facilities [53]. The  $\Phi$ 29 DNAP used in preliminary experiments was wild-type  $\Phi$ 29 DNAP obtained from Enzymatics and Epicenter. All proteins were stored at -20 °C until immediately before use.

##### A.1.2 *Nanopore Experiments*

All experiments used a single M2-NNN-MspA nanopore. Detailed description of our nanopore experiments is provided in [54]. Briefly, experiments were established on a Teflon platform containing two  $\sim 50 \mu\text{L}$  chambers (*cis* and *trans*). The two chambers are connected by a Teflon heat-shrink “u-tube”,  $\sim 30 \mu\text{L}$  in volume. The *cis* side of the u-tube narrows into a horizontal  $\sim 20 \mu\text{m}$  diameter aperture. Both chambers and the u-tube were filled with the operating buffers. The *cis* well was connected to the negative terminal of the amplifier (ground) via an Ag/AgCl electrode. A lipid bilayer was painted across the aperture using 1,2-diphytanoyl-sn-glycero-3-phosphocholine (DPhPC) or 1,2-di-O-phytanyl-sn-

glycero-3-phosphocholine (DOPC), obtained from Avanti Polar Lipids. An Axopatch 200B integrating patch clamp amplifier (Axon Instruments) applied the driving voltage across the bilayer (*trans* side positive) and measured the ionic current throughout the experiment.

Following bilayer formation, M2-NNN-MspA was added to the *cis* well to a final concentration of  $\sim 2.5$  ng/mL. A single pore insertion into the bilayer was recognized by a characteristic increase in the conductance. Upon single pore insertion, the *cis* well buffer was perfused out and replaced with MspA-free buffer to prevent the insertion of additional pores. The motor enzyme was added to a final *cis* well concentration of 50 nM, and DNA was added to a final concentration of  $\sim 5$  nM.

### A.1.3 Operating Buffers

All sequencing experiments (using Hel308) were conducted using symmetric *cis* and *trans* buffer conditions of 400 mM KCl with 10 mM HEPES at pH  $8.00 \pm 0.05$ . The *cis* buffer additionally contained 1 mM EDTA, 1 mM DTT, 10 mM MgCl<sub>2</sub>, and 100  $\mu$ M ATP. ATP-containing buffer was re-perfused into *cis* approximately once per hour to prevent depletion of ATP and accumulation of ADP. Hel308 experiments were performed with at 37 °C.

Preliminary  $\Phi 29$  DNAP experiments were conducted using 300 mM KCl, 10 mM HEPES buffer at pH  $8.00 \pm 0.05$  in both *cis* and *trans*. Again, 1 mM EDTA, 1 mM DTT, and 10 mM MgCl<sub>2</sub> were included in the *cis* buffer only.  $\Phi 29$  DNAP experiments can be designed so that the DNA is passed through the pore twice [100] [51], first proceeding toward *trans* (5' direction, unzipping mode) then being pulled back toward *cis* (3' direction, synthesis mode). For these dual-mode experiments, dATP, dCTP, dGTP, and dTTP were added to the *cis* buffer to a final concentration of 10 nM.  $\Phi 29$  DNAP experiments were performed at 22 °C.

### A.1.4 Data Acquisition and Analysis

Data were acquired with acquisition software written in LabView (National Instruments) at a sampling rate of 50 kHz using an Axopatch 200B amplifier low pass filtered at 10 kHz. Ionic current traces were analyzed using custom programs written in Matlab (the

Mathworks). Enzyme-controlled DNA-translocation events were detected via a thresholding algorithm. The open pore ionic current value is determined for the data, and an event is called whenever the ionic current drops below 75% of the open state value. The event end is called when the ionic current returns to greater than 94% of the open pore value. Events failing certain basic criteria (duration longer than 1s, an average current less than 10% or greater than 70% of the open pore ionic current) were automatically discarded. Remaining events were classified by-eye based on their quality.

Small variations in temperature, salt concentration, and electrode offsets day-to-day, pore-to-pore, and read-to-read cause changes in both the overall magnitude of the observed ionic currents (and conductances) (an “offset”) as well as the relative magnitudes of adjacent states (a “scale”). We calibrate each read back to the 6-mer model prior to sequencing using a scale and an offset calculated specifically for that read.

#### *A.1.5 DNA Sequences and Constructs*

Short DNA oligonucleotides were synthesized and purified using column purification methods at Stanford University Protein and Nucleic Acid.  $\Phi$ X-174 DNA was obtained from New England Biolabs.  $\lambda$  DNA was obtained from Promega. pET-28a was obtained from collaborators who used it as an expression vector for another DNA sequence not used in this work.

All experiments were conducted with the DNA threaded through the pore 5' first. Constructs for  $\Phi$ 29 DNAP experiments consisted of a template read strand, a blocking strand, and a cholesterol primer strand, as shown in Fig A.1a. The phosphate at the 5' end of the read strand increases the capture rate by MspA. The cholesterol tag at the 5' end of the cholesterol primer strand anchors the DNA constructs into the bilayer, increasing the local concentration near the pore and improving capture rate.  $\Phi$ 29 DNAP constructs were annealed by mixing the read strand, blocker strand, and primer strand to a relative molar concentration of 1:1:2, then heating to 95 °C for 5 minutes, cooling at 1 °C/s to 60 °C, holding there for 2 minutes, then finally cooling to 4 °C at 1 °C/s.

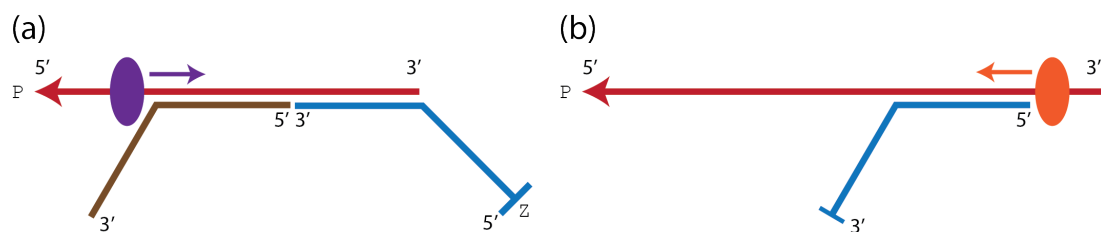


Figure A.1: DNA constructs for  $\Phi 29$  and Hel308. **(a)** The basic DNA constructs for  $\Phi 29$  experiments consist of 3 partially complementary strands. The template read strand (red) is the sequence that we read in the experiment. Its 5' terminal phosphate (P, arrowhead) facilitates threading into MspA. A blocker strand (brown) forms a y-tail at the 5' end of the template read strand where  $\Phi 29$  DNAP (purple) loads on. The cholesterol primer strand (blue) enables a second read of the DNA by  $\Phi 29$ 's synthesis mode. The 5' terminal cholesterol on this strand up-concentrates the DNA in the bilayer to improve capture rate. **(b)** The basic DNA constructs for Hel308 experiments consist of 2 partially complementary strands. Again, the template read strand (red) has a 5' terminal phosphate. The cholesterol blocker strand (blue) has a 3' terminal cholesterol. The template read strand has an 8 base 3' overhang where Hel308 (orange) loads on.

Constructs for Hel308 experiments consisted of a template read strand and a cholesterol-tagged blocking strand, as shown in Fig A.1b. Again, a 5' phosphate on the read strand facilitates capture by MspA. A 3' cholesterol on the blocking strand up-concentrates the DNA in the bilayer. Read and blocking strands were mixed at a 1.2:1 molar ratio, then annealed using the same procedure as for the  $\Phi 29$  DNAP constructs.

The preparation of long genomic DNA for building the variable-voltage signal-to-sequence model is covered in appendix E.

#### A.1.6 Constant-Voltage Sequencing Experiments

Constant-voltage sequencing experiments were conducted using Hel308, in standard Hel308 buffer conditions at 37 °C. A constant 180 mV voltage was applied.



### *A.1.7 Variable-Voltage Sequencing Experiments*

Variable-voltage sequencing experiments were conducted using Hel308, in standard Hel308 buffer conditions at 38 °C. The voltage was applied as a 200 Hz, 100 mV peak-to-peak symmetric triangle wave, offset to a mean voltage of 150 mV. The cycling frequency of 200 Hz gave an integer number of sampling points per cycle (250), as well as ensured that most enzyme steps would last for multiple full voltage cycles (section A.4). The 100 to 200 mV voltage range ensures that the DNA stays anchored in the pore (as the voltage is always positive) and provides over a full nucleotide of stretch (appendix C.4), giving good overlap between the conductance curves of sequential states.

## ***A.2 Using Hel308 as a Translocase***

In our Hel308 experiments, Hel308 operates as a translocase rather than a helicase, moving directionally (from 3' to 5') over a ssDNA track. Hel308 is a poor helicase in our operating conditions and does not readily unwind dsDNA. Translocase experiments proceed as diagrammed in Fig A.2.

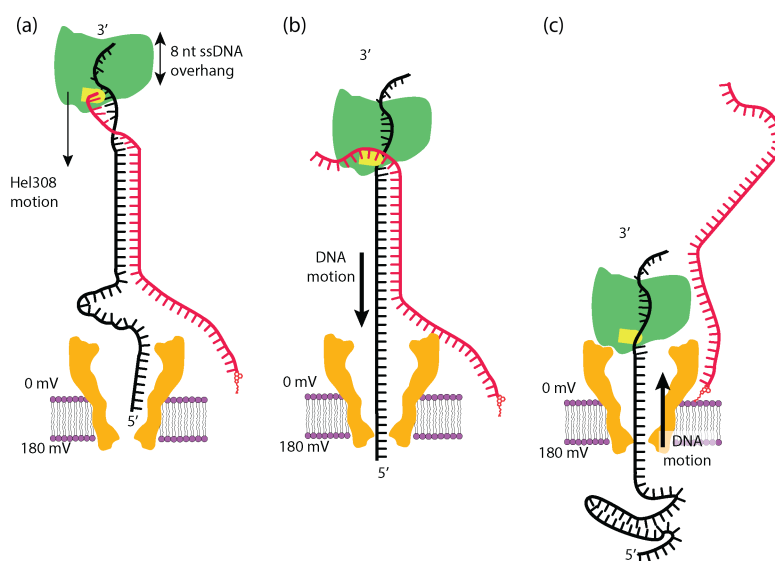


Figure A.2: Hel308-controlled DNA translocation through MspA. **(a)** Hel308 (green) binds on to the overhanging 3' end of the template DNA strand (black) at the ssDNA-dsDNA junction. Hel308 is an inefficient helicase in our experimental conditions and does not substantially unwind the dsDNA ahead of it. **(b)** The 5' end of the template strand is captured by the pore (gold). As the template strand is pulled through, the blocking strand (red) is sheared off, as dsDNA is too large to fit through the pore's constriction. **(c)** Once the blocking has been sheared off, the template strand is fully threaded through the pore, with the attached Hel308 blocking complete translocation. At this point, Hel308 is able to operate as a translocase rather than a helicase, and begins to walk towards the 5' end of the DNA. As Hel308 progresses along the DNA, the DNA is pulled out of the pore. This image is adapted from [53].

### A.3 *Hel308 Processivity*

The read length in both our constant-voltage and variable-voltage sequencing experiments is limited by the processivity of the Hel308 helicase enzyme we use to control DNA motion through the pore. The enzyme’s processivity is the typical number of nucleotides it translocates through the pore before it dissociates from the DNA, ending the event. Processivity can in principle be a function of various experimental conditions, including temperature, substrate and salt concentration, pH, and applied force (i.e. applied voltage). Hel308’s activity is insensitive to force over the range of forces (voltages) we apply in our experiments, so its stepping rate and processivity should not change with the variable applied voltage [53] [55].

We observed Hel308’s processivity under our variable-voltage conditions (section A.1.7) by looking at the read lengths obtained on our  $\Phi$ X-174 construct. Specifically, we looked at read lengths on the larger, 5042 bp fragment, as this fragment is long enough that nearly all reads terminated due to the helicase unbinding from the DNA prior to reaching the end of the strand. Based on alignments of the reads to the  $\Phi$ X-174 reference, we investigated 50 reads of the long fragment, all starting at the same location in the genome (at the *Ava*II cut site, appendix E.3.1). Hel308 shows little ability to unwind DNA in our experimental conditions, so all reads began at the loading site and only progressed once the duplex strand had been sheared away by the pore. The read survival fraction as a function of read length was calculated as the number of reads reaching a given position in  $\Phi$ X-174 over the total number of reads. We found that the survival fraction  $f$  as a function of read length  $l$  was well modeled by a single exponential function of the form  $f(l) = e^{-\frac{l}{l_p}}$ , where  $l_p$  is the characteristic processivity of the enzyme (Fig A.3). The single-exponential form of the survival fraction indicates that Hel308 dissociation from the ssDNA track in our experiments is dominated by a single rate-limiting step. From our data, we found a best fit processivity of  $l_p = 945 \pm 139$  nt.

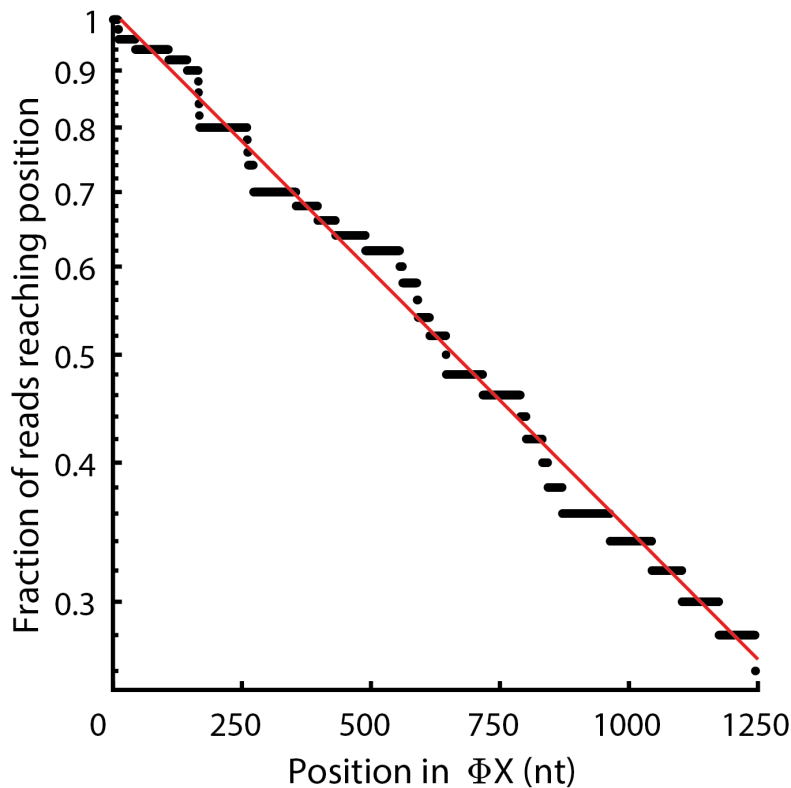


Figure A.3: Hel308 processivity in variable-voltage sequencing conditions. The fraction of reads (all starting from the same cut site in  $\Phi X$ -174) reaching a given position in the genome is plotted as block dots on a logarithmic y-scale. We see an exponential fall-off in read survival, indicating a read-termination process dominated by a single off-rate. The red line shows the best-fit single exponential model of the form  $f(l) = e^{-l/l_p}$ , with the best fit obtained with  $l_p = 945$  nt.

#### **A.4 *Hel308 Step Durations***

The distribution of step durations for Hel308 in our sequencing experimental conditions (section A.1.7) is shown in Fig A.4. We need 3 complete voltage cycles (15 ms) to accurately estimate the covariance of the 3 principal components for a state (appendix C.3), otherwise we must take a default value for the covariance for the state. Over 90% of states are long enough to accurately estimate the covariance (blue). A small fraction (<10%, red) are too short and are assigned a default covariance value.

States shorter than a full voltage cycle (5 ms) will not be detected by the change-point detection algorithm and ultimately manifest as skips in the final data. However, for these conditions such short states should make up only a small minority of the total Hel308 states. Long term, it will be desirable to use a faster enzyme (or experimental conditions in which Hel308 steps faster) in to increase throughput and decrease the per-read time. To accomplish this in variable-voltage setting, we will need to increase the variable-voltage cycle frequency. The primary limitation to increasing the cycle frequency is that the capacitive current (appendix C.1) increases with increasing rate of change in the voltage. If the capacitive current becomes too large, it could rail the amplifier (rail is  $\pm 1$  nA), resulting in a loss of signal.

This issue can be addressed in multiple ways. First, reducing bilayer capacitance is a straightforward way of reducing the capacitive current. A commercial sequencing device will need to be dramatically miniaturized compared to the experiments we run in our lab, and will require an automated method of bilayer formation. These automatically-formed bilayers can in principle be much smaller (lower capacitance) than the hand-painted bilayers used in this work. Second, we have some room to reduce the range of the voltage sweep without compromising the efficacy of the variable-voltage signal. The 100-200 mV swing currently in use gives us more than enough state-to-state overlap to identify and correct enzyme missteps. A smaller voltage range should still provide adequate overlap, while reducing the rate of voltage change and thus the size of the capacitive current. Finally, on-line methods can be used to compensate for the capacitive signal. The injection of an in-phase square

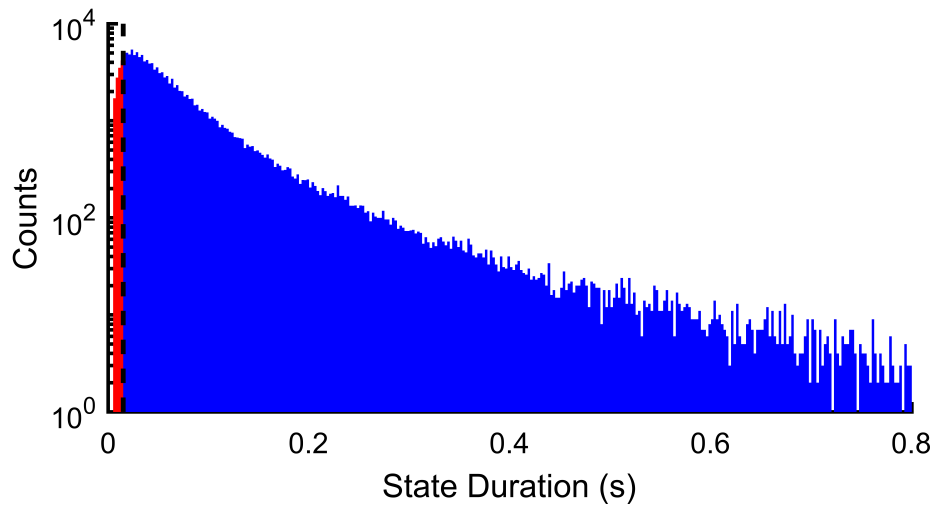


Figure A.4: Hel308 step durations in variable-voltage sequencing conditions. Marked in red are states too short to accurately estimate the covariance of the 3 principal components. States of sufficient length for this estimation are marked in blue.

wave current into the system to counteract the square wave contribution of the capacitance would allow us to take the variable-voltage method to much higher cycle frequencies.

### **A.5 Sequencing Verification Experiment**

We tested the relative performances of our constant-voltage and variable-voltage sequencing methods by using both methods to sequence DNA from the pET-28a vector. The pET-28a vector was chosen as it represented a readily available genomic DNA sequence and was not involved in our 6-mer model construction, thus avoiding the risk of over-training artificially boosting our sequencing numbers. Given Hel308's limited processivity (section A.3), an experiment in which all reads began from the same start point in pET-28a would be unlikely to generate good coverage throughout the sequence and instead concentrate most reads on the same  $\sim 1000$  base pairs nearest the start point. To get broad coverage throughout the sequence, and to get reads of both the sense and antisense strands, we fragmented the pET-28a sequence using a double restriction digest. Digestion gave us a variety of 100-1000 base fragments for our sequencing experiments (table A.5), which were generated, then prepared for Hel308 sequencing experiments as follows.

1. The pET-28a vector was digested using the NspI and Sau3aI restriction enzymes (New England Biolabs). We used  $3.5 \mu\text{L}$  of  $10000 \frac{\text{U}}{\text{mL}}$  NspI and  $7 \mu\text{L}$  of  $5000 \frac{\text{U}}{\text{mL}}$  Sau3aI per  $17.5 \mu\text{g}$  of vector DNA. Following digestion, fragments were cleaned on DNA Clean and Concentrator column (Zymo Research).
2. Following digestion, we prepared 4 distinct adapter constructs for ligation. The four constructs were
  - (a) Sau3aI threading adapter, composed of the Sau3aI threading strand and the Sau3aI cholesterol blocker (table A.3)
  - (b) Sau3aI loading adapter, composed of the Sau3aI loading strand and the Sau3aI loading blocker
  - (c) NspI threading adapter, composed of the NspI threading strand and the NspI cholesterol blocker



- (d) NspI loading adapter, composed of the NspI loading strand and the NspI loading blocker

We require four adapter constructs as each pET-28a fragment needs an threading adapter to facilitate capture into the pore and a loading adapter to facilitate Hel308 loading onto the DNA. Each of the two cutsites needs its own set of loading and threading adapters as the two restriction enzymes leave different sticky-end overhangs. Adapters were prepared individually by mixing equimolar portions of the two constituent oligos and annealing using standard annealing protocols (appendix E).

3. We ligated the several adapters to the pET-28a fragments by mixing the fragmented DNA with the annealed adapter constructs in approximately equimolar ratios <sup>1</sup>, then incubating with T4 DNA ligase. Following ligation, the final products were purified using another DNA Clean and Concentrator column.

There are a few important drawbacks to the above-described preparation procedure that must be considered in estimating the overall yield and in conducting downstream analysis. First, due to the palindromic nature of both the NspI and Sau3aI cutsites, we are not guaranteed to correctly get one each of the loading and threading adapters ligated to each fragment. Indeed, 25% of the total fragments will have the correct adapters for a sense strand read, 25% will have the correct adapters for an antisense strand read, and 50% will have either 2 loading adapters or 2 threading adapters and will be unlikely to produce reads. Even with this 50% drop off in the effective yield of this preparation procedure, we were still able to generate plenty of DNA to collect the data needed.

On this same note, the loading and threading adapters for each cutsite are themselves self-complementary at their overhanging sticky ends. This can lead to the formation of so-called adapter dimers, where two adapters ligate to each other. When a loading adapter

---

<sup>1</sup>The exact molarity of the pET-28a DNA is difficult to determine as it is not clear what fraction of which fragments survived the clean-up step on the column.

Fragment Length (bp)	Left Cutsite	Right Cutsite
826	Sau3aI	Sau3aI
570	Sau3aI	NspI
409	Sau3aI	Sau3aI
389	Sau3aI	Sau3aI
373	Sau3aI	Sau3aI
367	NspI	NspI
346	Sau3aI	Sau3aI
292	NspI	NspI
252	Sau3aI	Sau3aI
207	Sau3aI	Sau3aI
178	Sau3aI	Sau3aI
145	NspI	Sau3aI
132	Sau3aI	Sau3aI
130	Sau3aI	Sau3aI
104	Sau3aI	Sau3aI
shorter fragments		

Table A.1: pET-28a fragments from double digest with NspI and Sau3aI. Only fragments longer than 100 bp are shown, shorter fragments resulted in short sequencing reads that were not used for the validation experiment.

ligates to a threading adapter, we create a DNA construct that can both load Hel308 and thread into the pore, and so is likely to be read. We see a population of these dimers in our experiments, and discard them from later analysis based on their characteristically short length and recognizable pattern of states.

The final drawback also stems from the palindromic nature of the restriction cutsites. The sticky-end overhangs left on the pET-28a fragments after digestion are self-complementary, which can lead to chimera formation. Chimeras occur when different fragments from disparate parts of the pET-28a reference sequence ligate together. We see a population of these chimeras in our reads. There is nothing intrinsically wrong with the chimera reads, but determining the base calling accuracy for these reads is more difficult. In these cases, we must piece together the ground truth reference sequence by separately aligning the smaller fragments composing the chimera to find which parts of the reference sequence have been stitched together. The called sequence is then compared against this stitched-together ground truth

sequence to evaluate the read accuracy.

## **A.6 *Random Sequencing Accuracy***

We empirically determined the random accuracy baseline for our constant-voltage and variable-voltage sequencing reads by aligning random sequences of the same length as the called reads against the reference pET-28a genome. The pET-28a reference genome is 5204 bp, so the sense and antisense sequences together comprise a 10408 base reference. In general, the random accuracy of a local-to-global alignment, in which a shorter called sequence is aligned to the best-matching location of a longer reference sequence is a function of the called length and the reference length. Additionally, genomic DNA is highly non-random, so the distribution of accuracies of random called sequences against a genomic reference sequence may differ from the distribution obtained from alignment against a random reference sequence of the same length. Fig A.5 shows the random accuracy dependence on called sequence length for comparisons against the pET-28a reference sequence. Several random sequences were generated for a variety of lengths from 50 to 10408 bases and aligned against the sense plus antisense pET-28a sequence. We see that shorter reads result in higher random accuracies than long reads, with random accuracy falling from 64.5% for the 50 base reads to 54.3% to 10408 base reads (full length reads).

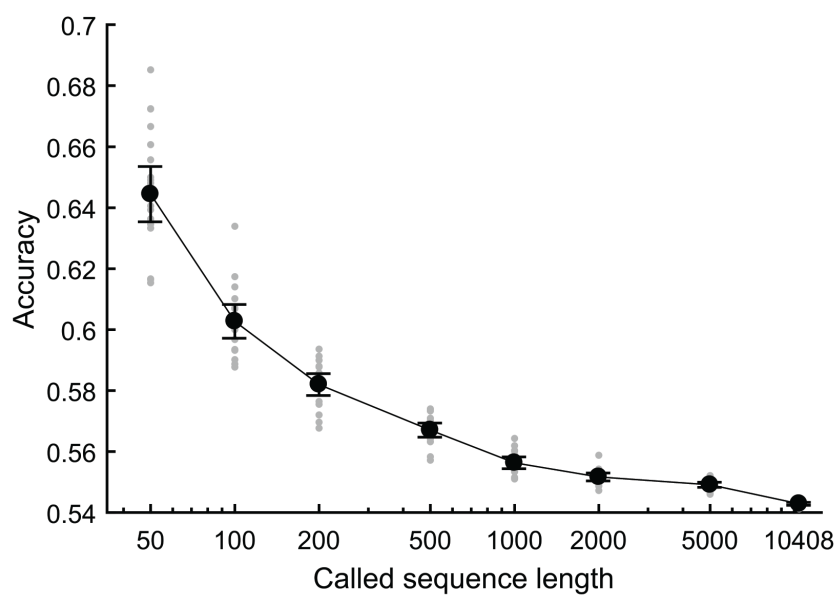


Figure A.5: Random sequencing accuracy as a function of called sequence length. The mean random sequencing accuracy generated by aligning random sequences of various lengths to the pET-28a reference sequence are shown (black markers). Error bars show one standard deviation around the mean. Gray markers show the individual trial results for each read length.

## A.7 DNA Sequences

Tables A.2 and A.3 contain a list of the short custom DNA sequences used in our sequencing and DNA stretching experiments. In addition to these short sequences, we used the  $\lambda$  phage (Promega) and  $\Phi$ X-174 genomes (New England Biolabs) as well as the pET-28a vector (from collaborators).

DNA Construct Name	Sequence (5'→3')										
<b><math>\Phi</math>X174 Experiments</b>	1	6	11	16	21	26	31	36	41	46	51
$\Phi$ X174 threading strand	FXAAA AAAAC CTTCX XCCAT CATCA TCAGA TCTCA CGCGG TGCA										
$\Phi$ X174 cholesterol blocker	PCCGC GTGAG ATCTG AAAAA TTATA ACCCA AAXZ										
$\Phi$ X174 loading strand	PGACC CGCCA AGTAC AAGTA AGCCT ACGCC TACGG TTTT TTTT TTTT TTTT										
$\Phi$ X174 loading blocker	CCGTA GCGGT AGGCT TACTT GTACT TGCGG G										
<b><math>\lambda</math>-phage Experiments</b>	1	6	11	16	21	26	31	36	41	46	51
$\lambda$ threading strand	PTACT ACTAC TACTA CTACX XTITT GAGCC TCTCA CTATC GCATT CTCAT GCAGG T										
$\lambda$ cholesterol blocker	PCCGT CATGA GAATG CGATA GTGAG ATCGT AGCCQ QQQZ										
$\lambda$ loading strand	PGGAC GTACT CTTAC GCTAT CACTC TTCGT AGCC										
$\lambda$ loading blocker	AGAGT GATAG CGTAA GAGTA CGTCC T										
<b>Missing 6-mer Experiments</b>	1	6	11	16	21	26	31	36	41	46	51
	56	61	66	71	76	81	86	91	96	101	106
	111	116	121	126	131	136	141	146	151	156	
Fill-in template 1	PTACT ACTAC TACTA CTACX XTITT TTGGC GCTTC ATACA GCGGC GCCGG CGAGA TTTTG GCGAG ACAGG CACGC GCGAG CCCAA TCTAT TTTCA ATCTA CGTAT ACTAG GGGGT TCTAG TACTT TTTCT CACTA TCGCA TTCTC ATGCA GGTGG TAGCC										
Fill-in template 2	PTACT ACTAC TACTA CTACX XTITT CTAGT ACACT AGACT AGTCC CTACT ACGAT TTTTG TACGA TTAGG GCGCT ATCTA ATCTA GAGTT TTTCT AGAGT AGGGA CCCCC GGACT CCGTT GTATT TTTCT CACTA TCGCA TTCTC ATGCA GGTGG TAGCC										
Fill-in template 3	PTACT ACTAC TACTA CTACX XTITT CCTTG TAGAT CCTAT ACGGA CGGGG TCTCT TTTTG GTCTC TAGCG CTCGA ATGTG TCGAC ACCCT TTTGA CACCT CAGAG ACCTA GCTAG GCTAG TGTTT TTTCT CACTA TCGCA TTCTC ATGCA GGTGG TAGCC										
Fill-in template 4	PTACT ACTAC TACTA CTACX XTITT CTAGT GTACA CCTCG GACCG GTGCC CTCGA TTTTG CCTCG AGAGG ACCAT GCTAG CCCCC CGCTT TTTCC CGGCT ATACA AGTAC CCGAG TTAGA ACTTT TTTCT CACTA TCGCA TTCTC ATGCA GGTGG TAGCC										
Fill-in template 5	PTACT ACTAC TACTA CTACX XTITT TAGAA CTAGG ATAGG GTGGG GCACA TACCT TTTTG ATACC TAGGT CCGAA TCGAT CTTAG CCTAT TTTTA GCGTA AGGAT AGACG TGATT GGGCC TACTT TTTCT CACTA TCGCA TTCTC ATGCA GGTGG TAGCC										
Fill-in template 6	PTACT ACTAC TACTA CTACX XTITT CATAC GTAGC ATTTT TCATA GGCCT CATTT TTGAT CCGGC GCATT TTTCA TCTA CGCAT TTTT TCTAC TATCG CATTC TCATG CAGGT CGTAG CC										
Fill-in cholesterol blocker	CTGCT ATGAG AATGC GATAG TGAGA QQQQZ										
<b>Legend:</b> P = phosphate, K = 3 carbon spacer, Q = 18 carbon spacer, X = abasic site, Z = cholesterol tag											

Table A.2: Table of short DNA sequences used for constructing the variable-voltage 6-mer model.

DNA Construct Name	Sequence (5'→3')
<b>pET28a Sequencing Experiments</b>	1 6 11 16 21 26 31 36 41 46 51 56 61 66 71
Sau3aI threading strand	PTACT ACTAC TACTA CTACT ACTAC TACTA CXXTT TTATT GAAGT GCAGT ACTTT ACTAA TTATT GCITT T
Sau3aI cholesterol blocker	PGATC AAAAG CAATA ATTAG TAAAG TACTG CACTT CAATQ QQQZ
Sau3aI loading strand	PGATC ATTGA AGTGC AGTAC TTTAC TAATT ATTGC TTTTT CTGAG CC
Sau3aI loading blocker	AAAAG CAATA ATTAG TAAAG TACTG CACTT CAAT
NspI threading strand	PTACT ACTAC TACTA CTACT ACTAC TACTA CXXTT TTATT GAAGT GCAGT ACTTT ACTAA TTATT GCITT TCATG
NspI cholesterol blocker	PAAAA GCAAT AATTA GTAAA GTACT GCACT TCAAT QQQZ
NspI loading strand	PATTG AAGTG CAGTA CTTTA CTAAT TATTG CTTTT TCTGA GCC
NspI loading blocker	AAAAG CAATA ATTAG TAAAG TACTG CACTT CAATC ATG
<b>DNA Stretching Experiments</b>	1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81
Stretching read strand	PAAAA AAACC TTCCX ACACC GATTC TCCCG AGTGG GCCGA ATCAA GCAC TAATA AAAGC ATTCT CATGC AGGTC GTAGC C
Stretching blocker strand	TTTTA TTAGT TGCTT GATTC GGCC XXXXXXXX K
Stretching cholesterol primer strand	TTTTT TTTTT TTTTT TTTTT TTTTT TTTTT TTTTT TTTTT TTTTT TXXXX XGGCT ACGAC CTGCA TGAGA ATGC
<b>Legend:</b> P = phosphate, K = 3 carbon spacer, Q = 10 carbon spacer, X = abasic site, Z = cholesterol tag	

Table A.3: Table of short DNA sequences used for measuring DNA stretching and for validating variable-voltage sequencing performance.

## A.8 Experimental Statistics

Statistics for the variable-voltage and constant-voltage experiments conducted to generate the 6-mer model, validate the performance of variable-voltage sequencing, and measure the stretching response of DNA in MspA in response to voltage are summarized in table A.4.

Experiment Description	Enzyme control mechanism	DNA Sequence	Number of Pores	Number of Events
6-mer model building, SVM training	Hel308	$\Phi$ X174	19	155
6-mer model building	Hel308	$\lambda$ phage	46	128
6-mer model building	Hel308	Fill-in template 1	3	18
6-mer model building	Hel308	Fill-in template 2	3	28
6-mer model building	Hel308	Fill-in template 3	7	38
6-mer model building	Hel308	Fill-in template 4	6	30
6-mer model building	Hel308	Fill-in template 5	4	25
6-mer model building	Hel308	Fill-in template 6	4	33
Variable-voltage sequencing	Hel308	pET28a	10	73
Constant-voltage sequencing	Hel308	pET28a	21	31
DNA stretching	$\Phi$ 29	Stretching strand	3	30

Table A.4: Experimental statistics. The number of pores run and the total number of enzyme-controlled DNA translocation events collected are summarized for the experiments underpinning the development and demonstration of variable-voltage sequencing.

## A.9 Work Fuel

Both the writing of this dissertation and the data analysis presented within were fueled by copious amounts of tea. A summary of the tea consumed to keep this work moving forward can be found in table A.5.



Flavor	Count	Flavor	Count
Lemon Lift	119	Orange and Spice	9
Darjeeling	51	Earl Grey	8
Constant Comment	47	Chai Green	5
China Black	44	English Breakfast (Ceylon)	5
China Green	38	Irish Breakfast	5
Green (Twinnings)	36	Pomegranate White	5
Lady Grey	35	Chamomile	4
Super Irish Breakfast	20	Mint Medley	2
Ginger Peach Green	18	Chai	1
Herbal	18	China Green Tips	1
Vanilla Chai	17	Ginger Green	1
Green (Private Selection)	16	Green (Stash)	1
French Vanilla	15	Lemon Black	1
Jasmine Green	12	Passion Fruit	1
English Breakfast (Twinnings)	10	Pomegranate Pizzazz	1
Matcha Green	9	Zen	1
—	—	<b>Total</b>	<b>556</b>

Table A.5: Summary of tea used to generate this dissertation. We found Lemon Lift to be most effective, but variety is indispensable.

## Appendix B

### CHANGE POINT DETECTION ALGORITHM

#### *B.1 Basic Description*

In both constant-voltage and variable-voltage sequencing, our first step is to partition the raw time-series ionic current data into segments corresponding to enzyme steps. Partitioning simplifies the data stream passed to the hidden Markov model by turning the many noisy measurements making up an enzyme step observation into a series of a few low-noise parameters describing each step. In the case of constant-voltage sequencing, each enzyme step is described by a mean ionic current and an associated variance. For variable-voltage sequencing, we use the coefficients of the top three principal components (appendix C.3), along with their associated covariance.

The data is partitioned into enzyme steps using a change point detection algorithm (algorithm 4). The same fundamental algorithm works for both constant-voltage and variable-voltage sequencing data. Simply, the change point algorithm chooses between two competing hypotheses. Given a segment of data  $\{x_i\}$ , is the data best modeled by a single model (parameterized as  $\theta_T$ ) or by two models ( $\theta_L, \theta_R$ ) each separately describing the data to the left and right of some transition point  $t$ ? If the single-model hypothesis proves better, no change point is present in the segment. If the two-model hypothesis is better, a change point is called at the best transition point.

The basic considerations in this type of algorithm are how to model the data, and how to prevent over-calling transitions. In the case of constant-voltage data, we use a mean ionic current and a variance to describe the individual states. We model the variable-voltage data by the five largest principal components of the periodic functions that represent the raw data of each enzyme state (Fig B.1). These principal components were determined by choosing

change points by-eye, then averaging each enzyme state into a single 250-sample period of the waveform <sup>1</sup>. We then treated each state as a separate measurement for the purposes of principal component analysis. The principal components provide a descriptive, concise basis with which we can describe the variable-voltage time series data.

The over-calling issue is a consequence of the fact that a model with more parameters can always describe a data set better than a model with fewer parameters, even if it is not actually more predictive. Consequently, the two-model hypothesis will always fit the data better—the question is rather is it sufficiently better to justify the addition of more parameters into our description of the data? We correct this bias by penalizing the addition of extra parameters, using the results of Lamont and Wiggins [101] to determine the appropriate penalty.

## ***B.2 Mathematical Description***

The following is a full mathematical description of the change point detection procedure.

The change point problem is formulated mathematically as follows: given a time series of  $d$ -dimensional data  $\{x_1, x_2, \dots, x_N\}$ ,  $x \in \mathbb{R}^k$ , choose a model consisting of some number of change points  $\{t_a, t_b, \dots\}$ , and a different set of parameters  $\{\theta_a, \theta_b, \dots\}$  describing the data between each change point. Our change point detection algorithm (algorithm 4) finds a close-to-optimal partitioning of time series data using this model.

We assume each state is a function  $f$  of time  $t$  and parameters  $\theta$  with normally distributed noise  $\sigma$ . Under these assumptions, the probability density of obtaining a measurement  $x(t)$  at a time  $t$  given a choice of parameters  $\theta$  for that time is

---

<sup>1</sup>We actually do not model the entire 250-point cycle period but instead discard the first and last 20 points in each cycle period for the purposes of change point detection. Thus, the principal components model a 210-point period. This is done to avoid anomalous capacitive behavior commonly present at the beginning and end of cycles (around the change from increasing to decreasing voltage) from confusing the change point detection procedure. These points are later re-incorporated after change point detection for down-stream data reduction and analysis.

---

**Algorithm 4** Change point detection
 

---

```

1: Input:
    $d$ -dimensional data  $\{x_i\}, i \in \{1 : N\}$ 
   Transition threshold  $\mathcal{T}$ 
2: Initialize:  $\{t\} \leftarrow []$  ▷ Initialize an empty list of transition points
3: function PARTITION( $\{x_i\}, \mathcal{T}, \{t\}$ )
4:   Score: assign a score  $\mathcal{S}_i$  for the placement of a transition at each point  $i \in \{1 : N\}$ 
5:    $\mathcal{S}_{best} \leftarrow \max(\{\mathcal{S}_i\})$ 
6:    $t_{best} \leftarrow i$  such that  $\mathcal{S}_i = \mathcal{S}_{best}$ 
7:   if  $\mathcal{S}_{best} > \mathcal{T}$  then ▷ The best transition point is good enough to call a transition
8:      $\{t\}[end] \leftarrow t_{best}$  ▷ Add the found transition to the growing list
9:      $\{t\} \leftarrow \text{PARTITION}(\{x_i\} \text{ } i \in \{1 : t_{best}\}, \mathcal{T}, \{t\})$  ▷ Recursively find partitions in
       the data to the left of the found transition point
10:     $\{t\} \leftarrow \text{PARTITION}(\{x_i\} \text{ } i \in \{t_{best} : N\}, \mathcal{T}, \{t\})$  ▷ Recursively find partitions in
       the data to the right of the found transition point
11:   else
12:     Output:  $\{t\}$ 
13:   end if
14: end function

```

---

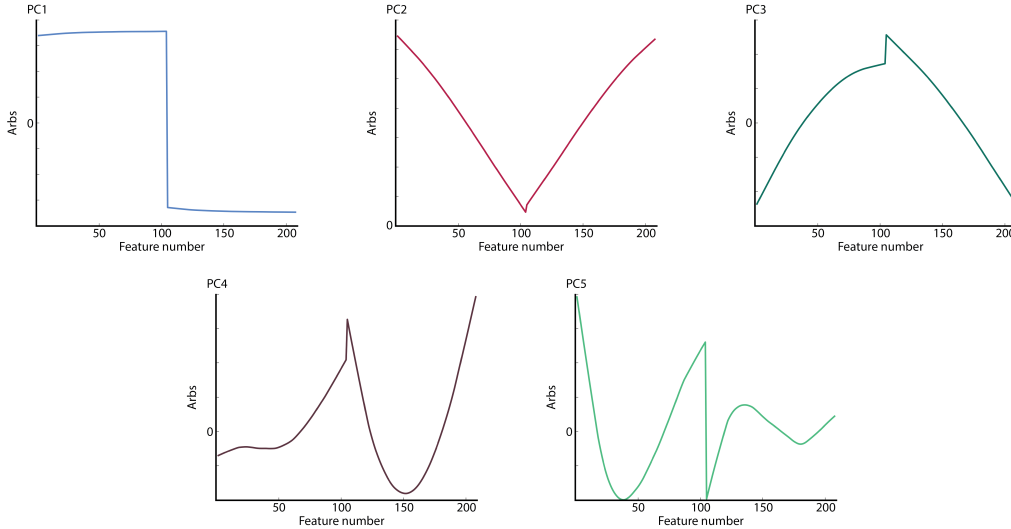


Figure B.1: Principal components for change point detection. The five principal component vectors used to model the variable-voltage time series data are shown. Linear combinations of these five vectors can describe the observed data.

$$p(x(t), t | \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f(t;\theta) - x(t))^2}{2\sigma^2}}$$

For a number of measurements indexed by time  $t = 1, 2, 3, \dots$ , the probability density is the product of the probabilities of each measurement:

$$p(x | \theta) = \prod_{t=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f(\theta)_t - x_t)^2}{2\sigma^2}}$$

We convert this probability into a log-likelihood  $L(\theta | x) = p(x | \theta)$  to simplify calculations, giving

$$\log \mathcal{L} = -\frac{1}{2} \sum_{t=1}^N \log 2\pi\sigma^2 + \frac{(f(\theta)_t - x_t)^2}{\sigma^2}$$

For change point detection, we are interested in the relative likelihood between using two different sets of parameters  $\theta_L$  and  $\theta_R$  to model the data to the left and right of a possible change point, versus using one set of parameters  $\theta_T$  to describe the total region in question. Defining the first time index of the region as  $L$ , the final index as  $R$ , and the number of points to the left, the right and in the whole region as  $N_L, N_R$ , and  $N_T = R - L + 1$  respectively, the relative log-likelihood is

$$\log \mathcal{L} = -\frac{1}{2} \left[ \sum_{t=L}^{L+N_L-1} \log 2\pi\sigma_L^2 + \frac{(f(\theta_L)_t - x_t)^2}{\sigma_L^2} + \sum_{t=L+N_L}^{N_T} \log 2\pi\sigma_R^2 + \frac{(f(\theta_R)_t - x_t)^2}{\sigma_R^2} - \sum_{t=L}^{L+N_T-1} \log 2\pi\sigma_T^2 + \frac{(f(\theta_T)_t - x_t)^2}{\sigma_T^2} \right]$$

If our maximum likelihood estimate for  $\theta$  given the data is  $\hat{\theta}$ , the residual variance is  $\hat{\sigma}^2 = \frac{1}{N} \sum_t (f(\hat{\theta})_t - x_t)^2$ . We find the maximum log-likelihood  $\log \hat{\mathcal{L}}$  by plugging in these estimators:

$$\begin{aligned}\log \hat{\mathcal{L}} &= -\frac{1}{2} \left[ N_L \log 2\pi \hat{\sigma}_L^2 + N_L \frac{\hat{\sigma}_L^2}{\hat{\sigma}_L^2} + N_R \log 2\pi \hat{\sigma}_R^2 + N_R \frac{\hat{\sigma}_R^2}{\hat{\sigma}_R^2} - N_T \log 2\pi \hat{\sigma}_T^2 - N_T \frac{\hat{\sigma}_T^2}{\hat{\sigma}_T^2} \right] \\ &= -\frac{1}{2} [N_R \log \hat{\sigma}_R^2 + N_L \log \hat{\sigma}_L^2 - N_T \log \hat{\sigma}_T^2]\end{aligned}$$

This is the correct expression for the log-likelihood of a fit, giving the most *descriptive* model of the data. However, we are not interested in the most *descriptive* but rather the most *predictive* model. To find this, we need to correct for the tendency to over-fit. We can always fit better by partitioning data and supplying more parameters, but we lose information by doing so. This leads to an over-fitting bias. To correct for this, we use the results of LaMont and Wiggins [101] to subtract this bias. In general, the bias is a function of the number of points  $N_T$  and the dimensionality of the data being partitioned  $d$ , and is calculated through Monte Carlo simulations and either fitted or used as a lookup table. The test statistic is

$$\text{CPIC} = -\log \hat{\mathcal{L}} + p(N, d) = \frac{N_R}{2} \log \hat{\sigma}_R^2 + \frac{N_L}{2} \log \hat{\sigma}_L^2 - \frac{N_T}{2} \log \hat{\sigma}_T^2 + p_d(N_T)$$

where  $p_d(N_T)$  is the penalty for adding parameters in modeling  $N_T$   $d$ -dimensional data points. The natural choice is to call a level transition if  $\text{CPIC}_p < 0$ . A simple way to tune the sensitivity of this score is to apply a multiplier  $\lambda > 0$  to  $p_d$ , which can be made higher to increase the penalty and find fewer levels. This is done to compensate for a model that does not exactly describe the data; we choose  $\lambda = 4$  because it provides empirically good results. So the final score used is

$$\text{CPIC}(\lambda) = \frac{N_R}{2} \log \hat{\sigma}_R^2 + \frac{N_L}{2} \log \hat{\sigma}_L^2 - \frac{N_T}{2} \log \hat{\sigma}_T^2 + \lambda p_d(N_T)$$

To calculate the  $\hat{\sigma}$ 's, we need to determine the maximum likelihood estimates of the model parameters  $\hat{\theta}$ . Obtaining these can in general be slow and difficult, possibly even requiring nonlinear optimization not guaranteed to converge. However, in certain situations

it is easy, and we can even take advantage of some tricks to avoid redundant calculation. The simplest example is the case of constant levels about a single mean. The maximum likelihood estimate of the mean in bounds  $[L, R]$  is

$$\hat{\mu} = \frac{1}{R - L + 1} \sum_{t=L}^R x_t$$

We can avoid continually re-adding the same points together by instead defining and pre-calculating the cumulate  $X_t = \sum_{s=1}^t x_s$ , in which case our expression for the mean is simply

$$\hat{\mu} = \frac{X_R - X_L}{R - L + 1}.$$

This difference is much more expedient to calculate than the mean, and the calculation of its value for many possible transition points may be vectorized. We can use a similar technique to calculate the variance,

$$\hat{\sigma}^2 = \frac{1}{R - L + 1} \sum_{t=L}^R (x_t - \hat{\mu})^2 = \left[ \frac{1}{R - L + 1} \sum_{t=L}^R x_t^2 \right] - \hat{\mu}^2$$

Again defining and pre-calculating the cumulate sum

$$X_t^2 = \sum_{s=1}^t x_s^2,$$

we quickly calculate the MLE variance as

$$\hat{\sigma}^2 = \frac{X_R^2 - X_L^2}{R - L + 1} - \hat{\mu}^2.$$

In general, the function  $f(\theta)$  may depend on time. One case is if we can write  $f(\theta)_t$  as a sum of  $p$  basis functions  $b_{it}$  with amplitudes  $\theta_i$ ,

$$f(\theta)_t = \sum_{i=1}^p \theta_i b_{it}.$$

Assuming again normally distributed random errors, we find maximum likelihood estimators

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \sum_{t=L}^R \left( x_t - \sum_{i=1}^p \theta_i b_{it} \right)^2$$

To this end, we set the derivative of the sum squared error to zero,

$$\begin{aligned} 2 \sum_{t=L}^R \left( x_t - \sum_{i=1}^p \hat{\theta}_i b_{it} \right) b_{jt} &= 0 \\ \sum_{t=L}^R x_t b_{jt} &= \sum_{i=1}^p \hat{\theta}_i \sum_{t=L}^R b_{it} b_{jt} \end{aligned}$$

Both of these sums over time are again precalculable from cumulates, which we define as

$$B_{ijt} = \sum_{s=1}^t b_{is} b_{js} \quad (\text{note: } B_{ijt} \text{ is symmetric in } i \text{ and } j.)$$

$$c_{it} = \sum_{s=1}^t x_s b_{is}$$

Then, in vector notation, interpreting  $\mathbf{c}_t$  as the vector with elements  $(c_{1t}, c_{2t}, \dots, c_{pt})$ ,  $\boldsymbol{\theta}$  as  $(\theta_1, \theta_2, \dots, \theta_p)$ , and  $B_t$  as the matrix with  $B_{ijt}$  as the element at row  $i$  and column  $j$ , the expression for  $\hat{\boldsymbol{\theta}}$  becomes

$$[\mathbf{c}_R - \mathbf{c}_L]^T = \hat{\boldsymbol{\theta}}^T [B_R - B_L]^T$$

$$[\mathbf{c}_R - \mathbf{c}_L] = [B_R - B_L] \hat{\boldsymbol{\theta}}$$

$$\hat{\boldsymbol{\theta}} = [B_R - B_L]^{-1} [\mathbf{c}_R - \mathbf{c}_L]$$

We can now calculate  $\sigma$  on that domain to be used in the CPIC calculation. The sum of



squared errors is

$$\hat{\sigma}^2 = \frac{1}{R-L+1} \sum_{t=L}^R \left[ x_t - \sum_{i=1}^p \hat{\theta}_i b_{it} \right]^2.$$

Expanding the squared term,

$$\hat{\sigma}^2 = \frac{1}{R-L+1} \sum_{t=L}^R \left[ x_t^2 - 2 \sum_{i=1}^p \hat{\theta}_i x_t b_{it} + \sum_{i,j=1}^p \hat{\theta}_i b_{it} b_{jt} \hat{\theta}_j \right].$$

Plugging in our expression for  $\hat{\theta}_i$ ,

$$\begin{aligned} \hat{\sigma}^2 = \frac{1}{R-L+1} & \left[ \sum_{t=L}^R x_t^2 - 2 \sum_{i,j=1}^p [B_R - B_L]_{ij}^{-1} [c_R - c_L]_j \sum_{t=L}^R x_t b_{it} \right. \\ & \left. + \sum_{i,j,k,l=1}^p [B_R - B_L]_{ik}^{-1} [c_R - c_L]_k \left( \sum_{t=L}^R b_{it} b_{jt} \right) [B_R - B_L]_{jl}^{-1} [c_R - c_L]_l \right]. \end{aligned}$$

Defining one more cumulate  $X_t^2 = \sum_{s=1}^t x_s^2$  and plugging in this as well as other cumulate expressions,

$$\begin{aligned} \hat{\sigma}^2 = \frac{1}{R-L+1} & \left[ X_R^2 - X_L^2 - 2 \sum_{i,j=1}^p [B_R - B_L]_{ij}^{-1} [c_R - c_L]_j [c_R - c_L]_i \right. \\ & \left. + \sum_{i,j,k,l=1}^p [B_R - B_L]_{ik}^{-1} [c_R - c_L]_k [B_R - B_L]_{ij} [B_R - B_L]_{jl}^{-1} [c_R - c_L]_l \right]. \end{aligned}$$

$$\begin{aligned} \hat{\sigma}^2 = \frac{1}{R-L+1} & \left[ X_R^2 - X_L^2 - 2 \sum_{i,j=1}^p [B_R - B_L]_{ij}^{-1} [c_R - c_L]_j [c_R - c_L]_i \right. \\ & \left. + \sum_{i,j=1}^p [B_R - B_L]_{ik}^{-1} [c_R - c_L]_j [c_R - c_L]_i \right]. \end{aligned}$$

$$\hat{\sigma}^2 = \frac{1}{R-L+1} \left[ X_R^2 - X_L^2 - \sum_{i,j=1}^p [c_R - c_L]_i [B_R - B_L]_{ij}^{-1} [c_R - c_L]_j \right]$$

Or, in vector notation,

$$\hat{\sigma}^2 = \frac{1}{R-L+1} \left[ X_R^2 - X_L^2 - [\mathbf{c}_R - \mathbf{c}_L] [B_R - B_L]^{-1} [\mathbf{c}_R - \mathbf{c}_L] \right]$$

At every possible division point we must invert a unique matrix, but these matrices are small, and with a reasonably small number of basis functions applying this algorithm is not too slow. For the variable-voltage data, we used the five largest principal components of the periodic ionic current signal, as described above.

## Appendix C

### VARIABLE VOLTAGE DATA REDUCTION

#### C.1 Capacitance Compensation

The bilayer separating the *cis* and *trans* wells acts as a capacitor. When operating the nanopore sequencer at constant voltage, the capacitor's presence in the circuit is unimportant. However, when operating using a time-varying voltage, the capacitor introduces an additional charging and discharging ionic current  $I_{cap}$  which must be removed from the signal  $I_{sig}$  we wish to observe. Thus, rather than directly reading out the sequence-dependent ionic current signal, the observed ionic current  $I_{obs}$  takes the form  $I_{obs} = I_{sig} + I_{cap}$

$I_{cap}$  depends on both the size of the capacitor formed by the bilayer (a constant value over the course of the experiment) and the size of the resistor formed by the pore and the translocating DNA (which varies as a function of the sequence present within the pore). Because the resistance is different at each ionic current state, capacitance compensation is conducted separately for each ionic current state.

As  $I_{cap}$  is proportional to the rate of change of the voltage  $\frac{dV}{dt}$ , our triangle wave applied voltage (section A.1.7) causes an in-phase square wave capacitive current, plus decaying exponential contributions around the ill-defined regions of  $\frac{dV}{dt}$  when the voltage transitions from up-slope to down-slope and back. The goal of our capacitance compensation procedure is to infer the  $I_{cap}$  from the asymmetry between the current values during the up-slope and down-slop voltage ramps, then subtract out this inferred signal to reveal  $I_{sig}$ .

The procedure is as follows.

1. The overall phase of the signal is calculated from the applied voltage signal for the entire read. Knowing the overall phase, along with the number of data points collected per voltage cycle (50  $kHz$  sampling rate, with the voltage cycling at 200  $Hz$  gives 250

points per cycle) allows us to assign an identification index between 1 and 250 to each point in the ionic current trace  $I(t)$  marking its phase.

2. For each ionic current state, all data points in the ionic current trace  $I(t)$  are grouped by their previously determined identification index, thus binning together all data points collected at the same location in the voltage sweep. For each ionic current state, the ionic current trace  $I(t)$  is divided into “up-slope” and “down-slope” based on the identification index previously determined (Fig C.1a,b).
3. For both the up-slope and down-slope data, we group and average all data points with the same identification index, thus finding the average ionic current value at each location in the voltage cycle. This yields the average current-voltage ( $I - V$ ) characteristic for both up and down:  $I_{up}(V)$  and  $I_{down}(V)$  (Fig C.1c).
4. Taking the difference between the two  $I - V$  curves, we get the asymmetry between the sweeps,  $H(V) = I_{down}(V) - I_{up}(V)$  (Fig C.1d).
5. To find the magnitude of the square wave component in the capacitive signal, which appears as a systematic offset  $m$  between the up and down  $I - V$  curves, we fit a parabola to the residual,  $H(V)$ , over the second and third quartiles in the voltage (125 to 175 mV). The x-coordinate of the parabola’s vertex is constrained to occur at the voltage midpoint (150 mV), and the y-coordinate is taken as the systematic offset  $m$ . Low and high voltages are omitted in order to isolate the offset, without interference from the sharp spikes appearing near the voltage turnaround points. A parabolic fit is used in lieu of a mean, as  $H(V)$  exhibits some curvature even over the middle voltage quartiles due to the decaying exponential current spikes generated at the voltage turnaround points.
6. Capacitive correction functions for up and down ( $Corr_{up}(V)$  and  $Corr_{down}(V)$ ) are generated from the left and right halves of the residual function  $H(V)$  (Fig C.1e, f).

The residual function is split around the midpoint voltage of the sweep  $V_{mid}$ , and the correction functions are given by:

For  $V < V_{mid}$ ,

$$Corr_{up}(V) = H(V) - \frac{m}{2}$$

$$corr_{down}(V) = -\frac{m}{2}$$

And for  $V > V_{mid}$

$$Corr_{up}(V) = \frac{m}{2}$$

$$Corr_{down}(V) = \frac{m}{2} - H(V)$$

Splitting the correction in this way attributes the spike at low voltage to the up-sweep and the spike at high voltage to the down-sweep. The overall offset  $m$  is attributed equally to both sweep directions. This assignment is justified, as the capacitive effect of an instantaneous change in  $V(t)$  falls off exponentially, with time constant  $RC$ . As the low voltage turnaround immediately precedes the up-slope region, the effect of this turnaround is strong in the up-slope, but negligible by the down slope. The opposite is true for the high voltage turnaround. The overall offset is the manifestation of the square wave current generated by the constant  $\frac{dV}{dt}$  throughout the rest of the triangle wave, and so appears equally in both up-slope and down-slope curves.

7. Applying the up and down correction functions to their respective  $I - V$  curves gives the corrected curves  $I_{up}^{cc}$  and  $I_{down}^{cc}$ :

$$I_{up}^{cc} = I_{up}(V) + Corr_{up}(V)$$

$$I_{down}^{cc} = I_{down}(V) + Corr_{down}(V)$$

The corrected curves show no residual hysteresis, and the spikes around the turnarounds have been eliminated (Fig C.1g).

8. Lastly, the correction is applied to all  $I(t)$ , at each point according to the identification index previously determined. This yields the capacitance compensated  $I(t)$  trace that will be used in all further analysis (Fig C.1h).

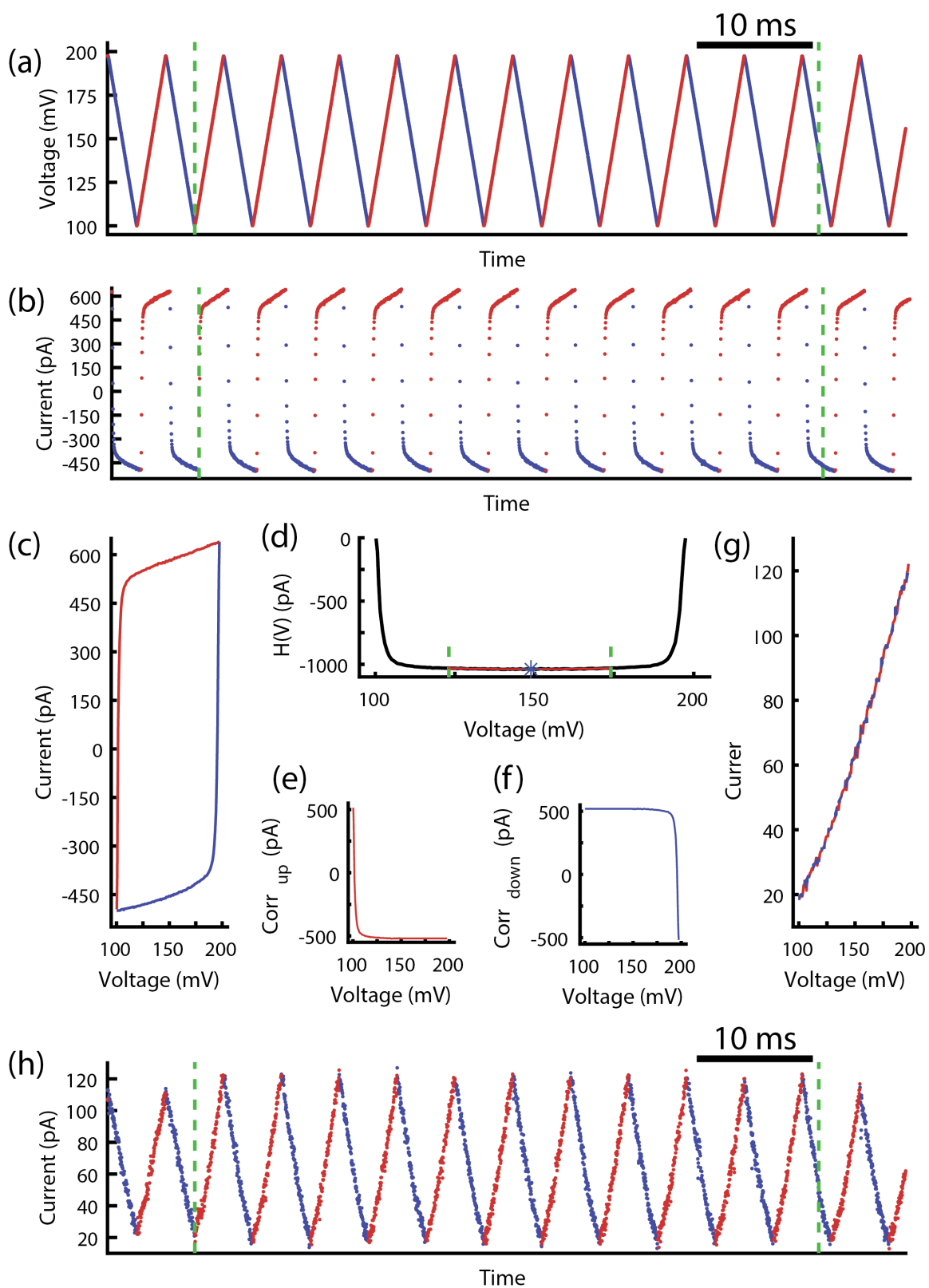


Figure C.1: Capacitance compensation. **(a)** Voltage time series for a single variable-voltage ion current state. Points at which the voltage is increasing are marked red, those at which the voltage is decreasing are marked blue. Dashed green lines mark the beginning and end of the ionic current state. **(b)** Raw ionic current time series for the single ionic current state in (a). Again, red points mark where the voltage is increasing, blue mark where the voltage is decreasing. Green dashed lines mark the beginning and end of the ionic current state. **(c)** Raw ionic current vs. voltage curve for the above ionic current state. Individual cycles have been averaged together. Red shows the average up-slope curve, blue the average down-slope curve. **(d)** Residual function  $H(V)$  for the above ionic current state. Black shows the residual as a function of voltage. Green dashed lines mark the first and third quartiles in the voltage over which the quadratic fit is calculated. Red shows the quadratic fit to these data. The blue asterisk marks the calculated vertex. **(e, f)** Calculated correction functions to be added in to the up-slope (e) and down-slope (f)  $I - V$  curves. **(g)** Corrected up-slope (red) and down-slope (blue)  $I - V$  curves. Dashed lines are used as both curves lie directly on top of one another after the capacitance compensation has removed all hysteresis. **(h)** Corrected current time series for the above ionic current state. Up- and down-slopes are marked in red and blue; dashed green lines mark the beginning and end of the ionic current state.

## C.2 Conductance Normalization

Following capacitance compensation (section C.1), the response to the changing voltage in the variable-voltage signal retains a nuisance component in addition to the DNAposition-dependent portion of the signal (which is the signal we are ultimately interested in for sequencing). This complicating component, dominated by the intrinsically non-ohmic character of the pore’s conductance when blockaded by a charged molecule, is mostly additive with the DNAdependent portion of the signal, but is not itself affected by DNA position. We need to remove this portion of the signal in order to arrive at the purely DNAposition-dependent conductance signal that changes smoothly as a function of DNA position. We refer to the process of removing the non-position-dependent portion of the conductance as “normalization” and refer to the final smooth conductance profile as the “normalized” conductance.



To find the normalized conductance curve  $g_i(V)$  of a state  $i$ , we take an average of the conductance at each voltage  $g_j(V)$  over each state in a read ( $j \in 1 : N$  where  $N$  is the number of states), and subtract this mean conductance from the measured conductance from each state:

$$g_i(V) = g_i(V) - \frac{1}{N} \sum_{j=1}^N g_j(V) \quad (\text{C.1})$$

In effect, this process estimates the position-independent contribution to each state's conductance curve as the portion of the curve found on average in all of the states, then removes this shared component.

Following this simple normalization, we require a further correction to fully realize the continuous conductance profile. We observe a “fraying” of the segments in the continuous curve (Fig C.2). That is, at high voltage, states with normalized conductances well above the mean tend to be exaggerated and take higher values than what is necessary for the curve to be continuous. Likewise, states with conductances below the mean take values lower than would be expected for the continuous curve. We attribute this effect to the stretching of the DNA at high voltages. The additional elongation of the DNA at higher voltage means that fewer bases on average will contribute to the instantaneous conductance through the pore, as fewer bases spend time near the constriction. So, the DNA-dependent signal is dominated further by a few bases at high voltage than low voltage. This effect serves to exaggerate the peaks and troughs in the normalized signal.

To correct for this effect, we note that there should be no correlation between the applied voltage and the DNA-dependent conductance. Therefore, we correct to the first order by fitting a linear model to each reduced conductance curve, obtaining from each  $g_i$  a slope  $m_i$ . These slopes are then linearly fit to the voltage means of the normalized conductances,

$$\langle g_i \rangle = \frac{1}{N_V} \sum_{j=1}^{N_V} g_i(V_j) \quad (\text{C.2})$$

where  $N_V$  is the number of voltages measured in each state ( $= 101$ ). This fit with slope

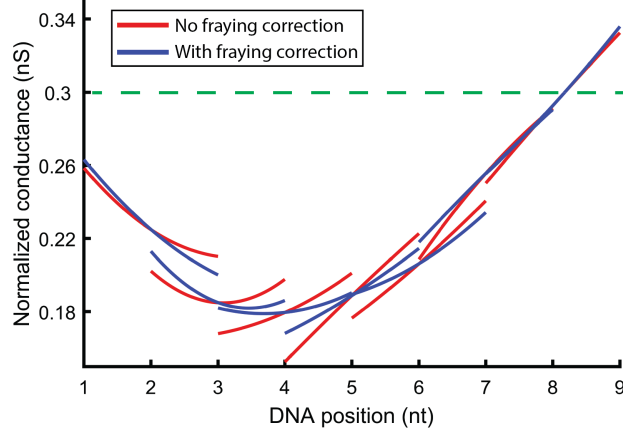


Figure C.2: Fraying correction. The linear fray correction accounts for the exaggerated effects of a few bases on the conductance at high voltage. The initial mean-only normalization (red) demonstrates systematic discontinuities around peaks (not shown) and troughs (shown here) where the high-voltage points (left on each segment) are too high (peaks) or low (troughs). The dashed green line shows the overall average conductance (for the whole read, of which only a short section is shown). The fray correction accounts for this and generates a more-continuous conductance profile (blue).

$\alpha$  represents the magnitude of the linear voltage response as a function of conductance. Subtracting this bias, we obtain the final normalized conductance which represents the DNA-dependent signal that will ultimately be used:

$$g_i(V) = g_i(V) - \frac{1}{N} \sum_{j=1}^N g_j(V) - \alpha V \left( g_i(V) - \frac{1}{N} \sum_{j=1}^N g_j(V) \right) \quad (\text{C.3})$$

### C.3 Feature Extraction

Following change point detection (appendix B) and capacitance compensation (section C.1), the sequencing data is in the form of a series of time-ordered ionic current-vs-voltage ( $I - V$ ) curves. These  $I - V$  curves are converted to conductance-vs-voltage ( $G - V$ ) curves by dividing out the voltage from the ionic current. Going forward from here, variable-voltage sequencing analysis is conducted using conductance in lieu of voltage.

Each  $G - V$  curve characterizes one enzyme step along the DNA, as determined during change point detection. Each  $G - V$  curve is made up of 101 conductance measurements taken at voltages between 110 and 190 mV, represented by a 101-dimensional feature (column) vector,  $\mathbf{g}$ . The sampled voltage points are chosen so that the shift in DNA registration between each consecutive pair of points is uniform—we sample the conductance uniformly over DNA position, but non-uniformly over voltage. Uniform sampling over position ensures maximum independence between the sampled conductances.

The 101 elements (features) in  $\mathbf{g}$  are largely not independent. Many of the features provide redundant information and serve only to introduce noise into our characterization of the states. We used principal component analysis (PCA) to reduce the dimensionality of the feature vectors describing each state. PCA revealed that the top 3 principal components explain nearly 98% of the variance between  $G - V$  curves. In light of this, we reduce the dimensionality of the feature vectors from 101 to 3 by replacing the 101 sampled conductances with the coefficients of the top 3 principal components (Fig C.3).

We calculate the reduced 3-dimensional feature vector  $\mathbf{p}_i$  for state  $i$  as

$$\mathbf{p}_i = [\boldsymbol{\pi}_1; \boldsymbol{\pi}_2; \boldsymbol{\pi}_3]^T * \mathbf{g}_i \quad (\text{C.4})$$

where  $\boldsymbol{\pi}_j$  is the  $j^{th}$  principal component (column) vector. This dimensional reduction allows us to satisfactorily characterize each state while dramatically de-noising our description (Fig SC.4).

Additionally, we are much better able to estimate the covariance amongst the features for these smaller feature vectors. Each full voltage cycle  $j$  (200 Hz) completed during a given state  $i$  provides two measurements  $\mathbf{g}_i^j$  of the state's conductance feature vector  $\mathbf{g}_i$ , one from the voltage up-swing, one from the voltage down-swing. Similarly, we can treat the 3 principal component coefficients for each half cycle  $\mathbf{p}_i^j$  as distinct measurements of the overall principal component feature vector  $\mathbf{p}_i$ . Given  $t$  half-cycle measurements, we can estimate the covariance in the state's conductance ( $\Sigma_i^g$ ) and principal component features ( $\Sigma_i^p$ ) as

$$\Sigma_i^g = \mathbb{E}_{j \in 1:t} [(\mathbf{g}_i^j - \mathbb{E}_{j \in 1:t} [\mathbf{g}_i^j])(\mathbf{g}_i^j - \mathbb{E}_{j \in 1:t} [\mathbf{g}_i^j])^T] \quad (\text{C.5})$$

and

$$\Sigma_i^p = \mathbb{E}_{j \in 1:t} [(\mathbf{p}_i^j - \mathbb{E}_{j \in 1:t} [\mathbf{p}_i^j])(\mathbf{p}_i^j - \mathbb{E}_{j \in 1:t} [\mathbf{p}_i^j])^T] \quad (\text{C.6})$$

The estimators  $\Sigma_i^{g,p}$  are only well defined if we have at least as many measurements as there are independent entries elements in the covariance matrix. As covariance matrices are symmetric,  $\Sigma_i^{g,p}$  has  $\frac{d}{2} * (d - 1)$  independent entries, where  $d$  is the dimensionality of the associated  $\mathbf{g}$  or  $\mathbf{p}$  feature vector. So, in order to get a good estimate of the covariance of the conductance features  $\mathbf{g}_i$  for a given state, we require 5050 half-cycle measurements, representing over 12.5 seconds spent in that state—far longer than the typical state duration. Conversely, we can estimate the covariance of the principal component features  $\mathbf{p}_i$  from just 6 half-cycle measurements, or 15 ms of data. Using the principal component dimensional reduction, we are thus able to accurately estimate the feature covariance for nearly all ( $> 90\%$ ) of the observed states (appendix A.4). For the  $< 10\%$  of states for which the covariance is not well estimated, we fill in the covariance with the 90th percentile largest (by value of the determinant) well-estimated covariance.

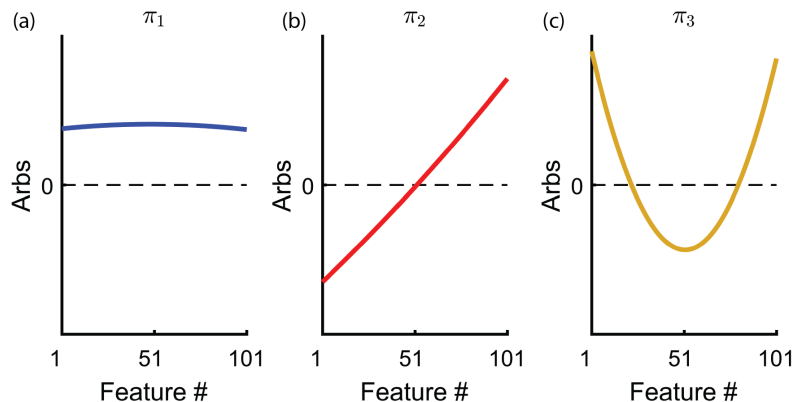


Figure C.3: Principal component vectors for feature extraction. **(a)**, **(b)**, and **(c)** show the first, second, and third principal component vectors for the variable voltage data, respectively. Linear combinations of these three vectors can describe all observed conductance vs. DNA position states. The three vectors roughly represent an offset (a), slope (b), and curvature (c) and thus primarily describe the states as quadratic curves.

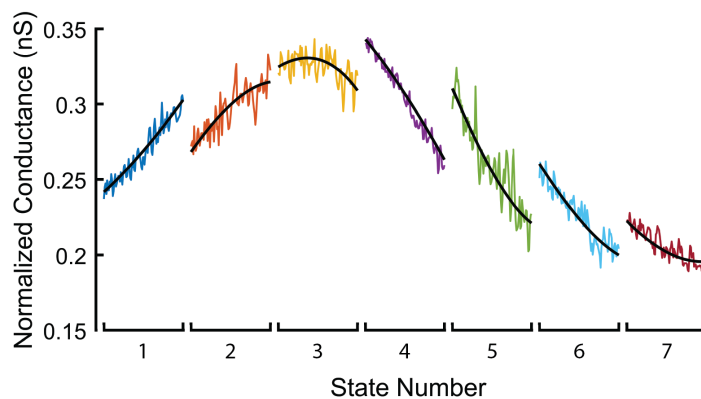


Figure C.4: Principal component description of conductance states. Linear combinations of the 3 principal components (black curves) satisfactorily describe the 101-dimensional conductance states (colored curves). The description preserves the state shape while discarding parameters describing only noise.

### C.4 Elongation of DNA in MspA

We hypothesize that the observed voltage-dependent shift in DNA position relative to MspA is due primarily to the elongation of the section of ssDNA between the enzyme and the pore's constriction in response to the force generated by the applied voltage. To confirm that DNA stretching is the main effect responsible for the position shift and that other effects (i.e. Brownian motion of the enzyme above MspA or deformation within the enzyme or pore under force) are less important, we compare our shift vs. voltage data to the extensible Freely Jointed Chain (ex-FJC) model of ssDNA elongation in response to force. The ex-FJC is an experimentally validated model [71] of ssDNA's elastic response to applied force which predicts the average end-to-end distance of the DNA ( $x$ ) as a function of the force ( $F$ ) applied to one end as

$$x = L_c \left( \coth\left(\frac{Fb}{k_B T}\right) - \frac{k_B T}{Fb} \right) \left( 1 + \frac{F}{S} \right) \quad (\text{C.7})$$

where  $L_c$  is the DNA contour length,  $k_B$  is the Boltzmann constant,  $T$  is the temperature,  $b$  is the Kuhn length of ssDNA, and  $S$  is the stretching modulus of ssDNA.

In the high force regime in which we operate our variable-voltage experiments  $Fb \gg k_B T$ , so the coth term can be well-approximated as identically equal to 1. With this approximation, the force-extension relation simplifies to

$$x = L_c \left( 1 - \frac{k_B T}{Fb} \right) \left( 1 + \frac{F}{S} \right) \quad (\text{C.8})$$

The Kuhn length of ssDNA is known to depend upon salt concentration. From Bosco *et al.* [73], we expect a Kuhn length of around 1.40 nm for the 400 mM KCl conditions in our variable-voltage experiments (section A.1.7). We take a reasonable value of the stretching modulus  $S$  to be 800 pN [73].

Following the analysis in Derrington *et al.* 2015 [53], we observe that in our system the end-to-end extension  $x$  is fixed as the distance between MspA's constriction and the point

where the DNA is anchored within the enzyme. With  $x$  fixed, it is the contour length  $L_c$  that changes with applied force. Assuming that the force on the DNA is proportional to the applied voltage as  $F = \alpha V$  ( $\alpha$  some proportionality constant) gives

$$x = L_c(1 - \frac{k_B T}{\alpha V b})(1 + \frac{\alpha V}{S}) \quad (\text{C.9})$$

Changing the applied voltage from  $V$  to  $\beta V$  will change the contour length of the DNA within the pore from  $L_c$  to  $\omega L_c$ :

$$x = \omega L_c(1 - \frac{k_B T}{\beta \alpha V b})(1 + \frac{\beta \alpha V}{S}) \quad (\text{C.10})$$

Here, the elongation ratio  $\omega$  is the ratio between the contour length of DNA in the pore at the two voltages  $V$  and  $\beta V$ . Solving equations C.9 and C.10 for  $\omega$  gives us a model predicting the elongation ratio  $\omega$  as a function of the voltage ratio  $\beta$  as

$$\omega_{model} = \beta \left[ \frac{(b\alpha V - k_B T)(S + \alpha V)}{(b\beta \alpha V - k_B T)(S + \beta \alpha V)} \right] \quad (\text{C.11})$$

We compare this  $\omega_{model}$  to the measured elongation ratio results ( $\omega_{meas}$ ) as a function of voltage. The measured elongation ratio is calculated from the position shift data as

$$\omega_{meas}(\beta) = \frac{N_{ref} + \delta(\beta)}{N_{ref}} \quad (\text{C.12})$$

where  $\delta$  is the measured position shift from 180 mV (Fig C.5) and  $N_{ref}$  is the number of nucleotides between the last hold point within the enzyme and MspA's constriction at the reference voltage of 180 mV. Position shift is calculated as described in section C.5.

From Bhattacharya *et al.*[102], we estimate  $N_{ref} = 12$  nt. Fitting equation C.11 to our data shows that a single parameter fit with  $\alpha = 1.32 \pm 0.10 \frac{e^-}{nm}$  describes the data well (Fig C.5). Uncertainties here are based on uncertainties in the DNA shift at different voltages and an assumed 0.5 nt uncertainty in  $N_{ref}$ . As the position shift can be well modeled by a

reasonable single parameter model of DNA elongation, we are confident attributing the shift observations to this effect.

The fitted  $\alpha$  parameter corresponds to a force of  $\sim 38$  pN at 180 mV. This force estimate has larger uncertainties than the uncertainty in  $\alpha$  as the estimate is critically dependent on the choices of ex-FJC parameters and ignores secondary effects contributing to the stretching. Potentially relevant secondary effects not accounted for by the ex-FJC model could include effects from the confinement of the ssDNA within the pore's vestibule, voltage dependence of the position of the enzyme relative to MspA, and voltage-induced deformation of the enzyme or the pore.



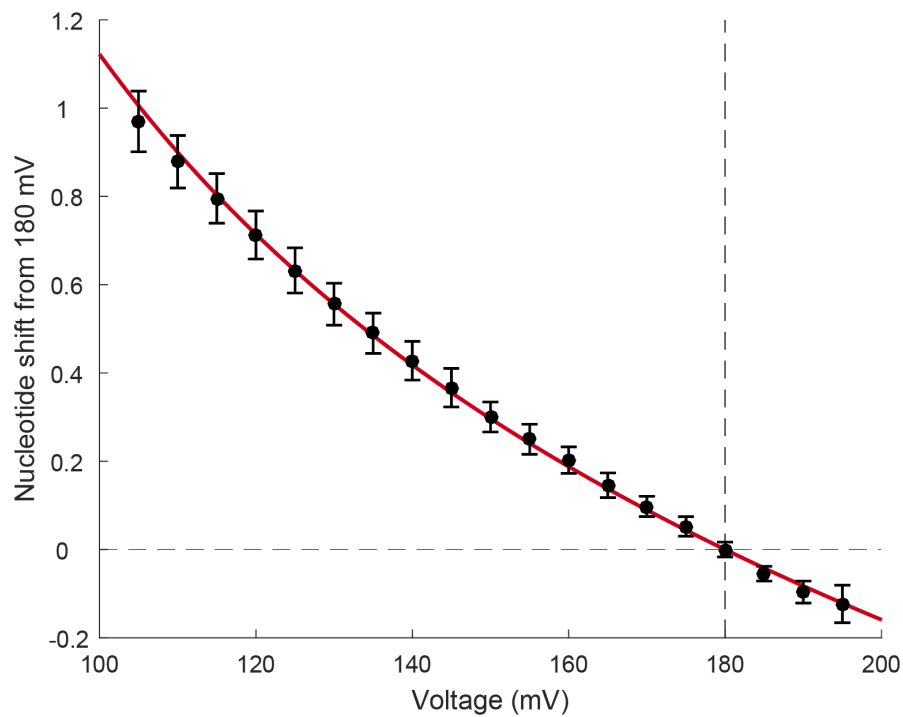


Figure C.5: DNA stretching in MspA. The DNA position shift data is plotted here (black points) as the shift to the position at 180 mV as a function of applied voltage. Error bars show standard error in the position shift measurement. Errors are correlated as subsequent shift measurements are calculated based on the previous shift value (e.g. shift at 140 mV is based on shift at 150 mV). An ex-FJC model is fit to the data with one free parameter  $\alpha$ : the effective charge per length on the DNA. The red curve shows the best fit, resulting from  $\alpha = 1.32 \pm 0.10 \frac{e^-}{nm}$ .

### C.5 Position Shift Calculation

During the voltage-to-position shift calculation, we calculate the position shift between the voltages  $V_1$  and  $V_2$  by finding the shift that best places the conductance profile measurements at both voltages along a single spline. The shift yielding the best single spline placement is calculated as follows.

After first normalizing the conductances (section C.2), we have the conductance profiles for each of the two voltages,  $G_1$  and  $G_2$ . We then calculate cubic spline interpolations ( $spG_1$  and  $spG_2$ ) to both of the transformed current profiles. The two splines are shifted left and right relative to each other in increments of  $\frac{1}{1000}^{th}$  nt. For each shift position  $\phi$ , we calculate a match score  $\mathcal{M}$  by taking the error-weighted sum-square difference between the two splines for the given shift:

$$\mathcal{M}(\phi) = \sum_{i=1}^{N_{pts}} \frac{(spG_1^{(i)} - spG_2^{(i+\phi)})^2}{2(\sigma_{spG_1^{(i)}}^2 + \sigma_{spG_2^{(i+\phi)}}^2)} \quad (\text{C.13})$$

The shift  $\phi_0$  giving the best (smallest) match score gives us the shift that makes the two splines most similar. This  $\phi_0$  is taken as the position shift between  $V_1$  and  $V_2$ .

The match score at a given shift  $\mathcal{M}(\phi)$  is interpretable as the negative log likelihood (up to a constant additive factor) of both spline curves being statistically identical. Thus, the minimum match score corresponds to the highest likelihood of the two curves matching. The uncertainty  $\delta\phi$  was determined by a Monte Carlo analysis of 1200 random perturbations of the data, with  $\delta\phi$  taken as the standard deviation of the  $\phi$  measurements at each voltage.

## Appendix D

### PRE-SEQUENCING STATE FILTERING

#### *D.1 Flicker Filter*

In Hel308-controlled DNA translocation data, we observe short-lived states of a particular character which we refer to as “flickers”. These states are milliseconds or less in duration, always have a lower ionic current than the state they start from, and always return to the state they started in. These flicker states cannot be mapped to any predicted ionic current state when reads are compared against the predicted signal for the known DNA sequence, and are thus not informative in decoding the DNA sequence. We remove these flickers prior to any data processing (including change point detection) as their presence decreases the performance and accuracy of downstream.

In variable-voltage data, flickers are easily identified and removed by the removal filter (section D.2.1) as their conductance curves look starkly different from normal data. To remove these transients from the constant-voltage data, we search for outlying low states of short duration. We score every individual ionic current measurement  $x_n$  in a read with a one-sample t-test against the data surrounding it:

$$t_n = \frac{x_n - \mu_{[n-k, n+k]}}{\sigma_{[n-k, n+k]}}$$

where

$$\mu_{[n-k, n+k]} = \frac{1}{2k} \left[ \left( \sum_{m=n-k}^{n+k} x_m \right) - x_n \right]$$

and

$$\sigma_{[n-k, n+k]}^2 = \frac{1}{2k} \left[ \left( \sum_{m=n-k}^{n+k} x_m^2 \right) - x_n^2 \right] - \mu_{[n-k, n+k]}^2$$

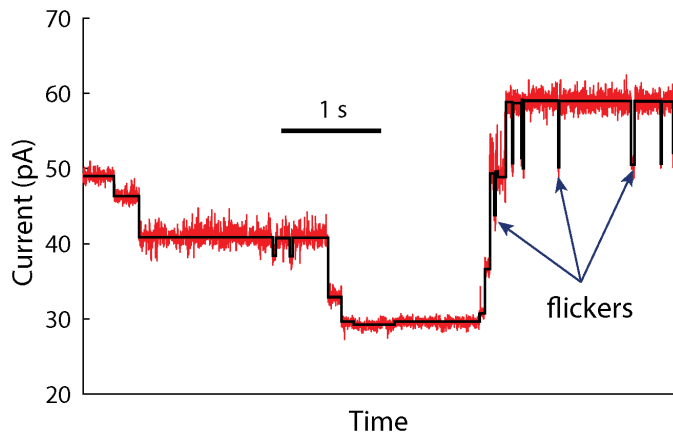


Figure D.1: Example of flicker states. The raw ionic current trace (downsampled to 5 kHz) for a Hel308-controlled DNA translocation event is shown in red, with the states found by the change point detection algorithm overlaid in black. The arrows identify several flickers—transient decreases in the ionic current that are not cannot be mapped to any sequence state. This image is adapted from [55].

are the mean and variance of the data  $k$  points to the left and right of the point being scored, not including the point itself. We discard any points  $|t_n| > e$ , where  $e$  is a threshold chosen to specify the desired aggressiveness of the filter. This procedure is iterated with a fixed threshold  $e$  and window  $k$  until no more points are removed, then repeated a second time with a larger window  $k$ . The the constant-voltage sequencing data in this work, we use  $e = 3$  for both iterations and  $k = 2$  for the first iteration and  $k = 5$  for the second. Filtering is done on the 5 kHz time series data.

## D.2 Variable-Voltage State Filtering

One of the primary advantages of the variable-voltage method is that it allows us to determine the correct ordering of the observed states prior to sequencing. We determine the best ordering of observed states via a three stage “state filtering” process prior to sequencing. The three stages are termed the removal filter (section D.2.1), the recombination filter (section D.2.2), and the reorder filter (section D.2.3). Each stage of state filtering aims to eliminate

a specific error mode common to the data.

### *D.2.1 Removal Filter*

The goal of the removal filter is to find and remove states that are not informative of the DNA sequence moving through the pore. These uninformative “bad” states are common in both constant-voltage and variable-voltage sequencing data and can arise from myriad sources. Common sources of “bad” states include:

1. **Pore Gating:** Protein pores such as MspA are well known to exhibit transient stochastic changes in their conductance, referred to as gating. Gating can occur during DNA translocation, resulting in an abrupt drop in the observed conductance of the observed states for the duration of the gating event. Although DNA translocation continues during the gating event, the conductance states measured in this time period will not match the ionic current-to-sequence model states of the translocating DNA due to the low overall conductance.
2. **Conductance Spikes:** We observe occasional transient spikes up in the conductance through the pore during DNA translocation events. These spikes may be attributable to brief openings of alternative conducting pathways through the bilayer. Regardless of origin, these spike states are not indicative of the translocating DNA sequence, and are not observed at the same DNA sequence position when comparing multiple translocation events of the same DNA sequence.
3. **Flickers:** As discussed in section D.1, we observe short drops in conductance within enzyme states termed “flickers” in both constant- and variable-voltage data. These drops in conductance are distinct from pore gating as they are far shorter-lived and return to the state that preceded them.
4. **Over-called States:** The change point detection algorithm (appendix B) occasionally calls too many transitions, partitioning a single state into multiple. This can be

caused by spontaneous changes in the electronic noise, flickers occurring faster than the variable-voltage cycling frequency, or other transient effects distorting the signal. Frequently, the over-called states exhibit higher noise than the true state. These high-noise over-called states are discarded for sequencing.

The removal filter works by iteratively assigning a “bad state probability”  $P_{bad}$  to each state in the event, then removing those where  $P_{bad}$  exceeds some threshold value for removal  $T_{remove}$ . This process is repeated until no more states are removed (algorithm 5). The process is iterated because  $P_{bad}^i$ , the bad state probability for a given state  $i$ , is a function not only of the state itself, but also of its flanking states  $i + 1$  and  $i - 1$ . So,  $P_{bad}^i$  can change following the first round of removal if either of its flanking states were removed. As there is no state preceding the first state or following the last state,  $P_{bad}$  cannot be evaluated for these two cases. To cope with this, the first and last state are kept as “good” until the final iteration of removal, at which time they are discarded.

---

**Algorithm 5** Removal Filter

---

```

1: assume we start with  $N$  states  $\{\mathbf{x}^i\}_{i \in 1:N}$ 
2:  $stop \leftarrow false$ 
3: while  $\sim stop$  do
4:    $anyremoved \leftarrow false$ 
5:   calculate the SVM feature vectors  $\{\xi^i\}_{i \in 2:N-1}$  from the states  $\{\mathbf{x}^i\}_{i \in 1:N}$ 
6:   calculate the bad state probabilities  $\{P_{bad}^i\}_{i \in 2:N-1}$  from the SVM feature vectors
      $\{\xi^i\}_{i \in 2:N-1}$ 
7:   for  $j \in 2 : N - 1$  do
8:     if  $P_{bad}^j > T_{thresh}$  then
9:       remove  $\mathbf{x}^j$  from  $\{\mathbf{x}^i\}$  ▷ Remove states above the removal threshold
10:       $anyremoved \leftarrow true$  ▷ Stop iterating if nothing is removed
11:    end if
12:  end for
13:  if  $\sim anyremoved$  then
14:     $stop \leftarrow true$ 
15:  end if
16: end while
17: remove  $\mathbf{x}^1$  and  $\mathbf{x}^N$  from  $\{\mathbf{x}^i\}$  ▷ Remove the first and last states

```

---

The  $P_{bad}$  values are calculated as follows. States are first evaluated using a support vector machine (SVM) with a quadratic kernel classifying between “good” states (those to be kept for sequencing) and “bad” states (those to be removed). The SVM takes as input 12-dimensional feature vectors for each state. The composition of the feature vector for state  $i$  is as follows:

*Features 1-3* are the 3 principal component coefficients (section C.3) for the previous state,  $i - 1$ .

*Features 4-6* are the 3 principal component coefficients for the state itself,  $i$ .

*Features 7-9* are the 3 principal component coefficients for the subsequent state,  $i + 1$ .

The first 9 features serve to quantify how continuous or discontinuous the state is with its neighbors. States that are discontinuous with both the previous and subsequent states are more likely to be “bad”.

*Feature 10* is the value of the single conductance measurement in the state’s conductance curve that most deviates from the overall mean conductance in the event. This helps to identify levels with short, extreme deviations from typical conductance values. Such deviations can indicate that a noise spike occurred during the state, likely causing an over-calling during change point detection.

*Feature 11* is the average mean square difference between the state’s 101-dimensional measured conductance curve and its 3-dimension principal component description. This quantifies how well the state is described by the principal components. States poorly described by the principal components are more likely to be “bad”.

*Feature 12* is the score of these state’s best match against the 6-mer model (section E). States that do not have any high scoring match within the 6-mer model are unlikely to represent good measurements of the DNA’s conductance profile and should be labeled “bad”.

To train the SVM, we hand-labeled states taken from the reads used for map building (section E) as either “good” or “bad”. The SVM was trained on a sample of 800 labeled “good” states and 800 labeled “bad” states. We then passed a hold-out validation set

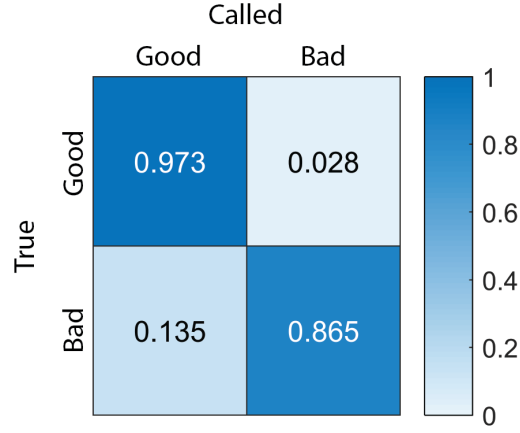


Figure D.2: Removal filter confusion matrix. Entries show the rate at which truly “good” or “bad” states are called as either “good” or “bad” by the SVM.

consisting of 400 labeled “good” and 400 labeled “bad” states to the trained SVM. The validation set showed that the SVM correctly classifies 97.3% of “good” states, 86.5% of “bad” states (Fig D.2), and 91.9% of validation states overall.

To generate the “bad state” probabilities  $P_{bad}$ , we looked at the scores output by the SVM, rather than the labels. The SVM score  $\mathbb{S}$  of a state is the distance of that state’s SVM feature vector from the decision boundary (Fig D.3a). This score serves as a proxy for how good (negative scores) or bad (positive scores) a state is. We want to assign higher  $P_{bad}$  to states with higher scores. We do this by plotting the true state labels (0 for good, 1 for bad) as a function of the state scores  $\mathbb{S}$  (Fig D.3b). These data are then fit by the logit function

$$f(\mathbb{S}|\alpha, \beta) = \frac{1}{1 - e^{-(\alpha\mathbb{S}+\beta)}} \quad (\text{D.1})$$

using a global likelihood maximization fit (Fig D.3).

Together, the SVM and the fit logit function give us a way to calculate  $P_{bad}^i$  for any state  $i$ . First, the 12 features are evaluated for this state. Then, the SVM is used to score the feature vector relative to the decision boundary, yielding a score  $\mathbb{S}^i$ . Finally, we evaluate



$f(\mathbb{S}^i|\alpha, \beta)$ , yielding  $P_{bad}^i$  for the state.

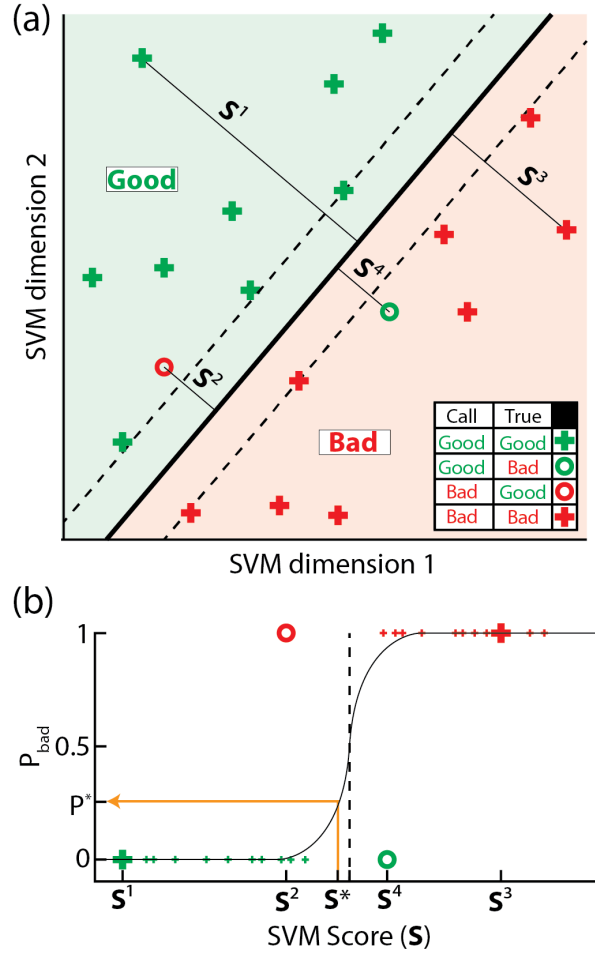


Figure D.3: Converting SVM outputs to  $P_{bad}$  probabilities. **(a)** In this classifier (contrived data), points occupying the space (shaded green) above the decision boundary (solid black line) are classified as good while those below (shaded red) are classified as bad. Points marked with a “plus” are classified correctly, while “circles” are classified incorrectly. Green markers denote truly good states and red markers denote truly bad states. Each point  $i$  has an associated score  $S^i$ , which is its distance from the decision boundary. **(b)** Each state in the validation set is plotted by its good (0) or bad (1) label as a function of its assigned SVM score  $S$ . The dashed black vertical line is at  $S = 0$ , representing points lying exactly on the decision boundary. The solid black curve shows the logit function fit to the validation states using a global likelihood maximization procedure. The SVM scores  $S$  are converted into probabilities that the state is bad using a logit function. During removal filtering, an unknown state is assigned a score  $S^*$  by the SVM. This score is then converted into a probability it is bad ( $P_{bad}^*$ ) using the logit function (orange arrow).

### D.2.2 Recombination Filter

The goal of the recombination filter is to find instances where multiple observed states represent repeated measurements of the same DNA position. Repeated state measurements can arise from two potential sources. First, over-called transitions during change point detection result in consecutive states representing the same DNA position. If these over-called states are not removed by the removal filter, they show up in this stage as “holds”. The second source of duplicate states is enzyme missteps in which the enzyme moves backwards (in the 3’ direction) along the DNA. These “back steps” result in non-consecutive duplicate states.

The recombination filter works by aligning an event against itself (self-alignment). Repeated states will match to their duplicates within the event nearly as well as they match to themselves. We conduct a Needleman–Wunsch-style alignment of the states  $\{\mathbf{x}^i\}$  with themselves,  $\mathbb{A}(\{\mathbf{x}^i\}, \{\mathbf{x}^i\})$  (algorithm 6). In this alignment, alignment of a state  $i$  to itself  $\mathbb{A}(\mathbf{x}^i, \mathbf{x}^i)$  can be thought of as establishing  $\mathbf{x}^i$  as a unique, previously unobserved state. Conversely, alignment of a state  $i$  to a different (previous) state  $j$ ,  $\mathbb{A}(\mathbf{x}^i, \mathbf{x}^j), i \neq j$  means that states  $i$  and  $j$  are repeated measurements of the same DNA position and should be recombined into a single state.

A state  $i$  will always match best with itself. However, we bias the alignment against aligning states to themselves by applying a “self-alignment penalty”  $P_{SA}$  to such cells in the alignment matrix (Fig D.4). Statistically, the self-alignment penalty is a penalty for adding parameters (states) to our model of the observed event and the self-alignment penalty is thus taken  $\frac{1}{2}$  the number of added parameters (3, for the 3 principal component coefficients characterizing each state (appendix C.3)).

With these considerations, we conduct a Needleman-Wunsch-style alignment of the measured states against themselves with the following modifications. First, to reduce the computational load and avoid recombining distant states that may look similar but too far apart to represent a likely duplication, we limit ourselves to a fixed lookback distance  $L$ , where we

only consider matches for state  $i$  within states  $i - L$  to  $i - 1$ .

Secondly, we assign unique step-type probabilities at each transition based on the conductance curve overlap information (section 3.7). At each transition between two states  $m$  and  $n$ , we calculate the relative probabilities that the transition between the two occurred via a single half-nucleotide step ( $P_S^{mn}$ ), skip ( $P_K^{mn}$ ), backstep ( $P_B^{mn}$ ), or hold ( $P_H^{mn}$ ). To calculate these probabilities, we use an ensemble of 3 SVMs (quadratic kernel),  $\mathbf{S}_{SK}$ ,  $\mathbf{S}_{SB}$ , and  $\mathbf{S}_{SH}$ . These three SVMs are all trained on labeled transitions generated from the  $\Phi$ X-174 data used to build the 6-mer model (appendix E.3.1). In the same manner as was described above for the good/bad SVM classifier (section D.2.1), we first trained these classifiers to determine their decision boundary, then conducted a global likelihood maximization fit to tune a logit function (characterized by two parameters,  $\alpha$  and  $\beta$ ) to their output scores on a held-out validation set. This fit logit function allows us to convert the output scores from the SVMs (distances from the decision boundary) into probabilities. All three SVMs take as input a 6-dimensional feature vector composed of the 3 principal component coefficients of state  $m$  and from state  $n$ .

$\mathbf{S}_{SK}$  differentiates between steps and skips (88.7% correct on the validation set),  $\mathbf{S}_{SB}$  differentiates between steps and backsteps (98.2% correct on the validation set), and  $\mathbf{S}_{SH}$  differentiates between steps and holds (95.4% correct on the validation set). The scores of these SVMs ( $\mathbb{S}_{SX}$ ,  $X$  one of  $K$ ,  $B$ ,  $H$ ), converted to probabilities through their associated logit functions, give us relative likelihoods between the different step types. The relative likelihoods of a step vs. a skip between states  $m$  and  $n$  is given by

$$\frac{P_S^{mn}}{P_K^{mn}} = \frac{\text{logit}(\mathbb{S}_{SK}, \alpha_{SK}, \beta_{SK})}{1 - \text{logit}(\mathbb{S}_{SK}, \alpha_{SK}, \beta_{SK})}$$

with similar relations for step vs. back and step vs. hold. These three relations, along with the overall normalization condition that

$$P_S^{mn} + P_B^{mn} + P_H^{mn} + P_K^{mn} = 1$$

give us a system of four equations for the four unknowns, allowing us to solve for the various step type probabilities. Skips longer than two half-steps and backsteps longer than a single half-step backwards are treated as independent processes, with their probability given as the product of the correct number of  $P_K$ 's or  $P_B$ 's. For example, the probability of a backstep of 3 half-steps  $P_{B3}$  is given as

$$P_{B3} = P_B^3$$

In the language of affine probabilities, the extension probability is set to be equal to the basic probability,

$$P_{B+} = P_B$$

We can enter a previously unmeasured (new) state through one of three transitions: a step, a skip, or a backstep. Consequently, our alignment matrix has dimensions  $N \times (L + 3)$  where  $N$  is the number of measured states. The columns 1 :  $L$  represent alignment of a state to the state  $L : 1$  states before it. The final 3 columns represent the creation of a new state via alignment of the state to itself, entered into via a step, skip, or backstep, respectively.

The final modification made in our self-alignment method is the above-discussed assessment of an additional self-alignment penalty  $P_{SA} = -\frac{3}{2}$  to these newly created states. The full matrix of transition penalties (penalties taken as log probabilities,  $S = \log(P_S)$ , etc.) is summarized in Fig D.5. Using this self-alignment method to identify repeated states, we conduct recombination filtering as described in algorithm 6.

---

**Algorithm 6** Recombination filter

---

```

1: Input: start with  $N$  observed states  $\{\mathbf{x}^i\}$ ,  $i \in 1 : N$   $\triangleright$  States are passed in after
   removal filter
2: function STEPPOBS( $\{\mathbf{x}^i\}$ )  $\triangleright$  Function to calculate the transition-by-transition
   step-type probabilities
3:   Calculate Get the scores  $\mathbb{S}_{SK}$ ,  $\mathbb{S}_{SB}$ , and  $\mathbb{S}_{SH}$  from the SVMs  $\mathbf{S}_{SK}$ ,  $\mathbf{S}_{SB}$ , and  $\mathbf{S}_{SH}$ 
4:   Calculate Convert SVM scores into relative likelihoods using the attached logit
   functions
5:   Solve Use the resulting system of 4 equations to find  $P_S$ ,  $P_B$ ,  $P_H$ , and  $P_K$  for each
   transition
6:   Output Transitions matrix  $\mathcal{T}$  contains the step-type probabilities for each transition
7: end function
8: function SELFALIGN( $\{\mathbf{x}^i\}$ )
9:    $\mathcal{T} \leftarrow \text{STEPPOBS}(\{\mathbf{x}^i\})$ 
10:   $P_{SA} \leftarrow -\frac{3}{2}$ 
11:  Calculate Alignment  $\mathcal{A}$  is the alignment of  $\{\mathbf{x}^i\}$  to  $\{\mathbf{x}^i\}$  subject to the self-alignment
   penalty  $P_{SA}$  and the transition penalties  $\mathcal{T}$   $\triangleright \mathcal{A}$  is a  $1 \times N$  array, where  $\mathcal{A}_i = j$  means
   that the  $i^{th}$  measured state is the  $j^{th}$  recombined state
12:  Output:  $\mathcal{A}$ 
13: end function
14: Initialize:
    $changed \leftarrow TRUE$ 
    $\{\mathbf{x}_{new}^i\} \leftarrow \{\mathbf{x}^i\}$ 
15: while  $changed$  do
16:    $\{\mathbf{x}_{old}^i\} \leftarrow \{\mathbf{x}_{new}^i\}$   $\triangleright$  Store the existing  $\{\mathbf{x}_{new}^i\}$  in a new variable
17:    $\mathcal{A} \leftarrow \text{SELFALIGN}(\{\mathbf{x}_{old}^i\})$   $\triangleright$  Conduct self-alignment
18:   if  $\max(\mathcal{A}) = \text{length}(\{\mathbf{x}_{old}^i\})$  then  $\triangleright$  We have the same number of recombined states
   as initial states, meaning nothing has been recombined
19:      $\{\mathbf{x}_{new}^i\} \leftarrow \{\mathbf{x}_{old}^i\}$   $\triangleright$  no states have changed so just pass on the old ones
20:      $changed \leftarrow FALSE$   $\triangleright$  Our recombination has converged, exit the while loop
21:   else
22:     Initialize:  $\{\mathbf{x}_{new}^i\}$  as a empty holder of size  $1 \times \max(\mathcal{A})$   $\triangleright$  Storage for the set of
     recombined states
23:     for  $i \in 1 : \max(\mathcal{A})$  do  $\triangleright$  Loop over old states and recombine into new states
     based on alignment
24:        $\mathbf{x}_{new}^i \leftarrow \text{mean}(\{\mathbf{x}_{old}^{\{j\}}\})$  where  $\{j\}$  is such that  $\mathcal{A}^j = i$  for all  $j \in \{j\}$ 
25:     end for
26:      $changed \leftarrow TRUE$   $\triangleright$  As long as things have changed, continue recombination
27:   end if
28: end while
29: Output  $\{\mathbf{x}_{new}^i\}$   $\triangleright$  Final output is the new set of recombined states

```

---

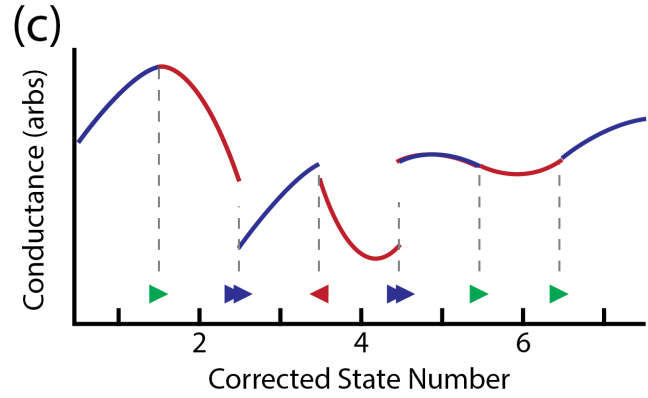
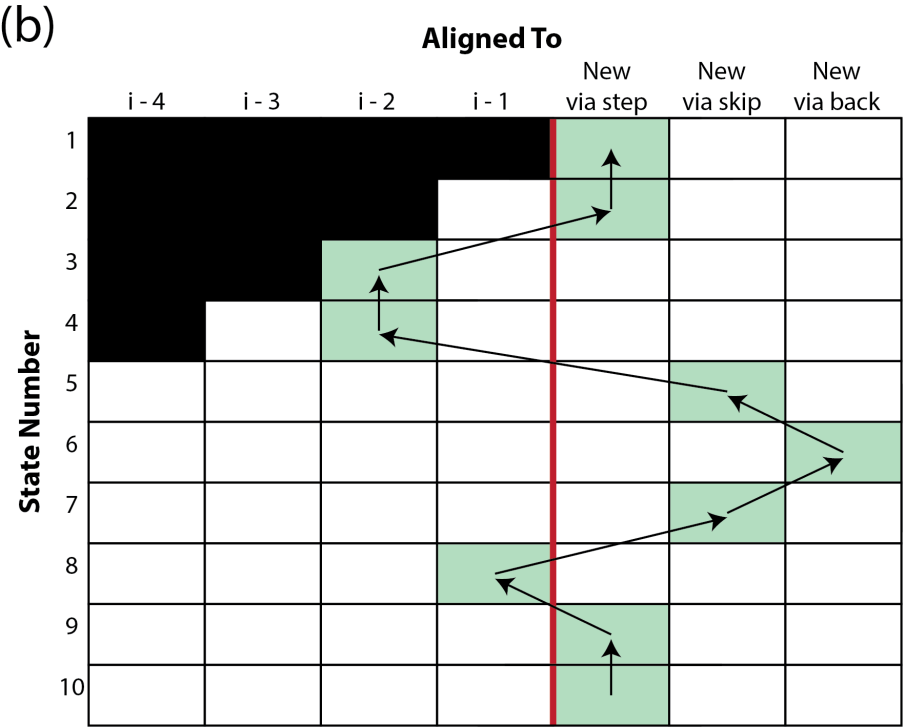
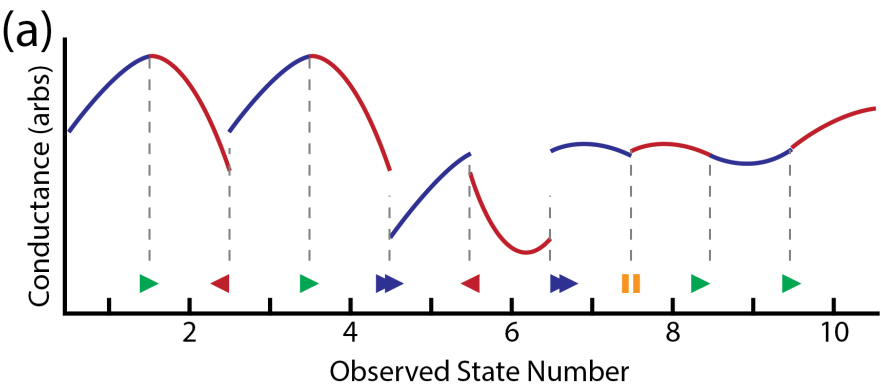


Figure D.4: Self-alignment procedure for recombination filter. **(a)** The recombination filter seeks to find repeated instances of the same  $k$ -mer conductance state in a sequencing read. Shown here is a toy example of a sequencing read with various missteps. Toy data is modeled on a single-nucleotide-stepping enzyme for simplicity. **(b)** The self-alignment of the above states to themselves reveals repeated  $k$ -mer states. States 1, 2, 5, 6, 7, 9, and 10 are unique states. States 3 (= state 1), 4 (= state 2), and 8 (= state 7) are repeated measurements of previously observed states. **(c)** Recombining the repeated measurements into single states dramatically reduces the errors in the signal. The remaining misordered states (3 and 4 should be swapped) will be treated by the reordering filter.

		To						
		$i-4$	$i-3$	$i-2$	$i-1$	new via step	new via skip	new via back
From	$i-4$	S	K	$K+K_+$	$K+2K_+$	$S+P_{SA}$	$K+P_{SA}$	--
	$i-3$	H	S	K	$K+K_+$	$S+P_{SA}$	$K+P_{SA}$	--
	$i-2$	B	H	S	K	$S+P_{SA}$	$K+P_{SA}$	--
	$i-1$	$B+B_+$	B	H	S	$S+P_{SA}$	$K+P_{SA}$	--
	new via step	$B+2B_+$	$B+B_+$	B	H	$S+P_{SA}$	$K+P_{SA}$	--
	new via skip	$B+2B_+$	$B+B_+$	B	H	$S+P_{SA}$	$K+P_{SA}$	$B+P_{SA}$
	new via back	$B+2B_+$	$B+B_+$	B	H	--	$K+P_{SA}$	--

Figure D.5: Self-alignment transition penalties. During self-alignment, the transition from a starting point in the alignment (rows) into a final point in the alignment matrix (columns) takes an additive penalty. Alignments of a state to a previously measured state take a penalty equal to the log probability of the enzyme step required to generate the states in that order. Alignments of a state to itself take a step-type penalty as well as the self-alignment penalty  $P_{SA}$ . Certain transitions (marked --) are not allowed.



### D.2.3 Reordering Filter

Some enzyme misstep errors can persist in the signal even after removal and recombination filtering. Particularly, complex error modes involving successive enzyme missteps (e.g. a skip, then backstep, then skip as in Fig D.4c) can result in out-of-order states even after bad states are removed and duplicate states are recombined. The reordering filter—the last of the three filters involved in the state filtering process—aims to identify and correct these out-of-order states prior to sequencing.

The reordering filter works by using by using an ensemble of SVMs with associated logit functions (as in the recombination filter, section D.2.2) to assign a probability that each transition was a single step (“*S*”), a skip (“*K*”), or a backstep (“*B*”). A dynamic programming algorithm is then used to find the most likely set of allowed transitions linking the observed states.

The calculation of step-type probabilities for the reordering filter uses the same SVMs and logits as the recombination filter. The only change here is we are no longer looking for holds (holds by definition result in duplicate states, and so should be entirely treated by the recombination filter) so we only use two of the three SVMs:  $\mathbf{S}_{SK}$  to decide between steps and skips, and  $\mathbf{S}_{SB}$  to decide between steps and backsteps. Using the same procedure as in the recombination filter, we use these two SVMs to calculate the probability that each state-to-state transition represents a step, skip, or backstep. In our notation, the  $n^{th}$  transition, from state  $n$  to state  $n + 1$  has a step probability  $P_S^n$ , skip probability  $P_K^n$ , and backstep probability  $P_B^n$ . For a read of  $N$  states, the step-type probabilities are summarized in the  $N - 1 \times 3$  matrix  $\mathcal{P}$ :

$$\mathcal{P} = \begin{bmatrix} P_S^1 & P_K^1 & P_B^1 \\ P_S^2 & P_K^2 & P_B^2 \\ \dots & \dots & \dots \\ P_S^{N-1} & P_K^{N-1} & P_B^{N-1} \end{bmatrix}$$

For convenience, we convert these probabilities to log-probabilities (denoted  $\mathbb{P}$ ) for further

use:

$$\mathbb{P} = \log(\mathcal{P})$$

Now with the step-type log-probabilities calculated, we use a dynamic programming algorithm (algorithm 7) to find the most likely path of transitions through the states, subject to certain constraints. Namely, we must choose a set of transitions reflective of a state ordering not requiring any repeated visits to the same state. For example, we cannot choose to take a step from state 1 to 2, then a backstep from 2 to 3. This hypothetical path would imply that state 3 is a repeated measurement of state 1. If this were the case, these states would have been recombined during the previous filtering step. As they were not recombined, this transition pathway must be ruled out, and is not allowed during reordering. To implement this “no repeated states” condition, we consider 4 “transition states”: steps ( $S$ ), backsteps ( $B$ ), skips where the previous transition was a step or a skip ( $K|SK$ ), and skips where the previous transition was a backstep ( $K|B$ ). The allowed linkages between these transition states are summarized as an allowed linkage matrix  $\mathbb{L}$ :

$$\mathbb{L} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

where a linkage from transition state  $i$  to transition state  $j$  is allowed if  $\mathbb{L}_{ij} = 1$  and is not allowed if  $\mathbb{L}_{ij} = 0$ . The 1st row and column in  $\mathbb{L}$  represents the step “transition state”  $S$ , the 2nd represents  $B$ , the 3rd represents  $K|SK$ , and the 4th represents  $K|B$ . So, for example  $\mathbb{L}_{1,2} = 0$  tells us that we cannot jump from a step into a backstep (as discussed above). Subject to these allowed transitions, we compute an alignment matrix  $\mathbb{A}$  and traceback matrix  $\mathbb{B}$  that provide us the most likely pathway through the allowed transitions. This pathway tells us what type of step was most likely taken at each transition. With this step-type information, we can optimally reorder the observed states to finally reconstruct the

most likely sequence order, completing the filtering process.

---

**Algorithm 7** Reordering filter

---

```

1: Input:
    $\mathbb{P}$                                  $\triangleright$  step-type log-probabilities for each of the  $N - 1$  transitions
    $\mathbb{L}$                                  $\triangleright$  matrix of allowed “transition type” linkages
2: Initialize:
    $\mathbb{A} \leftarrow \text{ones}(N - 1, 4)$   $\triangleright$  alignment matrix, 1st column is  $S$ , 2nd is  $B$ , 3rd is  $K|SK$ , 4th is  $K|B$ 
    $\mathbb{B} \leftarrow \text{ones}(N - 1, 4)$   $\triangleright$  traceback matrix
3:  $\mathbb{A}_{1,1:3} \leftarrow [\mathbb{P}_{1,1}, \mathbb{P}_{1,2}, \mathbb{P}_{1,3}, -\infty]$   $\triangleright$  Fill first row of alignment matrix
4: for  $i \in 2 : (N - 1)$  do  $\triangleright$  Loop over the rest of the transitions, filling the alignment and
   traceback matrices
5:    $A \leftarrow [\mathbb{P}_{i,1}, \mathbb{P}_{i,3}, \mathbb{A}_{i,2}, \mathbb{A}_{i,2}]$   $\triangleright$  Begin filling next row in alignment matrix
6:    $T \leftarrow [\mathbb{A}_{i-1,1}, \mathbb{A}_{i-1,2}, \mathbb{A}_{i-1,3}, \mathbb{A}_{i-1,4}]$   $\triangleright$   $T$  stores the scores of possible cells we can
   transition in from
7:   for  $j \in 1 : 4$  do  $\triangleright$  Loop over the 4 transition types
8:      $t \leftarrow T$   $\triangleright$  make a copy of  $t$  as we fill this particular cell
9:      $t_k \leftarrow -\infty$  for  $\forall k$  where  $\mathbb{L}_{k,j} = 0$   $\triangleright$  turn off disallowed transitions
10:     $t_* \leftarrow \max(t)$   $\triangleright$  take the best scoring path in
11:     $b_* \leftarrow k$  such that  $t_k = t_*$   $\triangleright$  record which transition had the best score
12:     $\mathbb{A}_{i,j} \leftarrow A_j + t_*$   $\triangleright$  fill alignment matrix cell
13:     $\mathbb{B}_{i,j} \leftarrow b_*$   $\triangleright$  fill traceback matrix cell
14:   end for
15: end for
16: Initialize  $\mathbb{R} \leftarrow []$   $\triangleright$  initialize storage for the best path through the alignment matrix as
   we conduct traceback
17:  $\mathbb{R} \leftarrow [r\mathbb{R}]$  where  $r$  is such that  $\mathbb{A}_{n-1,r} = \max(\mathbb{A}_{n-1,:})$   $\triangleright$  start traceback at best scoring
   cell in bottom row of  $\mathbb{A}$ 
18: for  $doi \in (n - 1) : -1 : 2$   $\triangleright$  conduct traceback over most likely pathway
19:    $r \leftarrow \mathbb{B}_{i,r}$   $\triangleright$   $\mathbb{B}$  tells us where we came from to get to the max cell in  $\mathbb{A}$ 
20:    $\mathbb{R} \leftarrow [r\mathbb{R}]$   $\triangleright$  append the location of the best score in the row to the start of the
   traceback
21: end for
22: Output:  $r$  contains the type of step ( $1 = S$ ,  $2 = B$ ,  $3$  and  $4 = K$ ) taken at each
   transition

```

---

## Appendix E

### CONSTRUCTING THE 6-MER MODEL

#### *E.1 General Considerations*

To sequence our variable-voltage reads, we determine the DNA sequence most likely to have generated the observed series of ionic current states. In order to decode this DNA sequence, we require a map relating the ionic current signal and the DNA sequence in the pore. We model the nanopore signal as being generated by the  $k$  nucleotides (i.e. the  $k$ -mer, with  $k$  an integer) nearest the pore's constriction. This model is described by a map of the  $4^k$  possible  $k$ -mers to the ion ionic currents typically observed when they are in the pore. The  $k$ -mer model has been previously validated as an effective model for nanopore signal prediction [58].

Our lab's previous work on constant-voltage nanopore DNA sequencing used a model with  $k = 4$ , but we found that a 4-mer model was insufficient for our variable-voltage signal. During variable-voltage sequencing, the nucleotides centered within the nanopore constriction at each enzyme registration are shifted forwards and backwards as the DNA is stretched by the changing voltage. This shifting means that the nucleotides both 5' and 3' of the central 4-mer have more of an effect on the observed signal when using variable-voltage than they had in the constant-voltage case. In order to better model this effect, we expanded our model from 4-mers to 6-mers, now including an additional nucleotide on both the 5' and 3' ends of the previous 4-mers, expanding our model from  $4^4 = 256$  4-mers to  $4^6 = 4096$  6-mers.

Each state in the variable-voltage model is characterized by more complex information than in the constant-voltage model. In the constant-voltage model, each  $k$ -mer state was characterized by its mean conductance value ( $G$ ), its typical conductance noise ( $dG$ ), and the variance of both of these quantities. In contrast, during variable-voltage sequencing, the

$k$ -mer state occupied at each enzyme step is not a constant conductance value characterized by a mean and noise, but instead a conductance vs. voltage ( $G - V$ ) curve. We found that each variable-voltage  $k$ -mer state is well characterized by its 3 principle component amplitudes  $\mathbf{p}$  and their covariance  $\Sigma^{\mathbf{p}}$  (appendix C.3).

In previous work, we used the  $\Phi 29$  DNAP as the motor protein controlling the DNA, which steps in full nucleotide increments. Our new method instead uses the Hel308 DNA helicase as the motor protein, which takes two distinct steps per nucleotide, an ATP-dependent step and an ATP-independent step [53]. As our signal now contains two distinct enzyme states per nucleotide, each single  $k$ -mer is now associated with two distinct states, and the 4096 6-mers in our model represent 8192 total states, two for each  $k$ -mer.

## ***E.2 Initial Model***

To construct the variable-voltage 6-mer model for the two-step-per-nucleotide Hel308 helicase motor protein, we refined the existing constant-voltage 4-mer model. We note that the 6-mer denoted  $N_1N_2N_3N_4N_5N_6$  (where  $N_i$  denotes a nucleotide  $A$ ,  $C$ ,  $G$ , or  $T$ ) is made up of 3 distinct 4-mers:  $N_1N_2N_3N_4$ ,  $N_2N_3N_4N_5$ , and  $N_3N_4N_5N_6$ . Additionally, we recall that the variable-voltage signal samples the translocating DNA's conductance as a function of position. This function should interpolate smoothly between the discrete samples taken at constant voltage. We approximate the smooth conductance vs. position curve interpolating the DNA positions between the three constituent 4-mers by a quadratic fit to the three conductances known from the  $\phi 29$  DNAP 4-mer model (Fig E.1).

Sampling this curve at the appropriate DNA positions gives us an estimate of the variable-voltage 6-mer states. The first of Hel308 helicase's two states is known to have the same DNA registration within the pore as  $\Phi 29$  DNAP [53]. The constant-voltage model was derived from experiments run at a bias of  $180mV$ , and we know from the DNA force-extension curve (appendix C.4) that the variable-voltage sweep stretches the DNA  $+0.1nt$  from the  $180mV$  position at its highest voltage and  $-0.9nt$  from the  $180mV$  position at its lowest voltage. So, to predict the state 1 conductance vs. position curve, we sample the interpolating curve

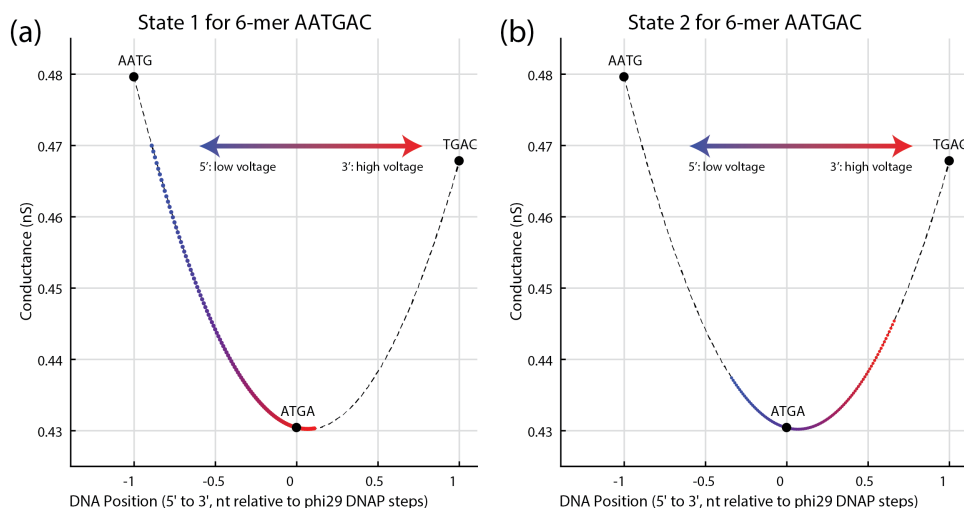


Figure E.1: Method of generating variable-voltage Hel308 helicase 6-mer predictions from the constant-voltage phi29 DNAP 4-mer model. Black points show the constant-voltage 4-mer model predictions for the 3 4-mers comprising the 6-mer of interest. Black dashed line shows the quadratic fit to the 4-mer model predictions, which acts as an estimate of the smooth conductance vs. position profile explored by variable-voltage. The blue to red points show the predicted conductance as a function of DNA position for the given 6-mer. Blue-er points correspond to lower voltages, red-er points to higher voltages. **(a)** Prediction for the first of the two Hel308 states. The first of the two Hel308 states has the same DNA registration within the pore as the phi29 DNAP full-nucleotide steps. The conductance value of the Hel308 state 1 prediction coincides with the phi29 DNAP conductance prediction for the central 4-mer in the 6-mer of interest. **(b)** Prediction for the second of the two Hel308 states. The second state is shifted  $0.55nt$  to the 3' of the first state.

at equally-spaced points from  $-0.9nt$  to  $+0.10nt$  (Fig E.1a). The second of Hel308's two states is  $0.55nt$  3' from state 1. So, state 2 is predicted by sampling the interpolating curve between  $-0.35nt$  and  $0.65nt$  (Fig SE.1b).

The amplitudes of the 3 principle components  $\mathbf{p}$  for each 6-mer state can now be calculated from the predicted  $G - V$  curve (appendix C.3). All 8192 states in the initial guess map were assigned the same default covariance for their 3 principle components. The 3 principle components are sufficient for this initial model to provide a framework on which to build an empirical 6-mer model based on measurements of DNA under variable-voltage conditions.

### ***E.3 Measuring Genomic DNA of Known Sequence***

To build the 6-mer model we will ultimately use for DNA sequencing, we measure the signal produced by all 4096 of the 6-mers during variable-voltage experiments. We read DNA of known sequence under the variable-voltage sequencing conditions (appendix A.1.7), then use the measured signals of this known DNA to update the initial 6-mer model.

#### *E.3.1 $\Phi$ X174*

We first measured the 5386 bp  $\Phi$ X174 genome (New England Biolabs). We prepared the circular genome for variable-voltage nanopore sequencing experiments as follows:

1. The circular genome was linearized via a double digest using the restriction enzymes PstI and AvaII (New England Biolabs).  $\Phi$ X174 was prepared in 20  $\mu$ g batches. For each batch, a mixture of
  - (a) 40  $\mu$ L of  $\Phi$ X174 DNA at 500  $\frac{ng}{\mu L}$
  - (b) 5  $\mu$ L of 10x CutSmart (New England Biolabs) Buffer
  - (c) 1  $\mu$ L of PstI-HF restriction enzyme at 100  $\frac{Units}{\mu L}$
  - (d) 1  $\mu$ L of AvaII restriction enzyme at 10  $\frac{Units}{\mu L}$
  - (e) 3  $\mu$ L of molecular biology grade water

was incubated at 37°C for 60 minutes, then heat inactivated via heating to 80°C for 20 minutes. Each of PstI and AvaII have a single cut site in  $\Phi$ X174, so the double digest yields two linear fragments, one of 5042 bp, the other of 344 bp (Fig E.2a, b).

2. The linearized fragments were purified from the heat-inactivated restriction enzymes on a DNA Clean and Concentrator column (Zymo Research), and eluted into 50  $\mu$ L of molecular biology grade water.



3. Two DNA adapters are attached to each of the fragments enabling reading by the nanopore (Fig E.2c, d). At one end, we ligate a threading adapter, which promotes capture a single strand into the pore, entering with the 5' end of the DNA threading into the pore. This threading adapter also features a cholesterol tagged 3' end. The cholesterol tagged 3' end inserts into the lipid bilayer, localizing the DNA strands near the pore and increasing the rate of 5' end capture. The threading adapter is made up of two partially complementary strands: the  $\Phi X174$  threading strand and the  $\Phi X174$  cholesterol blocker (Table A.2). The threading adapter is formed at high concentration by mixing equal volumes of  $12.5\ \mu M$  threading strand and cholesterol blocker, then annealing to yield a  $12.5\ \mu M$  solution of the fully formed threading adapters.
4. At the other end, we ligate a loading adapter which promotes loading of the Hel308 helicase onto the DNA construct. This adapter consists of two partially complementary strands: the  $\Phi X174$  loading strand and the  $\Phi X174$  loading blocker (Table A.2). The loading adapter is formed at high concentration by mixing equal volumes  $12.5\ \mu M$  loading strand and loading blocker, then annealing to yield a  $12.5\ \mu M$  solution of the fully formed loading adapters.
5. The threading and loading adapters are ligated to the sticky ends of the linearized  $\Phi X174$  DNA fragments. A mixture of
  - (a)  $48\ \mu L$  of  $100\ nM$   $\Phi X174$  DNA
  - (b)  $6\ \mu L$  of 10x T4 ligase buffer (New England Biolabs)
  - (c)  $2\ \mu L$  of  $12.5\ \mu M$  threading adapters (to give 5:1 ratio of adapters to target sticky ends)
  - (d)  $2\ \mu L$  of  $12.5\ \mu M$  loading adpaters (to give 5:1 ratio of adapters to target sticky ends)
  - (e)  $1\ \mu L$  of molecular biology grade water

(f) 1  $\mu L$  of T4 DNA ligase at 400  $\frac{Units}{\mu L}$  (New England Biolabs)

was incubated at 16°C for 60 minutes, then heat inactivated by heating to 65°C for 10 minutes.

6. The fully formed DNA constructs (Fig E.2e) were purified from the remaining unligated adapters and the heat-inactivated ligase on a DNA Clean and Concentrator column, and eluted into 50  $\mu L$  of molecular biology grade water.

These two fragments, now with the necessary adapters attached, were run using the standard variable-voltage nanopore sequencing conditions. In total, we observed 155 individual reads comprising 188543 enzyme steps, or 94272 nucleotides.

### *E.3.2 $\lambda$ Phage*

In order to get better coverage of numerous 6-mers not present in the  $\Phi X174$  genome, and to increase the context diversity of all of our measurements, we next decided to measure a larger genome. For this second round of measurements, we chose the 48502 bp  $\lambda$  bacteriophage genome. We chose a new approach to fragmentation for this experiment in order to provide uniform read coverage over the entire genome. Due to the limited processivity of our Hel308 helicase ( $\sim 1000$  nt, appendix A.3), restriction enzyme fragmentation results in most reads starting at the restriction site, but terminating prior to reading the entire fragment. Consequently, such a fragmentation gives excellent read coverage near the restriction sites, but poor coverage further away from them.

For uniform coverage, we instead use two separate Covaris products giving random shearing over the entire genome into fragments of a well-defined size range. In one  $\lambda$  library preparation, we used the Covaris Blue DNA miniTUBE, which yielded random fragments of on average 3 kbp in length. For our second library preparation, we used Covaris gTUBEs to get random fragments of on average 6 kbp in length. We switched from miniTUBEs to gTUBEs

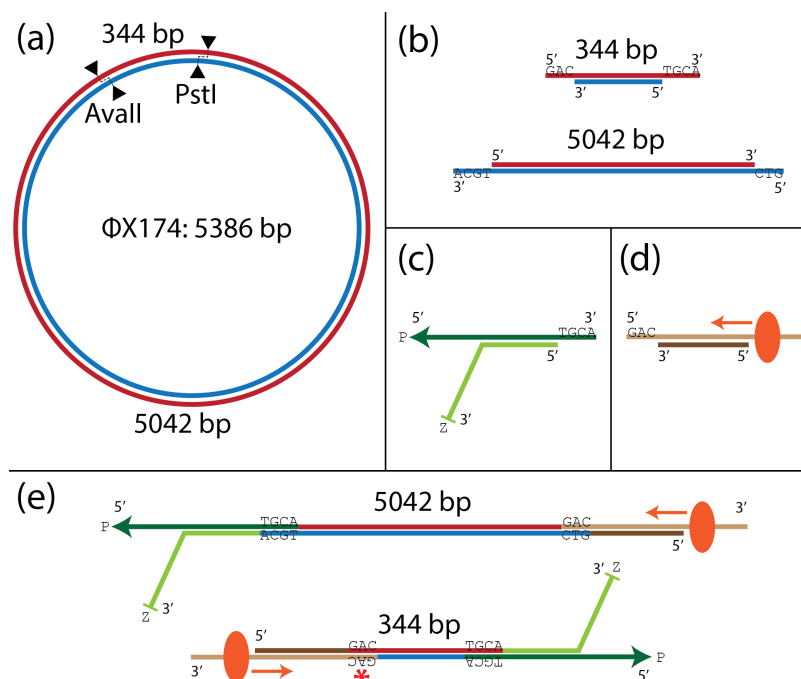


Figure E.2: Preparation of  $\Phi$ X174 for variable-voltage sequencing. **(a)** The circular 5386 bp genome is cut twice using the AvaII and PstI restriction enzymes. **(b)** Restriction results in two fragments of 344 and 5042 bp, with sticky ends of size 3 and 4 nt. **(c)** The threading adapter consists of a threading strand (dark green) featuring a 5' phosphate (P, arrowhead) which promotes capture of this end by the pore, and a cholesterol blocker strand (light green), featuring a 3' cholesterol tag (Z, crossbar) which associates with the lipid bilayer to concentrate the DNA constructs near the pore and increase the capture rate. **(d)** The loading adapter consists of a loading strand (tan) which overhangs the blocking strand (brown) at the 3' end to provide a loading site for the Hel308 helicase (orange ellipse). The helicase loads at the ss-dsDNA junction and proceeds to walk in a 3' to 5' direction along the loading strand (orange arrow). **(e)** After adapter ligation, the DNA constructs are now ready to be run in the variable-voltage sequencing experiments. Our adapters are designed such that we will read the sense (red) strand of the long fragment, and the antisense (blue) strand for the short fragment. The red asterisk marks a sticky end mismatch at the loading end of the short fragment, a byproduct of the non-palindromic AvaII cut site. Despite this mismatch, we still observe a population of reads of this smaller fragment, indicating that the loading adapter did still attach with some efficiency.

simply for easy of use, as these required only a centrifuge, and not the Covaris sonicator instrument. For both shearing methods, the library preparation proceeded as follows:

1. The full length  $\lambda$  DNA (Promega) was fragmented using either Blue miniTUBEs (in 20  $\mu g$  batches) or gTUBEs (in 30  $\mu g$  batches) (Fig E.3a, b).

For miniTUBE fragmentation, 20  $\mu g$  of  $\lambda$  DNA was suspended in Tris EDTA buffered at pH 8.0 to a total volume of 200  $\mu L$ . DNA was then fragmented using the Covaris M220 Focused-ultrasonicator, using the recommended settings for product fragments of  $\sim 3000$  bp in length.

For gTUBE fragmentation, 30  $\mu g$  of  $\lambda$  DNA was suspended in molecular biology grade water to a total volume of 150  $\mu L$ . The gTUBE was then centrifuged on an Eppendorf 5417R centrifuge for 30 seconds at 12400 rpm (corresponding to 16200 g), resulting in fragments of  $\sim 6000$  bp in length.

2. Following fragmentation, the DNA fragments have random 3' and 5' overhangs. Before proceeding with adapter ligation, we ensure that all DNA fragments are blunt-ended by running an end repair protocol (Fig E.3c). Using the NEBNext end repair module (New England Biolabs), a mixture of

- (a) 5  $\mu g$  fragmented DNA
- (b) 10  $\mu L$  of NEBNext 10x End Repair Reaction Buffer
- (c) 5  $\mu L$  of NEBNext End Repair Enzyme Mix
- (d) Molecular biology grade water to total volume of 100  $\mu L$

was incubated at 20°C for 30 minutes. The end-repaired fragments (now blunt-ended) were purified on a DNA Clean and Concentrate column.

3. After end repair, we used the NEBNext dA-tailing module (New England Biolabs) to attach a dA monomer at the 3' end of each strand as a target for adapter ligation (Fig

E.3d). A mixture of

- (a) 5  $\mu g$  of  $\lambda$  DNA
- (b) 5  $\mu L$  of 10x NEBNext dA-Tailing Reaction Buffer
- (c) 3  $\mu L$  of Klenow Fragment ( $3' \rightarrow 5'$   $\text{exo}^-$ )
- (d) Molecular biology grade water to a total reaction volume of 50  $\mu L$

was incubated at  $37^\circ C$  for 30 minutes, then purified on a DNA Clean and Concentrate column.

4. Similar threading and loading adapters are used for variable-voltage sequencing experiments on the  $\lambda$  DNA as were used for  $\Phi X174$ , differing only in the sequence at the sticky ends to be ligated onto the genomic DNA fragments. For the threading adapter (Fig E.3e), equimolar parts of the  $\lambda$  threading strand and the  $\lambda$  cholesterol blocker (Table A.2) were mixed and annealed to a final concentration of 10  $\mu M$ . Similarly, for the loading adapter (Fig E.3f), equal molar parts of the  $\lambda$  loading strand and the  $\lambda$  loading blocker were mixed and annealed to a final concentration of 10  $\mu M$ .
5. Adapters were ligated to the dA-tailed  $\lambda$  DNA fragments using T4 DNA ligase (New England Biolabs). A mixture of
  - (a) 10  $\mu g$  of  $\lambda$  DNA fragments
  - (b) 3  $\mu L$  of 10  $\mu M$  threading adapters (for a  $\sim 10:1$  adapter to dA-tail end ratio)
  - (c) 3  $\mu L$  of 10  $\mu M$  loading adapters (for a  $\sim 10:1$  adapter to dA-tail end ratio)
  - (d) 15  $\mu L$  of 10X Ligation Buffer
  - (e) 7.5  $\mu L$  T4 DNA Ligase
  - (f) Molecular biology grade water up to a total reaction volume of 150  $\mu L$

was incubated at  $22^{\circ}\text{C}$  for 125 minutes, then heat inactivated at  $65^{\circ}\text{C}$  for 10 minutes. The ligation products (Fig E.3g) were purified on DNA Clean and Concentrator columns to remove the inactive ligase and residual un-ligated adapters, and eluted into molecular biology grade water.

As all the 3' ends of the  $\lambda$  fragments have the same single dA overhang, not all ligation products will have the correct conformation of one threading adapter and one loading adapter. 25% of the population will have loading adapters at each end, and 25% will have threading adapters at each end. This reduces the overall effective yield of this library preparation by half, but a sufficient number of constructs were well formed to allow us to generate 128 individual reads comprising 120867 enzyme steps, or 60434 nucleotides.

#### ***E.4 Building the Empirical 6-mer Model from Genomic Reads***

Having measured a total of 309410 enzyme steps along genomic DNA tracks (120867 in  $\lambda$ , 188543 in  $\Phi\text{X174}$ ) representing 154705 measured nucleotides, we now organize these measurements to empirically update the initial model of the predicted nanopore signal for each of the 8192 model states (2 enzyme states for each of the 4096 6-mers). Each observed enzyme step is a measurement of one of the two Hel38 helicase states at one of the 4096 possible 6-mers.

To update the model, we must associate the signal at each enzyme step with the sequence that generated it. We get this association by aligning the measured signal to the predicted signal for the known DNA sequence being measured ( $\Phi\text{X174}$  or  $\lambda$ ). For the first construction of the empirical model, the predicted signal is given by the initial model described in section E.2.

Each read of genomic DNA is aligned to the predicted signal for its reference sequence using the BCJR alignment algorithm [103]. The alignment maps the  $G - V$  curve at each measured ionic current state to a state in the predicted signal, which represents a known location in the reference sequence. In addition to an alignment location, the BCJR algorithm

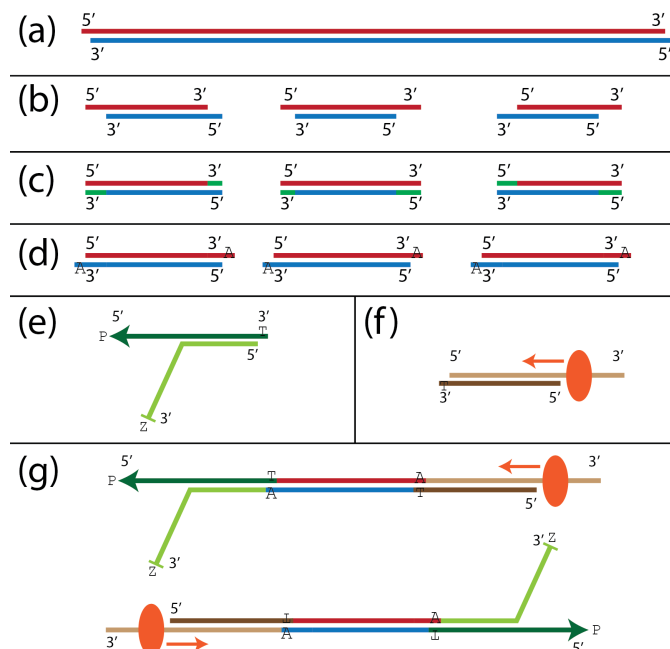


Figure E.3:  $\lambda$  DNA Fragmentation. **(a)** Full length double stranded  $\lambda$  DNA, 48502 bp. **(b)** The genomic DNA is sheared into random fragments of average length 3 kb (miniTUBE) or 6 kb (gTUBE). **(c)** The random 3' and 5' overhangs generated through the shearing are repaired (green segments) using the NEBNext end repair module. **(d)** A single base dA overhang is added to each 3' end using the NEBNext dA-tailing module. **(e)** The threading adapter is composed of two strands. The threading strand (dark green) has a 5' phosphate (P, arrowhead) to facilitate capture by the pore and a single base dT 3' overhang for ligation onto the  $\lambda$  fragment. The cholesterol blocker (light green) is partially complementary to the threading strand, with a non-complementary 3' end, and a terminal 3' cholesterol (Z, crossbar) which inserts into the lipid bilayer to up-concentrate DNA near the pore. **(f)** The loading adapter is also composed of two strands. The loading strand (tan) has an overhanging 3' end where the Hel308 helicase (orange ellipse) can load onto a ss-dsDNA junction, and proceed in a 3' to 5' direction along the strand. The loading blocker (brown) is complementary to the non-overhanging region of the loading strand, with a single base dT 3' overhang for ligation onto the  $\lambda$  fragment. **(g)** Our fully formed DNA constructs are now ready to be run in variable-voltage nanopore sequencing experiments. This library preparation can obtain reads of both the sense (red, top construct) and antisense (blue, bottom construct) strand.

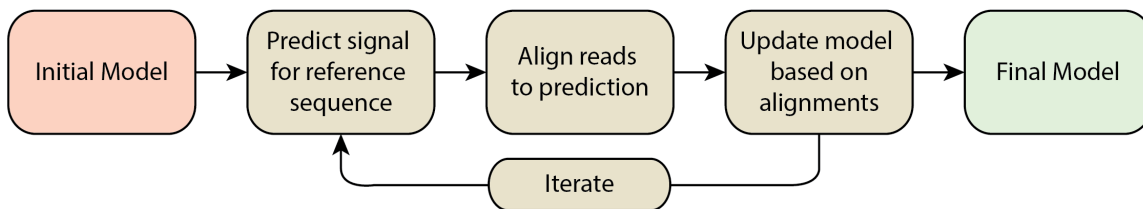


Figure E.4: Iterative map construction flow chart

also returns a likelihood that each alignment location is the true alignment location for the measured state. We update the mean values in the 6-mer model by filling each state in the model with the weighted average (weighted by the likelihood score of alignment) of all measured states aligning to locations in the reference corresponding to that enzyme and 6-mer state. Additionally, the covariance of each state in the model is updated with the covariance of all measured states aligning to reference locations corresponding to that state.

The above procedure of generating predictions, aligning reads, and updating the predictions can be iterated (Fig E.4). For the work presented here, we ran two iterations: one starting from the interpolated initial model and second aligning to the first version of the empirical model. Though we found that two iterations yielded a good quality model, it is possible that a larger data set of genomic DNA reads combined with further iterations of the model generation could result in an improved model.

### ***E.5 Filling in Unmeasured 6-mers***

After constructing the empirical 6-mer model from long reads of  $\Phi X174$  and  $\lambda$  DNA, we found that for a small fraction (168 out of 4096) of the 6-mers, one or both of the enzyme states had not been well measured. In order to efficiently measure the remaining states, we used a De Bruijn graph approach [34] to construct a minimal sequence of length 337 nt containing all 168 of the poorly measured 6-mers. We then split up this minimal sequence over a total of 6 short synthetic DNA oligos (Fill-in strands 1-6 in Table A.2, Fig E.5). We



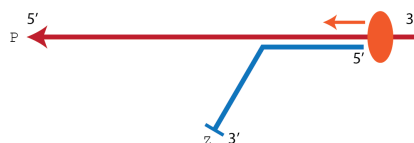


Figure E.5: DNA constructs for the fill-in data set are composed of two partially-complementary short oligos. The fill-in template (one of a possible 6) in red has a 5' terminal phosphate (P, arrowhead) facilitating threading of this end into the pore. The 5' phosphate is followed by a section of the minimal sequence containing the missing 6-mers, then by a target sequence for duplexing the complementary strand, and finally an overhanging non-complementary 8 nucleotide section at the 3' end for loading the Hel308 helicase (orange oval). The fill-in cholesterol blocker (blue) contains at its 5' end the complementary sequence to the target duplex sequence in the template strand. The complementary sequence is followed by a series of 4 18-Carbon spacers and a 3' terminal cholesterol tag (Z, crossbar). The 18-Carbon spacers give the construct a flexible fan-tail, and the terminal cholesterol tag associates with the lipid bilayer, up-concentrating the constructs near the pore.

ran these 6 strands using standard variable-voltage sequencing conditions, collecting a total of 172 reads comprising 16675 enzyme steps (8338 nucleotides) across the 6 strands. Using these reads, we filled in the remaining gaps in the empirical 6-mer model using the same approach detailed in section SE.4: we predicted the signal for the known sequence based on the existing 6-mer model, aligned the reads to this prediction, then updated the model based on the alignments and iterated the process. We iterated the predict/align/update cycle 10 times for the short strands in order to generate the final version of the 6-mer model which we ultimately use for sequencing.

## E.6 Constant-Voltage Model Extraction

The variable-voltage 6-mer map contains as a subset all the information required for a constant-voltage 6-mer model. In evaluating the performance of the two sequencing methods, we used the constant-voltage 6-mer model extracted from the variable-voltage model to provide a fair test. By doing this, any errors present in one model detracting from the sequencing accuracy will be present in both models, and will affect the accuracy of both

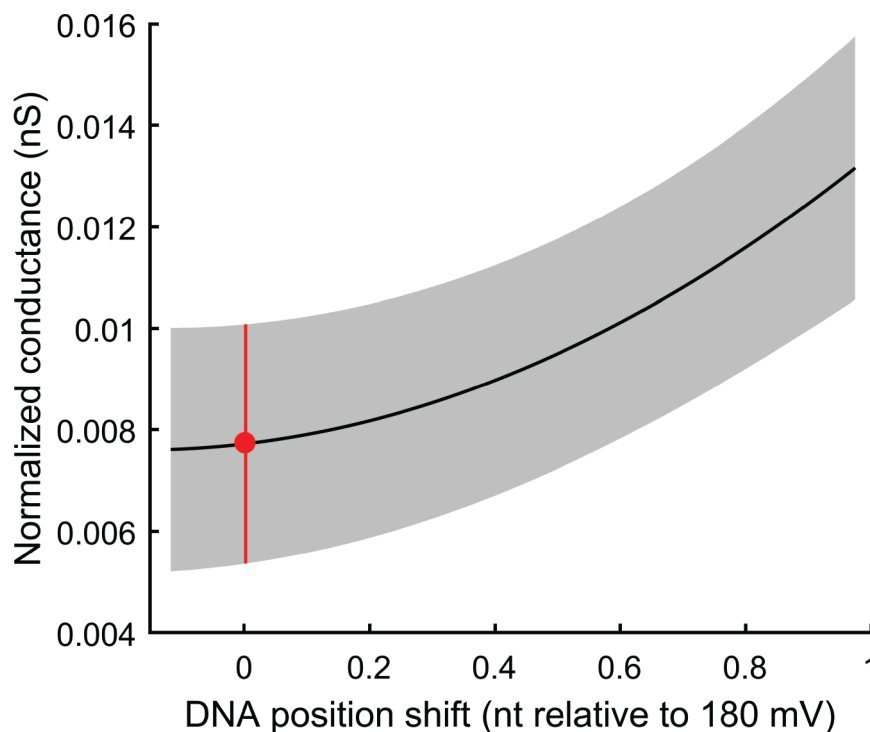


Figure E.6: Constant voltage model extraction. The constant-voltage model value for a given 6-mer (e.g. *ATGAGA*) is taken as the point (red) in the variable-voltage conductance curve for that 6-mer in the variable-voltage model (black) corresponding to 180 mV (0 nt shifted relative to 180 mV). The uncertainty (red line) is taken as the variation in the variable-voltage model prediction (gray shading) at the 180 mV point.

methods equally.

The constant-voltage model is extracted from the variable-voltage model as shown in Fig E.6. The constant-voltage mean conductance for each 6-mer is extracted from the corresponding variable-voltage conductance curve by taking the value of the curve at the point corresponding to 180 mV (the operating voltage for our constant-voltage sequencing experiments). The variance of the mean constant-voltage conductance is taken as the variance in the variable-voltage conductance curve's value at that same point.

## Appendix F

### SEQUENCING ALGORITHM

DNA sequencing is performed using a hidden Markov model (HMM) solver as described below. Simply, we decode the series of  $k$ -mers most likely to have generated the observed series of conductance states by conducting an alignment between the observed states and the 6-mer model states. In standard sequence-to-sequence (or conductance-to-conductance) alignment, the alignment proceeds from left-to-right in both sequences. In contrast, this sequencing alignment proceeds left to right in the measured states, but jumps around in the model states based on the allowed  $k$ -mer transitions. For example, *AAAAAT* (the 4th  $k$ -mer) can transition to *AAAATG* (the 15th  $k$ -mer) via a single nucleotide step, but requires a 6 nucleotide jump to reach the 5th  $k$ -mer, *AAAACA*. A simple implementation of this HMM-solving algorithm applied to nanopore sequencing data is formalized in algorithm 1 for the case where only single-nucleotide steps are allowed. Our full adaptation and calculation of the measured-to-model alignment is described in detail below.

#### ***F.1 Match Scores***

We first compute an score matrix  $S$  of match likelihoods  $S_{nj}$  between measured state  $n$  and reference 6-mer model state  $j$ . The measured state and model state are each characterized by their  $d$ <sup>1</sup> principal component amplitudes and the associated uncertainty (for measured states) or covariance (for model states) covariance matrix. The measured state is written as  $\mathbf{x}_n$  with uncertainty matrix  $\Sigma_{\mathbf{x}_n}$ , and the reference state is written as  $\mathbf{y}_j$  with covariance

---

<sup>1</sup>For our purposes here,  $d = 3$  (section C.3).

matrix  $\Sigma_{\mathbf{y}_j}$ . The match score between these states is given by

$$S_{nj} = \frac{1}{\sqrt{(2\pi)^d \frac{|\Sigma_{\mathbf{x}_n}^{-1}| |\Sigma_{\mathbf{y}_j}^{-1}|}{|\Sigma_{\mathbf{x}_n}^{-1} + \Sigma_{\mathbf{y}_j}^{-1}|}}} \exp \left[ -\frac{1}{2} \left( \Sigma_{\mathbf{x}_n}^{-1} \mathbf{x}_n - \Sigma_{\mathbf{y}_j}^{-1} \mathbf{y}_j \right)^T \left( \Sigma_{\mathbf{x}_n}^{-1} + \Sigma_{\mathbf{y}_j}^{-1} \right) \left( \Sigma_{\mathbf{x}_n}^{-1} \mathbf{x}_n - \Sigma_{\mathbf{y}_j}^{-1} \mathbf{y}_j \right) \right] \quad (\text{F.1})$$

The corresponding array of log-likelihoods is the natural logarithm of this,

$$s_{nj} = \log S_{nj} = \frac{1}{2} \left[ d \log 2\pi + \log |\Sigma_{\mathbf{x}_n}^{-1}| + \log |\Sigma_{\mathbf{y}_j}^{-1}| - \log |\Sigma_{\mathbf{x}_n}^{-1} + \Sigma_{\mathbf{y}_j}^{-1}| \right. \\ \left. - \left( \Sigma_{\mathbf{x}_n}^{-1} \mathbf{x}_n - \Sigma_{\mathbf{y}_j}^{-1} \mathbf{y}_j \right)^T \left( \Sigma_{\mathbf{x}_n}^{-1} + \Sigma_{\mathbf{y}_j}^{-1} \right) \left( \Sigma_{\mathbf{x}_n}^{-1} \mathbf{x}_n - \Sigma_{\mathbf{y}_j}^{-1} \mathbf{y}_j \right) \right]. \quad (\text{F.2})$$

## F.2 Hel308 Backstep Kinetics

We use the known backstep kinetics of the Hel308 enzyme to inform our sequencing. Specifically, previous work [55] found that Hel308 is far more likely to backstep when in its ATP-independent state (the “pre” states in our 6-mer model) than when in its ATP-dependent state (the “post” states in our 6-mer model). Consequently, measured states determined to have backstepped during enzyme step correction (appendix D.2.2) are more likely to have been generated with ATP-independent states in our 6-mer model. To use this information, we label which measured states backstepped, then incorporate the independent/dependent state probabilities into the score matrix  $S$  as follows.

We estimate the probability that a state that backstepped was an ATP-independent state  $P_{ind|b} = 0.975$  from Hel308 kinetics data [55]. The overall probability that a state will backstep is also estimated from Hel308 kinetics data as  $P_b = 0.025$ . From these, we can calculate the probability  $P_{ind|\sim b}$  that an ATP-independent state will *not* backstep as

$$P_{ind|\sim b} = \frac{\frac{1}{2} - P_b * P_{ind|b}}{P_{ind|\sim b}}$$

The probability that a given state is ATP-dependent given that it did ( $P_{dep|b}$ ) or did not

( $P_{dep|\sim b}$ ) backstep is simply 1 minus the complementary independent probability:

$$P_{dep|b} = 1 - P_{ind|b}$$

$$P_{dep|\sim b} = 1 - P_{ind|\sim b}$$

We incorporate these probabilities into the score matrix  $S$  by first converting them to log-probabilities:  $p = \log(P)$ . The odd-numbered columns in the score matrix ( $S_{ij}$  where  $j$  is odd) represent matches to ATP-independent states and the even-numbered columns ( $S_{ij}$  where  $j$  is even) are matches to ATP-dependent states. For every measured state  $i$  where we observed a backstep, we update the row  $S_i$  as

$$S_{ij} \leftarrow S_{ij} + p_{ind|b} \text{ if } j \text{ is odd}$$

and

$$S_{ij} \leftarrow S_{ij} + p_{dep|b} \text{ if } j \text{ is even}$$

Likewise, for every measured state  $i$  where we did not observe a backstep, we update the row  $S_i$  as

$$S_{ij} \leftarrow S_{ij} + p_{ind|\sim b} \text{ if } j \text{ is odd}$$

and

$$S_{ij} \leftarrow S_{ij} + p_{dep|\sim b} \text{ if } j \text{ is even}$$

This accounting tells our sequencer to preferentially call states for which a backstep was observed as ATP-independent states.

### ***F.3 Transition Probabilities***

We also determine transition probabilities between each pair of states. In the case of constant-voltage sequencing, the relative probabilities of different transitions (step, skip1, skip2, ...) between any two given states are fixed for all states. In variable-voltage sequencing, we can

use the overlap between two states' conductance curves in order to get a more informed estimate of the relative probabilities. Two states whose conductance curves overlap well are likely to be separated by a single step, whereas states whose conductance curves do not overlap are more likely to be skips. We find that we can differentiate effectively between steps and non-steps (88.9% correct calls on the labeled validation set), as well as between single skips and larger skips (79.1% correct calls on the validation set).

We use an ensemble of SVMs (similar to those described in appendix D.2) to assign each transition its own set of probabilities of being a step, skip1, or a larger skip. The SVMs take as input the principal components of the two measured states  $m$  and  $n$  to assign probabilities to the different types of transitions between  $m$  and  $n$ . The ensemble of SVMs is made up of two classifiers ( $\mathbf{S}_1$  and  $\mathbf{S}_2$ ), trained on labeled examples of steps and variously sized skips from the  $\Phi$ X-174 data collected during the 6-mer model construction (appendix E.3.1). The scores  $\mathbb{S}_i$  output by the SVMs  $\mathbf{S}_i$  are converted into probabilities using the same logit procedure as described previously in appendix D.2.

$\mathbf{S}_1$  differentiates between steps and non-steps, and assigns the probability that the transition from state  $m$  to state  $n$  was a single step as

$$P_{mn}^{(1)} = \text{logit}(\mathbb{S}_1, \alpha_1, \beta_1)$$

with the logit function as defined previously and using logit parameters  $\alpha_1$  and  $\beta_1$  determined from global likelihood maximization over a labeled validation set (appendix D.2).

Similarly,  $\mathbf{S}_2$  differentiates between single skips (involving two half-nucleotide steps) and larger skips (involving more than two half-nucleotide steps).  $\mathbf{S}_2$  gives us the probability that the transition between states  $m$  and  $n$  was a single skip given that it was not a step:

$$P_{mn}^{(2|\sim 1)} = \text{logit}(\mathbb{S}_2, \alpha_2, \beta_2)$$

The overall probability then that the transition between  $m$  and  $n$  was a single skip is then

$$P_{mn}^{(2)} = P_{mn}^{(2|\sim 1)} * P_{mn}^{(\sim 1)} = \text{logit}(\mathbb{S}_2, \alpha_2, \beta_2) * (1 - \text{logit}(\mathbb{S}_1, \alpha_1, \beta_1))$$

We set the probabilities of larger skips by an affine probability  $P_{mn}^{(+)}$  such that

$$P_{mn}^{(k)} = P_{mn}^{(k-1)} * P_{mn}^{(+)}$$

$P_{mn}^{(+)}$  is set so that the summed probability of all possible steps and skips sums to 1.

#### F.4 Transition Matrix

For each pair of measured states we wish to consider, we compute an 8192 x 8192 transition matrix  $T$  composed of the probabilities of transitioning between map states:

$$T_{mn,ij} = P\left(\begin{array}{c} \text{state } m \text{ is a measurement} \\ \text{of true map state } i \end{array} \mid \begin{array}{c} \text{state } n \text{ is a measurement} \\ \text{of true map state } j \end{array}\right) \quad (\text{F.3})$$

To calculate the transition matrix, we first find a matrix whose elements are the probabilities of having transitioned between states conditioned on a step size of a single half-step,

$$\tau_{ij1} = P\left(\begin{array}{c} \text{state } t \text{ is a measurement} \\ \text{of true map state } i \end{array} \mid \begin{array}{c} \text{state } t+1 \text{ is a measurement} \\ \text{of true map state } j \end{array}, \text{ step size} = 1\right)$$

$$= \begin{cases} 1 & i \text{ is a "pre" state and } j \text{ the corresponding "post" state} \\ 1/4 & i \text{ is a "post" state and } j \text{ a succeeding 6-mer} \\ 0 & \text{otherwise} \end{cases}$$

where we define two 6-mers as “successive” when they share 5 nucleotides shifted by one position, e.g. ACGTAC could be succeeded by CGTACT. We then define a similar matrix for larger sizes of step, which is calculated by taking powers of the single half-step matrix:

$$\tau_{ijk} = P(\text{state } t \text{ is a measurement of true map state } i \mid \text{state } t+1 \text{ is a measurement of true map state } j, \text{ step size} = k) = (\tau_{ij1})^k.$$

Finally, we define  $\tau_{ij(12)}$  to correspond to all transitions with step size greater than or equal to 12, which could be between any two states. Therefore it has uniform entries  $\tau_{ij(12)} = 1/8192$ . Now, we can compute the total transition probability matrix as the sum of the probabilities of each possible step size by which the measured levels could have advanced:

$$T_{mn,ij} = P_{mn}^{(1)}\tau_{ij1} + \sum_{k=2}^{12} P_{mn}^{(2)} (P_{mn}^{(+)})^{k-2} \tau_{ijk}. \quad (\text{F.4})$$

We also define the log transition likelihood,  $t_{mn,ij} = \log T_{mn,ij}$ .

### F.5 Markov Model

If we are sequencing a read of  $N$  measured states, we create an  $N \times 8192$  alignment matrix,  $\mathbb{A}$ . In each element of the array  $\mathbb{A}_{nj}$  we write an estimate of the log-likelihood that measured state  $n$  came from map state  $j$ , given the observation of measured states 1 through  $n-1$ :

$$\mathbb{A}_{1j} = s_{1j} + \log \left( 1 - P_1^{(\text{bad})} \right)$$

$$\mathbb{A}_{nj} = \log \sum_{k=1}^{8192} \sum_{m=1}^{n-1} \exp \left\{ s_{nj} + t_{mn,kj} + h_{mk} + \log \left( 1 - P_n^{(\text{bad})} \right) + \sum_{l=m+1}^{n-1} \log P_l^{(\text{bad})} \right\}, \quad n > 1,$$

where  $P_n^{(\text{bad})}$  is the probability that observed state  $n$  is an erroneous measurement that should be omitted from the sequencer. In constant-voltage sequencing,  $P^{(\text{bad})}$  is taken as a constant value for all states. In variable-voltage sequencing, we use the same bad state classifier as in the removal filter (appendix D.2.1) to assign a unique  $P^{(\text{bad})}$  to each state.

This is a forwards-propagating approximation of a MAP algorithm, which in practice gives similar results to a slower forwards-backwards algorithm relying on all observations to



determine likelihoods [104] [103]. We take two additional steps to increase speed. Firstly, using the approximation

$$\log \sum_i e^{a_i} \approx \arg \max_i a_i,$$

which is valid when one  $a_i$  is significantly larger than the others. We replace the logarithms of sums of exponentials in our alignment matrix  $\mathbb{A}$  with maxima, which are more expedient to calculate:

$$\mathbb{A}_{nj} = \max_{k,m} \left\{ s_{nj} + t_{mn,kj} + h_{mk} + \log(1 - P_n^{(\text{bad})}) + \sum_{l=m+1}^{n-1} \log P_l^{(\text{bad})} \right\}, \quad n > 1,$$

We also record a traceback array,

$$\mathbb{B}_{nj} = \arg \max_{k,m} \left\{ s_{nj} + t_{mn,kj} + h_{mk} + \log(1 - P_n^{(\text{bad})}) + \sum_{l=m+1}^{n-1} \log P_l^{(\text{bad})} \right\}, \quad n > 1.$$

Thus  $\mathbb{B}_{nj} = (m, k)$ , such that  $\mathbb{A}_{mk}$  is the maximum likelihood observed state-map state matching to have occurred just prior to the one described by likelihood  $\mathbb{A}_{nj}$ . This is a Viterbi algorithm [104], approximating the results of the MAP algorithm [103].

Additionally, we improve speed by restricting the max over  $m$  to only cases where  $m > n - q - 1$ , where  $q$  is the maximum number of sequential “bad” observed states allowed by the algorithm. We found good results taking  $q = 3$ , as cases of more than 3 consecutive “bad” states not removed by the removal filter (appendix D.2.1) are exceedingly rare. We also restrict the max over  $k$  to values of  $k$  such that  $s_{nk} > \max_j s_{nj} - c$ , where  $c$  is a score difference cut-off. Similarly,  $\mathbb{A}_{nk}$  with  $k$  subject to the same restrictions is left uncalculated, because it will not be used by the algorithm under any circumstances. This avoids spending time calculating the probability flow into and out of states unlikely to represent the optimal

alignment. Using  $c = 10$  provides identical results to the full calculation in all tested cases, while dramatically reducing the computational load.

Calculation of  $\mathbb{A}_{nj}$  and  $\mathbb{B}_{nj}$  requires knowledge of  $\mathbb{A}_{(n-1)k}$  for all  $k$ , so the array is calculated starting with the  $n = 1$  elements and proceeding upwards in  $n$ .

### ***F.6 Traceback and Sequence Construction***

Once  $\mathbb{B}$  has been calculated, we find sequence of map states with the maximum approximate-likelihood of having produced the observed states. We do this by starting at the maximum approximate-likelihood entry in alignment matrix, at  $\mathbb{A}_{n^*j^*}$ , and iteratively following the traceback array through the most likely sequence of transitions. In other words, if  $a$  is the sequence of indices of true map states and  $n$  is the sequence of indices of valid observed states,

$$(n_{\text{final}}, a_{\text{final}}) = \arg \max_{(n, j)} \mathbb{A}_{nj},$$

$$(n_i, a_i) = \mathbb{B}_{n_{i+1}, a_{i+1}}.$$

From  $a$  we calculate the most likely DNA sequence. Between  $a_i$  and  $a_{i+1}$ , we find the most likely (the smallest) step size that could transition between those two 6-mers, and fill in bases accordingly. For example, GTACAC (pre) could transition to ACACTT (pre) with four half-nucleotide steps, moving the GT outside of the pore's constriction and the TT into it. It could also make the transition by taking eight half-nucleotide steps, moving the GTAC outside and the ACTT in, or by taking twelve half-nucleotide steps, moving the entire sequence GTACAC out of the constriction and the entire sequence of ACACTT in. The four-nucleotide step is the most likely based on our empirical model of transition probabilities. Therefore, if these two 6-mers were  $a_i$  and  $a_{i+1}$ , they would be sequenced as GTACACTT, because that is more likely than the alternative choices GTACACACTT or GTACACACACTT. By performing this step for every state in  $a$ , we arrive at a close-to-optimal-likelihood sequence for the observed states.

## Appendix G

### SUPPLEMENTAL INFORMATION FOR BLAST

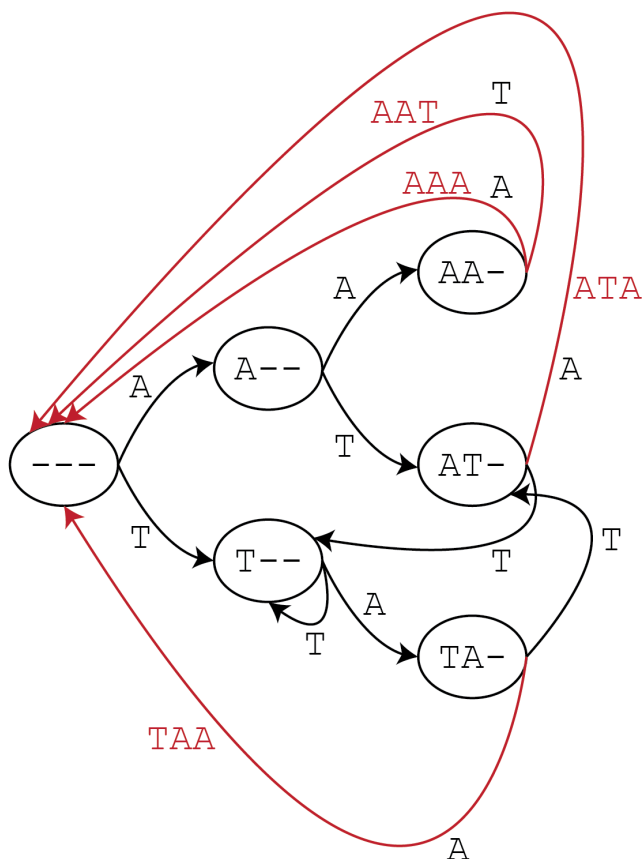
#### ***G.1 Seed Scanning***

Both ssBLAST and iiBLAST use a Mealy finite state machine (FSM) to scan for seeds within the reference database [95, 77]. The Mealy FSM can be graphically represented as a collection of states linked by transitions triggered by inputs (individual bases or ionic currents from the reference database), or mathematically represented by a “transition” and “emission” matrix (Fig G.1).

Simply, the FSM-based seed scan works as follows (algorithm 8). Starting in the null state (“— — —” in Fig G.1), we read in the bases from the reference database one-at-a-time. As each base is read in, we move along the corresponding pathway from the initial state to a new state. Each time the input base takes us along an emitting pathway (red pathways in Fig G.1), we report a seed and continue. This procedure continues for the entire length of the reference genome and results in a complete list of all seed locations within the reference. Computationally, this procedure requires  $N$  lookups from the transition matrix and emission matrix, where  $N$  is the length of the reference.

**Query:** AAATAAA

**List:** {AAA, AAT, ATA, TAA}



**Transition Matrix**      Input

State		A	T
	---	A--	T--
	A--	AA-	AT-
	T--	TA-	T--
	AA-	---	---
	AT-	---	T--
	TA-	---	AT-

**Emission Matrix**      Input

State		A	T
	---	x	x
	A--	x	x
	T--	x	x
	AA-	AAA	AAT
	AT-	ATA	x
	TA-	TAA	x

Figure G.1: Finite state machine for seed finding. This example demonstrates a simple case of a Mealy FSM for seed finding. In this example, our genetic alphabet is made up only of *A*'s and *T*'s, and we are using *k*-mers with  $k = 3$ . All 3-mers present in the query sequence are added to the list of possible seeds. The state machine has a distinct state (ovals) for each possible substring of the possible seeds ( $\{---, A--, T--, AA-, AT-, TA-\}$ ). Arrows show state-to-state transitions given that the next incoming base is an *A* or a *T*. The Mealy FSM emits on transitions, rather than from states. When a state gains an input that forms a completed seed, it emits a seed hit and returns to the initial blank state (red lines). The entire machine can be concisely expressed using a transition and emission matrix. The transition matrix summarizes which state an initial state will transition into given a particular input. The emission matrix summarizes what seed hit is emitted from each state given different inputs (red 3-mers) or if no seed is emitted (black "x"). Scanning each base in the reference genome amounts to a single lookup from both the transition and emission matrices.

---

**Algorithm 8** Mealy FSM seed scan. This example algorithm refers to the ssBLAST case of bases as inputs, but works identically in the iiBLAST case of ionic currents as inputs.

---

- 1: Start with a query sequence  $\mathcal{Q}$ , a reference database of known sequence  $\mathcal{R}$ , and an alphabet  $\alpha$  of entries in  $\mathcal{Q}$  and  $\mathcal{R}$   $\triangleright$  For ssBLAST,  $\alpha = \{A, C, G, T\}$
  - 2: **List**: generate a list  $\mathcal{L}$  of all k-mers existing in  $\mathcal{Q}$
  - 3: **Build**: build a FSM comprised of
    - a transition matrix  $\mathbb{T}$  where  $\mathbb{T}_{i,j}$  is the output state resulting from input  $\alpha_j$  into state  $i$
    - an emission matrix  $\mathbb{E}$  where  $\mathbb{E}_{i,j}$  is the emitted seed resulting from input  $\alpha_j$  into state  $i$
  - 4: **Initialize**: initialize variables prior to beginning scan
    - State  $S \leftarrow 1$
    - Seed list  $\Sigma \leftarrow \text{zeros}(1, \text{length}(\mathcal{R}))$
  - 5: **for**  $i \in 1 : \text{length}(\mathcal{R})$  **do**
  - 6:    $\alpha_{in} \leftarrow \mathcal{R}_i$
  - 7:    $S \leftarrow \mathbb{T}_{S, \alpha_{in}}$
  - 8:    $\Sigma[i] \leftarrow \mathbb{E}_{S, \alpha_{in}}$
  - 9: **end for**
  - 10: **Output**:  $\Sigma$  is a final list of which seed (if any) was found at each position in the reference  $\mathcal{R}$
- 

## G.2 Binning

To adapt the Mealy FSM seed scanning method (section G.1) to work with continuously valued ionic current inputs, we binned the observed ionic currents, as discussed in section 4.4.2.

The binning was constructed based on the composition of the ionic current-to-sequence model used in base calling. For the data used in this experiment, ONT used a k-mer model, with  $k = 5$ , comprising  $4^5 = 1024$  distinct sequence states, each with an associated ionic current. Given a number of bins  $N_{bins}$  (section G.4), the partitions between bins were chosen so that each bin contained the same number of sequence states.

As mentioned in section 4.4.2, binning may seem a bad way of adapting our seed scanning algorithm to work for continuously-valued inputs, given the overarching goal of this work to avoid destroying information. Indeed, we tested an alternative, bin-free method of seed scanning.

Briefly, this method also used a Mealy FSM, but with “fuzzy”, rather than deterministic,

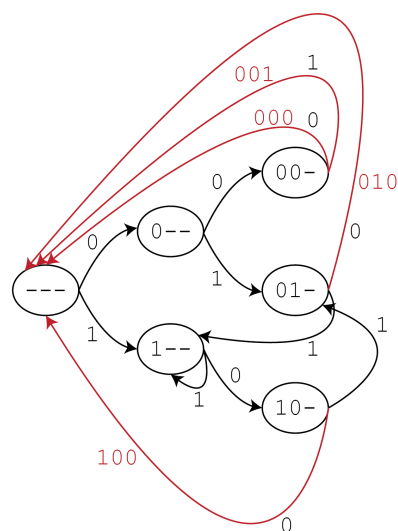
logic. The query is still binned to give a finite number of possible seeds and states, but the input from the reference is kept as continuously-valued. As such, the instantaneous “state” of the FSM is expressed not as a single state value but rather as a vector of the fractional occupation of all possible states. Likewise, the “inputs” from the reference database are vectors expressing the relative input contribution along the different alphabet elements. Seeds are reported when the build-up occupation of an emission state exceeds some threshold value. A simple example (Fig G.2) demonstrates this method for the case of continuously-valued inputs between 0 and 1.

We ultimately decided against the fuzzy FSM method in favor of the deterministic, binned standard FSM seed scan for two reasons. First, the fuzzy method failed to demonstrate meaningful improvement over the binned method. It seems that for the purposes of finding short, exact seed matches, binning does not appreciably diminish the information in the signal. Indeed, in many cases seed finding was easier with the binned method, as in this case we did not require tuning of the emission threshold parameter, which is strongly sensitive to the seed size and the chosen discretization of the query signal.

Secondly, the fuzzy method is considerably more costly in terms of computation. Rather than a simple lookup at each step in the scan of the database, the fuzzy method requires a matrix multiplication, where an  $N_{states} \times N_{states}$  size matrix multiplies the instantaneous state vector in order to progress the scan by a step. For typical iiBLAST run parameters,  $N_{states}$  is well over 500, so this sort of matrix multiplication is quite cumbersome. Overall, as the fuzzy method failed to improve performance while simultaneously increasing the computational load, it was abandoned in favor of the binned approach.

**Query:** 0001000

**List:** {000, 001, 010, 100}



State	Input									
	0.5	0	0.5	1	0.25	0.25	0.125	0	0	
---	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0--	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1--	0.00	0.50	0.00	0.00	0.50	0.25	0.06	0.03	0.00	0.00
00-	0.00	0.00	0.50	0.50	0.00	0.00	0.56	0.66	0.88	0.00
01-	0.00	0.00	0.00	0.50	0.50	0.00	0.19	0.09	0.00	0.00
10-	0.00	0.00	0.50	0.00	0.00	0.75	0.19	0.22	0.13	0.00

Emission	Input									
	0.5	0	0.5	1	0.25	0.25	0.125	0	0	
000	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.49	0.66	0.88
001	0.00	0.00	0.00	0.25	0.50	0.00	0.00	0.07	0.00	0.00
010	0.00	0.00	0.00	0.00	0.00	0.38	0.00	0.16	0.09	0.00
100	0.00	0.00	0.00	0.25	0.00	0.00	0.56	0.16	0.22	0.13

Figure G.2: Fuzzy FSM. This example presents the fuzzy FSM method for a short query and a brief excerpt of a reference database. The query sequence is used to construct a FSM comprising the possible states and inputs. However, inputs are now expressed as continuously-valued numbers between 0 and 1. These inputs can be thought of as 2-vectors with the two entries representing the “fraction” of the input along the **0** or **1** direction. For example, an input = 0.75 is the 2-vector (**0**, **1**) = (.25, .75). The instantaneous state is now expressed as an occupation vector of the fractional weight present in each of the possible states (upper table). A similar emission vector is tracked (lower table) and a seed is reported whenever a single component in the emission vector exceeds the emission threshold. In this example, the emission threshold is 0.75, and the seed **000** is reported at the last step, in response to the input sequence **0.125 – 0 – 0**.

### ***G.3 Seed Extension***

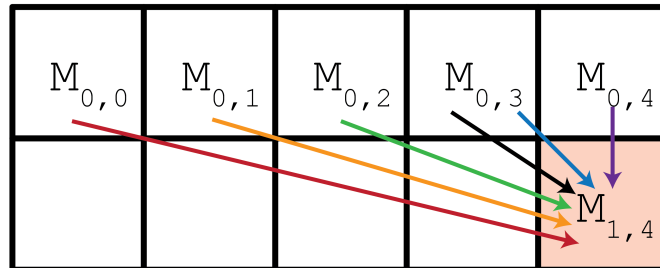
We use a non-standard seed extension method in both ssBLAST and iiBLAST in order to cope with the high rate of insertions and deletions (indels) in nanopore sequencing data. The main feature of our method is that we allow gapped alignments. In the simplest seed extension method, each cell  $(i, j)$  in the growing alignment matrix can only be entered via a step from cell  $(i - 1, j - 1)$ , corresponding to stepping one location forward in both the query and the reference. In our gapped alignment method [58], each cell can be entered from several initial cells, with each different pathway in weighted by an associated transition penalty (Fig G.3a). For each cell, all possible transition pathways in are scored, and the cell is populated with the best of the several scores. This gapped alignment strategy is able to handle the indels and poorly measured states common in nanopore reads.





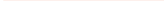

The downside of the gapped alignment method is that it dramatically increases the computational cost of the extension phase. Rather than a single calculation to fill each cell as in the ungapped alignment method, we now require a calculation for each potential path in. Furthermore, we must explore a larger swath of the alignment matrix, as the gapped alignment makes more cells conceivably reachable.

To cope with the computational cost and prevent it becoming intractable, we use a windowed extension method with a finite lookout distance. As the extension proceeds, at each extension step a new row and column are added to the alignment matrix. Rather than filling all of the newly added cells, the windowed method fills only those cells within a fixed lookout distance of the best scoring cell in the previously added row and column (Fig G.3b). This lookout cutoff prevents the computation time from blowing up as the extension gets long and the corresponding alignment matrix gets large. Instead of each subsequent extension adding linearly more cells to fill, the number of cells to fill now plateaus at a reasonable value, allowing extension to proceed in a tolerable time. The same fundamental extension algorithm (algorithm 9) extends seeds both to the right and the left, and is used for both ssBLAST and iiBLAST.



(a)



	$A = M_{0,4} + S(1, 4) + P_H$
	$B = M_{0,3} + S(1, 4) + P_S$
	$C = M_{0,3} + P_B$
	$D = M_{0,2} + S(1, 4) + P_K$
	$E = M_{0,1} + S(1, 4) + P_K + P_{K+}$
	$F = M_{0,0} + S(1, 4) + P_K + 2P_{K+}$
$M_{1,4} = \max(A, B, C, D, E, F)$	

(b)

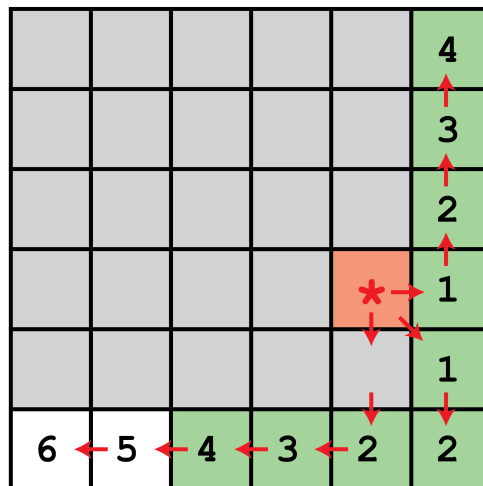


Figure G.3: Seed extension algorithm. **(a)** Method of gapped alignment. The gapped alignment algorithm evaluates multiple possible transitions into a given cell. Here,  $M(1, 4)$  is populated with the maximum score of 6 different potential transitions in. The  $A$  transition is a **hold**, where the query position progresses, but the reference position stays constant. The hold transition pays a penalty  $P_H$  in addition to the match score of query entry 1 and reference entry 4 ( $S(1, 4)$ ). The  $B$  transition is a **step**, where both the query and reference positions are incremented by 1. There is an associated penalty  $P_S$  along with  $S(1, 4)$ . The  $C$  transition is a **bad**, where we decide to discard a query entry at the cost  $P_B$ . This transition avoids paying the match score  $S(1, 4)$  at the expense of the bad penalty  $P_B$ . The  $D$ ,  $E$ , and  $F$  transitions are different size **skips**, where the reference position is incremented by a larger amount than the query position. These transitions pay a base penalty  $P_K$  to initiate a skip, and an additional penalty  $P_{K+}$  to extend the skip, as well as the match score  $S(1, 4)$ . Ultimately,  $M(1, 4)$  is populated with the best scoring option. **(b)** Method of windowed extension. The existing 5x5 alignment matrix (gray cells) is extended by adding a new row and column. Of the newly added cells, only those within the lookout distance of the best scoring cell in the outer rank of the 5x5 matrix (red cell, marked with star) are to be filled. Red arrows show what constitute steps of distance 1. In the case of a lookout distance of 4, the green cells would be filled while the white cells would be left unfilled.

---

**Algorithm 9** Extend seed to the right. A functionally identical algorithm is used to also extend seeds to the left)

---

```

1: Assume there is a seed found at query position  $q_i$  and reference position  $r_j$ 
2: Initialize variables
    $M \leftarrow [0]$  ▷ alignment matrix
    $S_{best} \leftarrow 0$  ▷ best score in M
    $S_{rank} \leftarrow 0$  ▷ best score in outermost row/column of M
    $loc_{best} \leftarrow (0, 0)$  ▷ indices of  $S_{best}$ 
    $loc_{rank} \leftarrow (0, 0)$  ▷ indices of  $S_{rank}$ 
    $q \leftarrow q_i$  ▷ current position in query
    $r \leftarrow r_i$  ▷ current position in reference
    $misses \leftarrow 0$  ▷ counter for consecutive misses
    $STOP \leftarrow FALSE$  ▷ flag to terminate alignment
3: while  $\sim STOP$  do
4:   if  $q > length(query)$  or  $r > length(reference)$  then ▷ make sure we have not
     overrun the end of the query or reference
5:      $STOP \leftarrow TRUE$  return  $S_{best}, loc_{best}$ 
6:   end if
7:   Add new cells  $\{(x, y)\}_{new}$  to  $M$  forming a new right-most row and new bottom-most
     column.
8:   for cell  $(x, y) \in \{(x, y)\}_{new}$  do ▷ check to see if the cell is within the lookout
     distance from  $loc_{rank}$ 
9:     if  $distance((x, y), loc_{rank}) \leq lookout$  then ▷ for distance, see fig. G.3b
10:       $M(x, y) \leftarrow \max\{\text{all alignment paths in}\}$  ▷ for paths in, see fig. G.3a
11:    else
12:       $M(x, y) \leftarrow -\infty$ 
13:    end if
14:  end for

```

---

---

```

15:    $S_{rank} \leftarrow \max\{M(\{x, y\}_{new})\}$     $\triangleright$  update  $S_{rank}$  with the best score of the newly filled
      cells
16:    $loc_{rank} \leftarrow loc(S_{rank})$   $\triangleright$  update  $loc_{rank}$  with the location of the best score of the newly
      filled cells
17:   if  $S_{rank} < S_{best} - T_{termination}$  then            $\triangleright$  have fallen too far below best score, stop
      extension
18:        $STOP \leftarrow TRUE$  return  $S_{best}, loc_{best}$ 
19:   else if  $S_{rank} \geq S_{best}$  then                        $\triangleright$  new best score, update and continue
20:        $S_{best} \leftarrow S_{rank}$ 
21:        $loc_{best} \leftarrow loc_{rank}$ 
22:        $q \leftarrow q + 1$ 
23:        $r \leftarrow r + 1$ 
24:   else                                                  $\triangleright$  intermediate score, record miss and continue
25:        $misses \leftarrow misses + 1$ 
26:       if  $misses > misses_{max}$  then  $\triangleright$  have exceeded the maximum allowed consecutive
      steps without improving alignment, stop extension
27:            $STOP \leftarrow TRUE$  return  $S_{best}, loc_{best}$ 
28:       end if
29:        $q \leftarrow q + 1$ 
30:        $r \leftarrow r + 1$ 
31:   end if
32: end while

```

---

### G.4 Algorithm Parameters

For the validation experiment, we ran both iiBLAST and ssBLAST with fixed parameters. These parameters gave satisfactory performance during pre-run trials, but were not extensively optimized. A subset of 19 reads of variable read accuracy were aligned against the reference viral database using a range of values for each of the run parameters. The parameters producing the best performance for this trial set (in terms of number of reads aligned and strength of alignment) were used for the full performance evaluation experiment. Although it is likely that further fine-tuning of the parameters of both algorithms could moderately improve performance, the alignment results were qualitatively similar within reasonable choices of the run parameters. The final parameters we used are shown in table G.1. The parameters work as follows.

1. Seed size ( $w_0$ ): number of bases (ssBLAST) or currents (iiBLAST) making up the seeds for the first phase of the BLAST algorithm. Typically, BLAST operates with a larger seed size (12 or 14 is common). However, since seeds are only found at exact matches, seeds must be smaller for the low accuracy nanopore data. Setting the seed size too small hurts run time as too many candidate seeds are found. See section G.5 for an in-depth discussion of this parameter.
2. Number of bins ( $N_{bins}$ ): the number of bins used to discretize the currents for the seed phase. Too many bins makes exact matches too rare, while too few can cause many falsely matched seeds. There is a balance between  $N_{bins}$  and  $w_0$  which yields good performance. Bins are chosen to partition the states in the sequence-to-current model into evenly-populated bins.
3. Match penalty ( $p_{match}$ ): how much a base-to-base or current-to-current match is rewarded (positive values) during the seed extension phase of the BLAST algorithm. In the case of iiBLAST and its continuously valued signal, matches between two currents  $i_1$  and  $i_2$  with variances  $\sigma_{i1}^2$  and  $\sigma_{i2}^2$  are scored as  $score = p_{match} - \frac{i_1 - i_2}{\sqrt{\sigma_{i1}^2 + \sigma_{i2}^2}}$ .

4. Mismatch penalty ( $p_{mismatch}$ ): how much base-to-base mismatch is penalized (negative values) during the seed extension phase. For iiBLAST, the scores are calculated as a rescaling of the distance between the currents (see previous), so  $p_{match}$  also sets  $p_{mismatch}$  and this second parameter is not needed.
5. Step penalty ( $p_S$ ): how much a forward step transition is penalized (negative values) during seed extension (fig G.3a). A value of zero for the step penalty means that there is no penalty for a step transition.
6. Skip penalty ( $p_K$ ): penalty for initiating a skip during seed extension (fig G.3a).
7. Skip extension penalty ( $p_{K+}$ ): penalty for extending an already-initiated skip during seed extension (fig G.3a).
8. Bad penalty ( $p_B$ ): penalty to completely discard a measured state during seed extension (fig G.3a). This is unimportant to the ssBLAST algorithm, as a measurement can only ever simply mismatch with the reference and take the penalty  $p_{mismatch}$ , which is capped. However for iiBLAST, a measurement can in principle be arbitrarily far from the reference. In this case, it is likely that the measurement was generated by some other, non-DNA-translocation-related phenomenon and is not indicative of the target DNA sequence. This  $p_B$  allows for a work around against these nonsense measurements.
9. Hold penalty ( $p_{hold}$ ): penalty to match a measured state to the same reference state as the previous measurement was matched to during seed extension (fig G.3a).
10. Longest skip ( $K_{max}$ ): upper limit on the longest skip ahead allowed during seed extension. This parameter mainly works to improve run time, as calculating longer skips adds computational requirements while the longer skips are only rarely the optimal transition.
11. Lookout ( $L$ ): sets the lookout window during the seed extension (fig G.3b).

12. Max misses ( $M_{max}$ ): maximum number of consecutive seed extensions allowed while failing to improve on the best alignment score before seed extension is terminated (see algorithm 9). Improves run time by forcing the extension to end if it is not proving productive.
13. Termination threshold ( $T_{term}$ ): how far the present alignment score during extension is allowed to fall below the best score observed previously in extension before extension is terminated. Improves run time by stopping extensions once they stop generating better scores.

Parameter Name	ssBLAST Value	iiBLAST Value
seed size ( $w_0$ )	7	7
number of bins ( $N_{bins}$ )	N/A	8
match penalty ( $p_{match}$ )	1	2
mismatch penalty ( $p_{mismatch}$ )	-1	N/A
step penalty ( $p_S$ )	0	0
skip penalty ( $p_K$ )	-1	-3
skip extension penalty ( $p_{K+}$ )	-1	-1
bad penalty ( $p_{bad}$ )	-10	-5
hold penalty ( $p_{hold}$ )	-10	-2.5
longest skip ( $K_{max}$ )	4	4
lookout ( $L$ )	7	7
max misses ( $M_{max}$ )	25	25
termination threshold ( $T_{term}$ )	-12.5	-7.5

Table G.1: Parameter values used in validation experiment

### ***G.5 Seed Size Parameter for Variable Read Lengths***

The validation experiment was conducted using fixed length 100 base reads, rather than the full length, several thousand base reads typical of nanopore sequencing data. As discussed in the main text, the use of smaller reads provided a wider range of sample sequencing accuracies over which to evaluate performance. Additionally, constant length reads were important for the purposes of this validation for a second reason; we found that the run parameters leading to good performance for both ssBLAST and iiBLAST were dependent upon the query length.

Specifically, the seed size parameter must change for long or short read lengths. For short reads, typical BLAST seed sizes of around 14 bases do not yield good results. This is because error-prone sequencing is unlikely to yield 14 consecutive correctly called bases within a short read. So, for shorter reads, a smaller seed size is necessary.

However, small seed sizes (i.e. 7) are not practical for longer read lengths. In this case, the number of seed words existing within the query read approaches the total number of possible seed words. The total number of unique seed words for a seed size  $w_0$  and an alphabet of size  $d$  is given by  $N = d^{w_0}$ . For ssBLAST,  $d$  is 4 (the alphabet is  $\{A, C, G, T\}$ ) and for iiBLAST  $d$  is equal to the number of bins used to partition the currents ( $N_{bins}$ ). So, as our read lengths grow into the thousands of bases (or currents), the list of seed words present in the read becomes a significant fraction of the total possible seed words. For example, a random 5000 base sequence can be expected to contain ~30% of all 7-letter seed words, meaning a seed will be found at ~30% of all positions in the reference genome. This density of found seeds is clearly too high to be useful, and results in the BLAST algorithm effectively calculating a complete Smith-Waterman gapped alignment against the reference genome.

Overall, the important parameter for variable-length reads providing good performance is the ratio of seed words present in the read to the total number of possible seed words. This ratio can be increased (reduced) by shortening (lengthening) the seed size  $w_0$  or by reducing (increasing) the number of bins  $N_{bins}$  in the case of iiBLAST.



Ultimately, the basic BLAST algorithm is not the best approach to error-prone nanopore data [91, 92, 93, 94]. The best application of this work on current-to-current alignment will be to integrate it into a nanopore-specific BLAST implementation. In such an integration, the better identity from current-to-current comparisons will still improve alignments, while the surrounding architecture should allow smoother application to typical nanopore data.

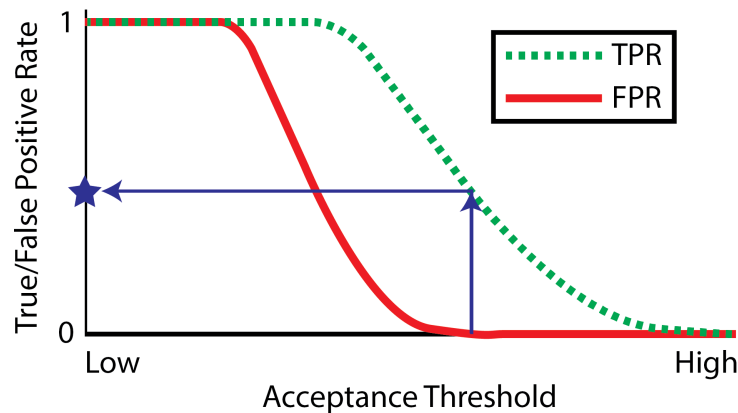


Figure G.4: TPR and FPR variation with acceptance threshold. As the acceptance threshold is tuned from low to high, both the  $TPR$  (green dashed line) and  $FPR$  (red solid line) fall from 1 to 0. The  $FPR$  falls off faster than the  $TPR$ . We choose the BLAST operational point to be the point where the  $FPR$  first reaches 0 (blue lines). The  $TPR$  at  $FPR = 0$  (blue star) is the metric reported in section 4.6.

### G.6 True Positive Rate at Zero False Positive Rate

The concept here is that we have the freedom to choose an acceptance threshold of alignment scores at which we will mark an alignment as meaningful. If we turn this threshold too low, we will simply accept all the candidates. At this operational point, we will keep all the true positives ( $TPR \rightarrow 1$ ) but also keep all the false positives ( $FPR \rightarrow 1$ ). Conversely, if we crank the threshold too high, we will reject all candidates. In this case, we will successfully discard all the false positives ( $FPR \rightarrow 0$ ) but simultaneously discard all the true positives ( $TPR \rightarrow 0$ ). If BLAST is generating useful results, the  $FPR$  should fall faster with an increasing acceptance threshold than does the  $TPR$ , allowing us to find a point where the  $FPR = 0$  while the  $TPR$  is still nonzero (Fig G.4). This point is the metric reported in section 4.6, Fig 4.6.

The performance metric of true positive rate provided zero false positive rate as a function of read accuracy (section 4.6, Fig 4.6) was extracted from the receiver operating characteristics of the iBLAST and ssBLAST algorithms over sets of reads binned by accuracy. The

aligned reads were binned by their sequencing accuracy into 10 bins each spanning a 4% range in accuracy (i.e. 74 – 78%). For each set of reads, we varied the acceptance threshold (how good an alignment score is required to report a match) and plotted the resulting true positive rate against the resulting false positive rate (fig G.5). True positive rate ( $TPR$ ) was calculated as  $TPR = \frac{TP}{TP+FN}$ . False positive rate ( $FPR$ ) was calculated as  $FPR = \frac{FP}{FP+TN}$ .  $TP$  is the total number of true positives,  $FN$  is the total number of false negatives,  $TN$  is the total number of true negatives, and  $FP$  is the total number of false positives.

We extracted the  $TPR$  value corresponding to  $FPR = 0$  for each set of accuracies to quantify performance as a function of read accuracy. We looked specifically at this  $FPR = 0$  point as this performance point best reflects the requirements of a typical BLAST-based experiment. In a typical experiment, the reference database will contain almost exclusively off-target genomes, meaning that we expect an overwhelming number of true negatives. Consequently, even a miniscule  $FPR$  will lead to most matches being false positives. Thus, a useful BLAST implementation must operate at  $FPR$  very close to 0 to generate useful results.

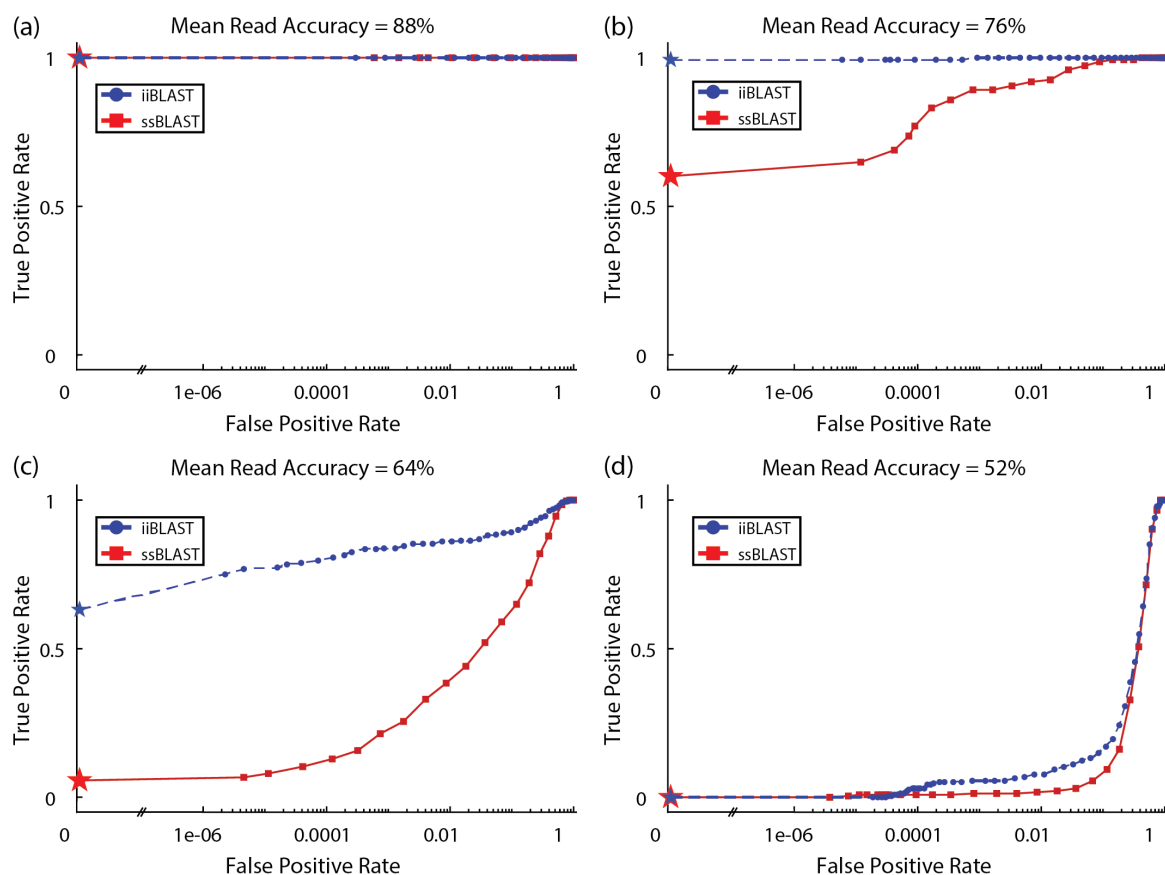


Figure G.5: Receiver operating characteristics for different read accuracies. The true positive rate is plotted against the false positive rate for reads with accuracies from 86 – 90% (a), 74 – 78% (b), 62 – 66% (c), and 50 – 54% (d). Red points show the performance of ssBLAST, blue points show the performance of iBLAST. The starred data points mark the true positive rate at zero false positive rate. The dashed blue line and solid red line connecting the data points are to guide the eye.

## G.7 Calculating Alignment P-Values

Alignment scores provide a relative ranking between alignments (higher scores are better), but the actual value of the score is not inherently statistically meaningful. The value of the alignment score is a function of the parameters used in the alignment algorithm and is not comparable to any alignment generated using different parameters. For example, the scores from iiBLAST and ssBLAST alignments cannot be directly compared, as alignment in the two algorithms uses different score parameters.

To compare alignment quality across algorithms, we need to assign a statistical significance to the alignment based on the score. We can do this by comparing the candidate alignment score against the distribution of scores observed for off-target alignments. We expect off-target scores to increase with increasing size of the off-target genome, as a longer genome provides more opportunities for a high scoring random alignment. Indeed, we observe that off-target scores increase logarithmically with genome length (Fig G.6). We normalized all off-target scores for genome length by shifting the scores along the logarithmic fit to the length of the M13mp18 genome (7249 bp).

For all of the reads, we took the best alignment score to each off-target reference genome and used the genome length correction function to bring all scores to the reference length of M13mp18. This provides us an empirical distribution of scores for random alignments (Fig G.7). We fit an extreme value distribution

$$f(s|\mu, \sigma) = \sigma^{-1} e^{\left(\frac{s-\mu}{\sigma}\right)} e^{-e^{\left(\frac{s-\mu}{\sigma}\right)}} \quad (\text{G.1})$$

to the tail of the observed distribution for both iiBLAST and ssBLAST. Using this fit, we can calculate the probability that a given alignment score  $s_*$  was generated by an alignment of a random sequence to the reference genome by

$$p_{\text{random}}(s_*) = \int_{s_*}^{+\infty} f(s|\mu, \sigma) ds \quad (\text{G.2})$$

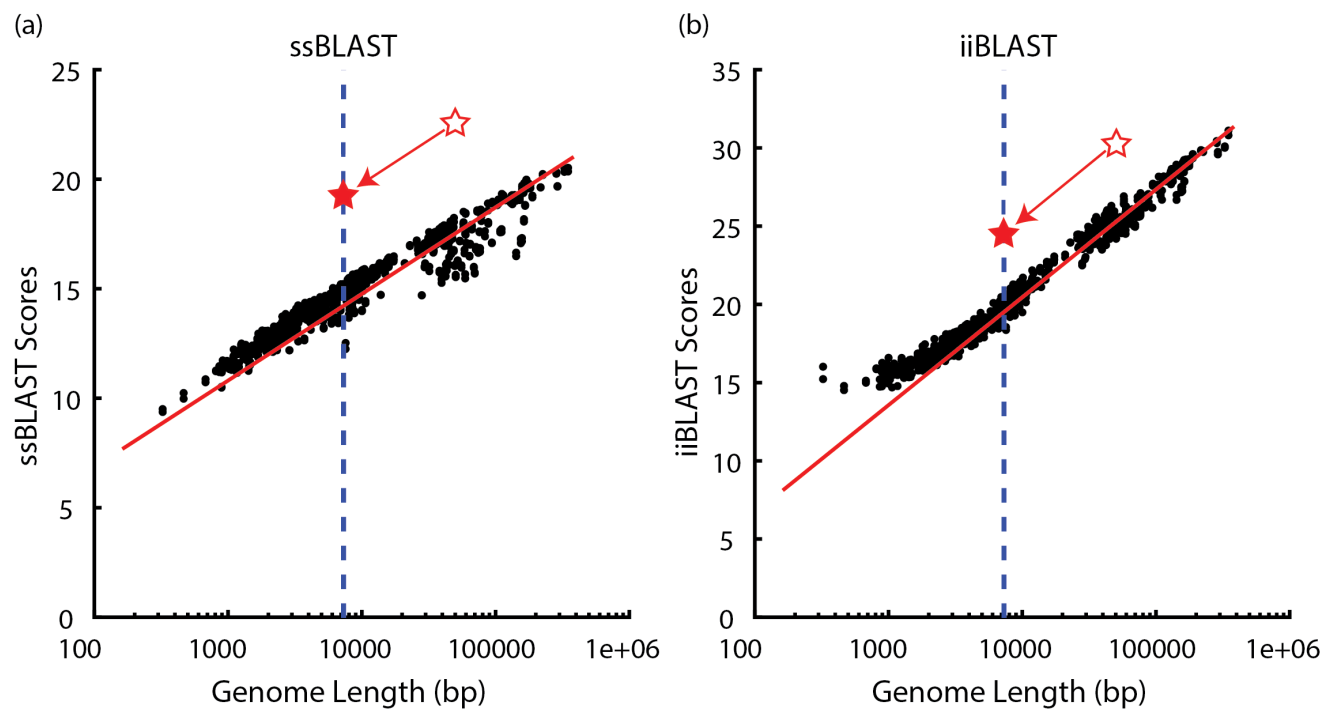


Figure G.6: Length correction for alignment scores. For each off-target reference genome, the mean best alignment score over all reads is plotted against the length of the genome in nucleotides (black points). Both ssBLAST **(a)** and iiBLAST **(b)** scores show a logarithmic dependence on the reference genome length (red line). The blue dashed line shows the M13mp18 genome length. An alignment score to a differently sized genome (unfilled red star) is shifted along the logarithmic fit (dashed red line) to a corrected score for the M13mp18 genome length (filled red star).

where  $f$  is defined by eq G.1 and  $\mu$  and  $\sigma$  are specific to the algorithm used to generate the alignment.

This  $p_{random}$  is the p-value for the alignment, and can be interpreted as a statement about how large of a reference database the read could be aligned against before a false positive alignment would be likely to generate a better score. For example, a p-value of  $10^{-4}$  means that a random alignment to a 7249 bp reference genome will generate a better score once in 10000 times. Therefore, this read could be aligned against a database of 72.49 Mb before we are likely to see a false positive outscoring the on-target alignment (eq G.3).

$$Size_{database} = \frac{Size_{M13mp18}}{p_{random}} \quad (G.3)$$

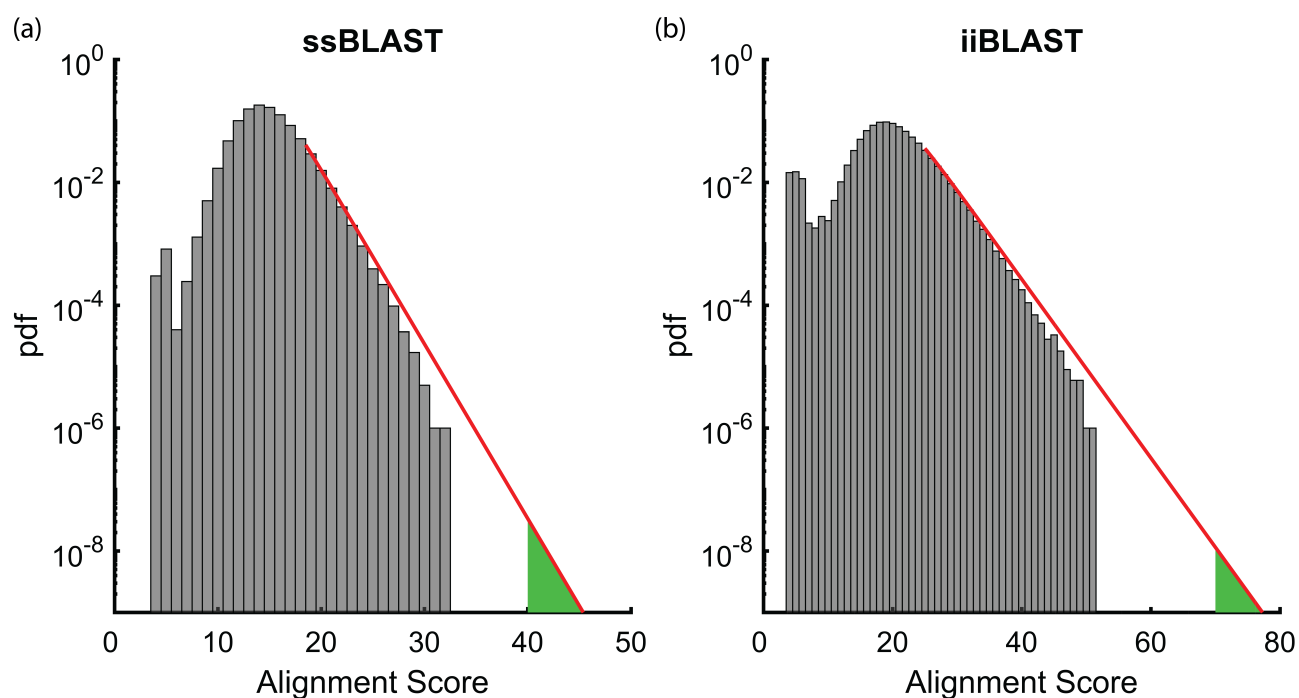


Figure G.7: Converting alignment scores to p-values. A maximum likelihood fit matches an extreme value distribution (red line, eq G.1) to the tail of the distribution of off-target alignment scores (gray) for both ssBLAST **(a)** and iiBLAST **(b)**. An alignment score  $s_*$  is converted to a p-value by integrating the fit distribution from  $s_*$  to  $+\infty$  (green shaded region). Histograms are pdf-normalized and displayed on a log y-scale.



Phase	iiBLAST (sec)	ssBLAST (sec)
Generate list of all seed words in the query	$9.0 * 10^{-4}$	$3.9 * 10^{-4}$
Build Mealy FSM for the seed search	0.23	0.034
Put currents into bins	0.0066	N/A
Scan database for seeds	0.45	20
Locate the found seeds in the query	0.058	0.19
Extend seeds into alignments	393.5	1787
Total	394	1807

Table G.2: Mean time spent per read (in seconds) during the various phases of the BLAST algorithm

### ***G.8 Computation Time***

Over the 1977 reads aligned in this study, the mean run time for iiBLAST was 394 seconds and the mean run time for ssBLAST was 1807 seconds. The improvement in run time for iiBLAST over ssBLAST is attributable entirely to spending less time on seed extension. In general, the seeds found during the scan phase were higher quality for iiBLAST than ssBLAST, meaning that a lower percentage of the candidate seeds led to uninteresting final alignments. So, ssBLAST spent much more time extending seeds into ultimately discarded alignments. Overall, the extension phase makes up nearly all of the computational burden. For iiBLAST, extension made up 99.8% of the total run time, with the next most costly step being the scan phase at 0.1% of the total. For ssBLAST, extension was 98.9% of total run time, with scanning taking up most of the remainder at 1.1%. The scanning phase was somewhat slower for ssBLAST as more seeds were found, and the computational cost to record a found seed was consequently larger than for iiBLAST. A complete accounting of the average per-read time spent during the various phases of the BLAST algorithm can be found in table G.2.