Getting the most out of a nanopore

Henry Brinkerhoff

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

Anton V. Andreev

Program Authorized to Offer Degree:
Department of Physics

University of Washington

Abstract

Getting the most out of a nanopore

Henry Brinkerhoff

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

Over the past three decades, the fields of biophysics and biotechnology have seen an era of unprecedented growth, bolstered by the development of new experimental techniques. Prominent within these techniques are "single-molecule" methods which enable the observation and manipulation of single biomolecules. These techniques allow for controlled experiments on the most fundamental structures composing life.

The function of living organisms is governed by statistical physics, and traditional bulk chemical methods report only an average of the rich and heterogeneous activity of these biological structures. Therefore, bulk methods provide an incomplete picture of the mechanical behavior of biomolecules, and of the way life stores, modifies and accesses information.

Early single-molecule experiments using flow cells and magnetic tweezers were initially used to study the passive mechanics of molecules like DNA and the behavior of stepping enzymes including myosin and kinesin. These enzyme experiments demonstrated the power of single-molecule techniques, showing that enzyme behavior is fundamentally statistical, moving randomly with a slight rectification provided by chemical potentials maintained by the cell. Soon thereafter, an expanding repertoire of single-molecule methods including superresolution micrscopy, single-fluorophore

microscopy and optical tweezers refined and expanded these results, making plain the diversity and complexity of the mechanical behavior of biomolecules.

Advances in genomics over this time period made it clear that the nucleic acids DNA and RNA, which store the information passed down and used by life to encode the sequences of every protein it produces, are also subject to the statistical physics governing biomolecules. Damage to DNA and its repair, the formation of secondary structure, the insertion of viral DNA fragments, replication, recombination, modification of bases, and regulation of gene expression: these are all fundamentally random processes. As recording and analyzing vast amounts of data has become more feasible with access to greater computing power, it has become clear that methods sequencing only large samples of many DNA molecules fail to recognize the variance crucial to the functionality of life's genetic library.

The scientific appeal of a single-molecule sequencing technique together with a push for longer read lengths and cheaper sequencing led to the development of nanopore sequencing, a method using a nanometer-scale hole in a thin membrane to trap and analyze DNA. Beginning with the demonstration of nanopores as single molecule "Coulter counters," through results proving that nanopore experiments can discriminate between trapped DNA strands with different base content, we now have arrived in an era where nanopores are used in commercial DNA sequencing platforms and high-precision single molecule biophysics experiments.

Within this dissertation, I provide a "user manual" of sorts for collecting and understanding the single-molecule information provided by nanopore experiments. Then, through two examples of concrete improvements to the nanopore DNA sequencing system, I demonstrate how a thorough understanding and adequate physical model of the system can motivate experiment and invention. My hope is that a

scientist wishing to perform nanopore experiments for the first time will find this to be a useful guide for executing the experiments, as well as for modeling and analyzing the rich and complex signals that they generate.

In part I, background is provided on the arena in which these experiments play out. I first introduce key properties of DNA and other biological molecules, as well as the history and future of DNA sequencing.

Part II contains a guide to the experimental setup and operation of a nanopore experiment. I also delve deeper into the biophysics of the experiment as it is performed at the University of Washington, discussing properties of the enzyme-DNA-nanopore complex, and I discuss the signal obtained from the experiment and its properties.

In part IV, describe the ways the nanopore signal can be modeled, reduced, analyzed, and interpreted, including introductions to some commonplace analysis tools used to study single molecule data.

Finally, part IV shows how using the results of this model, we developed extensions and modifications of the nanopore experiment, improve the accuracy and flexibility of nanopore DNA sequencing.

My work at the University of Washington is included primarily in part IV, much of which I completed in collaboration with primarily Dr. Andrew Laszlo and Dr. Brian Ross, and IV, in which the variable-voltage experiments were completed in collaboration with Dr. Matthew Noakes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

GLOSSARY

This glossary does not contain terms commonly used and defined in scientific literature. In fact, most of the terms here are unlikely to appear in publications. Rather, it is a list of jargon that is commonly used internally by nanopore scientists, especially those at the University of Washington. It also contains acronyms, words used in unconventional ways, and words that while technically correct could bear some added precision in their definition to be clear in a scientific context.

AC NANOPORE EXPERIMENTS: Another name for variable-voltage experiments.

AIC: Akaike information criterion.

ADAPTOR (DNA): A short oligo made to bind to the end of a strand to be read by the nanopore. Designed to provide enzyme binding sites, recognition sequences, or free ssDNA that can thread into the pore, among other applications.

ALIGNMENT: This functions both as an action-noun, as in "I made an alignment of the two sequences," and as a specific object, whose elements a_i are integers that indicate that the ith element of a measured sequence of states to the a_ith element of the reference sequence of states.

BACKWARDS AND FORWARDS PORE: MspA is said to be in the forwards orientation when the vestibule is on the *cis* side of the membrane, and the backwards orientation when the vestibule is on the *trans* side. This nomenclature has

a different meaning for each pore, with forwards being applied to the typical orientation of the pore in nanopore experiments.

BAD STATE: Any behavior classified as a unique state but not thought to relate to the underlying enzyme behavior or DNA sequence.

BIFURCATED STATE: An ion current state that reproducibly consists of two distinct ion currents.

BULLETPROOF BILAYER: A bilayer that is robust to high voltages but difficult to obtain insertions with. Thought to be caused by oversaturation with solvent.

CALIBRATION: The application of a scale (and sometimes offset) to an ion current sequence to account for systematic variation in temperature and salt concentration.

CALIBRATION SEQUENCE: A known sequence appended to the beginning of a DNA strand to be read. Used to provide a standard for calibration.

CAPACITANCE COMPENSATION (CAP COMP): The removal of capacitive currents in variable-voltage experiments. Can be performed in either software or hardware.

CLOG: Used alternately to refer to molecules clogging a pore, causing noisy and useless current blockades, and to refer to non-bilayer lipid completely obstructing an aperture.

CONSENSUS: A sequence of states believed to be the underlying true sequence, generated by comparison of many measured events.

CONSTRICTION: The narrowest part of the nanopore.

CONTINUOUS CURVE: A hypothetical smoothly changing ion current that would be observed if the DNA were moved smoothly through the pore at constant voltage.

DAQ: Pronounced /dăk/. Data Acquisition. Refers alternately to the hardware used for analog-to-digital conversion in the data, the computer it is installed in, and to the software used to run the experiment.

DEEP GATE: Low current states, characteristic of experiments using expired lipid.

DETERGENT: An amphiphilic chemical used to suspend protein. Disrupts bilayers at high concentrations.

DINUCLEOTIDE: A DNA strand consisting of two repeated nucleotides, e.g. "...ACACAC..."

DISAPPORANCE: When an inserted pore leaves the bilayer.

ENZYME SLIP: A large step taken by the nanopore over more than one or two bases in the read strand. Thought to be caused by enzyme dissociation.

EVENT: A blockage of the pore providing relevant data for an experiment.

FIC: Frequentist information criterion.

FLICKER: Time domain structure within an ion current state. Usually used to specify behavior not caused by enzyme stepping.

FRONT/BACK WELL: The two volumes of buffer on a puck. The front well is the well with the aperture end of the U-tube. The back well contains the open end of the U-tube.

HAND ALIGNMENT: Refers to either the manual construction of an alignment by eye, or to the alignment thus generated.

HAND CLICKING: Identifying state transitions by eye and marking them manually.

HAND CONSENSUS: A consensus built by deductive reasoning, observing several events with the same experimental conditions and building a reasonable sequence of true states.

HEADSTAGE: The preamplifier of the Axopatch 200B close to the experiment.

HOLD: Subsequent states where an enzyme has not actually progressed. Generally an artifact of gratuitous change point identification.

HOMOPOLYMER: A DNA strand consisting of a single repeated nucleotide.

INSERTION: A pore incorporated into a bilayer.

INTERACTION: Any observed blockage of the nanopore.

JUMPS: The structure containing information about change point locations.

KICKING A PORE/EVENT: Changing the voltage applied to the nanopore to induce either the termination of an event or the removal of a pore from the bilayer.

LEVEL: An ion current state. Note that this is used with some flexibility to mean any temporal unit into which we may divide nanopore data, and may contain internal structure.

MANO A MANO: Hand-to-hand, as in a physical confrontation.

NAN: Pronounced /năn/. Not-a-number, a programming tool to function as a placeholder for undefined values in an array.

ONT: Oxford Nanopore Technologies, a commercial producer of nanopore products.

PAINT: Dried down lipid, resuspended in hexadecane used for painting bilayers.

PARTIAL PORE: An insertion with unusually low current.

PHASE SHIFT: A shifting of the sampling points along the continuous curve caused by differences in experimental conditions, upstream DNA stiffness changes, and changes in enzyme attachment point or conformation.

PIPELINE: A series of analysis steps. E.g. "sequencing pipeline" or "variable voltage pipeline."

PIRANHA: A cleaning of the pucks using piranha solution.

POLYN: A homopolymer strand consisting of repeats of nucleotide N; e.g., polyA is AAAAA.

PREMIX PAINT: Lipid paint with MspA mixed in. Used for establishing backwards pores, or for situations where perfusion is not possible or desirable.

PUCK: The teflon armature with a U-tube used as the platform of the nanopore experiment at University of Washington.

QUADROMER: 4-mer or tetramer.

READHEAD: The number of bases affecting the ion current of the pore.

REDUCED DATA: Data that has been passed through the mean-and-decimate filter.

SCORE: A log-likelihood value assigned to a model or a portion thereof such as an alignment or a particular match within that alignment.

SELF-ALIGNMENT: An alignment of an ion current sequence to itself, used to identify backsteps and holds.

SEQUENCE: Does not exclusively mean nucleobase sequence. Commonly used to refer to an ordered set of ion current states or enzyme states.

STEP COUNTS: The number of each type of enzyme transition (skip, step, backstep, hold, bad level) in an event, or in a sample of many events. Used to compute step penalties.

STEP PENALTIES: The log-probabilities of each type of enzyme transition (skip, step, backstep, hold, bad level). Used by alignment or sequencing HMMs to assign scores to alignments.

SPIKE: An outlying data point of short (generally single-sample) duration within an ion current state.

TOGGLE: Rapid enzyme steps back and forth along a substrate.

UPGATE: Excursions to ion currents above the apparent open state.

UPSTREAM/DOWNSTREAM: Upstream refers to bases farther towards the *cis* side of the pore, usually to that side of the constriction. Downstream refers to bases on the *trans* side of the pore.

UTILIZE: To make practical and effective use of.

U-TUBE: The teflon tube connecting the *cis* and *trans* wells on a puck. The *cis* end is melted around a tungsten needle and shaved with a knife to produce an aperture.

VESTIBULE: The large, goblet shaped portion of MspA.

WIGGLE PLOT: Oxford Nanopore Technologies' term for ion current traces subdivided into ion current states.

Y-TAIL: A "split end" of DNA, where dsDNA bound to its complement breaks into two strands of ssDNA, because it has been engineered to be non-complementary

past the junction. Used, for example, in DNA adaptors to create single stranded threading ends.

ZAP: Brief application of large voltages to a membrane to break a bilayer. Used to test for bilayer existence.

ACKNOWLEDGMENTS

This work was supported by the National Institutes of Health, National Human Genome Research Institute $1,000 Genome Program Grants R01HG005115 and R01HG006321.

Part I

BACKGROUND

Understanding nanopore experiments and their present-day applications requires some background in molecular biology, particularly DNA sequencing and single molecule biophysics. A reader very familiar with fundamental biology may wish to skim this section, taking note only of the sections on nucleic acid sequencing and single-molecule physics.

Within the first chapter of this part, I provide a brief overview of the principal biological concepts underlying the materials used in and analyzed by these experiments: nucleic acids, proteins, biological membranes, and electrolytic buffers. This overview is meant to provide concrete definitions and some key physical clarifications for a scientific audience with only passing familiarity with biological concepts. I also provide some historical background of and enumerate present challenges in the field of DNA sequencing.

The second chapter includes a brief survey of single molecule techniques, a field of which nanopore experiments have gained a significant share in. Background in single molecule experiments is important for understanding how the behavior of nanopore systems is modeled and understood.

Chapter 1

BIOLOGICAL BACKGROUND

1.1 Nucleic acids

DNA is the blueprint for life, encoding the information about the proteins that ultimately determine the structure and function of organisms[1]. Its fundamental importance in every field of biology cannot be understated. This enormous molecule's structure affects the expression and regulation of genes on the time scale of seconds, while mutations in DNA that slowly accumulate over generations give rise to adaptation, genetic drift, and ultimately the evolution of species.

In organisms that exchange genes, DNA sequence is a molecular fingerprint that allows organisms to be uniquely identified and their heritage traced: it is simultaneously a technical manual and a historical tome. DNA and its code are therefore studied and used by molecular and organismal biologists, ecologists, epidemiologists, biotechnologists, physicians, pharmacologists, forensic scientists, farmers, anthropologists and historians.

Because of the breadth and importance of these applications, making the study and understanding of DNA accurate and accessible has been a central thrust of biology and biotechnology for half a century.

1.1.1 Structure of nucleic acids

DNA (deoxyribonucleic acid) and RNA (ribonucleic acid) are polymers, molecules consisting of a chain of repeating subunits called nucleotides. The structures of these

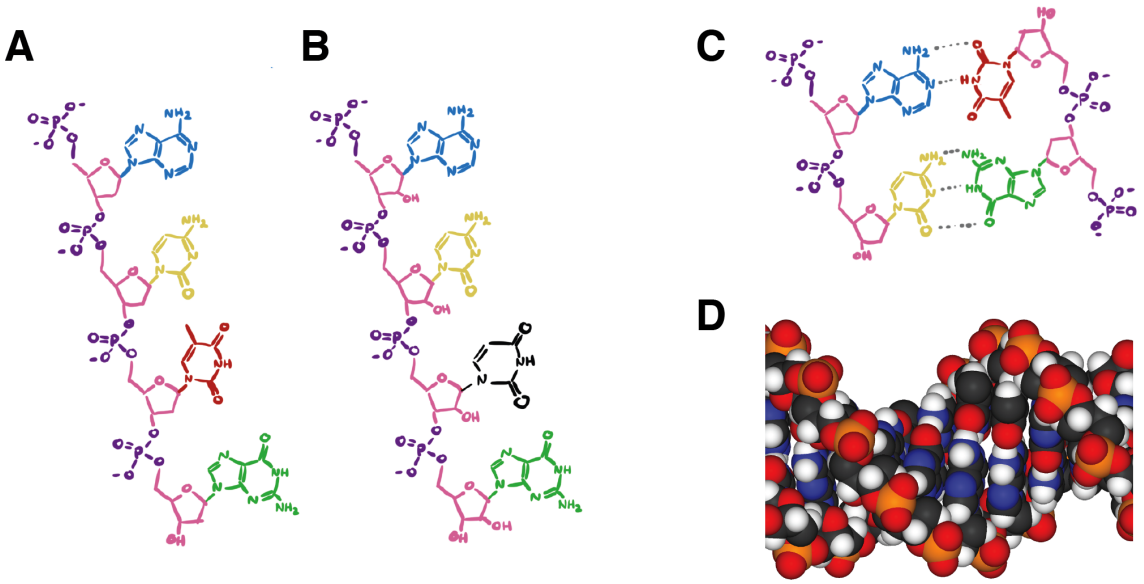

Figure 1.1: **The structure of nucleic acids.** A single-stranded nucleic acid consists of (a) deoxyribose (in DNA) or (b) ribose (in RNA) sugar rings (pink) alternate with phosphates (purple) forming a backbone. Each phosphate is bonded to one of four nitrogenous bases: adenine (blue), cytosine (yellow), guanine (green), thymine (red), or uracil (black). (c) Each base pairs through hydrogen bonds (gray dashed lines) with its complement, shown here for A-T and C-G pairing in DNA, forming a double-helical structure (d) of double-stranded nucleic acids. Space filling model adapted from wikipedia[2].

molecules are illustrated in figure 1.1.1. Each nucleotide consists of an alternating sugar-phosphate backbone with one negative charge per subunit at neutral pH. Attached to each subunit of the backbone is one of four variants called bases: adenine (A), cytosine (C), guanine (G) or thymine (T) for DNA, and adenine (A), cytosine (C), guanine (G) or uracil (U) for RNA.

Hydrogen bonding causes A to bind with T or U, and causes C to bind with G. Bases that bind to one another are called complementary bases. Two complementary single strands of DNA (called ssDNA for single-stranded DNA) bind together strongly

and specifically, running in opposite orientations, creating a double-helical structure called double-stranded DNA, or dsDNA[3]. The sequence of bases on a strand of DNA, typically written 5' to 3', is commonly referred to as DNA sequence. For genomic DNA, only one of the two strands codes for proteins and is called the "sense" strand. This is typically what is recorded in sequence databases.

In organisms, dsDNA is present as enormous molecules thousands to billions of bases long. In many organisms, the DNA is further structured and packaged to facilitate the replication of the DNA that is required for cell division. These molecules of DNA contain the information that codes for the proteins that take on functional or structural roles within the cell. DNA is copied by protein enzymes into complementary strands of RNA called messenger RNA or mRNA in a process called transcription (figure 1.1.1). The mRNA is then translated into a protein in a ribosome, a complex of RNA and protein.

Every three bases on the RNA, called a codon, corresponds to one of 20 amino acids in humans. A set of small RNA molecules called tRNA facilitates this translation, binding to both the codon and the amino acid building block it is keyed to. This correspondence is different in different organisms. RNA is less stable than DNA, and is even capable of catalyzing its own degradation. The mRNA therefore does not persist for a long time after its transcription; this allows cells to respond dynamically to their environments and change gene expression rates quickly, rather than continuing to transcribe one molecule. As it is released from the ribosome, the amino acid chain folds up to form a functional protein.

Each strand of DNA may contain the information about many proteins. Each protein in the cell is generally coded for by a contiguous segment of the DNA called a gene. However, one gene may code for multiple, similar proteins, whose final structure is determined during or after its transcription and translation. Additionally, some

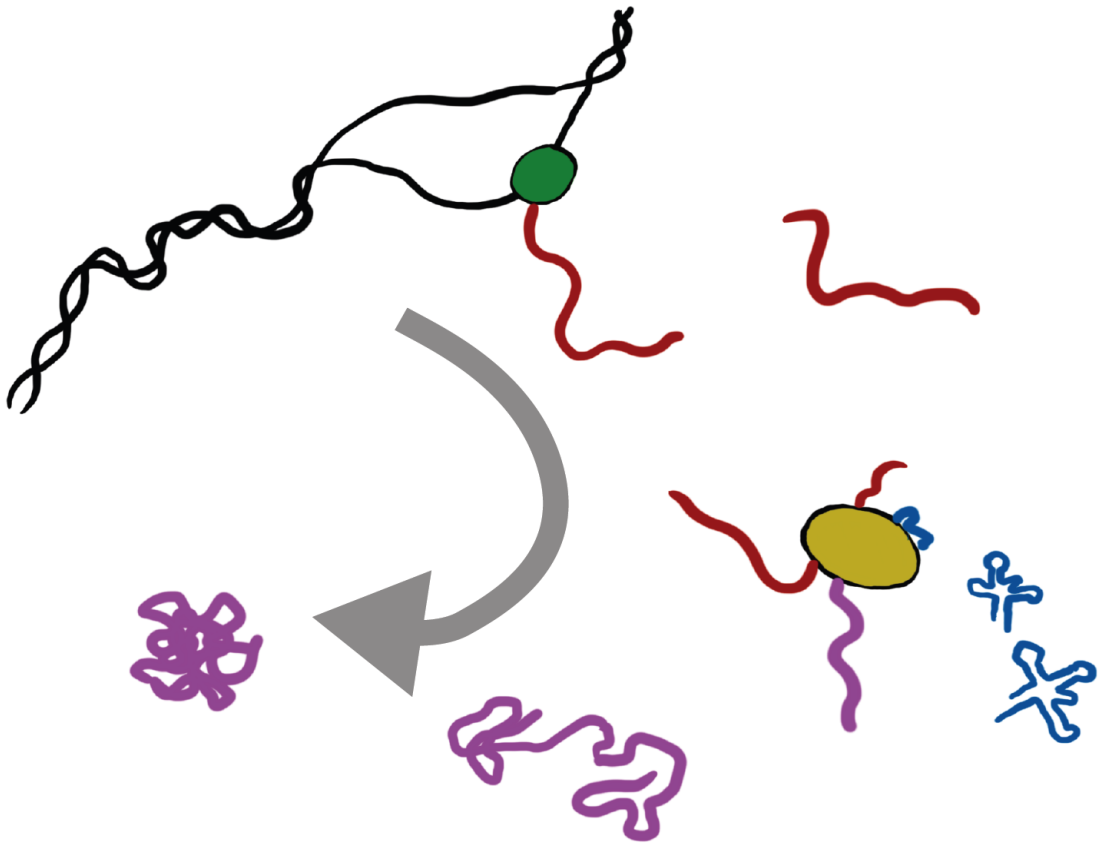

Figure 1.2: **Schematic of gene expression.** DNA (black) is transcribed by a polymerase (green) into mRNA (red), which is translated using tRNA (blue) by a ribosome (yellow) into a chain of amino acids (pink) which folds into a protein.

protein structures are actually composed of multiple protein subunits, each of which is encoded by a different gene. Genes are marked by sequences of DNA called stop and start codons, and are often adjacent to promoter sequences that enhance binding of the transcription complex. Between genes are often long sections of DNA that goes unexpressed. Although originally labelled and often still called "junk" DNA, it is now understood that these regions of non-coding DNA, called introns, play a role in regulating gene expression and structuring DNA molecules.

DNA molecules are also subject to modification. These modifications are generally small functional groups added to the bases (examples are illustrated in figure 1.1.1), often by regulatory enzymes, and are known to have effects on gene expression and replication, playing a roll in organism development and cell differentiation[4][5]. Some modifications have been implicated in diseases such as cancer, and some patterns of DNA modification have been demonstrated to be inherited from mother to daughter cell[6]. DNA can also be modified randomly through the process of mutation, with bases being inserted, deleted, replaced or modified. Mutations my be deleterious, causing cancer, genetic disease, or cell death. They may also have no significant impact, or even sometimes prove beneficial. Although every cell in an an organism typically has the same DNA[1], different cells have different modifications or mutations in their DNA, and modifications vary over the life of a cell.

1.1.2 DNA sequencing

DNA sequencing is the process of determining the sequence of bases on a strand of DNA. Sequencing is performed to study transcription and translation and the structure of DNA, to identify and study the protein that a gene encodes, to identify

[1]Some organisms, called chimeras, do consist of multiple populations of cells with different DNA sequences.

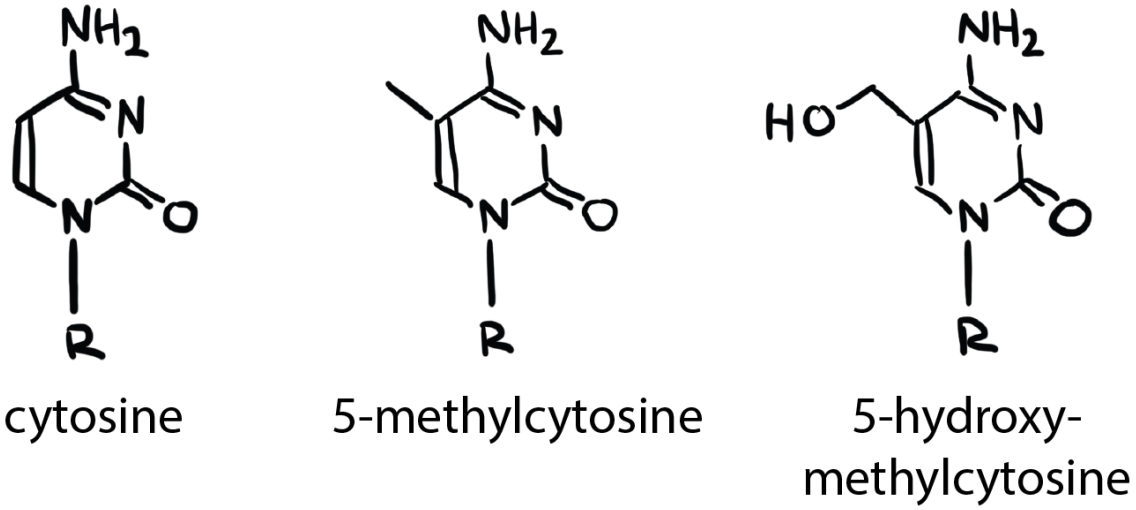

Figure 1.3: **Examples of epigenetic modifications.** Cytosine methylation and hydroxymethylation are commonly occurring epigenetic markers in eukaryotes serving to suppress gene expression. Typically these modifications occur on cytosines that have adjacent guanines on their 3' side, a motif called a CpG site.

organisms, to learn about an organism's ancestors and evolution, to facilitate medical care, or when using nucleic acids for bioengineering. Databases containing many sequences are used to study population genetics, the epidemiology of genetic disease, and the movement and history of populations.

DNA sequencing tasks fall broadly into two categories: de novo sequencing, where nothing at all is known about the strand to be sequenced; and reference sequencing, in which the sequencer is in possession of a "reference genome" that is presumed to be identical to the target sequence with the exception of a few individual base substitutions to be identified.

The early history of de novo DNA sequencing was dominated by Sanger sequencing [7]. In Sanger sequencing, a primer is attached to the target DNA molecule at the beginning of the region to be read. Four separate reactions are executed in the presence

of each of the four canonical nucleotides respectively. The nucleotide complementary to the next base after the ssDNA-dsDNA junction at the primer is incorporated to the primer strand by a polymerase. The other three nucleotides are not incorporated. These nucleotides are modified by the removal of the alcohol group from the deoxyribose, preventing further extension once they are incorporated. Of the products of these four reactions, one will be a single nucleotide longer than the others. The four products are separated by molecular weight using gel electrophoresis, and the reaction with the greater weight is identified as the one resulting in the added base.

Sanger sequencing has been steadily improved since its commercialization in 1977, incorporating the use of fluorescently labeled nucleotides and microfluidic automation. These developments have improved accuracy, cost, and read time. Sanger sequencing is a high-fidelity technique capable of reads as long as 1000 base pairs, and was the main technology used over most of the duration of the Human Genome Project.

From the mid-1990s onwards, the cost per base of DNA sequencing plummeted as a series of high-throughput "next-generation" sequencing platforms became available [8][9]. These methods parallelize the DNA sequencing process by creating numerous tagged or spatially separated colonies of cloned DNA and reading the sequence of each colony simultaneously. The methods presently used for most scientific and commercial sequencing applications are "shotgun" methods, such as Illumina sequencing. In these methods, many overlapping short reads are then combined to reconstruct a long target sequence. Next-generation sequencing has caused a precipitous drop in the cost and time required for sequencing a genome since 2007-2008 (figure 1.1.2)[10].

1.1.3 Challenges in the future of sequencing

Much of life's genetic code is comprised of repetitive DNA sequences[11]. Around two thirds of the human genome consists of repeat sequences, and in some organisms they

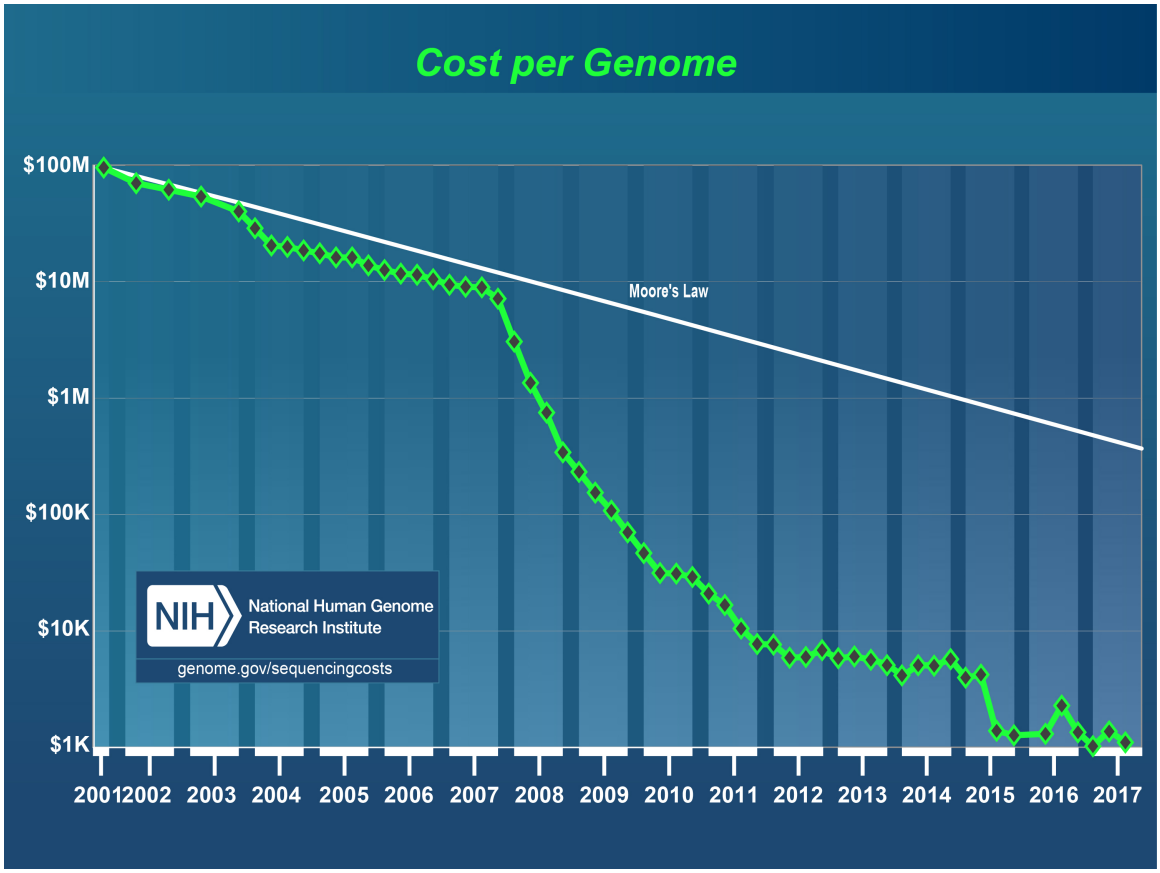

Figure 1.4: **Cost of sequencing a human genome over time.** The first fully sequenced genome was completed in 2001 at a cost of approximately $100 million US. Around 2007-2008, the cost per genome dropped precipitously, making it a reasonable goal first for research programs with large budgets, and now is cheap enough to be performed regularly by biological, genomic or medical scientists or in the clinic. Numbers are approximate, however, because of the difficulty in defining precisely what is meant by a fully seqeuenced human genome. For example, non-coding highly repetitive portions of the genome that serve a predominantly structural purpose have yet to be truly sequenced at all. Figure provided by the National Human Genome Research Institute[10].

comprise over 80% of the genetic code[12]. These include tandem repeats consisting of many adjacent repeats of 2 or more bases, with total lengths of up to a few million base pairs.

These tandem repeats challenge existing sequencing platforms that function by reconstructing overlapping short reads, because they cannot uniquely be re-assembled if the read length is shorter than the repeat region (figure 1.1.3). The failure to correctly assemble these regions is responsible for most of the remaining gaps in the sequencing of otherwise completed eukaryotic genomes. Despite the fact that these regions often do not code for proteins, and are highly variable (usually in number of repeats) compared to coding regions of the genome. For this reason, they are very useful in identifying lineage over short time scales at the individual level within a population. Being structurally such a large part of the genome, their identification and sequencing is also critical for understanding the way genetic material is regulated by structure.

Existing sequencing technologies are also not easily equipped to sequence nucleic acid modifications, because the PCR used to clone the DNA is either disrupted by or removes these modifications. Methods do exist for the detection of DNA modifications[13], but they must be tailored to the specific modification in question, and are often subject to further limitations in sequence length or content.

As more is revealed about how genome structure and DNA modifications function to regulate, protect, or harm genetic code, and as these discoveries are leveraged in medical science, a demand has grown for sequencing technologies capable of reading through long repeat regions and identifying base modifications in individual strands of unprocessed DNA. Nanopore sequencing, as a physical technique with no hard upper limit on read length, was developed as a tool to overcome these limitations in traditional sequencing[14][15].

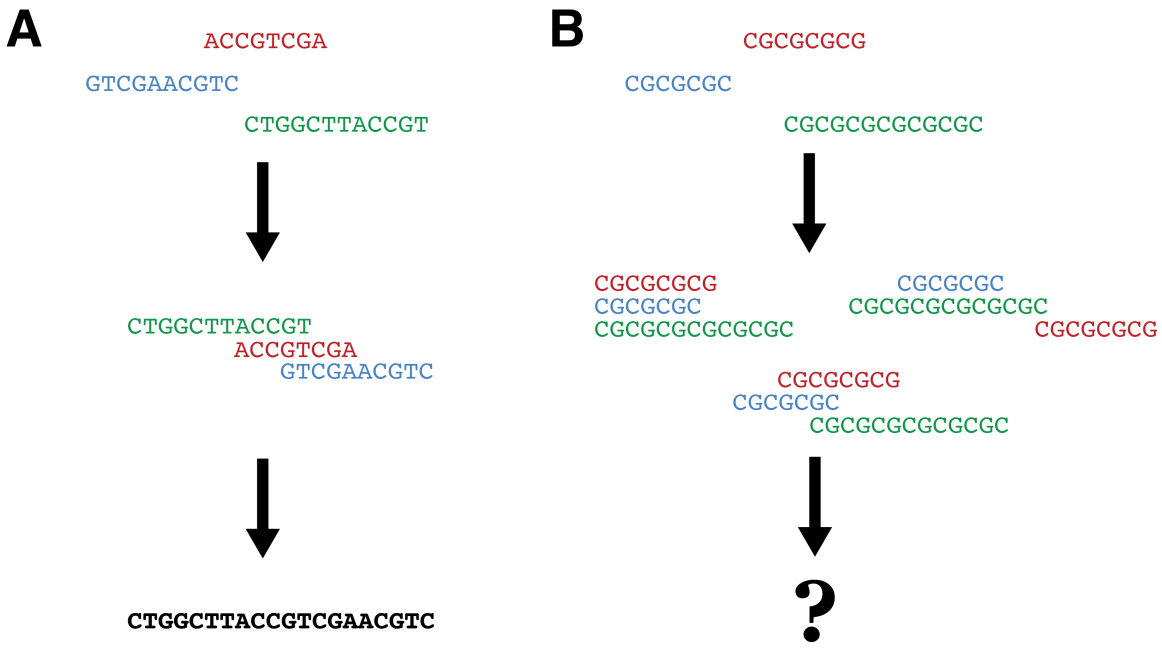

Figure 1.5: **Short read assembly.** (a) A number of short reads may be assembled into a larger sequence wtih sufficient coverage. (b) Repetitive DNA sequences introduce ambiguity into the sequence reconstruction process.

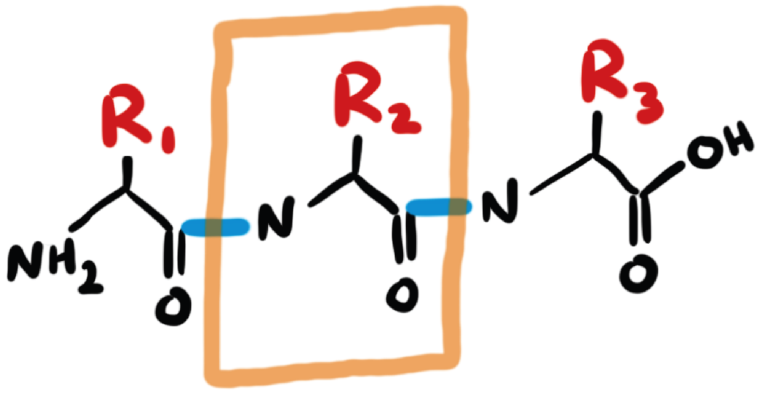

Figure 1.6: **The chemical structure of an amino acid chain.** Peptide bonds (blue) link together amino acid monomers (highlighted in orange), each of which has a *side chain*, also called a *residue* or *functional group*, marked R_1 through R_3 in red.

1.2 Proteins

As discussed in the previous section, the principal function of nucleic acids is to code for the proteins that carry out functions within an organism.

1.2.1 Protein structure

These proteins are composed of chains of amino acid monomers linked by peptide bonds (figure 1.2.1)[1]. Similarly to DNA, each of the twenty amino acid monomers encoded in the human genome has the same structure comprising its backbone. This backbone is directional, with one end terminating in an amine group (the "N-terminus") and the other terminating in a carboxylic acid (the "C-terminus"). The monomers are differentiated by their side chain, each of which has different physical properties. These side chains vary in physical size, charge, whether they are polar or non-polar, and their pH.

The amino acids of which a protein consists ultimately govern its structure and

function. A protein will adopt a confromation that at a local minimum of free energy. It does so by burying hydrophobic (nonpolar) side chains inside its volume and exposing hydrophilic (polar) side chains on its exterior, placing positively charged side chains near negatively charged side chains, and keeping bond angles and lengths close to their equilibrium values.

The sequence of amino acids in a protein is called the protein's primary structure. This sequence is what is encoded in the codons in a gene. Protein sequences are typically written from N-terminus to C-terminus. These sequences are of special interest to biologists seeking to understand the function or evolutionary history of a protein, since stable mutations to an species' genome are likely to be those that exchange similar amino acids[16]. For example, leucine and isoleucine are very similar in all their physical properties, and it is unlikely that a mutation that exchanges them at a particular site would be deleterious to that protein's function. Thus, these mutations are not selected against, and two proteins differing only by this mutation almost certainly are from the same lineage and have the same structure and function.

Certain patterns within the primary structure lead to a few common motifs in what is called the protein's secondary structure: α helices and β sheets (figure 1.2.1). These motifs are stiff compared to unstructured amino acids, and are generally the initial stage of broader structure formation in proteins. Because they rely on bonds between nearby amino acids in the primary sequence, they often begin to form before a nascent protein is fully extruded from the ribosome that is synthesizing it.

Tertiary structure is the larger scale structure giving a protein its overall shape. Secondary structure is preserved, but the flexible, less structure portions of the amino acid chain between the stiff α helices and β sheets bend to assemble a three-dimensional structure. Continuous sections of the amino acid sequence that form separately functional subunits of the tertiary structure are called "domains."

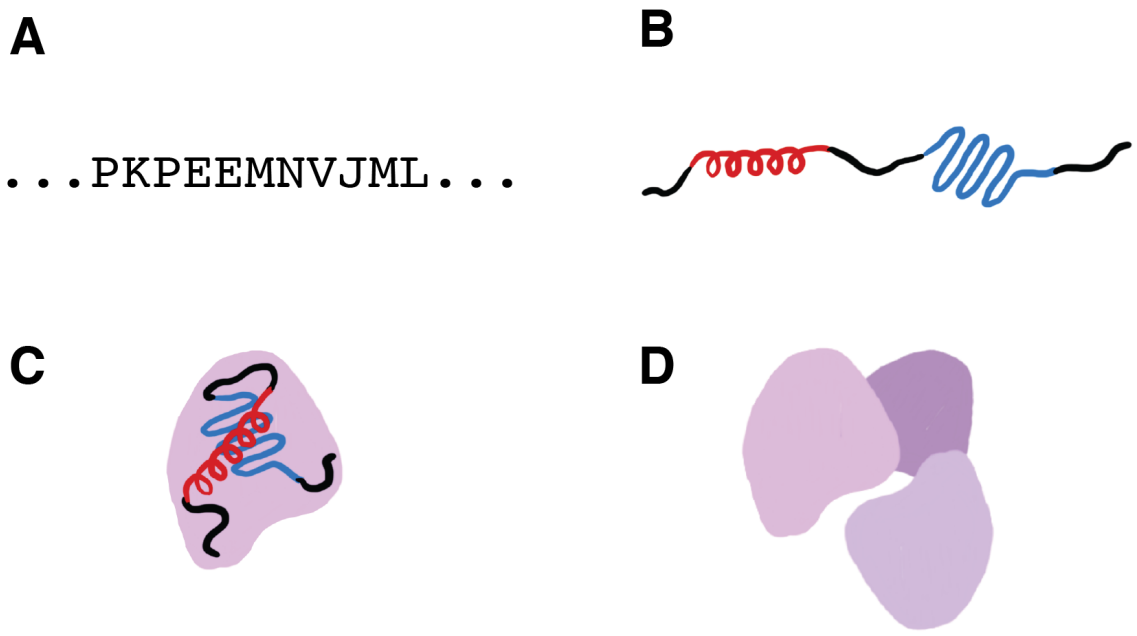

Figure 1.7: **Scales of protein structure.** (a) Primary structure consists of the amino acid sequence of the protein. (b) Secondary structure consists of helices and sheets formed by interactions of nearby bases. (c) Tertiary structure describes the folding of the protein as a result of longer-range interactions. (d) Quaternary structure describes the way multiple protein subunits come together to form protein complexes.

Quaternary structure is a yet higher level of organization, in which multiple protein subunits consisting of separate amino acid chains not connected by peptide bonds come together to form a complex. These can be multiple identical subunits, for example in the case of the membrane protein nanopore *Mycobacterium smegmatis* porin A, which consists of eight symmetrically arranged subunits. Different proteins can also come together, as in the GINS protein complex, consisting of four different subunits each with a different function.

1.2.2 Protein function

Protein structures are extremely diverse, allowing this fundamental design to produce molecules acting in many different roles within an organism. Two of these roles, that of an enzyme which catalyzes a chemical reaction, and that of a protein that associates with a biological membrane, are of particular interest for scientists studying nanopores.

Catalytic proteins

Life extracts energy from its environment to create order locally. Critical to this process is the ability to convert energy between and into chemical bonds in a coherent manner. This is enabled by proteins which catalyze chemical reactions, lowering the activation energy of certain chemical processes so they happen at a much greater rate, and favoring the formation of the products of these processes. A particular subcategory of catalytic enzymes are those which undergo directed "walking" motion along a substrate such as a nucleic acid,

The behavior of a catalytic protein operating on a single substrate can be described in broad generality through the chemical equation

$$E + S \xrightarrow[\text{binding}]{\text{substrate}} E \cdot S \xrightarrow[\text{and release}]{\text{catalysis}} E + P, \tag{1.1}$$

where E is the enzyme, S is the substrate molecule, and P is the product of the catalysis. Here, "+" denotes the separate presence of two molecules, and "$\cdot$" denotes a bound state of two molecules. The first step is the binding of the substrate to the enzyme, and the second step is the catalysis and release of the catalyzed product.

Proteins which catalyze chemical bond formation or breakage have an active site which binds directly to the substrate molecule(s). The affinity of an active site for its substrate (in other words, the rate constant associated with substrate binding) depends on how well the substrate "fits" into the active site. The substrate must be the right shape and size to both access and then fit into the active site. A high-affinity binding site will match exposed polar side chains to polar parts of the substrate, non-polar side chains to non-polar parts, and charged side chains to oppositely charged parts.

This principle is used in pharmacology to design drugs which target the behavior of certain proteins by binding strongly to their active site, possibly competitively inhibiting them from binding them to a substrate and decreasing the enzyme's activity, or alternatively functioning as a strongly binding cofactor that increases the enzyme's activity[17].

Membrane proteins

Cell membranes do not simply contain the contents of a cell. They are studded with a wide variety of proteins serving a variety of functions. Some of these proteins contribute to the shape and stability of the bilayer. Others function as sensors or beacons, allowing a cell to commuicate with other cells through the molecules in the intercellular medium. Still others function as gate keepers, selectively admitting

certain chemicals. Membrane proteins are also found in lipid bilayers other than the cell's outer membrane. For example, the membrane protein ATP synthase in mitochondrial membranes is the means by which cellular respiration extracts energy from the proton gradient across these membranes.

ATP synthase is an example of a subcategory of membrane proteins called ion channels. Ion channels' primary function is evident from the name: they are proteins that associate with membranes to form stable openings which allow small molecules through, while being too small for macromolecules to pass through (figure 1.2.2). These proteins are used to maintain physiologically optimal salt concentrations within cells, function as receptors that open and allow an ion species through only in response to certain external stimuli, establish electric potentials across membranes, and are even used as weapons to perforate the membranes of foreign cells or competing species. When ion channels are used in vitro for biotechnology applications, they are often referred to as biological nanopores.

These complex behaviors are often enabled by a feedback-based "gating" mechanism, in which certain stimuli such as electric potentials, heat, and chemical concentrations or gradients trigger a conformational change in the ion channel that causes it to either close or open. This is the means by which, for example, neurons build up electric potential in the form of ion species concentrations, and then release it or "fire" at systematically determined rates.

Other protein functions

The categories of protein discussed above are of particular relevance to nanopore experiments, but proteins perform almost every complex function within an organism. These include structural proteins that stabilize, organize, and transport at all scales within an organism; signalling proteins that send or receive messages by modifying

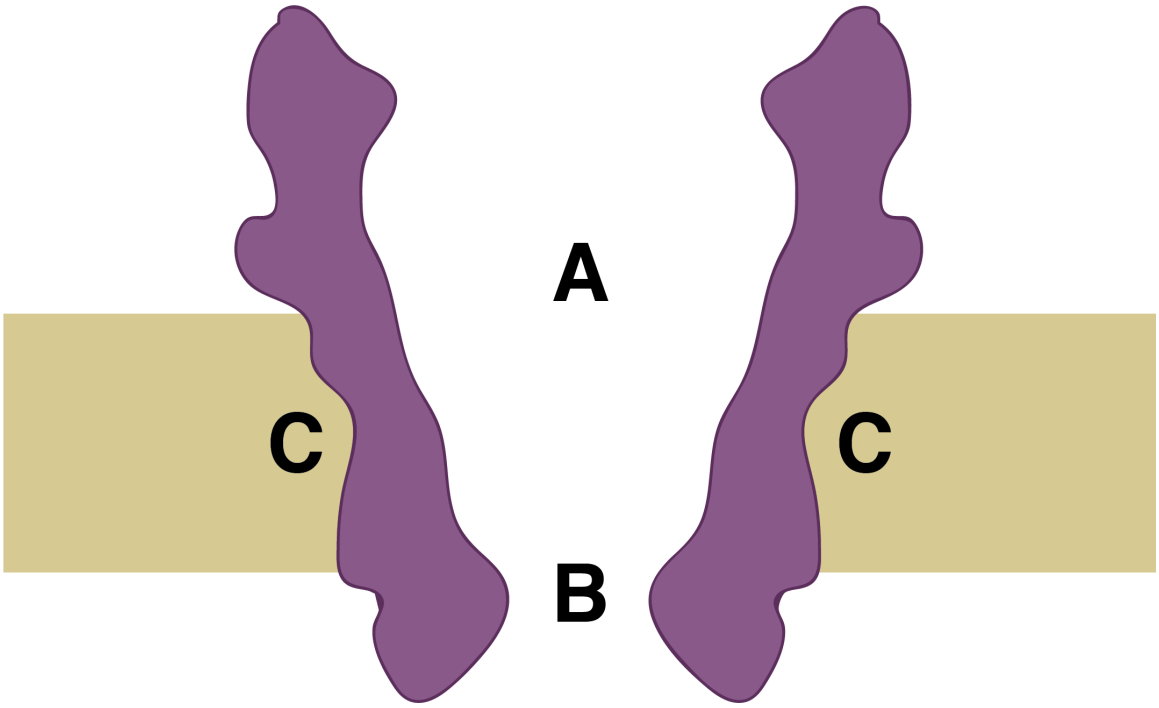

Figure 1.8: **Structure of an ion channel.** The ion channel itself typically consists of multiple protein subunits assembled into a tunnel shape (cross section in purple). The vestibule (a) or the constriction (b) may have charged surfaces or a sufficiently small diameter to selectively exclude or permit different species of molecules. Part of the exterior of the ion channel (c) is hydrophobic, allowing it to assemble within a phospholipid bilayer (tan).

other proteins to activate or deactivate them; shepherd proteins that facilitate important reactions or processes such as protein folding without directly catalyzing bond formation or breakage; fluorescent proteins and other light-sensitive molecules that provide coloration and protection from or use of light energy- providing a comprehensive list would be a thesis in itself.

1.2.3 Protein engineering

Although protein function is often rich and difficult to predict from first principles given only the primary structure, proteins are somewhat robust to small changes in protein sequence. If a protein is well-understood, it is possible to modify ("mutate") it by changing the base sequence the gene that codes for it. Changing this code means the translated protein will have a different amino acid sequence. This mutated gene is inserted ("transfected") into the genome of a host organism used for protein synthesis (a "vector").

The effect of a mutation depends strongly upon exactly which mutations are made. Introducing a large charged residue to an active site that previously lacked any charge might prevent the protein from having any of its original functionality at all. On the other hand, proteins are unlikely to change function if a mutation is made which does not significantly affect its active site(s) or secondary structure. For example, changing a polar side chain on a disordered strand on the exterior of a protein and far from an active site is unlikely to have any effect on protein activity.

It is common to also simply append genes together, so that the C- or N-terminus of the original protein is linked to the N- or C- terminus of another, often with a disordered chain inserted between the two proteins, linking them less disruptively. For example, a standard method in fluorescence microscopy is to use genetic engineering to attach a fluorescent protein such as GFP to a protein of interest. This makes it so

that every single copy of the protein in question will be tethered to a GFP molecule, and the concentration of the protein can be tracked using a fluorescence microscope. Compared to chemical labeling methods, this technique is minimally invasive and requires no foreign material to be introduced to cells after the transfection of the original cell line, and presuming that the GFP mutation does not disrupt the protein it modified, it generally has few off-target effects. It is also broadly applicable to many proteins using similar steps regardless of exactly what the target of observation might be.

Mutation of proteins is a key tool for understanding how proteins work. Using X-ray crystallography, cryo-electron microscopy, or nuclear-magnetic resonance images of the protein's structure, alongside known data about its functionality, a biologist can form hypotheses about the mechanisms by which it folds, performs its function, or is affected by other molecules. These hypotheses can be tested by making mutations expected to disrupt, strengthen, or otherwise affect these mechanisms. If the mutant protein behaves as expected, this is taken as evidence that the protein functions using the hypothesized mechanism.

1.3 Biological membranes

Cells use biological membranes for containment, transport, and the maintenance of homeostasis[1]. These membranes are also frequently used in bioengineering to establish separations of small volumes or conduct reactions in picoliter or smaller volumes.

A typical biological membrane consists of a phospholipid bilayer. A mixture of phospholipids in a polar solvent such as water will minimize free energy by taking a phase in which the hydrophilic parts are buried away from the solvent and the heads are exposed. This can happen by the formation of small single-layer lipid spheres, but more commonly lipids will form bilayer sheets with the tails pointing inward and the

heads forming the outward pointing surfaces. These sheets tolerate some degree of surface bending, and will typically form spheroidal bubbles in water so no lipids are exposed to the polar solvent. Phospholipid membranes are semi-permeable to small, uncharged polar molecules such as water and to non-polar molecules. This makes it possible to establish chemical or electric potentials across membranes, which are functionally capacitors.

In vivo, biological membranes are often composed of mixtures of lipids and are studded with numerous membrane proteins that serve to stabilize their shape. They also have a rich statistial physics, and form numerous ordered phases with different 2D geometries. This physics is simplified somewhat in vitro, where bioengineers often select a single species of phospholipid, and proteins not relevant to the system being studied are not present in the membrane. In bioengineering applications, unlike in vivo, membranes are sometimes used in geometries other than spheroid. Specifically, they are used to span small holes and separate volumes chemically and electrically, as in nanopore experiments; they are also used to coat surfaces, making them hydrophobic and enabling the adsorption of hydrophobic molecules of interest that a scientist may want to associate with a surface.

1.4 Electrolytic buffers

The cellular medium is full of ions that stabilize its pH, increase its conductivity by functioning as charge carriers, and are used by enzymes as cofactors to help bind their substrates. The properties of a medium and its effect on any biological process depends on the physical properties of the ions it contains: their effective size, their charge, their mobility in the medium, and their concentration[18].

The primary way that ion concentration affects biological systems is through a phenomenon called electric field screening. When a charge, such as a phosphate group

on a DNA molecule, a charged amino acid side chain, or an ion itself, is present in a medium with mobile and oppositely charged ions, those mobile ions will condense around the charge (figure **??**. Additionally, if surrounded by a polar solvent such as water, the polarization of this solvent with be affected by the presence of the charge. Both of these effects cause the strength of charge- or dipole-based bonds to decrease as salt concentration increases. It also causes electric field strength to fall off exponentially rather than as $1/r^2$, so the electric potential from a point charge is given by

$$\Phi = \frac{1}{4\pi\epsilon}\frac{Q}{r}e^{-r/\lambda_D} \tag{1.2}$$

where ϵ is the dielectric constant of the medium with no ions present, Q is the charge of the point charge, r is the distance from the point charge, and λ_D is a length scale called the Debye length,

$$\lambda_D = \sqrt{\frac{\epsilon k_B T}{\sum_i n_i q_i^2}}. \tag{1.3}$$

Here $k_B T$ is the Boltzmann constant times the temperature in Kelvin, and the sum in the denominator is over ion species. n_i is the concentration of ion i, and q_i is the electric charge of ion i. Debye lengths in physiological ion concentrations are typically on the order of 1 nm, the size of just a few water molecules. This means that long-range electrostatic attraction even between different domains of a protein is negligible in electrolytic media.

At very high salt ocncentrations, the disruption of electrostatic bonds can significantly affect the structure of biomolecules, causing some proteins to denature and affecting the biophysics of nucleic acids by reducing their degree of self-repulsion. This is discussed further in §4.2.

Part II

NANOPORE EXPERIMENTS

This part contains an introductory guide to nanopore experiments, the physics behind them, and their execution. My hope is that it reads as a methods section that together with some supplementary material will enable the reader to collect initial data for either single-molecule or sequencing experiments using the nanopore MspA. Detail on understanding and analyzing this data is provided in part IV.

Chapter 2
INTRODUCTION TO NANOPORES

Within this chapter is an introduction to nanopore experiments, from their inception through their present incarnations. The applications of nanopore experiments that have been studied at the University of Washington, in DNA sequencing and in single molecule biophysics, are given special attention.

2.0.1 Experimental basics

A schematic of a nanopore experiment is illustrated in figure 2.0.1. A thin barrier or membrane separates two volumes (termed the *trans* and *cis* volumes) of an electrolytic solution. Free ions are capable of carrying an electrical current through this solution. A single hole, or nanopore, on the order of nanometers across connects the two wells, allowing small ions and solution molecules to pass through.

Electrodes are placed into the *trans* and *cis* volumes and used to establish an electric potential difference across the membrane. This causes an ion current to flow through the nanopore, which is measured by the experimenter.

Because the only conducting pathway between the electrodes is the nanopore, variations in the measured ion current are dominated by physics in or very close to the nanopore. For example, if a large molecule obstructs the pore, it prevents ions from translocating through, and the measured ion current is reduced. Typically, the name *cis* is applied to the side of the membrane that a translocating particle starts on, and the *trans* side is the side that the particle travels to through the nanopore.

The precise degree to which the ion current is blocked varies depending on the

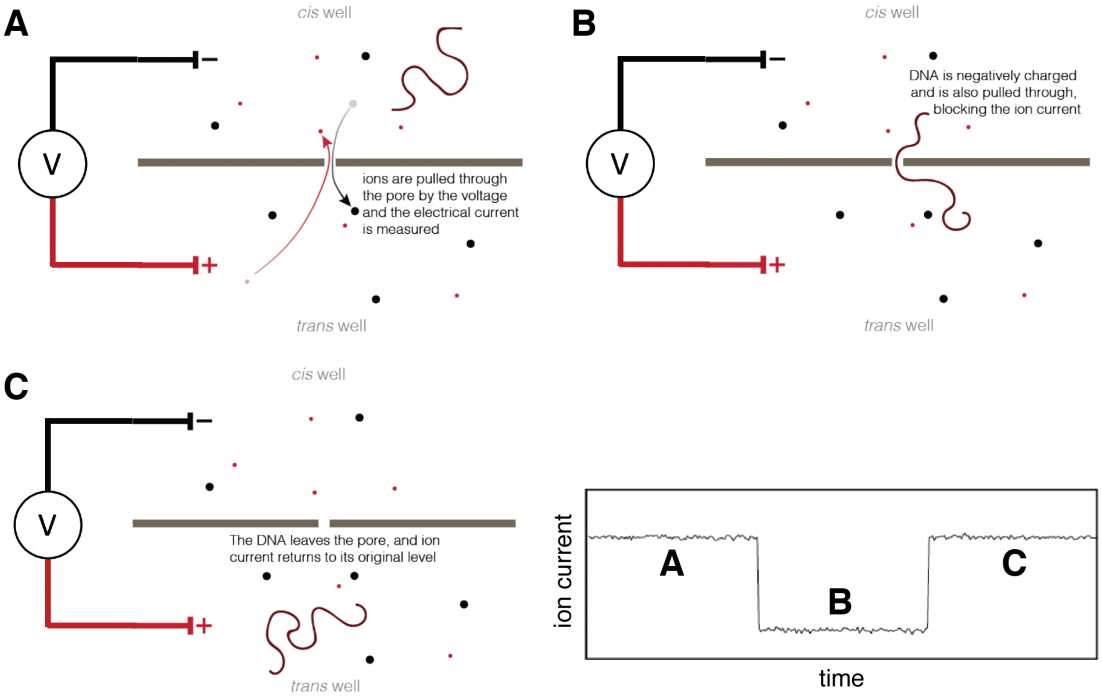

Figure 2.1: **Schematic of nanopore experiment principles.** (a) Ion current is driven through the nanopore (purple) by an applied voltage and measured. (b) When the nanopore is blocked by a large molecule, the ion current drops. (c) After the molecule leaves the pore, the ion current returns to the original value.

nature of the blockage: different molecules or conformations and orientations thereof may obstruct the pore differently. Different blockages may also be characterized by other criteria, such as how long they dwell in the nanopore before fully translocating or wandering back out from whence they came (if ever).

Nanopores provide a method for isolating single molecules, because the sensitivity of the experiment is only to physics close to the nanopore. Through electrophoretic or electro-osmotic force, they are capable of applying forces and manipulating these molecules. Finally, they provide a method for analyzing the molecules, as the measured ion current and its dynamics may correspond to the identity or behavior of the analyte.

2.0.2 Solid state nanopores

Solid state nanopores consist of holes drilled in thin solid membranes (figure 2.0.2). These membranes are most commonly composed of silicon nitride (SiN) or of 2D materials such as graphene or molybdenum disulfide (MbS_2). Membranes are typically established through epitaxial growth of a thin layer of membrane material on a substrate, followed by etching away of the substrate.

Compared to biological nanpopores, solid state pores are able to tolerate much higher voltages, often over a volt. This is because solid membranes are far more robust than lipid bilayers to electric fields.

Solid state pores presently suffer from several drawbacks when it comes to DNA sequencing applications. The foremost is that the high energy processes used to drill nanopores do not create atomistically reproducible pores. Any two pores might have different geometries. In a system intended for using a reproducible ion current signal to identify analytes such as individual DNA bases that differ by only a few atoms, having a few atoms out of place in the sensor can be catastrophic. Solid state

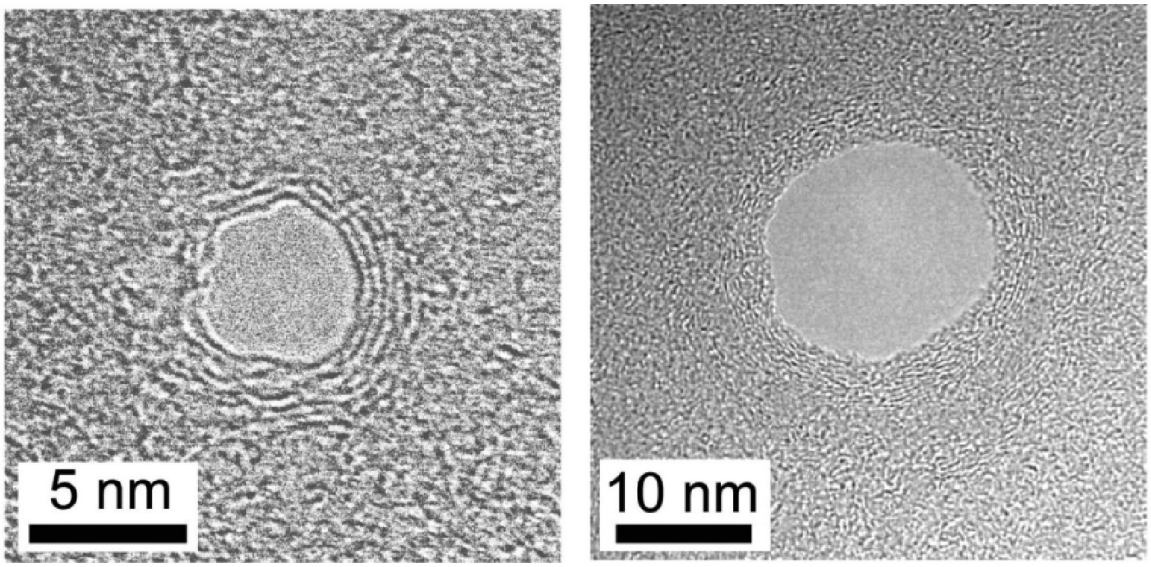

Figure 2.2: **A TEM image of a graphene solid-state nanopore.** Pore and image produced by the Dekker laboratory at TU Delft. Image credit: G. Schneider *et al*[19].

pores also tend to be unstable at their typical operating voltages, and slowly change shape over time. This means that even a perfectly uniform set of pores might differ in geometry and response to voltage after a few hours of operation. A less critical limitation, which may be addressed in the near future as manufacturing methods improve, include the difficulty, expense and time required to producing high quality pores.

Larger pores, 10+ nm across, are much easier to produce than 1 nm pores within an acceptable degree of reproducibility. These are often used for applications other than sequencing or distinguishing very similar analytes, such as in Coulter counters, identification of proteins, or as general-purpose single molecule trapping tools.

2.0.3 Biological nanopores

Biological nanopores consist of protein ion channels, or sometimes other biological materials[1], creating a path for ions to travel through a membrane. Most often, the membrane is a lipid bilayer or other hydrophobic polymer membrane. Occasionally, biological materials are combined with solid state materials in hybrid-biological-solid-state nanopore systems, such as in attempts to isolate an ion channel in a similarly-sized solid state pore[22], or in the insertion of solid state materials such as carbon nanotubes into biological lipid membranes[23].

Ion channels in lipid or polymer membranes have been thus far the most successful biological materials employed as nanopores for sequencing and single molecule analysis. There are a large variety of ion channels useful for trapping single molecules, but few have been shown to be suited for DNA sequencing. To explore why this is the case, it is helpful to examine two of the most commonly used ion channels in nanopore experiments, MspA and α-hemolysin.

α-hemolysin

Early experiments seeking to demonstrate nanopore sequencing using biological nanopores often used α-hemolysin (figure 2.0.3(a)). α-hemolysin is an ion channel from *Staphylococcus aureus* that is used by the bacteria to perforate the cell walls of competing organisms, causing a failure of chemical homeostasis and ultimately cell death. It is composed of seven identical protein subunits which bind to a membrane and then assemble to create an ion channel.

α-hemolysin was used in early nanopore sequencing experiments because it is a well-understood model system in electrophysiology, and the width of its channel is similar to the size of ssDNA. Additionally, an important early sequencing proof of

[1]Such as DNA origami[20][21].

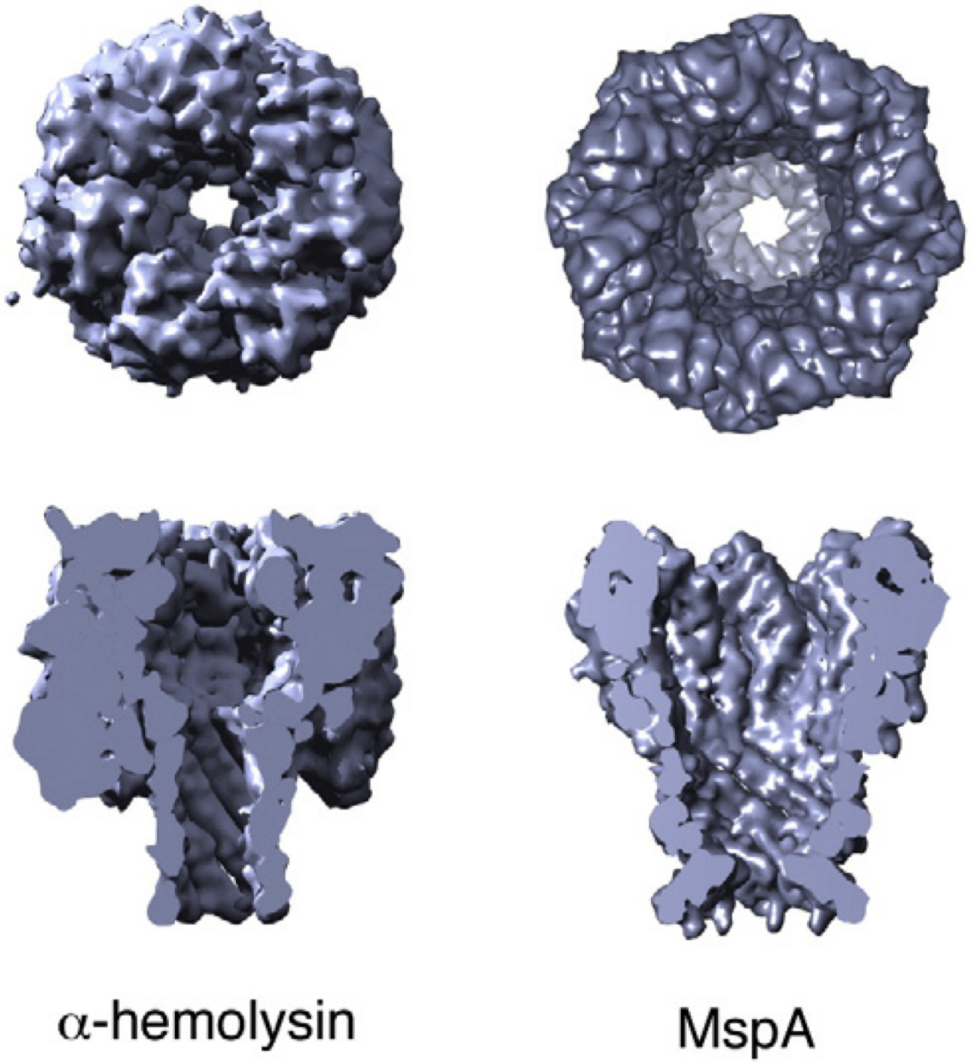

Figure 2.3: **Nanopores α-hemolysin and MspA.** Note the dimensions of the constriction of each pore. MspA's short constriction makes it ideal for analyzing only about 2-4 DNA bases at a time[24], while α-hemolysin's long tunnel-like constriction makes it sensitive to a span of approximately 10-12 DNA bases simultaneously[25]. Image credit: J. Kasianowicz *et al*[14].

concept was carried out on α-hemolysin showing that different homopolymer DNA strands could be identified based on the depth of the ion current blockage they caused while inside the pore[26], as well as being used in some of the first enzyme-controlled DNA nanopore experiments[27] (more on this in §2.1).

Unfortunately, although it is well-suited for Coulter-counter-like applications, α-hemolysin is not suitable for single base recognition or the detection of single-nucleotide steps. This is because of the geometry of the pore: its constriction consists of a tunnel the length of approximately ten ssDNA nucleotides. This means that the ion current is affected by at least ten nucleotides simultaneously. The limited space of measurable ion currents does not allow for specific identification of each of the 4^10 different 10-base DNA subsequences by observing the ion currents. Nevertheless, α-hemolysin's status as a model system has made it a persistently useful tool in nanopore physics.

MspA

The limitations caused by the geometry of α-hemolysin prompted a search for a nanopore better suited for single-base recognition. One such pore is *mycobacterium smegmatis* porin A (figure 2.0.3(b)), typically referred to as MspA. MspA is composed of eight identical protein subunits, and is highly stable: it still folds and self-assembles correctly in very high molarity salt solutions and is not denatured by extreme pH or high temperatures[28].

MspA's constriction is approximately 1.2 nanometers in diameter, just large enough for one strand of ssDNA. Additionally, unlike α-hemolysin, this constriction is only a few atoms tall, short enough that it can contain only a few ssDNA nucleotides at once. This makes it especially suited to nanopore sequencing.

Despite its advantageous geometry, wild-type MspA is unsuitable for nanopore sequencing because its constriction is covered in negatively charged amino acids. These

repel negatively charged DNA, making the translocation of DNA through the pore very unlikely. However, these negative charges can be engineered out of the protein, enabling DNA translocation.

2.1 Nanopore sequencing

Nanopores have the ability to trap individual nucleic acid molecules, and by monitoring the changes in ion current when the nucleic acid is trapped, inferences can be made about the its base sequence. These capabilities have prompted a sustained effort to sequence nucleic acids using nanopores.

In the standard biological nanopore experiment, a lipid membrane is established separating two volumes of a salt solution. A single protein nanopore is inserted into the membrane, and a voltage is applied across the membrane causing an ion current to flow through the pore.

DNA, being negatively charged, is drawn electrophoretically into the pore. While the DNA is trapped, it occludes the constriction of the nanopore, reducing the ion current. The depth of the ion current blockage depends primarily on the bases in the constriction. The principal data analysis problem in nanopore sequencing is to determine the base sequence of the trapped DNA by examining the ion current through the pore.

If the DNA is allowed to translocate freely through the pore under the electrophoretic force, it travels through far too quickly[2] (figure 2.1). Therefore, in order to sequence DNA, some method is needed to slow the DNA's translocation.

Many such methods have been attempted. These include, for example, methods of chemically "expanding" the DNA into a much large polymer[29], or interruption of

[2]At common laboratory conditions, the translocation is approximately 1 base/μs. This is too fast to resolve with the bandwidth of a typical nanopore experiment.

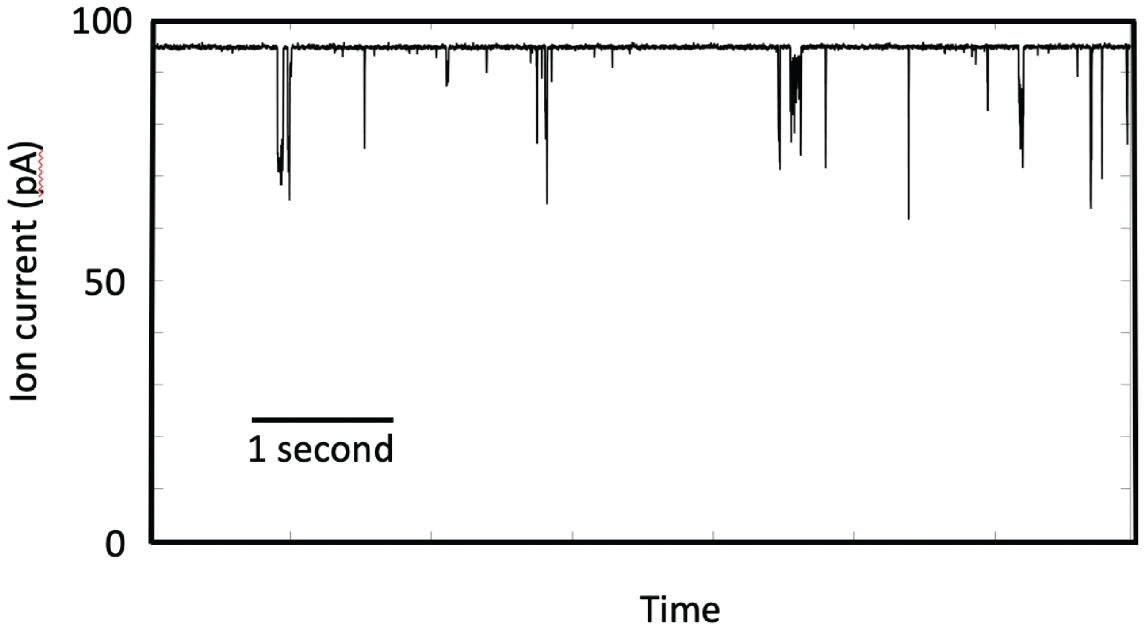

Figure 2.4: **Ion current recording of a nanopore in the presence of freely translocating DNA.** DNA's biased-random-walk translocation through the pore is very fast, typically on the order of 10^6 nucleotides per second for applied voltages on the order of 100 mV and pore diameters on the order of 1 nm. Since sampling frequencies for electrophysiology experiments are typically an order of magnitude less than this, these translocations result in short-duration blockages which have little or no resolvable internal structure, visible in this figure as transient downward spikes in the ion current.

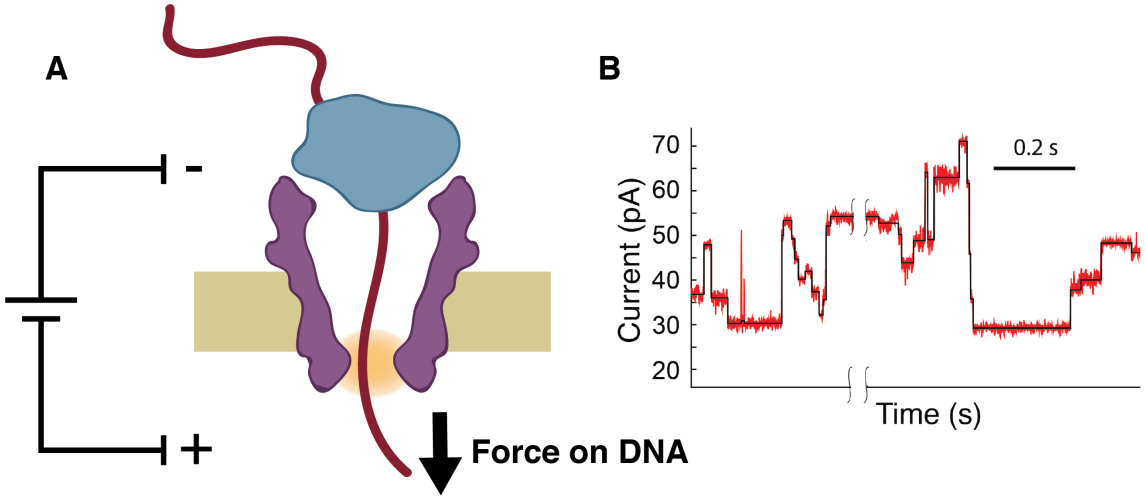

Figure 2.5: **Enzyme control of translocating DNA.** (a) A DNA-translocating enzyme such as a polymerase or helicase, too big to translocate through the pore, binds to the DNA and serves as an anchor arresting its translocation. As the enzyme steps along the DNA, it either releases the DNA or pulls it up through the pore. This slows the translocation of DNA and allows the ion current to be measured at each DNA position, enabling the discrimination of individual bases with a suitably chosen nanopore. (b) The resulting signal is a series of ion current levels, each corresponding to a different position of DNA in the nanopore. Each change in ion current corresponds to an enzyme step, and the sequence of ion currents depends on the bases on the DNA.

translocation by dsDNA duplexes which have to be separated for the DNA to continue through the pore[30].

Another method is to control the DNA with a DNA-translocating enzyme (figure 2.1)[27][31]. This is to date the only method for slowing DNA translocation that has been demonstrably useful for single-base resolution DNA sequencing.

2.1.1 Potential advantages of nanopore sequencing

The interest in nanopore sequencing has been spurred on by the needs of the next generation of genome scientists and the limitations of existing sequencing technologies.

Long read lengths

Existing DNA sequencing technologies are limited by their short read length. Sanger sequencing is limited to DNA fragments shorter than around 1000 base pairs. More modern high-throughput methods such as Illumina sequencing exchange read length for throughput, and typically limited to reads shorter than a few hundred base pairs. Short read lengths can make it difficult or impossible to reconstruct highly repetitive sequences, and may drastically increase the coverage required, offsetting the part of the benefit of a high throughput technology. Some sequencing platforms, such as SMRT sequencing, have demonstrated longer read lengths

One of the principle sources of interest in nanopore sequencing is its promise of long read lengths. There is no intrinsic limit to the read length in a nanopore sequencer. In practice, the read length is limited by the DNA-translocating enzyme falling off before reads are completed. The length of DNA which may be practically sequenced is also limited by DNA breakage occurring during the handling of the nucleic acid prior to sequencing. Strands of DNA of lengths greater than 100,000 base pairs must be handled very carefully.

Fortunately, both of these limitations may be circumvented by engineering: extremely processive enzymes have been selected and engineered, and methods for gently handling long DNA molecules have been developed. This has enabled reads up to nearly 1,000,000 base pairs, long enough to cover in a single read the longest repeat regions of the human genome.

Low overhead costs and robust design

A nanopore sequencer contains a minimal amount of equipment: some biological materials, a simple amplifier, and an analog-to-digital converter. This means that the overhead costs of a sequencer are in principle very low, possibly under $100 for

a device. This may be compared to the expensive optics required for sequencing by fluorescence- for example, Illumina sequencers typically cost in excess of $100,000.

Additionally, because of their simple construction, they can be built to be light and sturdy. This together with the minimal sample preparation typically required makes nanopore based sequencing devices feasible to use in the field, in handheld devices, or in extreme environments. The first DNA sequencing in outer space was performed on the international space station using a nanopore sequencer from Oxford Nanopore Technologies, and nanopore sequencers have been used by epidemiologists to rapidly test for and track the evolution of Ebola virus outbreaks in real time.

Physical, single molecule technique

One of the principal benefits of nanopore sequencing is that it is a physical technique, operating directly on a single molecule of genomic DNA. The molecule being sequenced does not need to be duplicated with error-prone PCR, and large sample sizes containing DNA from many cells are not required.

The single-molecule nature of the technique gives it some unique advantages. It makes possible the identification of low-concentration DNA sequences. For example, it is possible using nanopores to detect a trace amount of DNA from a particular virus present at low concentration in a sample. Additionally, a nanopore sequencer can detect cell-to-cell differences in DNA sequence or epigenetic information (discussed further in the following subsection).

The fact that the technique is purely physical means that new chemical reactions or materials do not necessarily need to be developed to study a polymer different from standard genomic DNA. Nanopores have been used to sequence RNA[32], DNA with modified bases[33][34][35], non canonical bases[36], and are even beginning to be used to study amino acid chains[37].

2.1.2 Present state of nanopore sequencing

Nanopore sequencing has developed from a complete hypothetical to an industrial sequencing technology over the course of around 15 years. Although nanopore sequencing has been limited by a low accuracy-per-base, making it unsuitable for high fidelity sequencing, it can be used to overcome a limitation of high fidelity sequencing platforms: short read lengths. This application was demonstrated in a 2014 paper by Andrew Laszlo proving the capability of nanopore sequencing for long reads of genomic DNA[24].

By aligning short, high-fidelity reads to long nanopore reads, we can determine with high confidence where they should be placed in the completed assembly, greatly simplifying the computation challenges and decreasing the error rate of short read reconstruction. A number of short, high-fidelity Illumina sequencing reads were aligned with confidence to a long nanopore read of the PhiX 174 viral genome. How precisely the Illumina reads overlap was constrained by this alignment to generate a fully assembled, long DNA sequence with high confidence.

Obviously, low-accuracy nanopore reads contain less information about the DNA being read than a high-accuracy sequencing read. However, even imperfect information can be used to uniquely identify organisms with a high degree of accuracy.

This fact was used in Laszlo's 2014 paper[24] as well as in later, more sophisticated experiments[3] to perform species identification. In Laszlo 2014, 250 bp sections of nanopore reads of the PhiX 174 viral genome were aligned to each genome in a catalogue of 5287 viral genomes including PhiX 174, totalling 156 megabases of candidate DNA. The highest likelihood alignment was the alignment to the PhiX 174 DNA sequence in all reads, with confidence ¿99.9996%, demonstrating the utility of nanopore reads for species identification.

---

[3]Led by Matthew Noakes, unpublished at the time of the writing of this thesis.

As a purely physical DNA sequencing technique, nanopore sequencing can also be used to detect epigenetic modifications to DNA sequences[35]. In a 2013 paper[33] published alongside complementary results from Mark Akeson's lab at UCSC[34], we demonstrated the capability of MspA nanopores in distinguishing 5-methyl-cytosine and 5-hydroxymethyl-cytosine modifications from each other and from the unmodified strand at single-base resolution.

Each of the three variants was observed in the context of a CpG motif, where the modified cytosine is followed in the 3' direction by a guanine. Each of the three modifications was observed in sixteen different contexts of flanking nucleotides of the form N_1CpGN_2, where N_1 and N_2 may be any two nucleotides, to cover the different possible signals as a result of the multiple-base sensitivity (for more on the multiple-base sensitivity of the pore, see §4.3). The distinguishability of the three different signals in each context is illustrated in figure 2.1.2.

Using a Bayes classifier trained on a withheld subset of the methylation reads, we performed a blinded identification of the remaining reads. methylated CpG sites were detected at a true positive rate of 97.5 0.7% and a hydroxymethylated CpG sites were detected at a true-positive rate of 97.0 0.9%. The true-negative detection rate for unmethylated CpGs was 98.4 0.6%. The details of this classification are discussed in the supplement of Laszlo's 2013 paper[33].

A very similar experiment involves sequencing of synthetic DNA bases, also called *non-canonical bases* or *unnatural base pairs* (*UBPs*). A 2015 paper by Jonathan Craig[36] showed the sensitivity of MspA sequencing to dNaM and d5SICS (figure 2.1.2, a pair of UBPs that have been demonstrated to not only pair, but to be replicated *in vivo* as long as the host organism has access to the UBP nucleotide monomers[38].

Nanopore reads were obtained of six DNA strands containing each of the six

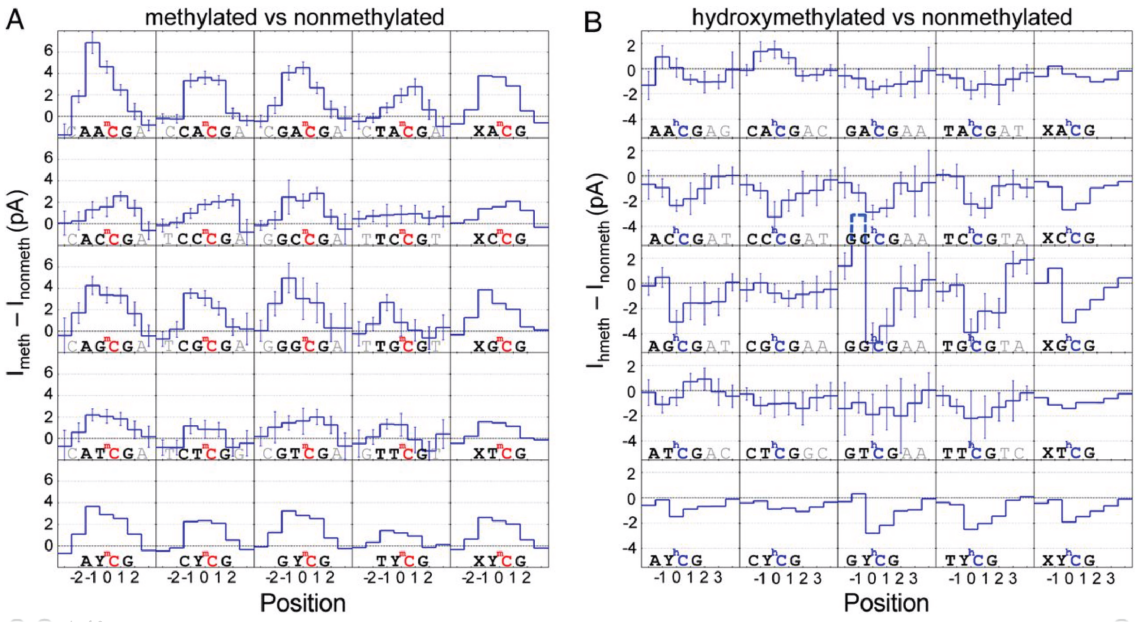

Figure 2.6: **Identification of epigenetic markers.** For each of the sixteen 4-mer contexts of CpG motifs, the differences between ion currents read using (a) methylated or (b) hydroxymethylated DNA and ion currents from unmethylated DNA. The six ion current levels centered on the CpG motif are plotted in each box, because these were most affected by the methylation. The fifth column in each matrix contains plots of ion currents averaged over the nucleotide in the 1st position, and the fifth row contains plots of ion currents averaged over the nucleotide in the 4th position. Figure adapted from Laszlo 2013[33].

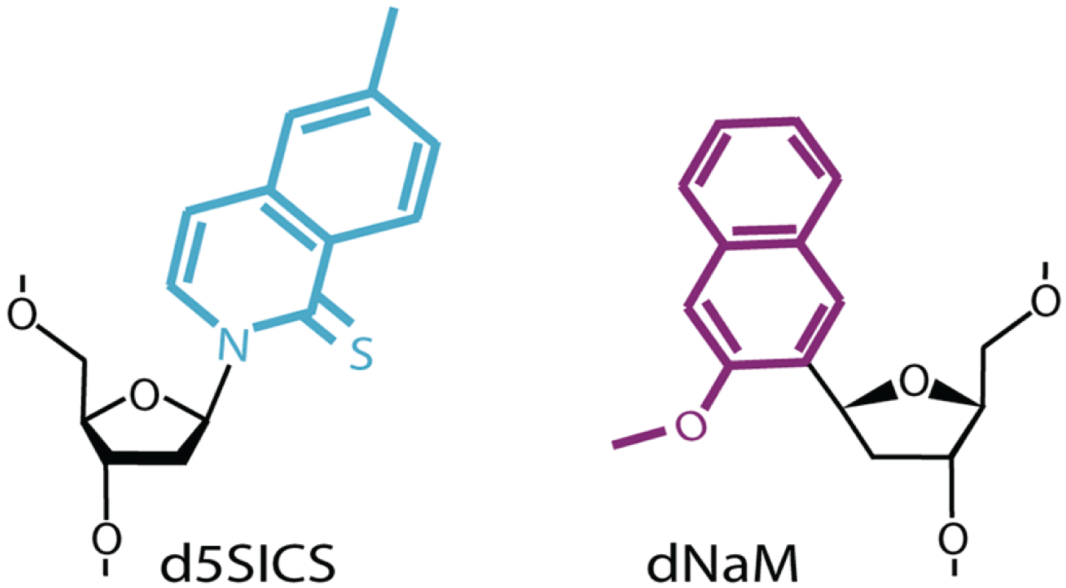

Figure 2.7: **Noncanonical DNA bases**. Chemical diagrams of dNaM and d5SICS, two noncanonical DNA bases that have been successively replicated *in vivo*, and that are identifiable through nanopore experiments.

bases in the expanded alphabet in the same surrounding sequence context. Significant differences between all six strands were observed, making reference sequencing identification of both UBPs possible.

2.2 SPRNT

Instead of using nanopores to sequence an unknown DNA strand using a well-understood enzyme to control its translocation, the experiment can be "inverted" in a sense, using a control strand of DNA with known sequence, and monitoring ion current to make inferences about the behavior of the protein controlling the translocation of the DNA. This experimental method has been termed SPRNT, an acronym for Single molecule Picometer Resolution Nanopore Tweezers.

The changing ion current signal then takes on new significance: every jump in ion current corresponds to a movement or change of conformation of the enzyme causing the DNA position within the pore to change. This data can be used to study the kinetics of a protein, by observing the dwell times of the enzyme in each state as it walks along DNA. Changes in kinetics can be observed as the experimenter changes the concentrations of substrates or cofactors, the temperature, the applied force to the DNA (by means of voltage across the pore), or any other number of experimental variables. This single-molecule spatiotemporal information can be determined with a resolution of approximately 900 $pm^2 \cdot ms$, about 30 picometers or 1/10 of a nucleotide at millisecond time scales.

Additionally, because the DNA bases in the pore at each observed time point can be determined from the ion current signal, the kinetic information about an enzyme can be correlated with the sequence of DNA in its active site, allowing for detailed analysis of sequence-dependent enzyme behavior.

Chapter 3
EXPERIMENTAL SETUP

In this chapter, I provide a guide to designing, setting up and running a nanopore experiment, focusing on the construction and operation of the physical structure built to contain the buffer, membrane, and nanopore.

3.1 Experimental apparatus

The engineering of the nanopore experiment is minimal. The experiment can be realized mostly with off-the-shelf components, with a few key exceptions in the armature used to contain the buffer and the micrometer-scale aperture used to support the lipid membrane. This section provides an overview of the design of the physical system used to run a nanopore experiment and its components. Additionally, it provides some detail on how to prepare, use, and troubleshoot all of these elements in the course of experimentation. A number of graphics are available depicting the construction of such a setup in Laszlo (2016) [39].

3.1.1 Design

The tabletop portion of the nanopore experiment must perform several tasks. It must support the lipid membrane and separate the *cis* and *trans* volumes of buffer. It must allow the two volumes to be accessible to the experimenter, so that materials may be added or removed, and addressable by electrodes. It must support a lipid membrane, and this membrane should also be accessible to the experimenter. Finally, it must isolate the experiment from ambient noise, temperature, and evaporation. This setup

is illustrated in the inset of figure **??**.

As a brief overview, a PTFE container is constructed with two separate wells of buffer, each in contact with a separate electrode. The two wells are connected by a buffer-filled tube, one end of which has been closed off except for a 20 micrometer hole. This device is fixed to a heavy table subject to minimal vibration. The entire setup is thermally isolated and temperature controlled using a Peltier element, and is electronically isolated from the environment using a Faraday cage.

PTFE armature

The container for the volumes of buffer needs to be made out of a non-reactive material that is durable, waterproof, and easy to clean. For these properties and also for the ease with which it may be machined, PTFE is an ideal choice.

The container is sometimes referred to as a *puck* because historically they were in shape of a short cylinder. Starting with a block-shaped PTFE blank, it is shaped with a bandsaw to have a "T" cross section. Two cylindrical wells with volumes of around 20-60 microliters are drilled in the top of this object with a drill press. The tops of these wells may be beveled to enable easier access for the tools used by the experimenter.

The choice of volume presents a tradeoff. A smaller well volume is more prone to evaporation and will suffer from a higher percent error in reagent concentration, and can be more difficult to work with manually. A larger well volume requires more materials in each experiment to achieve the correct concentrations, leading to more rapid consumption of potentially expensive reagents or samples.

For electrode access, a hole the width of the electrode insulation tubing is drilled in each end of the armature. Each of these holes is deep enough to connect to the well on that end of the device. For access to the connecting tube, another hole

with diameter equal to that of the connecting tube is drilled into the bottom of the armature, connecting to the wells from underneath.

Aperture

The aperture is the most critical part of the experimental apparatus, because it is the site of membrane and nanopore. An aperture must be made of a material that lipid will adhere to, so that a membrane may be established across it- PTFE, again, serves this purpose in our standard experiments.

In a standard nanopore experiment at University of Washington, a length of PTFE heat-shrink tubing is melted with a heat gun at one end around a very sharp tungsten needle. The needle is removed, and a very sharp knife is used to shave thin layers off of the melted end of the tube. After each removal of material, the aperture is examined under a measuring microscope to determine its width. The needle is conical, and therefore so is the void it leaves after removed from the tube. This means that the more length is removed from the end of the tube, the larger the aperture will be.

When the aperture reaches a sufficient diameter, the tube is trimmed to length at the unmelted end, and inserted connecting the two holes on the bottom of the PTFE puck. A properly inserted and trimmed tube will have both ends set back a few millimeters from the surface of the puck, but still accessible by pipette.

An aperture must pass two tests: it must be able to be fully wetted, and must support a bilayer. These requirements put restrictions on the diameter of an aperture. Small apertures support very stable and long-lived bilayers, but an aperture that is too small will be difficult to force buffer through and will not be able to maintain an electrical connection between the *cis* and *trans* buffer volumes. Small apertures also reduce the insertion rate of membrane proteins, because they have a reduced cross-sectional area with which to interact with the protein in solution in the buffer. Large

apertures are easier to wet and clean and also easier to see in a bench microscope, but apertures that are too large are difficult to establish bilayers on. Bilayers on large apertures are also fragile and short-lived, and they also have larger capacitance, leading to a noisier nanopore signal.

Other materials and methods may also be used to create apertures, which may be necessary if exploring alternative experimental geometries. For example, different solutions might be needed for a nanopore experiment using microfluidic volumes of buffer, or one in which both *cis* and *trans* buffer volumes must both be accessible to the experimenter, or an experiment that needs to be optically accessible. Micrometer-sized holes have been drilled in many materials, including glass, silicon, silicon nitride, and polyimide film[1], and using many methods, including laser or ion beam drilling, dielectric breakdown, and chemical etching.

When devising a new aperture production technique, a few considerations are necessary. First of all, the aperture must be flat on the side the lipid will be applied to. Any recession in the surface around the aperture makes it likely that the hole will become clogged with lipid when it is applied to the surface. This makes chemical etching difficult to use in aperture production for nanopore experiments, because it produces conical holes.

The geometry of the other side of the pore is also important, even though its details are not as rigidly prescribed. An excessively long and narrow channel to the membrane will increase access resistance and make wetting the aperture very difficult. Additionally, it makes diffusion away from very slow. This is of concern if the experimenter wishes to access both *cis* and *trans* buffer volumes, because it makes one volume essentially inaccessible. However conical cross-section is desirable in other respects, as this geometry reduces the capacitance of the aperture and increases its

[1]Trade name Kapton.

durability.

Electrodes

The University of Washington nanopore lab has found success in using silver chloride pellet electrodes[2]. To prepare these electrodes for use in a nanopore experiment, the exposed wire is soldered to a 1 mm brass pin. The length of the electrode from the base of the pin to past the silver chloride pellet is insulated by evenly melting PTFE shrink-wrap tubing. The tubing melts completely around the pellet. The melted tubing is cut with a sharp knife to expose the pellet, minimizing waste by cutting as close as possible to the end of the pellet.

Electrodes degrade over time, both from environmental corrosion and from depletion during experiments. Fortunately, these effects only degrade the exposed surface of the pellet, so the life of an electrode may be extended by shaving a very thin layer off of the surface, exposing unreacted silver chloride.

Electrode degradation is a common and inevitable occurrence for nanopore experimenters. Trouble establishing a bilayer may be caused by severe electrode degradation, when the electrode potential is high. Additionally, unusual signals, particularly drifting or noisy ion currents and unusual voltage biases, may be caused by a corroded or depleted electrode. Therefore, replacing or refreshing the electrodes is a typical first step in troubleshooting a non-functioning setup.

Environmental isolation

There are several sources of environmental noise and artifacts. These can be addressed through experimental design considerations and observation of certain laboratory pro-

[2]1.0 mm diameter, 2.5 mm length pellet electrodes with 70 mm exposed wire; sourced from AM systems.

tocols.

EM shielding. Radio-frequency electromagnetic radiation is ubiquitous anywhere on earth. Nanopore experiments are effective enough antennae that an unshielded experiment will not provide useful data. Fortunately, a Faraday cage is an effective tool for eliminating this interference.

The Faraday cage should be a conductive box containing the patch clamp headstage and the nanopore armature. It is advisable to place some minimal cushioning on the base of the box where it contacts the table, to prevent sharp impacts which may disturb the experiment. The cage should be grounded to the same reference as the rest of the experiment.

Although it is effective at isolating the experiment from most ambient radiation, including the 60 Hz[3] signal from AC power, the Faraday cage does not eliminate all sources of intereference. Strong local radio sources, such as cell phones, can be problematic, so it is advisable to either keep cell service off, or keep such devices at least five meters away from experiments.

Temperature control. Temperature may be controlled both actively and passively. Passive control is achieved by placing the puck in a small, sealable box, within the Faraday cage, preventing convective heat transfer. Conductive heat transfer may also be limited passively by building this container out of or lining it with a heat-resistant material. This interior container is advisable for any nanopore experiment, even those operating at room temperature, because it also provides additional electronic shielding and minimizes evaporation of buffer.

Active control of temperature may be provided by a Peltier element and a thermistor. The Peltier is placed under the base of the passive temperature control box,

[3]In the United States.

and controlled with a power supply. A third, very small well is drilled in the center of the top of the puck, just large enough to fill with buffer and submerge a thermistor sensitive to the temperature range being studied. This well is not electrically connected to the wells containing buffer. The resistance of the thermistor is monitored by the experimenter and , who adjusts the Peltier power as necessary to keep it stable.

Temperature control can also be automated if the power supply is computer-controlled and the thermistor resistance is measured by the computer using an analog-to-digital converter.

Mechanical isolation. The meniscuses of the two buffer wells on the puck contribute to the capacitance of the system. When they are disturbed, this capacitance changes, causing a relatively large electrical signal in the highly sensitive patch clamp. Unfortunately, the drops of buffer tend to behave as mechanical oscillators due to their surface tension's restoring force. This means mechanically isolating the experiment to a reasonable degree is necessary.

The most important factor in reducing mechanical noise is using a small enough amount of buffer that the meniscuses of the wells are approximately flush with the surface of the puck. This eliminates the lowest mode of vibration and the one most likely to be induced by motion of the table it is attached to, where the entire mass of the meniscus moves side-to-side.

Additionally, a heavy and stable work surface reduces the magnitude and frequency, and therefore the total power in mechanical oscillations. Finally, some behavioral protocols should be observed while running a nanopore experiments: avoid loud noises[4] and avoid any physical contact with the experiment table while recording data.

---

[4]Even raised voices or high fives can introduce artifacts into the nanopore signal.

Hardware

It is advantageous to limit the mobility of each element used in the experiment, as this reduces clumsiness-related errors. As such, the temperature isolation cage and Faraday cage should be on hinges fixed to the table. The puck itself should be held into the temperature isolation cage by a clamp. Plastics and anodized aluminum are good choices for hardware material, as they are easily machined and do not corrode when exposed to buffer.

3.2 Preparation and Operation

This section contains a guide to preparing materials for nanopore experiments, setting them up, and operating the experiments while recording data.

3.2.1 Preparation

Before collecting data with a nanopore experiment, several preparatory steps are required.

Cleaning

Keeping the buffer volumes free of contamination is important to make sure experiments operate smoothly, and is critical in experiments meant to compare different conditions. Most obviously, this necessitates using fresh pipette tips and syringes for every experiment. Between experiments, it is also important to clean the pucks, electrodes, and U-tubes.

The protocol found most effective for cleaning pucks is, leaving the U-tubes in place, to stir them in a mixture of 3 parts concentrated sulfuric acid and 1 part 30% hydrogen peroxide, also known as *piranha solution* for an hour. This solution, while highly effective at cleaning the pucks, is also exothermic when mixed and highly

corrosive. Nonreactive gloves and eyewear as well as a nearby neutralization bath are required. After cleaning in piranha solution, the pucks are boiled in deionized water and drained twice.

To clean the inside of the U-tube after the boiling, a syringe is used to force first water, then ethanol, and finally hexane through the aperture by applying pressure to the open end of the U-tube. This dissolves and expels any lipid that might remain. A final rinse with deionized water removes any residual hexane.

Electrodes and lipid brushes are cleaned simply by rinsing first with ethanol and then with DI water, wiping and drying between each rinse on a laboratory wipe.

Priming with lipid

To facilitate adhesion of the lipid to the surface of the PTFE aperture tube, it can be helpful to "prime" the surface with dried lipid. The priming solution consists of lipid is suspended at 3% by mass in hexane. To make this solution, 20 microliters of lipid-chloroform solution (concentration 10 milligrams/milliliter) is transferred using a glass pipette into a glass test tube. The tube is loosely covered to prevent contamination and placed in vacuum to dry. After 15 minutes, the chloroform will have evaporated. The tube is placed on a mass scale and 66 mg of hexane are added by weight.

1 microliter of the priming solution is applied to the aperture end of the U-tube of each cleaned puck with a pipette. The primed pucks are dried in a vacuum chamber for 15 minutes. This process is repeated, for a total of two layers of priming lipid.

An excess of priming lipid is visible under a benchtop microscope as hairy or flakey debris which disperses when the puck is filled with buffer. If too much priming lipid is added, it may be necessary to lightly brush the surface of the aperture with a fine tipped paintbrush to remove the excess lipid, then perfuse fresh buffer into the front well to remove the debris.

Preparing lipid paint

To prepare the lipid used for establishment of a bilayer, dried down lipid is mixed with a solvent. Because of the means by which this lipid is applied to the aperture, we call this mixture *lipid paint*. To prepare the dried lipid on slides, lipid suspended in chloroform is dropped onto the slide to create dots of lipid about 2 millimeters in diameter. These slides are covered and dried in a vacuum chamber for 15 minutes to evaporate the chloroform. The mixture of this lipid with solvent is detailed in the following section.

Because lipid oxidizes in open air, it is recommended that both priming and paint preparation be done within a few days of the experiment, ideally the same day. Lipid stock should be kept at -20°C. It also should be replaced two or three weeks after opening, and kept under argon gas in a glove bag whenever it is accessed. Oxidized lipid causes mid-event gating in MspA experiments. This gating severely hampers data throughput. Careful handling and consistent refreshment of lipid is essential to preventing this problem.

3.2.2 Wetting the aperture

When beginning an experiment, first the puck is electrically connected. The magnitude of the applied voltage is not important, other than that it be nonzero so a blocked aperture may be distinguished from an open aperture as the current drops to zero, and that it be less than roughly 200 millivolts to avoid stressing the bilayer.

Next, the puck is wetted by filling the front and back wells with buffer, and then forcing buffer through the back of the U-tube with a syringe until it goes through the aperture. A good connection is indicated by the measured ion current going from 0 (no connection) to the amplifier maximum (because the low resistance of the conductive buffer admits a current much greater than the dynamic range of the patch

clamp amplifier).

After a good connection is achieved, excess buffer is removed until the meniscuses on the front and back wells are approximately flush with the top surface of the puck.

3.2.3 Establishment of bilayer

There are many methods of establishing a bilayer over an aperture, but the most common technique used with the apparatus described above is referred to as *painting*. All steps in this section are ideally performed under a benchtop microscope, which helps to visualize the paint mixing, painting, and bubbling processes described here.

To paint a bilayer, a paintbrush with a very fine point, usually cut down to only a few bristles, is dipped in hexadecane solvent. The brush is used to work this solvent into the dried down lipid paint described in the previous section.

The greater the ratio of hexadecane to lipid, the thinner the paint will become. Ideally, the paint will be just wet enough to form a smooth globule that sticks to the end of a single bristle of a paintbrush. When paint is too dry, containing not enough solvent, bilayers will fail after a short time. When the paint contains too much solvent, it will spontaneously clog the aperture with solvent-rich paint rather than form a bilayer, and even if a bilayer is established, pore insertions are unlikely. Slightly drier paint is required for smaller apertures, and slightly wetter paint is required for larger apertures.

Once the appropriate consistency of paint has been achieved, it is transferred using the brush to the surface of the aperture, and spread in a thin layer around the aperture of a wetted puck. Often this painting process results in paint entering the aperture and forming a clog. Therefore, after painting the aperture, if the measured ion current has dropped to zero, buffer may be forced through the back of the U-tube to blow excess paint out of the aperture, and then excess buffer removed.

Once paint has been applied and an electrical connection re-established, a pipette is used to blow a small bubble over the surface of the aperture, then pull it back off of the surface without popping it. The air-buffer interface promotes formation of a membrane. After blowing and removing the bubble, if the measured ion current is still non-zero, the paintbrush is used to redistribute paint and the bubbling process is repeated.

If after bubbling the ion current has dropped to zero, a bilayer may have formed. If the aperture blockage is a bilayer, it will break when briefly exposed to high voltage around 1 volt, or *zapped*. If ion current does not return to the amplifier maximum after the puck is zapped, the blockage is not a bilayer, but a clog. As before, clogs are cleared by forcing buffer through the back of the U-tube. After clearing the clog, the paint is redistributed and the bubbling and testing are repeated.

If the bilayer does break when zapped, it is reformed by bubbling again. To check if it is a stable bilayer before adding reagents, it is advisable to set the applied voltage to the operating voltage of the experiment and wait approximately five minutes before doing so. Bilayers that survive for several minutes under these conditions are likely to last long enough for the duration of an experiment.

3.2.4 Isolation of single nanopores

Once a bilayer has been established, nanopore protein is added to the buffer in the front well. The objective is to obtain a single nanopore insertion. Too much protein will lead to multiple insertions occurring too quickly to isolate one, and too little protein makes waiting time prohibitively long, so a moderate amount is required. The exact amount required depends on the volume of buffer in the well, the consistency and size of the bilayer, and the quality of the protein, but can generally be determined by trial and error. Generally, the amount should be on the order of 1 nanogram of

protein in a 40 microliter well volume.

Nanopore insertion can be expedited by refluxing the buffer with a pipette, focusing on increasing the flow of buffer over the aperture, or by repeatedly bubbling the aperture to re-form the bilayer.

Chapter 4
NANOPORE BIOPHYSICS

In this chapter, I seek to provide a qualitative overview of the issues at play in the biophysics of the nanopore-DNA-enzyme complex. For some concepts, I provide back-of-the-envelope calculations indicating orders of magnitude for different effects.

I begin by discussing the effect of Brownian motion on the behavior of the DNA and the enzyme, and how it affects nanopore measurements. These ideas are combined with models of DNA stretching. Later, I discuss a few other important concepts relevant to nanopore physics including access resistance.

4.1 Brownian motion

At the scale of molecular biophysics, thermal motion of the molecules being studied and of the surrounding medium is an omnipresent consideration. In the nanopore sequencing and SPRNT experiments specifically, we must consider the way thermal motion affects the behavior of the trapped DNA, the enzyme it is anchored to, and the ions that carry electrical current through the nanopore.

With a few assumptions, we can estimate the probability distribution of the center of mass of the DNA-enzyme complex. The enzyme-DNA complex is confined by the pore, which excludes DNA and other amino acids and can be considered an infinite potential wall at nanppore experimental conditions. It is also trapped by the applied voltage. The shape of the potential caused by this trapping forces is a infinite potential wall on the left and a constant slope on the right (figure 4.1). Since we are only considering the translational motion of the center of mass of the enzyme-DNA

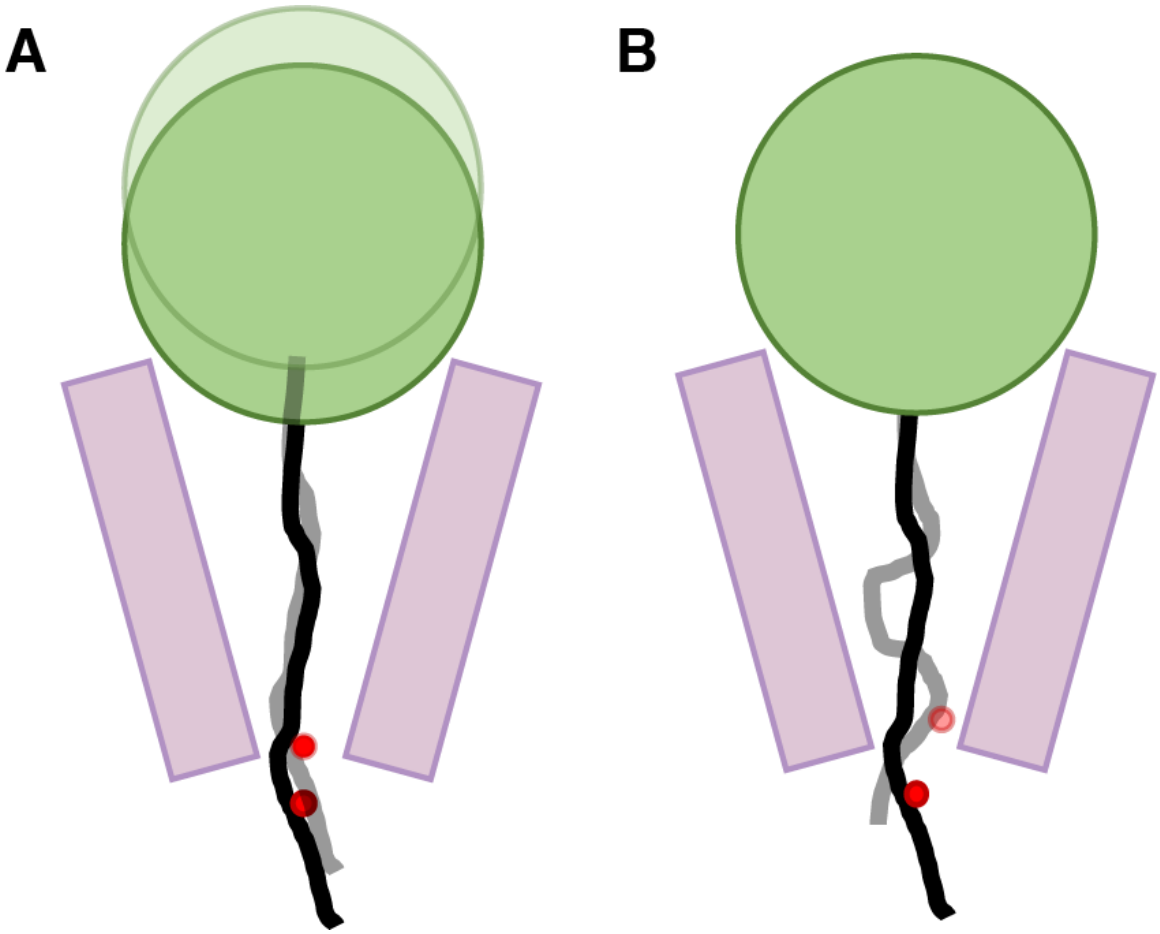

Figure 4.1: **Brownian motion in the nanopore experiment.** (a) The entire enzyme-DNA complex moves randomly relative to the pore. (b) The DNA conformation changes randomly.

complex as a whole here, we neglect entropic effects, and the position of the complex should be Boltzmann distributed:

$$p(x) \sim e^{-U(x)/k_BT} \tag{4.1}$$

We can find the correct normalization by plugging in the correct expression for $U(x)$ and only integrating up to the infinite potential barrier:

$$1 = \int_0^\infty \mathrm{d}x; Ae^{-(7.5\ k_BT/\mathrm{nm})\cdot x/k_BT} \quad \rightarrow \quad A = 7.5\ \mathrm{nm}^{-1} \tag{4.2}$$

$$p_{\mathrm{COM}}(x) = 7.5e^{-(7.5\ \mathrm{nm}^{-1})\cdot x}\ \mathrm{nm}^{-1} \tag{4.3}$$

where x is the position in nanometers. The standard deviation of this distribution is (1/7.5) nm = 0.13 nm, about one third of the 0.34 nm contour length of one ssDNA base.

Meanwhile, the changing configuration of the DNA apart from center-of-mass motion also contributes to the fluctuations in its position relative to the constriction of the pore. If it adopts a more bunched-up configuration, the base in the constriction will be closer to the feeding end of the DNA, and its configuration is more extended, the opposite is the case. In approximating the potential landscape of DNA configurations, we neglect the lateral confinement caused by MspA's vestibule, because the tensional forces on the DNA cause lateral motion to be relatively small[40]. In this approximation, we can estimate the potential as the sum of the approximately quadratic spring potential from ssDNA extension and the linear potential from the electrophoretic force. $k \approx 300$ pN/nM-base pair at 30 pN applied force[41], which for a $\sim$ 10 base pair stretching length of DNA in the vestibule of MspA yields a spring constant of 30 pN/nM. This back-of-the-envelope calculation yields a normally

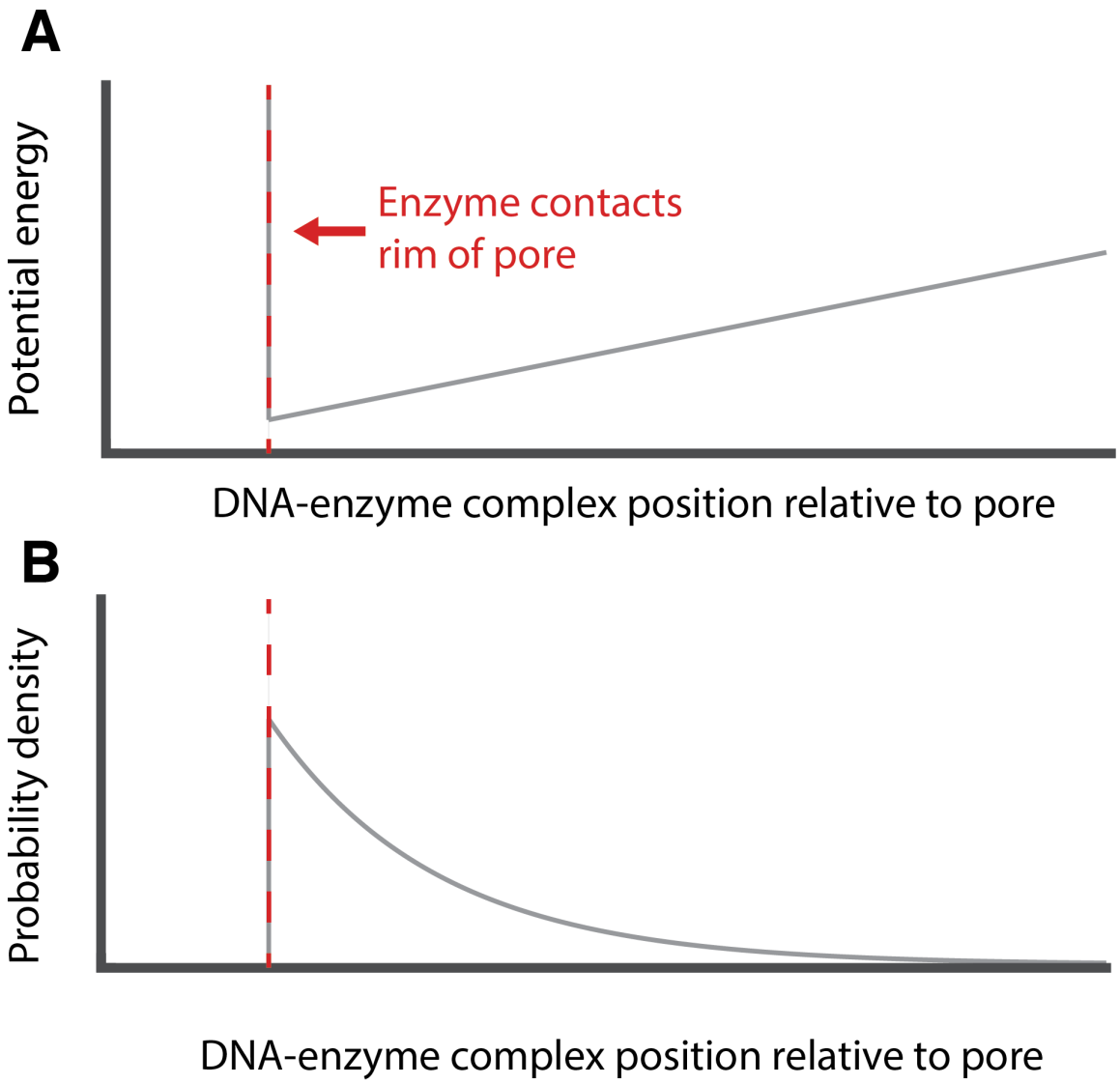

Figure 4.2: **Trapping potential for DNA-enzyme complex.** (a) A hard wall potential (black) is at $x = 0$ representing the point of collision between the enzyme and the rim of MspA (red dashed line), and linear for $x > 0$ representing the constant force applied to the DNA as it translocates through the pore. (b) The Boltzmann-distributed position of the center of mass of the DNA-enzyme complex is a decaying exponential for $x > 0$ and 0 elsewhere.

distributed DNA position x,

$$p_{\text{stretch}}(x) = \frac{1}{\sqrt{2\pi}} \sqrt{\frac{k}{k_B T}} e^{-kx^2/k_B T} \tag{4.4}$$

whose standard deviation is $\sqrt{k_B T/k} \approx 0.37$ nm.

The standard deviation of the DNA position relative to the constriction is then the standard deviation of the sum of these two random variables, obtained by computing the square root of the variance

$$\delta x_{\text{DNA-constriction}} = \left[\int_0^\infty \mathrm{d}x \int_{-\infty}^\infty \mathrm{d}y \, (x + y - \bar{x} - \bar{y})^2 \, p_{\text{COM}}(x) \, p_{\text{stretch}}(y) \right]^{1/2} \tag{4.5}$$

$$= \sqrt{\sigma^2_{\text{COM}} + \sigma^2_{\text{stretch}}} = 0.19 \text{ nm} \approx 0.56 \text{ ssDNA bases.} \tag{4.6}$$

These results omit several potential sources of fluctuation: the access resistance of the pore (see §4.5) may change as a result of enzyme-DNA complex COM motion. This, as well as the random presence of ion species in the constriction or differing forces depending on the portion or orientation of the DNA within the constriction may all contribute to picosecond scale fluctuations in force which may contribute to greater fluctuations in position.

The relaxation time scale of all parts of the trapped enzyme-DNA complex is shorter than nanoseconds[42], and the position distribution is continuous and predominantly unimodal. Therefore, it is reasonable to go forward considering every observed ion current value to be an average, weighted by this distribution, over all DNA positions relative to the constriction of the pore.

This effect is actually beneficial to data analysis. We don't generally have to consider the conformational state of the enzyme-nanopore system at each observation,

since each observed ion current is the result of average behavior as the system ergodically explores its state space. The fact that the time scales of brownian motion and the time scales of the more coherent biological activity of enzymes are so different greatly reduces the complexity of the data analysis.

4.2 DNA stretching

From force spectroscopy experiments on DNA (or any flexible microscopic polymer), it can be seen that DNA is extensible under force[43]. The more force is applied to the DNA, the further it is stretched, similarly to a spring. However, the mechanism of DNA's elasticity is quite different from that of a macroscopic mechanical spring.

In a macroscopic mechanical spring, the restoring force is a result of intermolecular forces within the solid comprising the spring, and acts to minimize internal stress and strain, the dominant component of free energy in this relatively low-entropy system. In a microscopic polymer, on the other hand, entropy dominates the free energy. There is a much greater number of possible polymer configurations with a small end-to-end extension (in other words, "bunched up") than with a large extension ("stretched out"). This entropic preference for a small end-to-end extension competes with the stiffness and the self-repulsion of the polymer: for a stiff or self-repulsive polymer, there is an energy cost for a bunched up configuration requiring a great deal of bending or compaction of the polymer.

In terms of forces, the tendency for bunching is the result of molecules from the surrounding medium colliding with the polymer (figure 4.2). The role of entropy in the free energy ($G = H - TS$) is modulated by temperature, with the behavior of systems at greater temperature being driven more towards high-entropy states. In the case of polymer extension, higher temperature results in a greater rate of higher energy collisions, further randomizing configuration of the polymer and increasing the

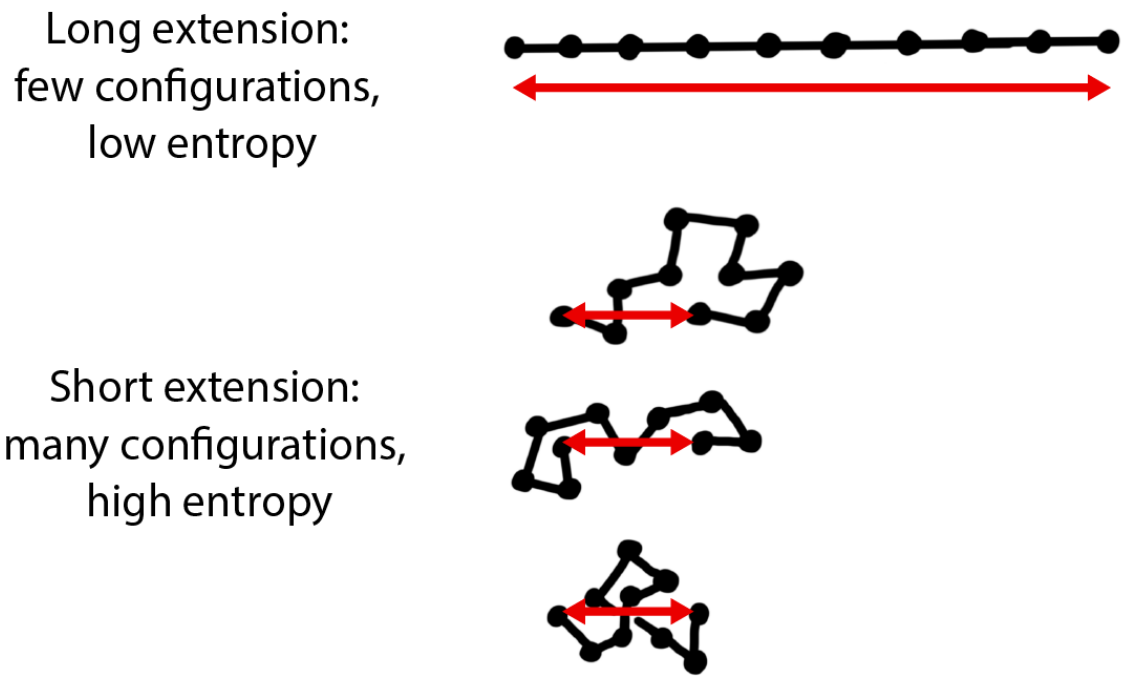

Figure 4.3: **DNA functioning as an entropic spring.** (a) Large extensions are low-entropy, with full extension requiring all bond angles and lengths to be at maximal extension. (b) Smaller extensions have a greater multiplicity of possible configurations of DNA and are therefore higher-entropy and thermodynamically favored. Physically, this is the result of energy exchange with the surrounding medium as its particles collide with the DNA.

likelihood of high-entropy states.

4.2.1 ssDNA stretching curve

In most nanopore experiments, the DNA between the pore constriction where force is applied and the anchoring enzyme is a single-stranded.

Unfortunately, most studies of ssDNA stretching measure the behavior of longer (at least 100 bp) strands of DNA. When these studies are performed on long het-

eropolymer DNA strands, this means that secondary structure formation significantly impacts the stretching curve. It is therefore important to reference curves taken from homopolymer DNA strands, as these are likely to more closely represent the physical reality of the short DNA and reasonably high applied force in the nanopore experiment, in which there is little.

In a review of DNA stretching measurements, Frey (2012) provides a homopolymer ssDNA stretching curve obtained from atomic force microscopy experiments (figure 4.2.1)[41][44]. This curve is measured using λ-phage DNA and presented in terms of extension-per-nucleotide. This is because in any reasonable model of microscopic polymer stretching, the effective spring constant of the DNA is approximately inversely proportional to the length of the polymer.

The curve shows two regimes of stretching with drastically different spring constants, visible as two different slopes in different ranges of applied forces. The lower spring constant is a result of the entropic restoring force as the DNA is pulled out from its high-entropy, bunched-up configuration into a straight line. Most of the changes to the configuration here are in the bond angles between nucleotides. The high-force spring constant is after the bond angles have rearranged themselves to bring the DNA close to full extension, and is a result of the restoring force that maintains the interatomic spacing in the bonds between nucleotides in the DNA backbone. Nanopore experiments are typically at forces below 100 pN, and therefore primarily explore the lower spring constant.

4.2.2 Extensible freely jointed chain model

The rich theoretical physics of chainlike polymers such as DNA has a history stretching back to the work of John Paul Flory in the 1930s [45], originally being developed in industrial laboratories studying polymers and plastics such as those of chemical and

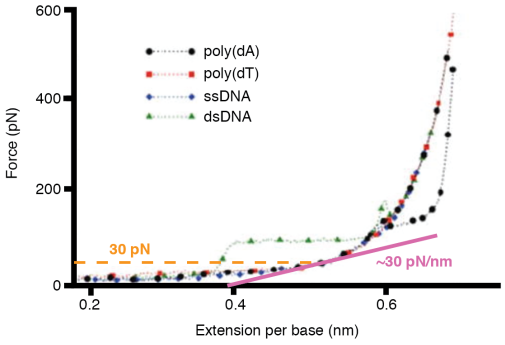

Figure 4.4: **Force-extension curve for different ssDNA strands.** DNA extension shows two different regimes of spring constants as force increases. At low forces, the DNA is functioning as an entropic spring. At higher forces, angles between statistically independent segments have straightened out and energy is being put into stretching the covalent bonds linking the DNA backbone. Figure is a reproduction from Frey (2012)[41] using data from Chen (2010)[44].

materials manufacturer DuPont. The core idea linking most physical modeling of polymers is the notion of a random walk.

Each polymer consists on a microscopic level of a set of N linked monomers. Each of the $N-1$ links' configurations are characterized by some set of degrees of freedom, typically bond angles and distance (figure 4.2.2). The path that a polymer traces out is thus characterized by a discrete set of choices of these parameters for each of the links in the chain.

In general, the physical link configurations are not statistically independent, instead sharing some degree of mutual information. However, it is always possible to replace our consideration of physical bond angles with the configurations of *statistical segments*, portions of the chain[1]that are sufficiently large that the displacements of their endpoints are statistically independent. The probability of a particular

[1]The length of these segments is termed the *Kuhn length*.

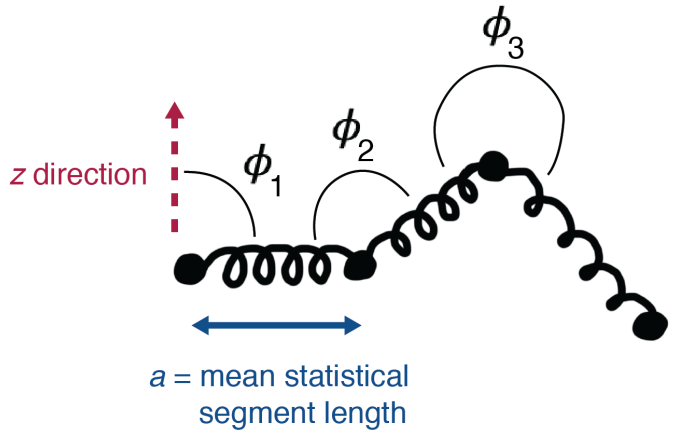

Figure 4.5: **Extensible freely-jointed chain model of DNA.** Bond angles between statistically independent segments are uniformly distributed, and bond lengths are subject to a harmonic potential around an average extension[43]. The first bond angle ϕ_1 is arbitrarily chosen, for a free polymer, in this cartoon relative to the chosen z direction.

microstate characterized in such a fashion can be broken into the product of the probabilities of each link configuration. These probabilities depend only on the potential energy landscape of the bond's degrees of freedom. However, in asking about macrostate properties of the polymer, entropy may also play a role. For example, only one configuration of link parameters leads to a polymer of maximal end-to-end extension, while many configurations lead to some variation of bunched-up polymer with a small end-to-end extension. Therefore, high-entropy small extensions tend to be more likely than low-entropy large extensions.

For nanopore sequencing or single-molecule biophysics using MspA, we have been interested particularly in only a few macrostate properties: extension length and fluctuation in that extension at moderate extending forces. These quantities can be estimated with basic thermodynamics and empirical data as discussed in the previous section, but a model of polymer physics allows us to further dissect quantities such as

the spring constant used for the DNA's extension with force, allowing us to consider how these quantities may change with temperature or length of DNA.

The Gaussian or freely-jointed chain is a simple model appropriate for short segments of DNA, and accomodates results obtained without taking a large-N limit corresponding to polymer chains with a large number of statistically independent segments [43][46][47]. It approximates the bond separations as completely fixed and rigid. This simple model can be extended by modeling the link between each statistical segment instead as an elastic spring, which is a more reasonable assumption given the relatively high applied forces in a typical nanopore experiment that pull the DNA close to full extension of bond angles. Of particular interest in the following section is the expressions for the end-to-end extension of the extensible freely jointed chain under applied force,

$$x(F) = L\left[\coth\left(\frac{Fa}{k_BT}\right) - \frac{k_BT}{Fa} + \frac{F}{\kappa a}\right]. \tag{4.7}$$

Here L is the contour length of the polymer, a is the length of a statistically independent segment, and κ is the effect spring constant between statistically independent segments[2], T is temperature, and k_B is Boltzmann's constant. The spring constant for the extensible freely jointed chain may also be written as a function of force,

$$k = -\frac{\mathrm{d}F}{\mathrm{d}x} = -\left(\frac{\mathrm{d}x}{\mathrm{d}F}\right)^{-1} = Lak_BT\left[\left(\frac{k_BT}{F}\right)^2 + \frac{k_BT}{\kappa} - a^2\mathrm{csch}^2\left(\frac{aF}{k_BT}\right)\right] \tag{4.8}$$

[2]Or equivalently, the elastic modulus of each segment.

4.3 Pore "readhead"

The statistical properties of microscopic polymers and the resultant averaging over DNA positions relative to the constriction has consequences for DNA sequencing. It means that despite the short length of the MspA constriction, we are convolving its few-base effect on ion current over DNA positions around the equilibrium extension, about with a standard deviation of more than half a DNA base in either direction. This means that rather than a single base or just two bases affecting the ion current through the pore, a greater number of them will play a role in determining ion current blockage levels. The size of the region that influences the ion current is called the *readhead* of the pore, in analogy to the functioning of a tape cassette reader.

Indeed, when we measure the sequence of ion currents from an enzyme-controlled heteropolymer DNA strand blocking the nanopore, we see more than just four different ion current levels. Additionally, the states that are observed demonstrate some autocorrelation (figure 4.3). The length of the autocorrelation is commensurate with the expected thermal broadening in combination with the height of the narrowest constriction of MspA.

This multiple-base effect is treated in analysis by associating each observed ion current state with a "k-mer" rather than a single base. A k-mer is a subsequence of DNA bases of length k. For example, the sequence ACGTCCA consists of 4-mers ACGT, CGTC, GTCC, and TCCA. The requirement that adjacent observed k-mer states overlap by $k-1$ bases makes sequencing possible, despite the possibility of different k-mers having indistinguishable ion currents. The details of the construction of a k-mer model for DNA sequencing are explored in §9.2.

Understanding the physics behind the breadth of the readhead helps us make predictions about which changes to experimental conditions might affect it. Applying a greater force to the DNA reduces the jitter of the entire enzyme-DNA complex,

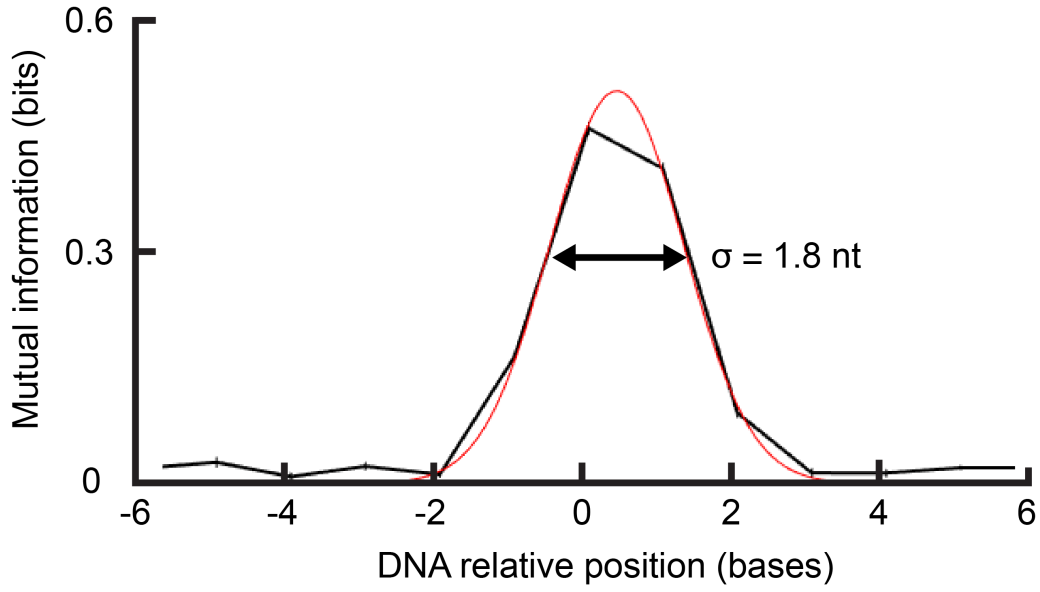

Figure 4.6: **MspA readhead.** Visualizing the readhead: a plot of mutual information between DNA base and ion current versus position relative to the constriction center. Mutual information estimated from long reads of PhiX 174 viral genome. Positive x values correspond to the *trans* side of the constriction. Red line shows a normal distribution of the same width, approximating the shape of the expected fluctuations in DNA position.

σ_{COM}, thus narrowing the readhead. Similarly, shortening the length of the DNA by operating MspA in a "backwards" orientation (see §11) or reducing salt concentration in the buffer increases the spring constant of DNA, reducing $\sigma_{\mathrm{stretch}}$ and also narrowing the readhead.

4.4 "Phase shift"

The averaging effect of the readhead has an interesting consequence. Since the observed ion current is the expectation value of the ion current over the smooth probability density function of the DNA position, the ion current will be a continuous function of DNA position[3]. Mathematically, this amounts to the statement that

$$\bar{I}(x) = \int I(x - x')p(x')\mathrm{d}x' \tag{4.9}$$

is continuous so long as the function $I(x)$ is finite and the function $p(x)$ is smooth.

This means that given a particular DNA strand and the fluctuations in its position, if we were to move this DNA smoothly and continously through the pore, we would expect to see a certain continuous curve. Using an enzyme to control the DNA's translocation, we instead sample this curve at a series of discrete, equally space points, leading to the steplike ion currents we actually observe. Shifting of the sampling points by changing the extension or displacement of the DNA will result in systematic variation of the observed ion currents (figure 4.4). We call both the phenomenon of sampling point shifting and the magnitude of this shift (measured in nucleotides) *phase shift*, the use of the word *phase* being in analogy to the phase of a waveform.

To calculate the phase shift for observations in a particular read, we first create an interpolating curve from reference reads, either through convolution of the discrete

[3]The concept of phase shift was originated by Ian Derrington and Andrew Laszlo. Quantification of force-dependent phase shift was performed by Ian Derrington.

observations with a smoothing envelope or through the calculation of a smooth spline. The observations which we wish to estimate the phase shift for are aligned to this interpolating curve. The calculation is not highly sensitive to the interpolating curve in regions where $|dI/dx|$ is large, because any reasonable interpolation method through these regions will be well approximated by a straight line connecting the points on either side. When calculating the phase shift in this way, experimental changes that change the DNA position indeed do cause shifts in the ion currents that appear to move the sampling points along this theoretical curve.

4.4.1 Force-dependent phase shift and DNA stretching

Since the DNA acts as an entropic spring, the equilibrium position of the DNA will change as the parameters of this spring (the spring constant and the applied force) change. If a greater force is applied by increasing the voltage across the nanopore, the DNA will be more extended and the center of the readhead will shift. Measuring phase shift in this context is slightly complicated by the scaling of the ion current with the voltage. Therefore, we can replace the measurement of the ion current with a measurement of pore conductance. This removes the scaling of ion current with voltage up to the nonlinear response function, which is small in the range of voltages over which we may observe a phase shift.

Experiments establishing the reality of the voltage dependent phase shift are reported in Derrington 2015[48]. In this publication, we measure the sequence of observed ion current conductances for the same strand of DNA at 180 mV and at 140 mV. When the 140 mV conductances are uniformly shifted by 0.29 nucleotides forward, they lie within measurement uncertainty on the interpolating curve generated by the 180 mV data.

The concept of phase shift is fundamental to understanding the resolution of single-

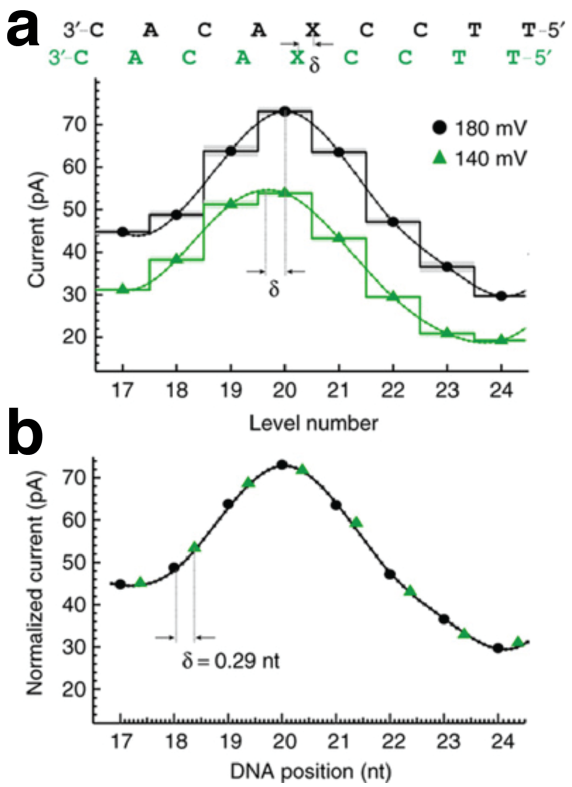

Figure 4.7: **Systematic phase shift in ion currents.** (a) Nanopore ion current at 180 mV (black) and 140 mV (green) applied voltage. At these two different voltage, different forces are applied to the DNA, causing different extensions relative to the pore. The conductance of the DNA is therefore sampled at slightly shifted positions along the strand. (b) Conductance versus DNA position. By applying a shift to the x-coordinates of the 140 mV conductances, they lie within error on an interpolating spline through the 180 mV conductances. Figures adapted from Derrington (2015)[48].

molecule studies using MspA. The slope of the underlying continuous ion current curve determines the spatial resolution of this technique, as it indicates what the expected change in ion current is for a given change in position. In high slope regions of the continuous curve, we can measure slopes as high as 15 pA/nt with a noise of 1 pA RMS at 5 kHz sampling frequency. This makes it possible to measure changes in DNA position of as small as 1/15 of a nucleotide, or about 20 picometers, over the course of a few hundred microseconds, an unsurpassed level os spatiotemporal resolution for single-molecule biophysics.

By repeating this experiment at many voltages, we can also create a DNA stretching curve, measuring displacement as a function of voltage. This idea is foundational to the variable voltage sequencing technique explored in §12 of this thesis. The DNA stretching curve specifically is discussed in §12.2.2.

4.4.2 Upstream bases and DNA stiffness

In figure 4.2.1, it can be seen that different homopolymer DNA strands yield different effective spring constants. This spring constant can change even more due to stacking or pairing interactions between bases on the strand of DNA being stretched[49]. Changing the base content in the vestibule of the nanopore therefore changes the spring constant, and by extension[4] changes the DNA bases in the constriction of the pore.

A simple model of this effect only takes into account nearest-neighbor interactions. In this case, each 2-mer composing the DNA sequence in the vestibule is associated with a particular mean extension, and the sum of these extensions gives the mean total extension of the DNA. This model suggests that the mean DNA extension should change slowly, because only one out of the 12 links in the vestibule DNA is changing

[4]No pun intended.

with each enzyme step. The effect of changing DNA stiffness in this way has not been thoroughly quantified.

Of course, other effects may cause a phase shift. For example, in certain DNA processing enzymes, it may be that different bases are held in slightly different positions in the active site. This could cause a repositioning of the DNA relative to the enzyme and pore that changes discretely with each enzyme step, as the bases within the enzyme change. This is more difficult to generalize about, however, because it changes for each enzyme used with the nanopore.

4.5 Access resistance

Access resistance is a property of nanopores indicating the voltage drop over regions other than the constriction of the pore (figure 4.5). Effectively, it results in a voltage divider, making it so only a fraction of the applied voltage causes an electrophoretic force on the DNA. Analytic results have been obtained for simple circular pore geometries[50], and computational results have been obtained for more complicated solid-state geometries.

However, computation of access resistance in our system is more complicated. Precise computation requires knowing exactly how the enzyme seats itself atop the rim of the pore, and how it changes position due to thermal motion. This may be different for every enzyme. Access resistance therefore serves as a complicating factor in many calculations dependent on the voltage across the constriction, such as those involving applied force.

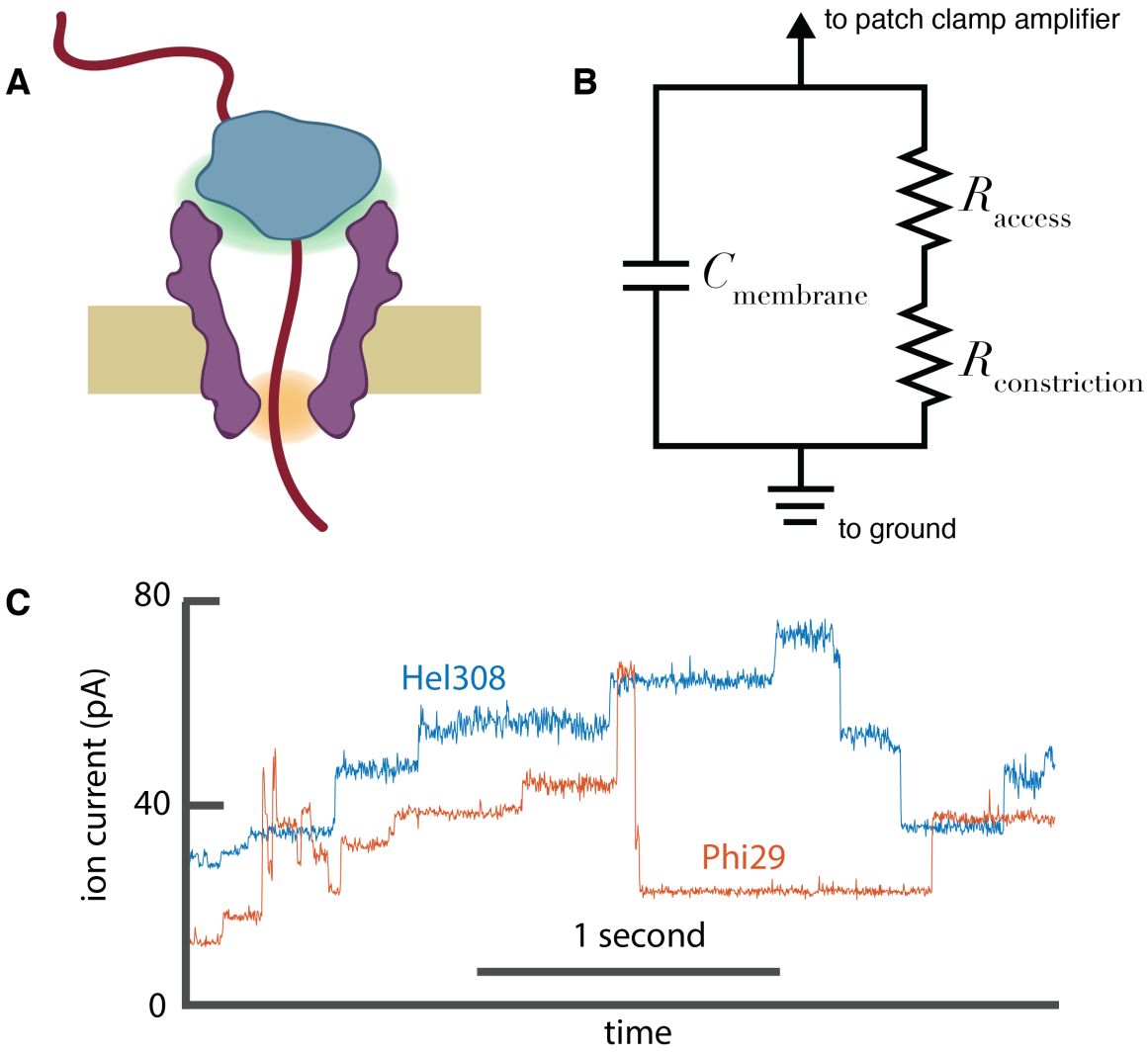

Figure 4.8: **Illustration of access resistance effects.** (a) Access resistance reflects the drop in electric potential over the region of the medium outside the nanopore's constriction (green highlight). While always present for a nanopore, it can be affected by the DNA-bound enzyme blockading the nanopore without translocating, introducing uncertainty in the force applied to the DNA. (b) Equivalent circuit for pore with access resistance separated out from constriction resistance. Access resistance results in an additional resistance, not dependent on DNA sequence, always being measured by the patch clamp. (c) Different enzymes may cause different access resistance, seen as an overall ion current scaling between experiments on the backwards nanopore (more in §11) controlled by Hel308 DNA helicase (blue) versus Phi29 DNA polymerase (red).

Part III
MODELS AND ANALYSIS

The way nanopore data should be handled, from experimental design through preprocessing all the way to final statistical analysis, depends on the scientific questions being asked.

Notation conventions

Throughout this chapter it regularly useful to discuss probabilities, probability densities, likelihoods, and the logarithms thereof. I will adhere to the following conventions:

The probability of event A is denoted $P(A)$, the probability of A given B is $P(A|B)$, and the probability of A and B is $P(A;B)$. If the meaning of the probability is clear from context, and especially if a set of probabilities needs to be indexed, I opt for simply P, so that the indexing will appear, for example, as P_i. Probability densities also use the letter p, but in lower-case.

Likelihoods are written with a similar convention when explicitly stating their conditions, but with $\mathscr{L}$ replacing p. Because I often deal with many likelihood quantities with complicated conditions, in general I use capital letters other than P to name particular likelihoods. The logarithms of these likelihoods are written in lower-case versions of the same letter. For example, I might write that $S = \mathscr{L}(x|\theta)$, and that $s = \log S$.

Chapter 5

HIDDEN MARKOV MODELS

Many randomly changing systems in physics and other sciences may be described using a framework called a *Markov model*. Such systems, which include nanopore data, are called *Markov processes*. In particular, nanopore data is well-described by a variant of a Markov model called a *hidden Markov model*, or *HMM*.

Within this chapter I explore the structure, design, and fitting of these models, and how they may be applied to nanopore experiments. §5.1 provides a formal definition of a Markov process. §5.1.2 discusses the ways in which the nanopore signal may be interepreted as one. Finally, §5.2 discusses *expectation maximization*, or *EM*, an algorithm commonly used to find local optima in the goodness-of-fit function when fitting HMMs, and goes into detail on the way it can be practically implemented for nanopore experiments.

5.1 Markov processes

Traditionally[1], a *Markov process*, also termed a *Markov chain*, is defined as a sequence of probabilistic changes in the discrete state of a system in which the probability of each state change depends only on the present state (figure 5.1). This "lack of

[1]Many variations of and extensions to Markov processes have been described which do not obey this definition. For example, a Markov-like process called a *Harris chain* may be considered in which the state space is continuous and the transitions between states are described by a probability density. A continuous-time markov process may also be considered in which the state changes continously with time or in which a system may be in a superposition of discrete states. These have not been notably applied to nanopore data and therefore are not discussed in this thesis.

memory" in the system is called the *Markov property*.

The countable (but possibly infinite) set of states the system may take may be assigned numerical indices. The same is true of the discrete series of times. This makes it convenient to write terms such as $P(\mathcal{S}_t = i)$, the probability that the system's state $\mathcal{S}_t$ at time t is state i. The Markov property above can then be stated formally as

$$P(\mathcal{S}_t = i_t|\mathcal{S}_1 = i_1,\ ..., \mathcal{S}_{t-1} = i_{t-1}) = P(\mathcal{S}_t = i_t|\mathcal{S}_{t-1} = i_{t-1}), \tag{5.1}$$

that the state at time t is independent of the state at all times prior to $t-1$. The Markov property allows the state transition probabilities to be expressed as a *transition matrix* T, whose elements are

$$T_{ij} = P(\mathcal{S}_t = j|\mathcal{S}_{t-1} = i), \tag{5.2}$$

the probability that the process transitions from state i to state j. The elements of T are subject to the constraints of probability: that they must be positive, $T_{ij} \geq 0$, and that they must be normalized, $\sum_j T_{ij} = 1$.

The set of states together with these transition probabilities fully characterize the Markov process. If the sequence of states in a Markov process may be directly observed, the Markov model may be fitted- that is, the maximum likelihood estimates of the elements of T can be estimated as ratios of counts of observed transitions:

$$\hat{T}_{ij} = \frac{\text{number of transitions from state } i \text{ to state } j}{\text{number of transitions out of state } i}. \tag{5.3}$$

These counts are multinomially distributed, so the uncertainty in these estimates is

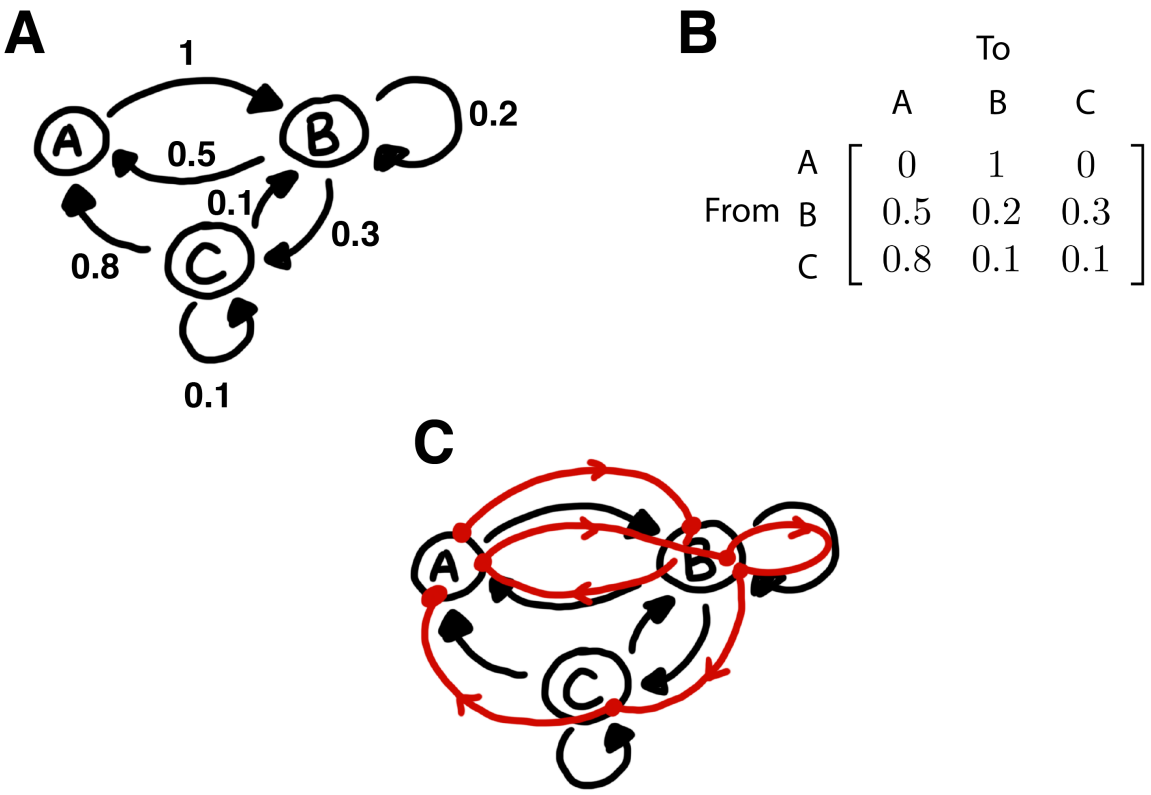

Figure 5.1: **A Markov process.** (a) Markov processes are commonly represented through weighted directed graphs, where nodes represent the discrete states that the system can take, and arrow weights represent the probabilities of transitions between states. (b) The transition matrix representing this Markov process. (c) An instance of a Markov process comprises a random walk along the edges of the graph subject to these transition probabilities. Shown in red is a graphical illustration of the Markov chain $\{A, B, A, B, B, C, A\}$.

$$\delta\hat{T}_{ij} = \sqrt{\frac{\hat{T}_{ij}(1-\hat{T}_{ij})}{\text{number of transitions out of state } i}}. \tag{5.4}$$

5.1.1 Hidden Markov models

Often, the realities of experimental science are such that direct observation of the state of a Markov process is not possible. Instead, the observation is determined probabilistically by the current state. These systems are well modeled by *hidden Markov models*. An example of such a system is illustrated in figure **??**.

Like an observable Markov model, a hidden Markov model is equipped with an indexable set of system states (called *true states*) and transition probabilities between these states. Additionally, a hidden Markov model has a set of *observable states* distinct from the true states. Each true state is associated with a probability distribution on the set of observable states. The observed state depends only on the current state.

A particular instance of a Markov process consists of a sequence of true states, where as before we call $\mathcal{S}_t$ the true state at time t. Associated with each true state $\mathcal{S}_t$ is an observation X_t drawn from the observed state distribution associated with $\mathcal{S}_t$, $P(X_t = x|\mathcal{S}_t)$. We refer to these probabilities as *observation probabilities*.

Therefore, training a hidden Markov model involves estimating not only the transition matrix T, but also the observation probability distributions[2]. Since the true states are in general unknown, before computing the observation and transition probabilities, we must either have a "training set" whose true states are known, or we must assign observations to their likely true states.

A "solution" to an HMM is an estimate of all unknowns: the true state identity of each observation, the transition probabilities, and/or the observation probabilities. Methods for solving HMMs for sets of observations are elaborated upon in §5.2.

---

[2]More precisely, we need to estimate the parameters defining the distributions.

5.1.2 Nanopore ion current as a Markov process

There are several ways in which we can model the nanopore signal as a Markov process. The lowest-level of these is considering each ion current sample in the time series as an observation, while the state space is the physical configuration of the DNA-enzyme-nanopore complex. This method is used in reference-based state segmentation, described in §6.1.1. Alternatively, data can first be segmented. Then we may consider statistics such as the mean ion current of each segment to be observations, while the true state is determined by DNA sequence in the constriction of the nanopore. Transition probabilities between these states are governed by the known sequence of the strand being read, if available, or the structure of the k-mer map when sequencing DNA. The nanopore signal in both these cases is indeed a Markov process: in the first case because of the thermal nature of the noise, and in the second case because of the enzyme's random stepping. Room temperature is sufficiently high that inertia is not generally of concern in molecule-scale systems.

5.2 Solving hidden Markov models with expectation maximization

If we understand the structure (that is, the space of true and observed states) of a system with Markovian behavior, we can construct a Markov model describing it. However, we need a prescription for actually assigning the state transition probabilities and observation probabilities (or probability densities), as well as for assigning the most likely true state to each observation.

The typical approach to these problems is the use of a modeling procedure called expectation maximization, or *EM*. EM is commonly used for fitting mixture models, where an observation may be classified as being part of one of some number of discrete states, each with a separately parameterized probability density function.

5.2.1 Discrete mixture models

In order to better understand expectation maximization, it is convenient to study a simple application that is not a hidden Markov model. One such algorithm is the problem of fitting a data set x drawn from a distribution expressed as the sum of multiple different distributions,

$$p(x|\Theta) = \sum_i w_i p_i(x|\theta_i). \tag{5.5}$$

Here $w_i \in [0, 1]$ are mixture parameters establishing the relative weights of each distribution, subject to the constraint $\sum_i w_i = 1$, and $\Theta = \{\theta_1, \theta_2, ...\theta_M, w_1, w_2, ..., w_M\}$, are the w_i together with the parameters characterizing each component distribution $\{p_1, p_2, ...p_M\}$.

Fitting this type of model with a likelihood maximation through variation of the parameters is often difficult, even if the p_i are simple functions such as normal distributions whose maximum likelihood parameters are easily found. The gradient of the likelihood function is not necessarily linear in the parameters, and the likelihood is not even guaranteed to be convex. However, as is frequently possible with these sorts of models, if we assume that each measurement was generated by just one of the component distributions, the problem is effectively decomposed into N separate unmixed problems. This means that if each component distribution is efficiently maximizable, we can efficiently maximize the entire likelihood function.

However, we do not know *a priori* to which distribution we should assign each of the observations. We therefore introduce unobserved, discrete "latent variables" Z, such that Z_t indicates the distribution X_t was drawn from. We then can compute and maximize the expectation value over Z of the likelihood function of Θ,

$$\mathbb{E}_{Z\sim z}\left[\mathscr{L}(\Theta|X)\right] = \prod_t \left[\sum_i z_{ti}\, w_i\, p_i(X_t|\theta_i)\right], \tag{5.6}$$

$$\hat{\Theta} = \arg\max_{\Theta} \mathbb{E}_{Z\sim z}\left[\mathscr{L}(\Theta|X, Z)\right]. \tag{5.7}$$

Here $\hat{\Theta}$ is the MLE for the parameters Θ, and $z_{ti} = P(Z_t = i)$ is the probability that X_t was drawn from the component distribution i.

However, this computation relies on knowing the probabilities z_{ti}. Given a set of parameters Θ, we can find estimates $\hat{z}_{ti}$ of these values straightforwardly as the relative likelihood that the observation X_t came from the distribution p_i:

$$\hat{z}_{ti} = \frac{w_i p_i(X_t|\theta_i)}{\sum_j w_j p_j(X_t|\theta_j)}. \tag{5.8}$$

It seems now as though we have constructed a circular process- in order to estimate the parameters Θ, we need to know z, and in order to estimate z we need Θ. Although this process is circular, if it is initiated with guess values of z and θ, it does converge to a (not necessarily global) maximum likelihood. Therefore, we proceed with an iterative process, alternately updating z and θ:

1. Choose $\hat{\Theta}^0$ as an initial guess for Θ at step $n = 0$.

2. Compute the estimates of probabilities for the next step,

$$\hat{z}_{ti}^{n+1} = \frac{\hat{w}_i^t p_i(X_t|\hat{\theta}_i^n)}{\sum_j \hat{w}_j^n p_j(X_t|\hat{\theta}_j^n)}. \tag{5.9}$$

3. Compute the estimates of parameters for the next step,

$$\hat{\Theta}^{n+1} = \arg\max_{\Theta} \mathbb{E}_{Z\sim z^{n+1}}\left[\mathscr{L}(\Theta|X, Z)\right]. \tag{5.10}$$

4. If converged, return the estimate $\hat{\Theta}^{n+1}$. Otherwise, increment n, and return to step (2).

5.2.2 Solving hidden Markov models with expectation maximization

A hidden Markov model is also amenable to fitting using iterative expectation maximization: the observations are each drawn from one of a discrete set of distributions, but which distribution they are drawn from is unknown to the experimenter. In the most general case of solving an HMM, neither the parameters of the observation probability distributions, the true states of each observation, nor the true state transition probabilities are known. In the framework from the previous subsection, the model parameters Θ consist of the transition probabilites $T_{ij} = P(\mathcal{S}_t = j|\mathcal{S}_{t-1} = i)$ and the observation probability function parameters θ, where $p(X_t = x|\mathcal{S}_t = i) = p(x|\theta_i)$. The latent variables z_t are the true states of the observations $\mathcal{S}_t$, so $z_{ti} = P(\mathcal{S}_t = i)$. Thus, the steps to find a maximum-likelihood model become

1. Choose $\hat{T}^0$ and $\hat{\theta}^0$ as initial guesses at step $n = 0$.

2. Compute the estimates of true state and transition probabilities for the next step

$$\begin{aligned} \hat{z}_{ti}^{n+1} &= P(\mathcal{S}_t = i|X, \hat{T}^n, \hat{\theta}^n) \\ \hat{T}_{ij}^{n+1} &= \frac{1}{N}\sum_{t=1}^{N-1} \hat{z}_{t-1,i}^{n+1}\, \hat{z}_{tj}^{n+1} \end{aligned} \tag{5.11}$$

   To understand the expression for T^{n+1}, note that it is the expectation value over all observations of the probability of seeing a transition from state i to state j.

3. Compute the estimates of parameters for the next step,

$$\hat{\theta}_i^{n+1} = \arg\max_{\theta} \mathbb{E}_{Z\sim z^{n+1}} \left[\mathscr{L}(\theta|X, Z)\right], \tag{5.12}$$

4. If converged, return the estimated transition probabilities $\hat{T}^{n+1}$, the estimated true state parameters $\hat{\theta}^{n+1}$, and the estimated true state sequence $\hat{Z}_t = \arg\max_i \hat{z}_{ti}^{n+1}$. Otherwise, increment n, and return to step (2).

T and θ are explicitly calculated at each step in the above procedure, so the only remaining task is to determine how to assign the true state probabilities z given a guess at T and θ. This problem is of great relevance to nanopore data analysis, because we often know T (given by the enzyme kinetics) and θ (given by the k-mer map), and are in search of the sequence of k-mer or reference states in the pore, so we can sequence bases or make an alignment to a reference respectively.

There are two subtly different approaches to this question. In one, we identify the most likely true state to have generated each *individual* observation, conditioned upon all observations; the solution to this problem is provided by the BCJR, or maximum *al posteriori* algorithm (§5.2.3). This solution maximizes the probability that any given true state is identified correctly, and is the correct choice for iterations of expectation maximization and for maximizing base calling accuracy when sequencing using an HMM. Alternatively, we can identify the most likely *sequence* of true states to have generated our data; the solution to this is provided by the Viterbi algorithm (§5.2.4). This solution maximizes the probability that our entire solution is correct without any error, and is often an effective approximation to the BCJR algorithm, although it is not as precisely correct for most nanopore applications.

5.2.3 BCJR maximum al posteriori *algorithm*

In this section, we use the notation that X_s^t is the subsequence of observations from s to t, $\{X_s, X_{s+1}, ..., X_t\}$, and similarly, $\mathcal{S}_s^t$ is the subsequence of true states $\{\mathcal{S}_s, \mathcal{S}_{s+1}, ..., \mathcal{S}_t\}$. We label the states with integers, so $\mathcal{S}_t \in \mathbb{Z}$.

The BCJR algorithm maximizes the likelihood of any given true state being correctly assigned[51]. The estimate of the true state probabilities of each observation depends on the entire sequence of observations X_1^N, so the probability of a sequence of states $\mathcal{S}_1^N$ is written $P(\mathcal{S}_1^N|X_1^N, T, \theta)$. Because here we are assuming we know or have estimates of T and θ, we will omit these conditions from probability expressions here, and simply write $P(\mathcal{S}_1^N|X_1^N)$.

To assign true state probabilities to each observation, we can take advantage of the probabilistic structure of a Markov process, namely that the probability of observing X_t is conditioned only on the present state $\mathcal{S}_t$, and the identity of $\mathcal{S}$ is conditioned only on the previous state $\mathcal{S}_{t-1}$. We can decompose the probability of the t'th true state being state i as

$$\begin{aligned} P(\mathcal{S}_t = i|X_1^N) &= P(\mathcal{S}_t = i, X_1^t)\ P(X_{t+1}^N|\mathcal{S}_t = i, X_1^t) \\ &= P(\mathcal{S}_t = i, X_1^t)\ P(X_{t+1}^N|\mathcal{S}_t = i) \end{aligned} \tag{5.13}$$

where the condition on X_1^t dropped out of the second term because of the property of Markov processes that X_{t+1} is independent of $X_{s<t+1}$, except for through the identity of the immediately preceding state $\mathcal{S}_t$. We next make two definitions,

$$\begin{aligned} A_{ti} &\equiv P(\mathcal{S}_t = i, X_1^t), \\ B_{ti} &\equiv P(X_{t+1}^N|\mathcal{S}_t = i). \end{aligned} \tag{5.14}$$

So that $P(\mathcal{S}_t = i|X_1^N) = A_{ti}B_{ti}$. In words, A_{ti} is the likelihood of observed t belonging to true state i conditioned on all the observations prior to and including t. B_{ti} is the posterior of the same likelihood, except conditioned on all the observations following t. By taking their product, we obtain the likelihood conditioned on every observation, which is what we are trying to calculate.

We can expand A_{ti} further as

$$\begin{aligned} A_{ti} &= \sum_j P(\mathcal{S}_t = i, \mathcal{S}_{t-1} = j, X_1^t) P(\mathcal{S}_{t-1} = j) \\ &= \sum_j P(\mathcal{S}_t = i, X_t|\mathcal{S}_{t-1} = j) P(\mathcal{S}_{t-1} = j, X_1^{t-1}) \\ &= \sum_j P(\mathcal{S}_t = i, X_t|\mathcal{S}_{t-1} = j)\ A_{t-1,j}. \end{aligned} \tag{5.15}$$

Similarly, we can expand B_{ti}:

$$\begin{aligned} B_{ti} &= \sum_j P(X_{t+1}^N, \mathcal{S}_{t+1} = j|\mathcal{S}_t = i) P(\mathcal{S}_{t+1} = j) \\ &= \sum_j P(X_{t+1}, \mathcal{S}_{t+1} = j|\mathcal{S}_t = i) P(X_{t+2}^N|\mathcal{S}_{t+1} = j) \\ &= \sum_j P(X_{t+1}, \mathcal{S}_{t+1} = j|\mathcal{S}_t = i)\ B_{t+1,j}. \end{aligned} \tag{5.16}$$

From these expressions, we see that we presuming we can calcuate terms of the form $P(\mathcal{S}_t = i, X_t|\mathcal{S}_{t-1} = j)$, we can recursively find every A by starting at $t = 1$ and advancing through the entire sequence of observations one at a time, and find B by starting at $t = N$ and stepping through backwards through the observations. To find the missing terms, we observe that again because of the Markov property,

$$P(\mathcal{S}_t = i, X_t|\mathcal{S}_{t-1} = j) = P(X_t|\mathcal{S}_t = i)P(\mathcal{S}_t = i|\mathcal{S}_{t-1} = j). \tag{5.17}$$

In general we may have a probability density for $P(X_t|\mathcal{S}_t = i)$, but because we are computing a relative likelihood, the prior $P(X_t)$ may be ignored and we can just replace this probability with $p_i(X_t)$, the pdf for true state i evaluated at the observed value X_t. Additionally, note that $P(\mathcal{S}_t = i|\mathcal{S}_{t-1} = j) = T_{ji}$. the known transition probabilities between states. We therefore can write

$$\begin{aligned} A_{ti} &= p_i(X_t) \sum_j T_{ji}\, A_{t-1,j}, \\ B_{ti} &= p_i(X_t) \sum_j T_{ij}\, B_{t+1,j}, \\ P(\mathcal{S}_t = i|X_1^N) &= A_{ti}\, B_{ti}, \end{aligned} \tag{5.18}$$

completing the prescription for assigning true state probabilities to each observation.

BCJR algorithm: software implementation

An analysis of the asymptotic cost shows that the BCJR algorithm runs in $O(NM^2)$ time with little overhead, where N is the number of observations and M is the number of possible true states. For N and M typical of nanopore experiments, this computation can typically be performed on a desktop computer in a few seconds or minutes.

There are a few practical details to software implementation that address computational issues. Foremost is the fact that the numerous products of probabilities make many of the A_t's approach 0 exponentially as t gets larger, and that for large M, any individual A_t may have machine precision problems. To address this issue, we deal with logarithms of A and B instead of with their values. Defining $s_{ti} = \log p_i(X_t)$,

$\tau_{ij} = \log T_{ij}$, $a_{ti} = \log A_{ti}$, and $b_{ti} = \log B_{ti}$, equation 5.19 becomes

$$\begin{aligned} a_{ti} &= s_{ti} + \log \sum_j e^{\tau_{ji} + a_{t-1,j}}, \\ b_{ti} &= s_{ti} + \log \sum_j e^{\tau_{ij} + b_{t+1,j}}, \\ P(\mathcal{S}_t = i | X_1^N) &= e^{a_{ti} + b_{ti}}. \end{aligned} \tag{5.19}$$

Two other issues to be considered in a software implementation are running time and the structures used to representation of data. Generally, these two are closely related, as careful selection of data structures can also often prevent unnecessary calculation and reduce running time.

The objects in the algorithm as described mathematically are all two-index arrays. This structure is natural for computer representation. Typically we will represent a, b, and s as arrays of size (number of observations) $\times$ (number of true states), and τ as a an array of size (number of true states) $\times$ (number of true states). In addition to being easy to visualize as an array being calculated one row at a time, allowing for easy debugging and intuitive extension of the method, these structures avoid redundant computation by precalculating important quantities used over and over again in s and τ. An algorithm to calculate the BCJR solution to a hidden Markov model is provided in algorithm 5.2.3.

5.2.4 Viterbi maximum likelihood sequence algorithm

The Viterbi algorithm[52][53] may be throught of as a further simplification of the BCJR algorithm. This method, rather than summing over all possible previous states, propagates only the probability of the most likely previous state. It is the result of the approximation that

Algorithm 5.1 BCJR algorithm.

procedure BCJR(x, y, T)

Compute the score matrix and initialize the forwards and backwards matrices.

$n \leftarrow \text{Length}(x)$
$m \leftarrow \text{Length}(y)$

for $i \leftarrow 1...n;\ \ j \leftarrow 1...m$ **do**
    $s_{ij} \leftarrow \log \mathscr{L}(x_i|y_j)$
end for

for $j \leftarrow 1...m$ **do**
    $a_{1j} \leftarrow s_{1j}$
    $b_{nj} \leftarrow s_{nj}$
end for

Fill out the alignment matrix by assuming that the model transitioned from the most likely state at each step. Keep track of which state this was for each true/measured state pair.

for $i \leftarrow 2...n$ **do**
    for $j \leftarrow 1...m$ **do**
        $a_{ij} \leftarrow s_{ij} + \log \sum_j\ e^{\tau_{ji} + a_{t-1,j}}$
        $b_{ij} \leftarrow s_{ij} + \log \sum_j\ e^{\tau_{ji} + b_{t+1,j}}$
    end for
end for

Starting at the most likely final state, walk back through the matrix by recalling the most likely transition each time.

$\tilde{u}_n \leftarrow \arg\max_j$

for $i \leftarrow 1...n$ **do**
    $u_i \leftarrow \arg\max_j(a_{ij} + b_{ij})$
    $L_i \leftarrow \max_j(a_{ij} + b_{ij})$
end for

Calculate and return the approximate maximum likelihood true states and the approximate total log-likelihood of the model.

return $\boldsymbol{a},\ \boldsymbol{L}$

end procedure

$$\log \sum_i e^{a_i} \approx \max_i a_i. \tag{5.20}$$

The Viterbi algorithm also solves a slightly different optimization problem than the BCJR algorithm. Instead of minimizing the expected per-observation error rate, Viterbi algorithms maximize the likelihood that the entire solution is error-free.

Notably, the same results are obtained from using only either forwards or backwards probabilities with the Viterbi algorithm. Therefore, only one probability matrix h is required for the calculation. Its elements are calculated similarly to the a matrix in the BCJR algorithm, replacing the sum with the maximum:

$$h_{ti} = s_{ti} + \max_j \{\tau_{ji} + h_{t-1,j}\} \tag{5.21}$$

However, each time an entry of the matrix is computed, we need to preserve the information of the maximum likelihood transition. This is easy to do computationally by tracking a matrix r of the same dimensions as h, in which we store the index of the most likely previous state the match assumed in the calculation of h_{ti}. Mathematically,

$$r_{ti} = \arg\max_j \{\tau_{ji} + h_{t-1,j}\}. \tag{5.22}$$

Once r has been calculated, to find the maximum likelihood path through the states, we start at the maximum likelihood final match (the maximal entry in the final row of h) and iteratively trace back to the previous most likely match based on the entries of r. The solution is ultimately the sequence of states traversed by this walk back from the highest likelihood final state through the most likely transitions to have led there.

The Viterbi algorithm is a constant factor faster than the BCJR algorithm due

Algorithm 5.2 Viterbi algorithm

procedure VITERBI(x, y, T)

Compute the score matrix and initialize the alignment matrix.

$n \leftarrow \text{Length}(x)$
$m \leftarrow \text{Length}(y)$

for $i \leftarrow 1...n;\ \ j \leftarrow 1...m$ **do**
    $s_{ij} \leftarrow \log \mathscr{L}(x_i|y_j)$
end for

for $j \leftarrow 1...m$ **do**
    $h_{1j} \leftarrow s_{1j}$
end for

Fill out the alignment matrix by assuming that the model transitioned from the most likely state at each step. Keep track of which state this was for each true/measured state pair.

for $i \leftarrow 2...n$ **do**
    for $j \leftarrow 1...m$ **do**
        $h_{ij} \leftarrow s_{ij} + \max_k \{t_{kj} + h_{i-1,k}\}$
        $r_{ij} \leftarrow \arg\max_k \{t_{kj} + h_{i-1,k}\}$
    end for
end for

Starting at the most likely final state, walk back through the matrix by recalling the most likely transition each time.

$\tilde{a}_n \leftarrow \arg\max_j h_{nj}$

for $i \leftarrow n-1...1$ **do**
    $\tilde{a}_i \leftarrow r_{i+1,\tilde{a}_{i+1}}$
end for

Calculate and return the approximate maximum likelihood true states and the approximate total log-likelihood of the model.

$\tilde{l} \leftarrow \sum_i h_{i,\tilde{a}_i}$

return $\tilde{\boldsymbol{a}},\ \tilde{L}$

end procedure

to the shorter time required to calculate a maximum than a logarithm of the sum of exponentials, and in practice yields almost identical results for nanopore alignments or sequencing.

"Bad observations" in a hidden Markov model

It is a possibility in nanopore data, and many other real-world Markov processes, that spurious observations are made which do not correspond to the sequence the experimenter is attempting to observe. These are often termed *bad observations* in the UW nanopore laboratory. In contrast, states that do correspond to relevant observations are termed *good* observations. Ideally, these observations should be addressed by the Markov solver itself, being identified as bad states and removed in the process of solving the model.

Assume that we assign observation i a probability p_i of being bad. The probability that observation i is good is then $1 - p_i$. The probability of an observation being bad may be a constant $p_i = p_{\text{bad}}$ over all observations equal to the expected fraction of bad observations. Alternatively, certain qualities of each measurement may be used to inform the probability. In nanopore data for instance, enzyme states that are highly noisy, short in duration, or far outside the expected range of ion currents are more likely to be bad. These qualities are mapped to probabilities of bad observations using a support vector machine trained on a set of correctly identified good and bad observations. More on this process is provided in §12.2.9.

A solution to the hidden Markov model taking into account the possibility of bad observations has likelihood equal to the likelihood $\mathscr{L}'$ for the solution to the smaller model from which the bad states have been excised, multiplied by the probability of that configuration of good and bad states:

$$\mathscr{L} = \mathscr{L}' \prod_{i \in B} p_i \prod_{i \notin B} (1 - p_i) \tag{5.23}$$

where B is the set of indices corresponding to bad observations. In general then, the maximum likelihood solution to the HMM is $\max_B \mathscr{L}$, this likelihood maximized over all possible sets of bad observations.

To find this solution, we need to consider the probability that each observation was actually a bad state and should be ignored, and instead consider transitions into each state from the states for last good observation. In the general case, we might be considering transitions from state $i - l$, corresponding to any prior observation to the one whose probability is being calculated at the present step i. This is most easily done with a forwards-only algorithm such as the BCJR algorithm only including the forwards probabilities, or the Viterbi algorithm.

Using the graphical understanding of the alignment matrix, this means we are now considering transitions from all rows before the present row (figure 5.2.4). Additionally, we penalize jumps to row i from row $i - l$ by including the probability that each observation between $i - l$ and i is bad. It also means that any time we apply a match probability, we also need to multiply by $(1 - p_i)$, the prior probability that the observation is good and a match should occur at all.

Mathematically, this means that our calculations of h and r in the Viterbi algorithm will now come to

$$h_{ti} = s_{ti} + \max_{j,\, 1 \le l \le i-1} \left\{ (\tau_{ji} + h_{t-l,j}) \prod_{k=1}^{l-1} p_{i-k} \right\} \tag{5.24}$$

$$r_{ti} = \arg\max_j \left\{ \tau_{ji} + h_{t-1,j} \right\}. \tag{5.25}$$

where r is now an ordered pair indicating both the row and column of the most likely

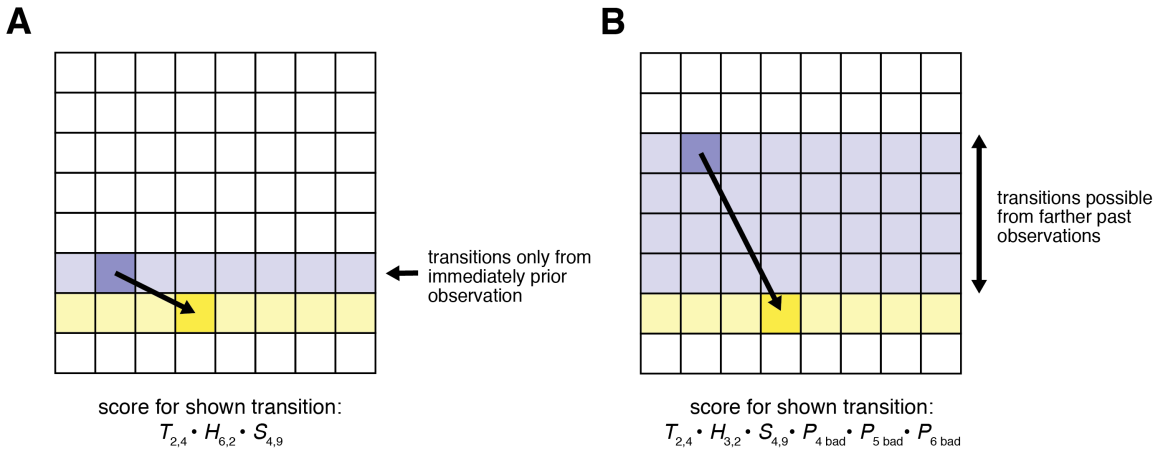

Figure 5.2: **A Viterbi algorithm step accounting for bad observations.** Instead of only considering transitions from the previous row, we record the maximum likelihood transition from all previous rows in the alignment matrix. Transitions from observations farther back are penalized by the interstitial observations' bad observation probability. In practice we can limit the number of rows we look back depending on the maximal number of sequential bad observations expected in a read.

previous matched state in the h matrix.

Efficiency measures for bad observation identification It is reasonable to cap the number of sequential bad observations likely to be seen, allowing us to only check back up to the maximal number of sequential bad observations we expect to see in the read. This saves time, especially for reads with a large number of observations, for which the bad observation checking would ordinarily lead to scaling with the number of observations as $O(N^2)$. By capping the number of observations, the algorithm instead scales as the normal HMM solvers, with computation time going as $O(N)$. For well-segmented nanopore reads, we find that a reasonable choice for the maximal number of sequential bad observations for reads as long as several kilobases is 3.

Chapter 6

DATA SEGMENTATION

The lifetime of the transitional configuration of an enzyme-DNA complex between stable DNA positions is generally very short, resolving far too quickly to be observed by sampling frequencies typical of nanopore experiments. Therefore, the observed time series of ion currents can be described as a sequence of segments, each corresponding to a different state of the trapped molecule. Since the experimenter does not typically have any precise control over when the changes of state occur, the transitions between these states, or *change points* are in general unknown. Identification of the separate segments is a procedure called data segmentation (figure 6).

Partitioning the measured ion currents into segments corresponding to enzyme steps is necessary before sequencing. It drastically simplifies the data stream that is passed to the hidden Markov model (described in §10) by turning the many noisy measurements that make up an enzyme step observation into a sequence of a few low-noise parameters describing each step

Detecting change points where abrupt changes of state occur is a problem that spans many fields, with applications in natural sciences, medicine, engineering, and even economics and finance. As such, many methods for change point detection have been developed. The first section in this chapter discusses several algorithms, each of which is applicable in different cases, depending on what information the data analyst has access to. The second section discusses the way that data is analyzed after segmentation, such that each state is represented by some statistics or observables within the measured segment.

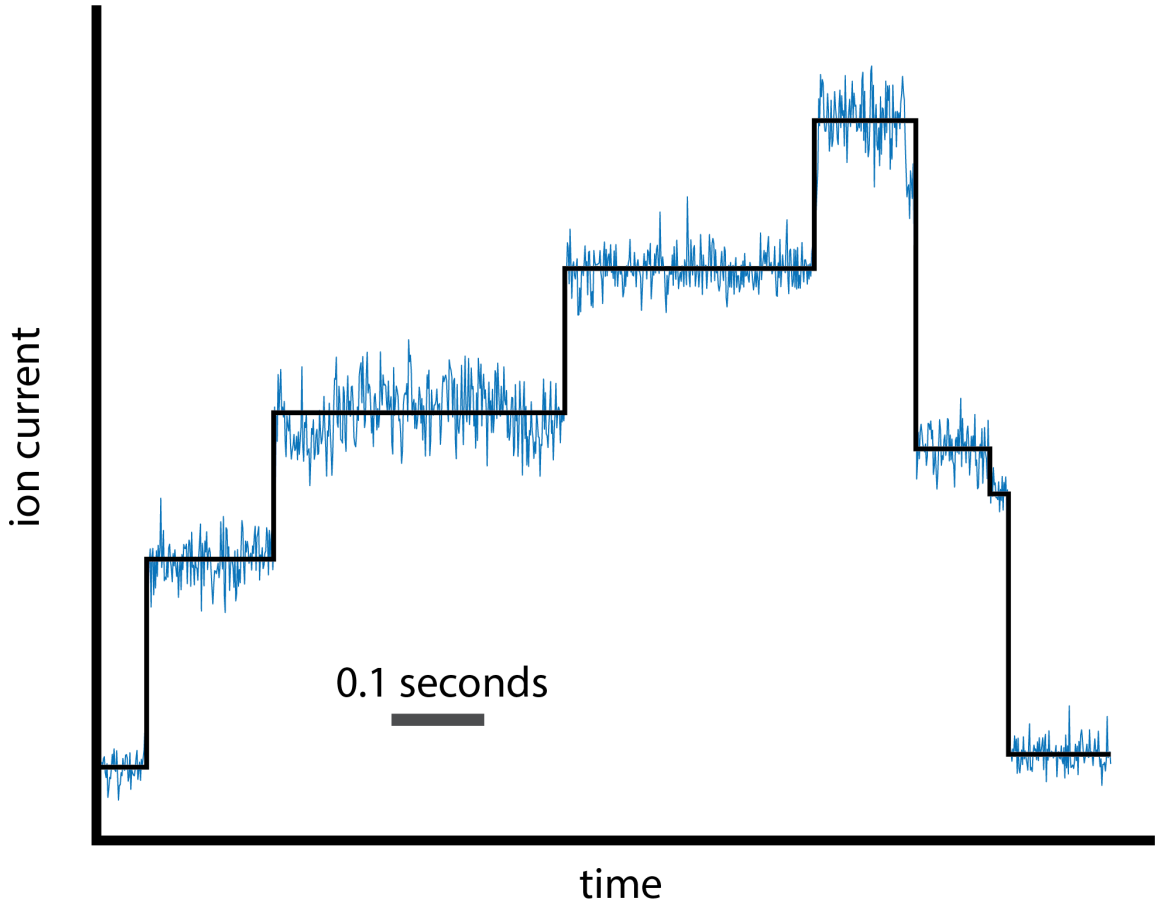

Figure 6.1: **Data segmentation.** Data segmentation is the problem of identifying transition points in time series data (blue). After segmentation, statistics can be extracted from each segment, such as the median ion current in a nanopore state (black). Data shown is from Hel308-controlled DNA on a backwards-orientation MspA nanopore (more on backwards MspA in §11).

6.1 Data segmentation algorithms

The phrasing of data segmentation problem depends on exactly which information is known about the time series.

Specifically, if something about the statistics or durations of the states that we expect to observe, this information can be supplied to the segmentation algorithm, and the appropriate tool in this type of case is often a hidden Markov model, as described in §5. In the case where this information is not available, it is helpful to appeal more directly to information theory and the theory of model selection.

6.1.1 Reference known

If the sequence of expected ion currents corresponding to successive enzyme steps is known, the problem of data segmentation lends itself to the direct application of the hidden Markov model solution techniques described in the previous chapter. Here, each observed ion current is an observation and each expected ion current is a true state.

Reference-informed segmentation is therefore just the maximally granular form of reference alignment, where each observation is just a single observed ion current. Reference alignments in general have a scope beyond data segmentation, and are discussed further in §7. Extensions and computational techniques discussed there apply equally to hidden Markov models implemented for data segmentation.

The transition probabilities for the segmentation HMM are computed through expectation maximization or from training sets. The remaining work then is to define a match likelihood function, $S_{ti} = p(X_t|\mathcal{S}_t = i)$.

This can be a little more involved than the normally distributed likelihood functions defined in §7, as the observations X are not necessarily normally distributed. One approach is to construct an empirical distribution for each true state using tech-

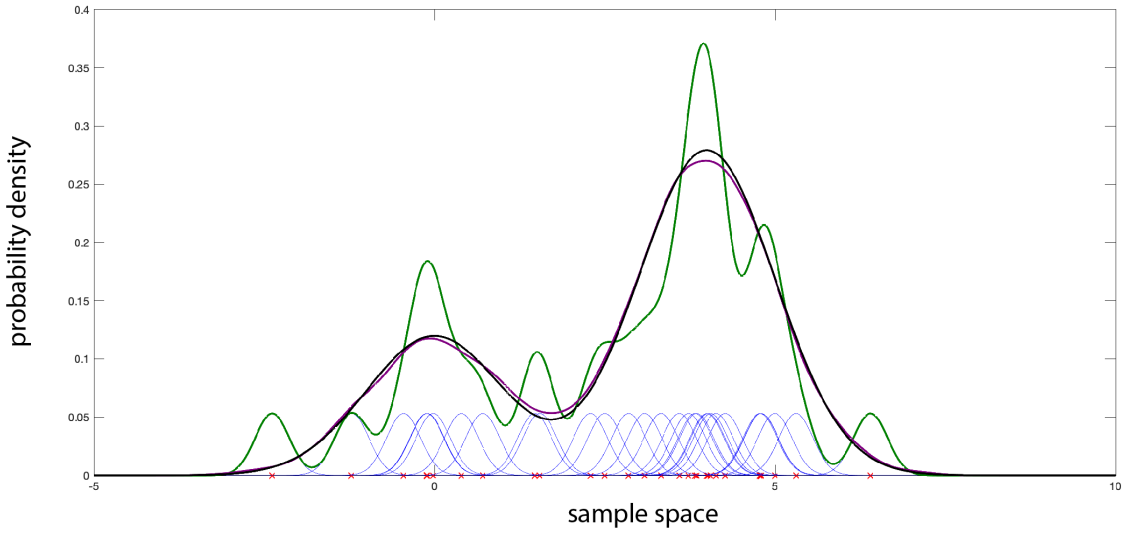

Figure 6.2: **Kernel-based density estimation.** A 1D data set (30 samples plotted as red X's) is associated with a set of kernel density functions (blue lines) centered on the samples. These kernels are summed to generate a continuous empirical probability density function. The green and purple lines show the sum of the 30 shown kernels and a better approximation generated by larger set of 10000 samples, respectively. The black line is the true distribution used to generate the data.

niques of non-parametric statistics.

Kernel-based density estimation methods are commonly used for this purpose (figure 6.1.1)[54]. These methods involve using a set of observations $Y = \{y_1, \ ..., \ y_n\}$ together with a probability distribution $K(x)$ called a *kernel* or *window function* to generate an empirical estimation $\hat{p}(x)$ of a probability density function $p(x)$,

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} K(y_i - x). \tag{6.1}$$

$K(x)$ must have the properties of a probability density function: real, positive, normalized. It is typical to choose kernels that have properties amenable to analysis:

continuity, differentiability, finitude.

A reasonable choice satisfying these requirements is the normal distribution with mean 0 and a standard deviation that is appropriate for both the measurement uncertainty of the individual observations y_i, and the sparseness of the data set used to generate the distribution. The choice of this value is comparable to choice of bin size in a histogram, another non-parametric statistical method for estimating a pdf. In nanopore experiments, we have obtained reasonable results using a standard deviation of 0.25% of the ion current of the unblocked pore. A smaller value may be used if a large number of samples are obtained, but should have a floor of the measurement uncertainty for an individual observation.

We then have everything in place to compute the solution to the HMM, using as a match likelihood $\hat{p}_i(X_t)$, the evaluation at the observed ion current of the estimated probability density function for state i generated with the kernel method.

6.1.2 Reference unknown

The most conceptually challenging case of the data segmentation problem is when no information about the states, the number of states, or the rate of transitions between states is available. In this case, aligning each observed point to a true state is impossible, because the set of true states and their corresponding observations are unknown.

Instead, some general assumptions must be made about the statistics of the data and about the form of a model which might describe it. For constant-voltage nanopore experiments with DNA controlled by a stepping enzyme, we use a model in which

$$x_t = \mu + \sigma\xi_t, \tag{6.2}$$

where x_t is the t'th observation in the time series, μ is the mean ion current of the

segment to which x_t belongs, σ is the standard deviation of the noise within the segment, and ξ_t is a standard normally distributed, uncorrelated random variable:

$$\xi_t \sim \mathcal{N}(0, 1), \tag{6.3}$$

$$\langle \xi_t \xi_s \rangle = \delta_{ts}. \tag{6.4}$$

With this as a model for each segment, a complete model for a time series consists of a choice of $N - 1$ change points dividing the time series into N segments, and for each of these segments choosing a mean and standard deviation to describe the data it contains.

Likelihood-based change-point detection

The core of the change-point algorithm is a recursive partitioning step. The stages, illustrated in figure 6.1.2, are as follows:

1. Find the most likely location in the time series for a change point.

2. If the change point is sufficiently likely, mark it, and repeat steps 1 and 2 on the data to the left and right of the change point separately.

To complete the specification of the algorithm, we need to understand how to estimate the likelihood of a change point existing at a particular time, and we need to precisely understand the notion of "sufficiently likely" described in step 2 above.

Given a data set $\{x_1, x_2, ..., x_T\}$, we can assess the likelihood of a model consisting of state means $\{\mu_1, \mu_2, ..., \mu_N\}$ and standard deviations $\{\sigma_1, \sigma_2, ..., \sigma_N\}$ together with the change points between states $\{t_1, t_2, ..., t_N, t_{N+1}\}$ where $t_1 \equiv 1$ and $t_{N+1} \equiv T$. Because each data point within a segment is independent, this is likelihood is simply the

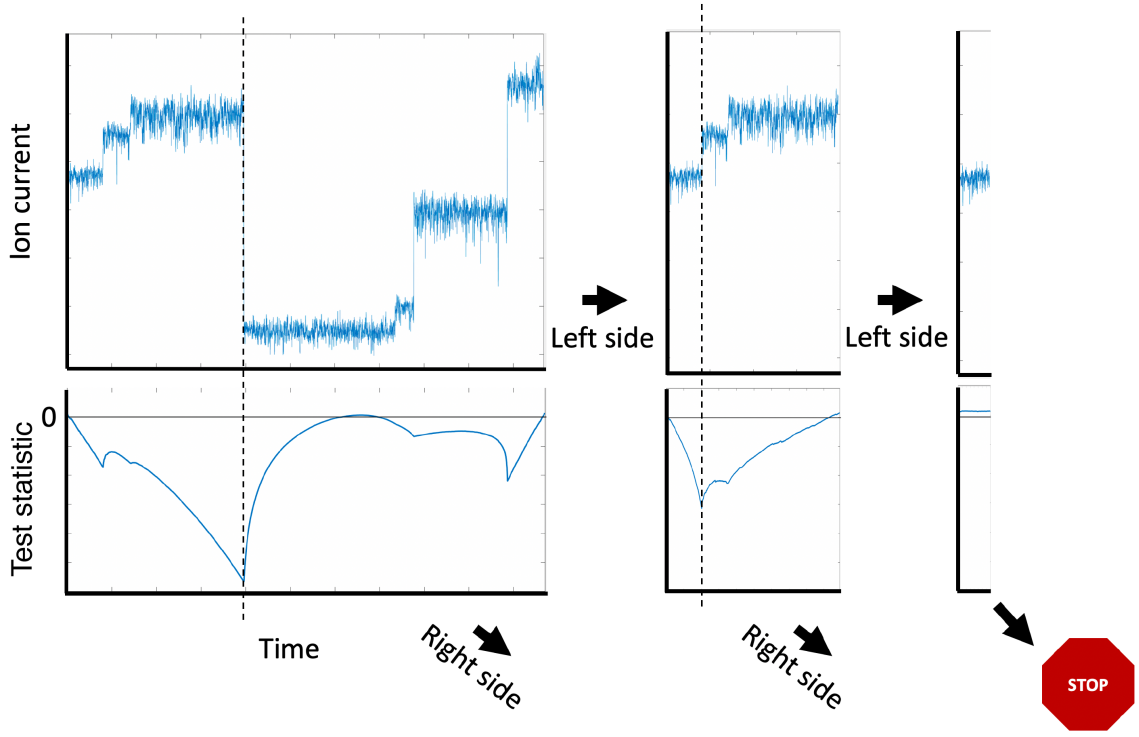

Figure 6.3: **Illustration of the maximum likelihood/minimum information loss approach to change point detection.** (a) A set of time-series data to be segmented. (b) The relative likelihood, or alternatively the information loss, for modeling the data with a change point at each candidate change point location to the likelihood/information loss from modeling it with no change point. (c) A change point is identified (red dashed line) and the steps are repeated on the left and right partitions independently.

product of the probability densities of the model evaluated at each data point within the segment. The total likelihood of the model can be expressed mathematically as

$$\mathscr{L} = \prod_{i=1}^{N} \prod_{t=t_i}^{t_{i+1}} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x_t-\mu_i)^2}{2\sigma_i}}. \tag{6.5}$$

Since the logarithm is a monotonic function, maximizing the likelihood is equivalent to maximizing the log likelihood. The log likelihood, where the products are replaced by sums, is easier to take derivatives of in order to maximize the likelihood, and is less likely to result in large or small number errors when stored in a computer. The log likelihood is

$$\log \mathscr{L} = -\frac{1}{2} \sum_{i=1}^{N} \sum_{t=t_i}^{t_{i+1}} \left[\log 2\pi\sigma_i^2 + \frac{(x_t - \mu_i)^2}{\sigma_i^2} \right]. \tag{6.6}$$

Taking the partial derivative of this expression with respect to each mean and variance, we find that the maximum likelihood estimates of these parameters are, as expected, the mean and variance of the data segment they describe:

$$\hat{\mu}_i = \frac{1}{t_{i+1} - t_i + 1} \sum_{t=t_i}^{t_{i+1}} x_t \tag{6.7}$$

$$\hat{\sigma}_i^2 = \frac{1}{t_{i+1} - t_i + 1} \sum_{t=t_i}^{t_{i+1}} (x_t - \hat{\mu}_i)^2 \tag{6.8}$$

Since we don't know the true values of the variance or mean in each segment, we can only estimate the likelihood function for the purpose of optimizing the model. The best estimate we can make is using the maximum likelihood estimates of the mean and variance written above. Doing so, we see that the approximate log likelihood is

$$\begin{aligned}
\log \hat{\mathscr{L}} &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{t=t_i}^{t_{i+1}}\left[\log 2\pi\hat{\sigma}_i^2 + \frac{(x_t-\hat{\mu}_i)^2}{\hat{\sigma}_i^2}\right] \\
&= -\frac{1}{2}\sum_{i=1}^{N}\left[(t_{i+1}-t_i+1)\log 2\pi\hat{\sigma}_i^2 + \frac{\sum_{t=t_i}^{t_{i+1}}(x_t-\hat{\mu}_i)^2}{\hat{\sigma}_i^2}\right] \qquad (6.9) \\
&= -\frac{1}{2}\sum_{i=1}^{N}(t_{i+1}-t_i+1)\left[\log\hat{\sigma}_i^2 + 1 + \log 2\pi\right]
\end{aligned}$$

Unfortunately, we cannot use calculus to straightforwardly find the maximum likelihood change points, because the likelihood function does not have a unique minimum. Instead, we use the approximate likelihood function to score candidate change points in the recursive algorithm stated above.

At any given step, the recursive algorithm only ever asks the simpler question of whether a section of data is better fit by a single state, or by two states separated by a change point. The likelihood of a change point being at a particular time t within a section of data is approximated by plugging the maximum likelihood estimates for the means and standard deviations into the likelihood function, and computing the relative likelihood of each candidate model. The relative log likelihood is also called the *score*, S_t, of the transition point. In other words, we can write a ratio of likelihoods as a difference of log likelihoods,

$$S_t = \log \mathscr{L}_L + \log \mathscr{L}_R - \log \mathscr{L}_T. \qquad (6.10)$$

Here $\mathscr{L}_L$ and $\mathscr{L}_R$ are the likelihoods of the fits to data on the left and right sides of the transition point t when modeling the system with two states. $\mathscr{L}_T$, meanwhile, is the likelihood of the fit to the entire section of data, assuming it is modeled by only one state with variance σ_T^2. Plugging the simplified result of equation 6.9 into

equation 6.10, we find that the score is

$$S_t = N_L \log \hat{\sigma}_L^2 + N_R \log \hat{\sigma}_R^2 - N_T \log \hat{\sigma}_T^2, \tag{6.11}$$

Where N_L and N_R are the numbers of data points in the section to the left and right of t, and $N_T = N_R + N_L$ is the total number of data points in the section. Note that the constant terms from equation 6.9 cancel out, because the number of data points in the left and right add up to to the total number of data points.

Computation of the variance for every possible subinterval of a data set naïvely takes $O(N^3)$ time, but this can be reduced with some clever precalculation. If we define the cumulate sums

$$X_t = \sum_{s=1}^{t} x_s, \tag{6.12}$$

$$V_t = \sum_{s=1}^{t} x_s^2, \tag{6.13}$$

we can compute the sums of these quantities on any subinterval bounded by indices L on the left and R on the right as differences,

$$= \sum_{t=L}^{R} x_t = X_R - X_{L-1}, \tag{6.14}$$

$$\sum_{t=L}^{R} x_t^2 = V_R - V_{L-1}. \tag{6.15}$$

Using the identity that

$$\mathrm{Var}(x) = \langle x^2 \rangle - \langle x \rangle^2, \tag{6.16}$$

we can write the MLE estimate of the variance of the data on an interval bounded by L and R as

$$\hat{\sigma}^2_{[L,R]} = \frac{1}{R-L+1}(V_R - V_{L-1}) - \frac{1}{(R-L+1)^2}(X_R - X_{L-1})^2. \tag{6.17}$$

This makes computation of the variance on any interval simply a difference of pre-computed quantities, drastically reducing run time.

The final piece of the change point problem is the threshold used to decide whether to place a change point at a particular time. This decision making is subject to a perennial issue in model selection: by placing a change point in the data, and thus adding an additional two parameters in another mean and another variance to the model, we will always improve the likelihood. So the question is not whether the score improves by adding parameters, but whether it improves enough to justify adding parameters.

The classic approach to this problem is to use some sort of information criterion instead of a likelihood. An information criterion is a quantity estimating the information lost when describing data with a model. These consist of the negative log likelihood function of parameters given the measured data (the information loss of the model) added to a penalty κ for the complexity (generally dependent on the number of parameters) of the model:

$$\mathrm{IC} = -\log \mathscr{L} + \kappa. \tag{6.18}$$

To use an information criterion for change point decisions, we choose the one with the lowest value of the information criterion. At any step, we are only choosing between two models, so we can rephrase the question as whether the difference in information criterion is less than or greater than 0: we choose model A if $\mathrm{IC}_A - \mathrm{IC}_B <$

0, and model B otherwise. Based on equation 6.18, this difference amounts to the relative log likelihood of the two models, plus the difference in complexity between the two models.

For the change point problem in particular, we calculate this information difference between using the relative log likelihood in equation 6.11 for the most likely change point and compare it to 0. If the information difference is greater than 0, it means that information is lost by adding the new parameters, and we should not add to the complexity of the model by adding a change point.

A common approximation for the information loss is the Akaike information criterion (AIC), where the complexity κ is approximated as the number of parameters in the model d,

$$\text{AIC} = -\log \mathscr{L} + d. \tag{6.19}$$

Since at each change point addition, we are adding two parameters in the mean and standard deviation of the new data segment, this means that we should add a change point at the point

$$t = \arg\max_{\tau} S_\tau \tag{6.20}$$

if $-S_t + 2 < 0$. In other words, the Akaike information criterion suggests we should simply apply a constant threshold $k = 2$ to S_t to determine whether to add a change point.

In practice, this threshold may be changed to make the change point detection more or less sensitive. Instead of the AIC value of $k = 2$, by using a greater threshold $k > 2$ we can we decrease the number of change points found, and by using a smaller threshold $k < 2$ we increase the number of change points found. Note that with

the threshold equal to zero, this is the case where we are applying no penalty to overfitting. Therefore, when $k \leq 0$, the algorithm will continually subdivide the data until every single data point is modeled by its own mean and variance.

The sensitivity may be tuned to accomodate a few effects, generally arising from the inadequacy of certain assumptions being made. First of all, the noise on the data may not be normally distributed, in which case the likelihood function will be inaccurate, and the algorithm will call more steps than really exist. Increasing the threshold will reduce the number of overcalled steps. This can result in acceptable results if all true steps are clearly distinguishable above the level of the noise, but when the transitions are subtler this method will be prone a high rate of mistakes-even when the correct number of change points are identified, many will be in the wrong places. When the noise on the observations is not well described by the model, the correct approach is to return to the model used to calculate the likelihood function and re-evaluate how the MLE parameters are calculated.

Second of all, the AIC is not a precise approximation. Its accuracy relies in part on the likelihood function being twice-differentiable in the neighborhood of the maximum likelihood parameters. In the case of change point detection, although it is suitable for the information contained in the mean of the data segment, this approximation is violated especially by the choice of change point, where the likelihood function is not differentiable.

In general, the correct expression for the complexity depends on the length of the interval being divided. Subdividing a larger interval requires specifying more information, because there are more possible change points. Changing the value of the constant threshold may make perform better on different segment sizes, but this is a clumsy fix. For a general-purpose change point finder, we need a rigorous choice of κ chosen to vary correctly with the number of data points.

Following work[55][56] by Colin LaMont and Paul Wiggins, also at University of Washington, we can arrive at a better estimator for the complexity than the one chosen by the AIC. In general, this can be done by computing the bias in the information loss estimator for every possible value of the parameters. That is, for every possible value of the parameters, we simulate the system for all possible values of the parameters, and determine what the appropriate value of k is so that the algorithm calls neither too few nor too many change points.

For the change point problem, the information loss from the mean and standard deviation are well approximated by the AIC, but the change point indices are not. Therefore, this problem amounts to running this simulation and calculating k for every possible length of the segment in which we are searching for a change point.

However, it turns out that simulating large amounts of nanopore data is not necessary. LaMont and Wiggins show[55][56] that this calculation is equivalent to a simpler one: the estimator for complexity is reducible to

$$k(d, N) = d + \mathbb{E}_B \max_{1 \leq j < N} \left\{ \frac{N}{j(N-j)} B_j^2 \right\}. \tag{6.21}$$

Here, d is the dimensionality of the model for each segment (an AIC-like term), N is the number of data points in which a change point is to be found, B_j is the j'th element of a d-dimensional Brownian bridge, and $\mathbb{E}_B$ indicates the expectation value over all such Brownian bridges. A Brownian bridge is a random walk of length N where each step is uncorrelated with the previous steps, has an identity covariance matrix, and has mean zero, and is conditioned such that $B_0 = B_N = 0$.

Brownian bridges may be generated as the difference between two random walks:

$$B_j = u_j - \frac{j}{N} u_N \tag{6.22}$$

where u_j is an uncorrelated random walk of length N with identity covariance matrix and mean zero. This calculation is easy to perform on a computer using Monte Carlo methods. In practice, nanopore data is segmented by using a piecewise function defined to fit calculations of these functions for varying choices of N, which is evaluated at run time in the segmentation algorithm each time a change point is assessed.

The derivation of this approximation of the complexity and its relationship to the expected maximal value of the Brownian bridge are beyond the scope of this thesis, which aims to provide a how-to guide for readers interested in applying it to nanopore data. More information about this technique is available in the Wiggins lab's publications.

With every component of the algorithm now specified, we can write it in full (algorithms 6.1.2, 6.1.2).

Algorithm 6.3 Minimum information loss change point algorithm.

```
1: procedure findChangePoints(x)
2:     n ← Length(x)
3:     X_1 ← x_1
4:     S_1 ← x_1^2
5:     for i ← 2...n do
6:         X_i ← X_{i−1} + x_i
7:         S_i ← S_{i−1} + x_i^2
8:     end for
9:     t ← partitionData(X, S, 1, n) return t
10: end procedure
```

The algorithm can be extended or changed in many ways, most readily by changing the model used to generate the data. For example, a model in which each segment is a Brownian walk with different characteristics will have an additional parameter for

Algorithm 6.4 Data partitioning function. Finds most likely transition point in the region of data bounded by indices $[l, r]$. Recursively called in the change point detection algorithm.

procedure PARTITIONDATA(X, S, l, r)

  $n \leftarrow$ r-l

  if $n \leq 2$ **then return** $\{\}$
  end if

  for $i \leftarrow l + 1...r - 1$ **do**
    $L \leftarrow i - l - 1$
    $R \leftarrow r - i - 1$
    $V_L \leftarrow \frac{1}{L}(S_i - S_l) - \frac{1}{L^2}(X_i - X_l)^2$
    $V_R \leftarrow \frac{1}{R}(S_r - S_i) - \frac{1}{R^2}(X_r - X_i)^2$
    $V_T \leftarrow \frac{1}{n}(S_r - S_l) - \frac{1}{n^2}(X_r - X_l)^2$
    $H_i \leftarrow L \log V_L + R \log V_R - n \log V_T$
  end for

  $t^* \leftarrow \arg\max_i H$
  $H^* \leftarrow \max_i H$

  if $H < 0$ **then return** partitionData$(X, S, l, t) \cup \{t^*\} \cup$partitionData$(X, S, t, r)$
  elsereturn $\{\}$
  end if

end procedure

each segment, defining the mean difference between each data point in that segment. Another example is the model of a system with time-dependent structure, where the noise is normally distributed about some fitted function other than a constant. This is the case when running nanopore experiments with time-varying voltage, and is discussed at length in §12.2.7.

Chapter 7

REFERENCE ALIGNMENT

In many cases, nanopores are used to read strands of DNA with known sequence. These include k-mer map generation, control experiments, and SPRNT experiments, in which the enzyme rather than the DNA is the object of study. In these cases, it is often necessary to compare nanopore observations to a *reference* sequence of predicted states. This comparison is typically done by first matching each observation to a single reference state that is likely to have produced it.

The matching procedure, which amounts to a special case of solving a hidden Markov model as discussed in §5.2.2, is called *reference alignment*, in analogy to the alignment procedure used to relate DNA or protein sequences. A reference alignment is represented mathematically and *in silico* by a sequence of integers $\{a_1,\ a_2,\ ...,\ a_N\}$, where $1 \leq a_i \leq M$, with N being the the length of the sequence of observations being aligned to a reference sequence of length M. Here, a_i is the index of the state in the reference sequence that the ith observed state is to be matched to. If the reference alignment allows for the possibility than an observation does not match to any reference state, a_i can also take a null value, which in this text will be written $\varnothing$, and in software implementations is generally represented by a NaN (not-a-number) value.

Reference alignment of nanopore reads is significantly different in a few respects from Needleman-Wunsch alignment algorithms used for aligning discrete sequences of DNA bases or amino acids. The first of these is match scoring, the way we assign the likelihood that one observation was drawn from a particular reference state. This is

explored in §7.1. The second of these is the choice of transition probabilities between reference states, explored in §7.2. Finally, because we are aligning to a reference, there is an asymmetry to this alignment that is not present in the nucloetide or amino acid alignments, where both sequences are on the same footing- a gap in one sequence is a deletion in another, and vice versa. Finally, considerations on how to represent a reference sequence as opposed to an observed sequence are provided in §7.3.

7.1 Match scoring

Segmented states are represented using statistics calculated from the segment of data they are identified with. These statistics depend on the model used to segment the states, and may include for example a mean, standard deviation, slope or drift rate, autocorrelation, or amplitudes of basis functions. The maximum likelihood estimates of these all these statistics have normally distributed and sometimes correlated errors.

These statistics have normally distributed errors because they are each computed as sums of independent, identically distributed random variables, so for any segment of sufficient length, the central limit theorem indicates that a normal distribution will accurately reflect the uncertainty in the maximum likelihood estimates of these statistics. Reference states are in turn represented by average statistics of many observations, and so the errors in reference state statistics are also normally distributed.

We therefore represent both observations and reference states with a vector consisting of these statistics. Each observation is paired with the Fisher information matrix representing the uncertainty in the maximum likelihood estimates of its statistics, and each reference state is paired with the inverse of the covariance matrix representing the distribution of observations it produces. Informally, these matrices can both be thought of as inverse covariance matrices representing the distribution of values associated with the observed and reference states respectively.

To solve the hidden Markov model, as shown in §5.2.2 we need the match likelihoods, $p(X = \boldsymbol{x}|\mathcal{S})$. This may be considered the probability that the true value of the observation statistics estimated by $\boldsymbol{x}$ with Fisher information K_x is drawn from the distribution for state $\mathcal{S}$, which has mean $\boldsymbol{y}$ and inverse covariance K_y. This may be written integrating over "true" values of X as

$$p(X = \boldsymbol{x}|\mathcal{S}) = \int p(\boldsymbol{x}|\boldsymbol{z}, K_x)p(\boldsymbol{z}|\boldsymbol{y}, K_y)\ \mathrm{d}^d z. \tag{7.1}$$

Using Bayes' theorem on the first term in the integral, we obtain

$$p(X = \boldsymbol{x}|\mathcal{S}) = \int \frac{p(\boldsymbol{z}|\boldsymbol{x}, K_x)p(\boldsymbol{x}|K_x)}{p(\boldsymbol{z}|K_x)} p(\boldsymbol{z}|\boldsymbol{y}, K_y)\ \mathrm{d}^d z. \tag{7.2}$$

The prior probability of any particular observation should be identical to the prior probability of any particular true value, so $p(\boldsymbol{x}|K_x) = p(\boldsymbol{z}|K_x)$ and the Bayes terms cancel, leaving

$$p(X = \boldsymbol{x}|\mathcal{S}) = \int p(\boldsymbol{z}|\boldsymbol{x}, K_x)p(\boldsymbol{z}|\boldsymbol{y}, K_y)\ \mathrm{d}^d z. \tag{7.3}$$

As stated above, the two terms in the integral are normal distributions:

$$p(X = \boldsymbol{x}|\mathcal{S}) = \int \sqrt{\frac{\det K_x}{(2\pi)^d}} \sqrt{\frac{\det K_y}{(2\pi)^d}} e^{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{x})^T K_x (\boldsymbol{z}-\boldsymbol{x}) - \frac{1}{2}(\boldsymbol{z}-\boldsymbol{y})^T K_y (\boldsymbol{z}-\boldsymbol{y})}\ \mathrm{d}^d z \tag{7.4}$$

$$\begin{aligned} = \sqrt{\frac{\det K_x \det K_y}{(2\pi)^{2d}}} e^{-\frac{1}{2}\left(\boldsymbol{x}^T K_x \boldsymbol{x} + \boldsymbol{y}^T K_y \boldsymbol{y}\right)} \\ \int e^{-\frac{1}{2}\left(\boldsymbol{z}^T (K_x + K_y)\boldsymbol{z} - \boldsymbol{z}^T (K_x \boldsymbol{x} + K_y \boldsymbol{y}) - (K_x \boldsymbol{x} + K_y \boldsymbol{y})^T \boldsymbol{z}\right)}\ \mathrm{d}^d z. \end{aligned} \tag{7.5}$$

because the matrices K_x and K_y are symmetric, the second and third terms in the exponential inside the integral are identical, and the expression reduces to

$$p(X=\boldsymbol{x}|\mathcal{S}) = \sqrt{\frac{\det K_x \det K_y}{(2\pi)^{2d}}} e^{-\frac{1}{2}\left(\boldsymbol{x}^T K_x \boldsymbol{x} + \boldsymbol{y}^T K_y \boldsymbol{y}\right)} \int e^{-\frac{1}{2}\left(\boldsymbol{z}^T (K_x+K_y)\boldsymbol{z} - 2(K_x\boldsymbol{x}+K_y\boldsymbol{y})^T \boldsymbol{z}\right)} \, \mathrm{d}^d z. \tag{7.6}$$

This is a well-known gaussian integral, and carrying it out we obtain

$$p(X=\boldsymbol{x}|\mathcal{S}) = \sqrt{\frac{\det K_x \det K_y}{(2\pi)^{2d}}} e^{-\frac{1}{2}\left(\boldsymbol{x}^T K_x \boldsymbol{x} + \boldsymbol{y}^T K_y \boldsymbol{y}\right)} \sqrt{\frac{(2\pi)^d}{\det(K_x+K_y)}} e^{\frac{1}{2}(K_x\boldsymbol{x}+K_y\boldsymbol{y})^T (K_x+K_y)^{-1}(K_x\boldsymbol{x}+K_y\boldsymbol{y})} \tag{7.7}$$

$$= (2\pi)^{-d/2} \sqrt{\frac{\det K_x \det K_y}{\det(K_x+K_y)}} e^{-\frac{1}{2}\left(\boldsymbol{x}^T K_x \boldsymbol{x} + \boldsymbol{y}^T K_y \boldsymbol{y} - (K_x\boldsymbol{x}+K_y\boldsymbol{y})^T (K_x+K_y)^{-1}(K_x\boldsymbol{x}+K_y\boldsymbol{y})\right)}, \tag{7.8}$$

which is a function only of observed and reference statistics. The log-likelihood used in solving the hidden Markov model may then be expressed as

$$\log \mathscr{L} = -\frac{1}{2}\left[d \log 2\pi + \log \frac{\det(K_x+K_y)}{\det K_x \det K_y} + \boldsymbol{x}^T K_x \boldsymbol{x} + \boldsymbol{y}^T K_y \boldsymbol{y} - (K_x\boldsymbol{x}+K_y\boldsymbol{y})^T (K_x+K_y)^{-1}(K_x\boldsymbol{x}+K_y\boldsymbol{y}) \right]. \tag{7.9}$$

This expression simplifies considerably when the K matrices are diagonal, reducing to

$$\log \mathscr{L} = -\frac{1}{2}\left[d \log 2\pi + \log \det \Sigma^{-1} + (\boldsymbol{x} - \boldsymbol{y})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{y}) \right], \tag{7.10}$$

where $\Sigma^{-1} = (K_x^{-1} + K_y^{-1})^{-1}$ is the combined covariance. In many cases the diagonality of the K matrices is a given. When we are only comparing one statistic, $d = 1$ and clearly the K matrices must be diagonal because they are 1×1. In fact, in the 1D case the expression simplifies further to

$$\log \mathscr{L} = -\frac{1}{2}\left[\log 2\pi + \log \sigma^2 + \frac{(x-y)^2}{\sigma^2} \right], \tag{7.11}$$

where $\sigma^2 = \sigma_x^2 + \sigma_y^2$ is the combined variance of the observation and state.

It is frequently the case that we are aligning one-dimensional data, as when analyzing constant-voltage experiments we typically only align mean ion current values for each observed state.

Even when analyzing higher-dimensional descriptions of states, many commonly computed statistics have no covariance with one another. For example, the errors in the maximum likelihood estimates of the principle component amplitudes, the mean value, and the magnitude of the noise are all uncorrelated. This means we can use the simpler match score 7.10 even for variable voltage experiments as discussed in §12.

A further refinement can be made to these scores. The expression given above is arrived at given no information about which other possible matches are available: essentially, we are adopting a uniform prior on the true value of the observation. We can convert the likelihood into a proper probability if we consider the match probability relative to all other possible states in the reference we match to. In the case of aligning a sequence of observations $\{x_1, x_2, ..., x_N\}$ to a reference sequence of

states $\{y_1, y_2, ..., y_M\}$, we should use a relative match probability

$$P_{ij} = \frac{\mathscr{L}_{ij}}{\sum_{k \neq j} \mathscr{L}_{ik}}, \tag{7.12}$$

where $\mathscr{L}_{ij}$ is the match likelihood derived above between observation i and reference state k. In an HMM used for sequencing, the sum in the denominator should instead be over all k-mer map states except the one in the numerator.

Intuitively, using these relative probabilities reflects that we have greater confidence when matching to states without any degeneracy. For example, if I observe a 30 pA mean ion current, but my reference contains 20 states with mean ion current very close to 30 pA, the match to each of these states will be of lower confidence than if there was only a single mean ion current close to 30 pA in the reference. In other words, a low match score can still lead to a confident identification of a match, as long as all the other candidate matches have significantly worse scores, but this is only taken into account using the relative match probability.

7.2 Transition penalties

In addition to determining the match likelihoods, in order to obtain a complete solution for the HMM we also need a way of calculating the transition matrices. This section provides a prescription for these calculations.

7.2.1 Ordering of reference states

For simplicity, a reference sequence of k-mers is represented in the sequential order of the k-mers appearing on the DNA. For example, the DNA sequence ACCGTCG consists of the sequence of 4-mers

{ACCG, CCGT, CGTC, GTCG},

which correspond in turn to the sequence of ion current means and uncertainties

$$\{0.534(9),\ 0.459(24),\ 0.246(15),\ 0.372(21)\}$$

in units of fraction of open pore ion current[1].

In some situations, such as when analyzing reads performed in new experimental conditions, when studying an enzyme with unknown stepping behavior, or when sequencing an analyte other than canonical DNA bases, the reference cannot be built from the known DNA using a k-mer map. In these situations, the consensus sequence (see §8 for more information) is used as a reference.

In either case, a reference alignment uses a reference sequence which contains no duplicate states, contains every state that may be observed, and the ordering of states is the most likely order of their first observation.

7.2.2 Translational invariance of transition probabilities

In general, the transition matrix can specify transition probabilities between any two states. However, in practice, there are only a few likely transitions out of each state. With the ordering of states as described above, this can be thought of as each state transitioning only to those nearby in the reference sequence. Intuitively, this is because the motor enzyme controlling the DNA does not typically make large leaps along the DNA, and it is highly unlikely that a large number of sequential observations will be completely missed.

Under most circumstances in which we choose the previously defined ordering for the reference sequence, it can be approximated that probabilities in the transition matrix T_{ij} can be reduced to numbers of the form T_{j-i}, depending only on the size

[1]These values are for DNA fed 5' end first into forwards-orientation MspA at room temperature, 500 mM *trans* well KCl, 150 mM *cis* well KCl, 180 mV applied voltage.

$j-i$ of the step taken through the reference to get from state i to state j. T_1 can then be thought of as the probability of the enzyme taking a regular single step forward through the reference. T_2 is the probability of either the enzyme slipping more than one step, or the data segmentation missing an observation, each of which results in a transition from a reference state to the state two positions after it in the reference sequence. T_0 is the probability that a state is observed twice in a row due to an incorrect segmentation of a single true state into multiple observed segments. T_{-1} will be the probability that the enzyme steps backwards by one nucleotide. In a sense, this is a statement that enzyme behavior and errors in data segmentation are both invariant with respect to translations along the DNA strand.

Realistically, this is not perfectly true. Many enzymes have sequence-dependent behavior, and data segmentation is more likely to miss those state transitions that have a low signal-to-noise ratio, and introduce erroneous state transitions on noisy states. But as nanopore experiments often operate without prior knowledge of these effects, the maximum likelihood estimates for the T_{ij} elements reflect the transition probabilities averaged over all reference states, which are translation invariant over the reference position and can be represented using the formalism above.

Because of the connection between the elements T_{i-j} and the stepping behavior of the enzyme, these numbers are often referred to as *step probabilities*[2]. Specifically, T_1 is the probability of a "forward step" or simply "step," T_0 is the probability of a "hold," T_2 is the probability of a "skip" (with T_3, T_4, etc. called "skip 2," "skip 3," and so on) and T_{-1} is the probability of a "backstep" (again with T_{-2}, T_{-3} being called "back 2," "back 3," and so on).

Exceptions to this model include, for example, SPRNT experiments studying transitions into and out of states with unusual behavior. One such experiment is the study

[2]The term *step probabilities* may also include the probability of a "bad" state, an observation not corresponding to any reference state.

of the arrested state of the E.Coli RNA polymerase at a pause site. Transitioning through this state involves traversing several substates with more complex transition probabilities that are not identical to the typical behavior of the enzyme on arbitrary non-pause-inducing DNA sequences, so calculating the transition probabilities based only upon the step size is not appropriate in this case. In such situations, it may be necessary to empirically determine the step probabilities for the reference states in question separately from the rest of the transition matrix.

Another case to consider is when the observations tell us something about the stepping behavior of the enzyme. This is the case in variable-voltage experiments, which provide an advantage to sequencing in part because of the information the observed data provide about the motor enzyme's stepping. This can also be the case in constant voltage data. For example, observations with larger noise make short states less likely to be observed. Therefore, an experimenter can relate the noise in a observation to the probability of a missed observation, seen in the transition matrix as an increased likelihood of a step size greater than 1. In these cases, a separate transition matrix is calculated for each observed state, so transition probability $T_{ij} = P(\mathcal{S}_t = i|\mathcal{S}_{t-1} = j)$ in equation 5.19 is replaced by T^t_{ij}, conditioning it upon which observation t we are considering transitions into. This calculation is discussed further in section §12.2.11 about methods for variable voltage data analysis.

7.2.3 Calculation of step probabilities

The step probabilities can be calculated empirically using a training set. With a sufficiently large training set, the full matrix T_{ij} may be computed using maximum likelihood estimates for each entry, as discussed in §5.2.2 covering the calculation of solutions to hidden Markov models. In practice, though, this is rarely an option. Training sets consist of reads that are confidently aligned to a reference, and this align-

ment often must be. It is much more reasonable to obtain enough reads to estimate the average stepping behavior of the enzyme, computing the average, translationally invariant stepping probabilities T_{j-i}.

To compute T_{j-i} from a set of well-aligned observations, we first compute the step sizes seen in the alignments by taking the difference between successive values of the alignment indices a_i after removing null values. In other words, we find the alignment index sequence with unaligned states removed, $\tilde{a} = \{a_i \in a | a_i \neq \varnothing\}$, and compute differences $\Delta_i = \tilde{a}_i - \tilde{a}_{i-1}$. If there are M elements in the sequence $\tilde{a}$, we can compute Δ_i from $i = 2$ to $i = M$. We can then compute probabilities of step types T_{j-i} as

$$T_k = \frac{N(\Delta_i = k)}{M}, \tag{7.13}$$

where $N(\Delta_i = k)$ indicates the number of elements of Δ equal to k. Construction of the entire transition matrix is then straightforward; we set $T_{ij} = T_{j-i}$.

Affine step probabilities

In experiments where enzyme "skips" are extremely unlikely, such as those where the enzyme is very tightly bound to the DNA or is pulling the DNA up through the nanopore, T_{j-i} for $j - i \geq 2$ is dominated by errors in the data segmentation, where states are simply too short or noisy to be accurately identified and segmented. This is similarly the case for the $j - i \leq -1$ elements of T_{j-i} in experiments where the DNA is being let towards the *trans* side of the pore by the enzyme.

In these cases, since the absence of observations is caused only by a uniform effect over all data, the probability that any given state is not observed is equal[3]. In this case, we can obtain a better and more stable estimate of the step probabilities, especially

[3]Presuming we have no prior knowledge of the accuracy of the data segmentation algorithm on each part of the observed data.

extrapolating to unlikely steps of which there may be zero observations in the training set. This effect leads to an exponential falloff of probabilities of larger and larger steps in each direction. In other words, if p is the probability of an forward step occurring but being missed, we can write that $T_{j-i} = T_1 p^{j-i-1}$ for $j - i \geq 2$, and if q is the probability of a backwards step occurring but being missed, then $T_{j-i} = T_{-1} q^{i-j-1}$ for $j - i \leq -2$. p and q can be found by computing the maximum likelihood exponential fit of these functions to the values of T_{j-i} computed using equation 7.13. p and q may then be used to compute the more robust values for the estimators of T_{j-i}. It is important that this fitting take into account the normalization condition,

$$\sum_{k=-k_{\max}-1}^{k=k_{\max}+1} T_k = T_0 + T_1 \sum_{k=0}^{k_{\max}} p^k + T_{-1} \sum_{k=0}^{k_{\max}} q^k = 1, \tag{7.14}$$

where $k_{\max}$ is the maximum number of sequential missed observations considered by the alignment algorithm. In principle, $k_{\max}$ can be infinite, but in practice to simplify computation, it is limited to a value of 10 or less, as for reasonable values of p or q in nanopore experiments[4], steps as large as 10 are already prohibitively unlikely.

7.3 Reference sequence construction

To construct a reference sequence, we need to determine the distribution of observed ion current values corresponding to each state in the reference. This is done using a weighted mean and standard deviation calculation.

In general, when training a hidden Markov model, we may not be completely certain which state an observation should be assigned to. However, even absent absolute certainty, we often have estimates of state probabilities for each observation. For example, the BCJR algorithm (§5.2.3) outputs the probability the ith observation

[4]These numbers are almost never larger than approximately 0.2.

came from the jth reference state, so if using a BCJR algorithm to align an observed sequence to a reference sequence, these probabilities can be used to weight the means.

Including both the probabilistic weights as well as the traditional weighting by the inverse covariance (or more precisely, by the Fisher information) of the observation uncertainty, we obtain that for a set of n observations with estimated means $\{\mu_1,\ ...,\ \mu_n)$ and standard deviations $\{\sigma_1,\ ...,\ \sigma_n)$ that are identified with a state with probability $\{p_1, ..., p_n\}$, we can then estimate the distribution that generated all of these observations as having mean μ and standard deviation σ such that

$$\mu = \frac{\sum_{i=1}^{n} \frac{p_i}{\sigma_i^2}\mu_i}{\sum_{i=1}^{n} \frac{p_i}{\sigma_i^2}}, \tag{7.15}$$

$$\sigma^2 = \frac{\sum_{i=1}^{n} \frac{p_i}{\sigma_i^2}(\mu_i - \mu)^2}{\sum_{i=1}^{n} \frac{p_i}{\sigma_i^2}} + \frac{\sum_{i=1}^{n} p_i\sigma_i^2}{\sum_{i=1}^{n} p_i}. \tag{7.16}$$

Chapter 8
CONSENSUS GENERATION

The data from a single enzyme-controlled read with a nanopore will suffer from pathological issues that are fundamental to the nanopore experiment. Those errors which are systematic may be removed by thorough and accurate modeling of the system; for example, by using a sequencer that accounts for different types of enzyme steps, and by using a well-trained k-mer map with sufficiently large k. However, many of these pathologies are random, differing from read to read. Random errors in reads may be caused by enzyme steps that are too short or noisy to be detected by the change-point algorithm, enzyme backsteps or slips, or the read beginning at an intermediate point along the strand of DNA. Because these errors are different for each read, they may be identified and eliminated by cross-referencing the information from multiple reads. A nanopore read free of random error is called a consensus.

8.1 Consensuses for different applications

Consensuses have uses in both sequencing and SPRNT analyses, although the workflow and methods used to obtain a consensus differ depending on the application the consensus is being used in, and and the information available to the experimenter.

8.1.1 Sequencing consensuses

Randomness of observed enzyme state transitions reduces the accuracy of sequencing. This is for two reasons: firstly, missed ion current states ("skips") result in information being deleted from the sequence. Relatedly, increasing the Shannon entropy of the

transition matrix in the Markov model,

$$H = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}\log T_{ij}, \tag{8.1}$$

makes the transition probabilities less informative, providing less of a tight constraint to the final maximum-likelihood base calls. For example, if a read were known to take only single-nucleotide forward steps, any k-mer state will only transition to one of four possibilities, generally reducing the number of k-mers with similar ion current being compared, and therefore reducing the number of base calling errors. At the other extreme, if backsteps and skips are large and very frequent, to the extent that transitions to any k-mer are equally likely, a sequencer will be unable to use the Markovian structure of the data and will just call each k-mer based on a single enzyme state observation. Given the variation in ion currents associated with each k-mer, this would lead to very large sequencing errors.

Sequencing a set of averaged observations that contains one observation corresponding to each k-mer in the true state sequence, and through which the stepping probabilities are very specifically known (i.e., T_{ij} is a low-entropy distribution of state transition probabilities), eliminates this source of uncertainty. Even combining two reads can eliminate some random error and improve sequencing, and the error rate improves asymptotically to a minimum value caused by systematic errors in the sequencing.

Consensuses for *de novo* sequencing (and relatedly, for k-mer map generation in new conditions for which an sufficiently accurate prediction does not exist) are challenging to construct because the underlying sequence is completely unknown. We refer to the process of building a consensus of the sequence of observations for a particular DNA strand and enzyme with no prior knowledge about its structure *de novo* consensus generation. These methods are beyond the scope of this thesis,

but the nanopore lab has had success generalizing approximate methods to nanopore data[57].

8.1.2 SPRNT consensuses

Obviously, it is not meaningful to study protein behavior using a consensus, because all time and enzyme step information has been removed from such a sequence. However, in order to understand how enzyme behavior varies from state to state, it is necessary to assign a true state to each observation. This way, the base sequence in the enzyme can be inferred. Additionally, in cases where SPRNT is capable of observing multiple enzyme substates, the substate corresponding to each observation can be determined.

Grouping observations by enzyme state is necessary because of the stochastic nature of single molecule physics. State duration, for example, is described by a probability density function of durations, and state transitions are described by probabilistic branching ratios. Many measurements of each state and each kinetic pathway are needed in order to characterize these distributions, so accurate identification of the enzyme state for each observation is necessary. Enzyme states are identified through reference alignment (as in §7) to a consensus. An accurate consensus is required to do this with confidence. so consensus generation is always the first step to detailed SPRNT analysis.

As often as possible, SPRNT is performed in conditions for which an accurate k-mer map has been produced, and almost always uses a template DNA of known sequence. This means that the observed sequence of ion currents may be predicted, and small corrections to systematic errors made by using the measurements to update the consensus. These consensuses are typically what we refer to as "seeded consensuses" and are further described in §8.2.

In certain situations, however, SPRNT consensuses are required to be constructed *de novo*. This is necessary, as mentioned in the previous section, when constructing a k-mer map for the first time in new conditions. It may also be necessary when the stepping behavior of the enzyme being studied is unknown. For example, it isn't possible to seed a consensus for an enzyme which has two observable steps in SPRNT with a single-step prediction, because the number of states will not match.

8.2 Seeded consensus generation

Seeded consensus generation is the process typically used to generate consensus reads of DNA strands of known sequence. In seeded consensus generation, we initialize the consensus with a guess or *seed*, and iteratively align reads and update the consensus. Seeded consensus generation is thus an expectation-maximization algorithm, with the likelihood-maximizing step being the solution of the alignment's hidden Markov model.

To generate a consensus, a number of nanopore reads must be obtained. The number of reads required can vary depending on the error rate in the reads: enzymes with highly uncertain stepping behavior, or those which step rapidly leading to short observations of each state could require as many as 30 reads to generate a reasonably confident consensus. On the other hand, it is possible to generate consensuses from low error-rate data using highly rectified enzymes that progress slowly along the DNA with only a few reads.

To begin, the consensus is set equal to the seed generated using a k-mer map. The ion current values in the consensus and their variances comprise the true state parameters of the HMM that are updated at each EM step. After collecting and segmenting the reads into observed states, each read is aligned using reference alignment to the consensus. The consensus is then updated by replacing the parameters of each

state with the mean and variance of those observations that were aligned to the that state.

8.2.1 Simulated annealing

If an initial guess is too inaccurate, the EM algorithm will not converge to a meaningful consensus. Highly erroneus reference states with small variances will never have any measurements aligned to them, and so will not be updated. It is not possible to fix this simply by placing a large variance on the reference sequence. This causes too many measurements to align to the state with large variance, making its own updated value revert to the mean and variance of the measurements. This effect also reduces the number of accurate alignments to other states. Ultimately, a false minimum is found where every measurement is aligned to one or a few large-variance reference states.

It is possible to address this problem by using small errors even for reference states whose means are highly uncertain by using a common tactic in optimization called simulated annealing. After each iteration of the expectation maximization, in which the parameters (consensus reference state means and variances) are updated, an additional noise term is applied, randomly perturbing the parameters. At each EM step, the magnitude of the noise is reduced (i.e., the "temperature" is lowered). This is in analogy to the metallurgical process of annealing, in which a material slowly brought down from high temperature is better able to achieve a low-energy, stress-free configuration.

Simulated annealing can be adapted in several ways. One option is to improve performance by avoiding overlarge or unnecessary perturbations to already confidently converged states. The perturbations ξ_n at step n can be made inversely dependent on the confidence in the correctness of a state, so states with larger variance in aligned

measurements or states that are infrequently aligned to will have larger perturbations.

8.2.2 Dependence on seed

Using the above algorithm for seeded consensus generation, it is clear that not any choice of seed will yield a reasonable consensus, even with many high-quality measurements. Seeds that do not resemble the measurements whatsoever will not generally converge. Poorly chosen transition probabilities can also prevent meaningful convergence just as they can with any other reference alignment. Additionally, to obtain an accurate consensus it is important that the seed contain the correct number of reference states, as the seeded consensus generation algorithm does not add or remove states. Incorporating an *ad-hoc* state-addition and state-removal mechanism is not advisable, as a more complete and accurate solution is provided through algorithms for *de novo* consensus generation.

Chapter 9

K-MER MAP

The k-mer map is central to most of the analysis carried out on nanopore data at the University of Washington. It forms the link between the true sequence of DNA bases passing through the nanopore and the observations of ion current. Therefore, it is important to carefully determine precisely how to construct such a map. The motivation for the use of k-mers in nanopore data analysis was introduced in §4.3, which also provides some motivation for a natural choice of k. The present chapter endeavors to build on this motivation to describe how in practice a k-mer map is designed and constructed.

9.1 Properties of the k-mer map

The k-mer map has one entry for every possible k-base word using the alphabet consisting of {A, G, C, T}. For computational and mathematical convenience the entries may be alphabetized such that each may be assigned an index, such that for example AAAA would be the 4-mer with index 1, AAAT would be the 4-mer with index 4, and TTTT would be the 4-mer with index 256.

Each of these entries contains some statistics describing the observations seen in nanopore reads when that k-mer is in the constriction of the pore. At minimum, this is usually the mean or median ion current associated with that k-mer, although it often also may include the noise level for constant-voltage data. Higher dimensional data may be obtained in more sophisticated experimental setups, such as when applying variable voltage waveforms as described in §12. The distributions of these statistics

over observations is typically approximately normal, but potentially covariant, in nanopore data. Therefore, they are well-described by normally distributed random vectors, and we associate each k-mer with a mean vector and a covariance matrix.

9.2 Map training using expectation maximization

Training a k-mer map amounts to an expectation maximization algorithm very similar to seeded consensus generation (described in §8.2).

It differs firstly in the data requirements. It is necessary to obtain a *training set* consisting reads of sufficient quantities of different sequences such that every k-mer in the map is represented in the observations within those reads. Unlike in seeded consensus generation, the reads do not need to be of the same strand of DNA. To fulfill these requirements, it is sufficient to simply fragment a large and widely available known DNA sequence, such a genome. This is the approach used with the PhiX 174 viral genome and pBlue plasmid in Laszlo 2014[24] to generate the 4-mer map using Phi29 DNA polymerase, and using the lambda phage genome in Noakes 2019[58] to generate the variable-voltage 6-mer map using Hel308 DNA helicase (§12). PhiX 174 and pBlue were also used to generate the backwards pore 4-mer map (§11).

As in seeded consensus generation, the reads are aligned at each iteration to a reference. However, instead of updating the reference states directly, the k-mer map is updated, and the reference is regenerated from the new k-mer map at each iteration. To update the k-mer map, the new mean and variance equally weight the reference state means, rather than the observation means, to avoid biasing the k-mer map based on differing levels of read coverage of different portions of the training set.

Chapter 10

DE NOVO SEQUENCING

De novo sequencing of DNA is the determination of its base sequence without any prior knowledge of any part of that sequence. This is to be contrasted with reference sequencing, in which isolated variations in base sequences are to be identified. *De novo* sequencing is challenging with nanopore reads because the information about each base is distributed over many observations. The typical approach is, similarly to reference alignment, to model the system with a hidden Markov model to match observations (the sequence of measured ion current states) to true states (the k-mer subsequences). Also just as in reference alignment, no iterative expectation maximization procedure is necessary for sequencing, because the transition probabilities between states are known (based on enzyme kinetics and the transition restrictions enforced by the k-mer model).

Another approach is to use a machine learning algorithm that includes little information about the model of the pore. Recurrent neural networks (RNNs) have been used to generate sequences of similar quality to HMMs. RNNs require substantially more training than HMMs (on the order of megabases rather than kilobases), which is infeasible for a laboratory without a high-throughput nanopore sequencer. Additionally, RNNs function as "black boxes" that output DNA sequence given nanopore input, and for that reason are difficult to troubleshoot or modify without acquiring a new data set and completely retraining the network.

Typically, the objective of sequencing is to minimize the per-base error rate, because this makes letter sequence alignment and protein sequence prediction more

accurate. The BCJR algorithm discussed in §5.2.3 is technically the correct choice, because it calls bases minimizing this per-base error rate. In contrast, the Viterbi algorithm in §5.2.4 finds the maximum likelihood sequence, minimizing the likelihood that there is even one mistake in the called sequence.

However, to save computation time and more importantly, to incorporate the capability for handling spurious bad observations, it is desirable to ultimately use the modified Viterbi algorithm described in §5.2.4.

10.1 Sequencing with a hidden Markov model

The likelihood function used for match scoring in a sequencer is identical to the one described in §7.1, as the reference states consist of the entire k-mer map, the same one used to generate reference states when aligning to known sequence. The primary difference is in the transition matrix defining the probability of transitioning from one k-mer state to another.

These transition probabilities T_{ij} can be related to the probability of a missed observation p_{skip}. Specifically, to calculate T_{ij} we need to enumerate all the ways k-mer i can transition to k-mer j. For example, the 4-mer AAGA could be followed by the 4-mer AGAC as a result of a step forward by one nucleotide, a step forward by three nucleotides meaning that two observations were missed, or a step forward by greater than three nucleotides, which will completely replace the nucleotides in the 4-mer. To illustrate the way these step sizes are consistent with those two k-mers, we can write them as overlapping sequences, along with the true sequence that generated those observations:

```
0 missed observations:
k-mer 1                      AAGA
k-mer 2                       AGAC
```

```
combined sequence           AAGAC

2 missed observations:
k-mer 1                     AAGA
k-mer 2                        AGAC
combined sequence           AAGAGAC

3+ missed observations:
k-mer 1                     AAGA
k-mer 2                         AGAC
combined sequence           AAGAAGAC
```

Because each of these numbers of missed observations is a distinct and disjoint possibility, the probability of this k-mer transition should be the sum of the probabilities of each times the uniform prior probability of each transition, $1/4^k$, in this case equal to $1/256$. Therefore,

$$T_{\text{AAGA, AGAC}} = \frac{A}{[}p(0 \text{ missed observations}) + p(2 \text{ missed observations}) + p(3+ \text{ missed observations})], \tag{10.1}$$

where A is a proportionality constant chosen to ensure that the sum along each column of T is normalized to 1. The set of possible numbers of missed observations associated with each transition is a property of the k-mer map only. If the probability of a missed observation can be given a more sophisticated, nonuniform prior, as it is in the variable voltage sequencing technique described in §12, this means that for each observation the transition matrix T must be recalculated.

Now that we have defined the quantities of interest in the HMM, we can assign k-mer states to observations using a modified Viterbi algorithm, detailed in algorithm 10.1.

However, our work is not done. Even provided the k-mer identities of each observation, we must determine how to stitch them together to obtain a final sequence. To do so, we simply assume the most likely transitions occurred between k-mers corresponding to successive good observations. Define the matrix K whose elements K_{ij} is the jth base in the ith k-mer in the k-mer map used for sequencing, and define $K_{i,j\ldots k}$ as the string consisting of the jth through kth bases in k-mer i. Additionally, define a pre-computed matrix v_{ij} of the most likely number of missed observations in a transition from k-mer i to k-mer j. Finally, define s as the final sequence being computed, and $s_{i\ldots j}$ as the subsequence from the ith to jth bases. With these definitions, algorithm 10.1 constructs the final called base sequence.

Algorithm 10.5 k-mer assignment algorithm for *de novo* sequencing.

procedure ASSIGNKMERS(x, y, t, p)

Compute the score matrix and initialize the alignment matrix from the observations x, the k-mer map y, and the bad observation probability p.

$n \leftarrow \text{Length}(x)$
$m \leftarrow \text{Length}(y)$

for $i \leftarrow 1...n;\ \ j \leftarrow 1...m$ **do**
    $s_{ij} \leftarrow \log \mathscr{L}(x_i|y_j) + (1 - p_i)$
end for

for $j \leftarrow 1...m$ **do**
    $h_{1j} \leftarrow s_{1j}$
end for

Fill out the alignment matrix by assuming that the model transitioned from the most likely state at each step. Keep track of which state this was for each true/measured state pair. $t_{i,i-l,kj}$ is a 4-index array of the log transition matrix t_{kj} evaluated using the probability of a missed observation between observations $i - l$ and i in the observed sequence. Track bad observations by looking back multiple rows in h and using the appropriate bad observation probabilities.

for $i \leftarrow 2...n$ **do**
    for $j \leftarrow 1...m$ **do**
        $h_{ij} \leftarrow s_{ij} + \max_{k,1\le l\le i-1} \left\{ (t_{i,i-l,kj} + h_{i-l,k}) \sum_{u=i-l+1}^{i-1} p_u \right\}$
        $r_{ij} \leftarrow \arg\max_{k,1\le l\le i-1} \left\{ (t_{i,i-l,kj} + h_{i-l,k}) \sum_{u=i-l+1}^{i-1} p_u \right\}$
    end for
end for

Starting at the most likely final k-mer, walk back through the matrix by recalling the most likely transition each time.

$a_n \leftarrow \arg\max_j h_{nj}$

for $i \leftarrow n - 1...1$ **do**
    $a_i \leftarrow r_{i+1,a_{i+1}}$
end for

Calculate and return the approximate maximum likelihood true states and the approximate total log-likelihood of the model.

$l \leftarrow \sum_i h_{i,a_i}$

return $\mathbf{a}, l$

Algorithm 10.6 *De novo* sequence reconstruction algorithm.

1: **procedure** CONSTRUCTSEQUENCE($\boldsymbol{a}$)

2: $n \leftarrow Length(\boldsymbol{a})$
3: $s_{1...k} \leftarrow K_{a_1,1...k}$
4: $j \leftarrow k+1$

5: **for** $i \leftarrow 2...n$ **do**
6: $s_{j...j+v_{a_{i-1}a_i}+1} \leftarrow K_{i,(k-v_{a_{i-1}a_i})...k}$
7: **end for**

8: **return** s
9: **end procedure**

Part IV

EXPERIMENTAL METHODS

In this part, I describe more specific, less general-purpose contributions to the field of nanopore science than in part . These are particularly aimed towards improving the accuracy of DNA sequencing, but also have applications in SPRNT.

Nanopore sequencing has been limited by a low single-passage de novo sequencing accuracy relative to other established sequencing platforms. High accuracy de novo nanopore sequencing can be achieved by combining multiple high-error nanopore reads into low-error consensus sequences. However, this approach exchanges throughput for accuracy, and is still fundamentally limited by systematic errors. Additionally, some nanopore applications, such as pathogen detection at low concentrations and metagenomics studies, are most effective if they are able to identify a single molecule of DNA with only one read. Therefore, the path toward fully realized nanopore sequencers requires improvement of the baseline single-passage accuracy.

Many of the single-passage sequencing errors can be attributed to two primary error modes: distinct sequences with indistinguishable conductance signals and irregular steps by the motor enzyme. To decode an observed signal, the base calling algorithm must use the k-mer map, a model of the blockaded pore conductance that maps observed conductance values to the likely generating DNA sequence. The conductance through the nanopore is influenced by several nucleotides near the constriction of the pore, resulting in a complex map of conductance to the underlying sequence. In many instances, different sequence segments generate statistically indistinguishable conductance values, thereby leading to error-prone base calls.

Irregular stepping by the DNA-controlling motor enzyme can also introduce sequencing errors. Ideally, the enzyme would move DNA unidirectionally through the pore in discrete steps of uniform length. However, the stochastic stepping of enzymes frequently diverges from this ideal behavior. In addition to regular forward steps, "skips" can occur when multiple forward steps occur in quick succession, too fast to electronically resolve the intermediate step or steps. Additionally, "backsteps" can occur when the enzyme backtracks to a previously observed position along the DNA. The existence of irregular enzyme steps means that the observed time order of conductance states does not necessarily match the base order in the DNA, and no part of the nanopore signal clearly distinguishes these steps from regular processive behavior. This complicates finding the true sequence from the observed conductance states and causes sequencing errors.

The methods described in this part address these error modes. In chapter 11, I describe how the pore and DNA orientations relative to voltage and relative to one another affect the ion current, and how this different k-mer map can be used to the advantage of a sequencer or enzymologist. In chapter 12 I provide a complete guide to the development, implementation, and analysis of a variable-voltage nanopore sequencer which drastically improves the accuracy of base calling.

Chapter 11

VARIANT PORE ORIENTATIONS FOR SEQUENCING AND SPRNT

Early work with MspA at the University of Washington used an impure preparation of the nanopore protein. In addition to making it much more difficult to obtain clean MspA insertions, this sample of protein also tended to introduce insertions with open pore ion currents other than the expected value for MspA. Typically, these had undesirable characteristics: they would be noisy, drift in ion current value, gate excessively, not allow translocation of DNA, or simply destroy the membrane after some period of time.

After switching to a cleaner source of nanopore protein, most of these "false" insertions were no longer seen. However, one of these types of insertion conspicuously remained even in the high-quality protein preparation, at around 135% of the ion current seen by the standard MspA insertion. This insertion was seen with a frequency comparable to that of a standard pore.

These unusual "type 2" pores were ignored for several years, until their identity was discovered through a simple observation: their ion current-voltage characteristic curve was mirrored from a standard pore (figure 11(a)). More explicitly, if the function describing the ion current response to voltage in a standard pore is $I = f(V)$, the equation relating ion current to voltage for one of these type 2 pores is $I = -f(-V)$.

This means that if an opposite polarity of voltage is applied, and the ion current is measured oppositely from normal, going from *cis* to *trans*, the pore is indistinguishable from a standard pore. It follows that these type 2 pores are actually just

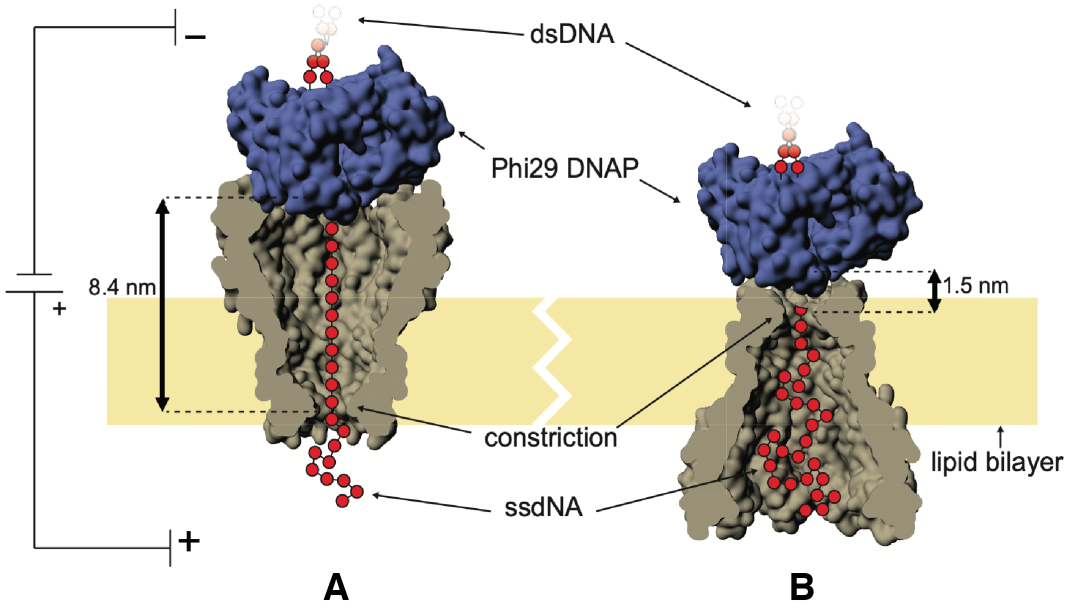

Figure 11.1: **Forwards (a) and backwards (b) pores.** In the backwards pore, the length of stretching DNA is much smaller, reducing the fluctuations contributing to the size of the nanopore readhead. Additionally, ion current depends on base content differently for opposite orientations of the DNA relative to MspA.

standard pores, oriented in the opposite direction (figure 11(b)). We call this oppositely oriented pore the *backwards pore*. The asymmetry in the MspA's conductivity is expected, as it is necessary for the *in vivo* functionality of an ion channel.

The remainder of this chapter explore the effects of pore and DNA orientation on the nanopore system. First, I discuss how DNA orientation affects the ion currents in a blocked pore. Next, I discuss some biophysical properties of the backwards pore with a trapped DNA-enzyme complex. I then discuss experimental considerations for practical use of the backwards pore, and detail its applications in both DNA sequencing and SPRNT.

11.1 DNA orientation and blockage current

When DNA is introduced to the nanopore system, there is another relative direction that needs to be tracked by the experimenter. In addition to the effect of pore orientation relative to the ion current, the orientation of DNA relative to the pore.

As discussed in §1.1.1, DNA is directional, having two different ends labeled 5' and 3'. To understand the effect of DNA orientation on the blocked ion current, Manrao *et al* performed experiments on both orientations[59].

ssDNA strands were labeled with biotin and attached to Neutravidin, a globular protein that very tightly binds to biotin molecules (figure 11.1(a)). Similarly to an enzyme, Neutravidin is too large to fit through MspA, and arrests the DNA's translocation. These ssDNA strands were trapped in MspA and the ion currents were observed. Since Neutravidin is not a DNA translocating enzyme, no steps in ion current are observed during the blockages.

The ssDNA molecules used were homopolymer A (also termed *polyA*), polyC, and polyT. PolyG was not used because of its tendency to form G quadruplices. These strands were attached to the neutravidin by both the 5' and 3' ends, allowing the 3' and 5' ends respectively to feed into the pore.

A marked difference in ion current is seen depending on the orientation of the DNA (figure 11.1(b)). Perhaps most remarkably, the ordering of ion currents for homopolymers is different: polyT has the lowest ion current when the 5' end of DNA is fed through the forwards pore, and polyC has the lowest current when the 3' end is fed in.

All things considered, there are four possible geometries in which the pore may be operated: forwards pore 5' feeding, forwards pore 3' feeding, backwards pore 5' feeding, and backwards pore 3' feeding. The DNA orientation relative to the pore is the most important component of this choice- forwards pore 5' feeding experiments

see similar ion currents to backwards pore 3' feeding experiments with the same DNA, and forwards pore 3' feeding experiments have similar ion currents to backwards pore 5' feeding experiments.

11.2 Effect of pore orientation on Brownian motion

Part of the contribution to the size of the readhead (and accordingly, one determinant of the appropriate k used to construct a k-mer map) is the Brownian motion of the DNA within the nanopore. As the DNA randomly changes its extension due to thermal motion, the portion of the sequence inside the constriction shifts. This motion occurs on a nanosecond time scale, much more quickly than the kilohertz sampling rate of a typically used patch clamp. Therefore, the observable result of this motion is that the observed ion current for any mean DNA position is a weighted average of the ion current with the DNA at nearby positions.

Approximating the anchored and stretched DNA as a spring in a thermal environment, the magnitude of the flossing position fluctuations is expected to be proportional to $\sqrt{l}$, the square root of the length of the spring. In the case of the nanopore experiment, this is the distance from the constriction, where the electrical force is applied, and the anchor point of the DNA within the enzyme.

In forwards MspA, the anchor point is separated from the constriction of the pore by the goblet-shaped vestibule. The separation from the constriction to the surface of a typical globular protein used for DNA control is approximately 11 nucleotides. This, together with the length of any nucleotides within the protein but not bound to it, contribute to the elastic length of the DNA "spring," which typically totals to a length somewhere in the range of 7 to 9 nanometers.

When using MspA in the backwards orientation, however, the constriction is much closer to the enzyme, often directly adjacent to its surface. In an enzyme-DNA-

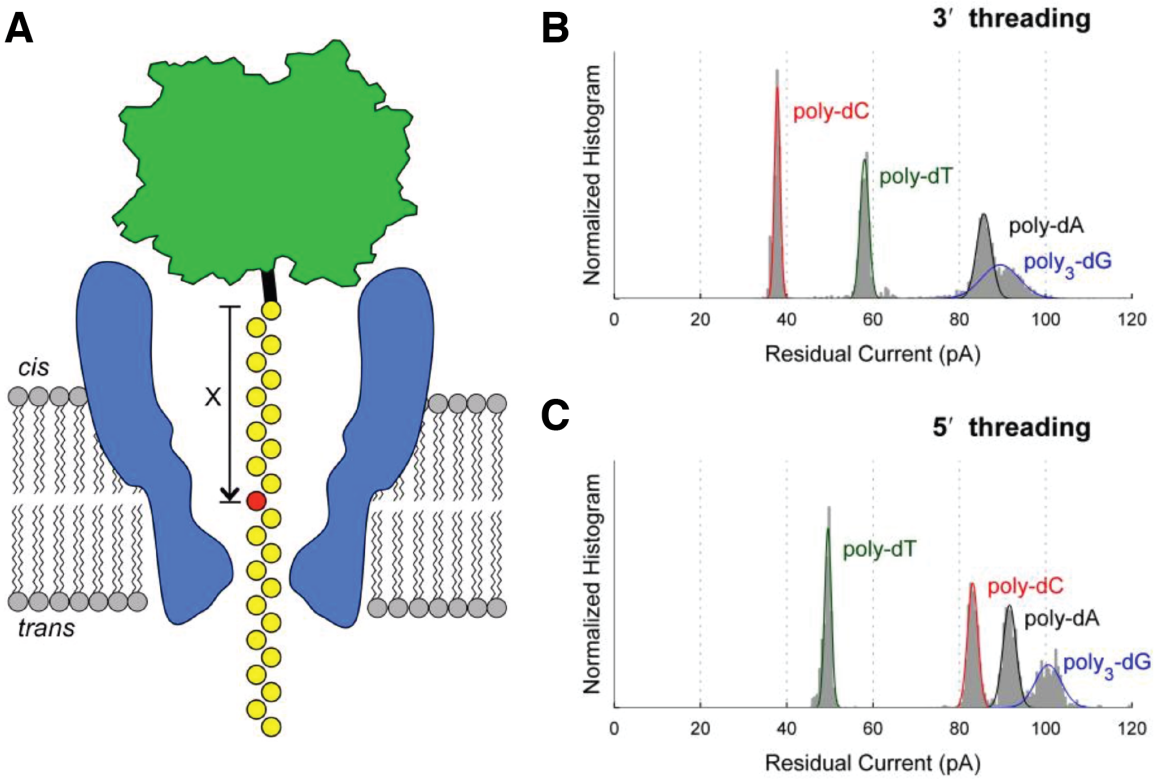

Figure 11.2: **Effect of DNA orientation on blockage current.** (a) ssDNA was attached through a biotin linker at either the 3' or 5' end to a Neutravidin molecule, and pulled into the nanopore. Histograms of ion current for 5' feeding (b) and 3' feeding (c) ssDNA bound to neutravidin. The ion currents are distinctly different, even changing the ion current level ordering of the different DNA bases. Figure adapted from Manrao 2011[59].

backwards pore complex, then, the elastic length of the DNA is much shorter, only a few nucleotides at most. We therefore expect the fluctuations in position of the DNA within the constriction of the backwards pore to be smaller than in the forwards pore.

Because part of the contribution to the size of MspA's readhead is due to these rapid fluctuations in DNA position, it follows that the readhead, and thus the minimal k in a reasonable k-mer map, of the backwards pore should be smaller than for the forwards pore.

Although it certainly requires less data for training, having a k-mer map with smaller k does not necessarily translate to increased sequencing accuracy. After all, one could imagine an utterly uninformative k-mer map with $k = 1$ but all four ion currents identical with large error bars. However, it is likely that a smaller k will improve sequencing given that it does not have a particularly pathological structure.

One reason for this is that with a narrower readhead, each measurement will be less strongly correlated with the previous one. This means that if the dynamic range of backwards pore ion currents is similar to that of the forwards pore, then for a random DNA strand larger jumps will be observed between adjacent ion current states in the backwards pore. Larger ion current transitions separating enzyme states makes the data segmentation algorithm more accurate, decreasing the number of extraneous or missing states in the string of observations passed to the sequencer. The size of jumps between adjacent states in the backwards pore is indeed seen to be significantly larger than in the forwards pore details on these results are provided in §11.4.

Secondly, the decreased number of states in the map reduces k-mer degeneracy. That is, fewer k-mers will have indistinguishable ion currents. When the enzyme stepping is known precisely, map degeneracies do not necessarily negatively impact the calling accuracy. However, they rapidly become problematic as enzyme stepping becomes more uncertain. Having a map with few degeneracies drastically improves

the quality of backstep filtering, and reduces the negative impact of skipped states. The reduction in k-mer map degeneracy is also observed in the backwards pore and detailed in §11.4.

11.3 Pore orientation experiments

Within this section I describe the methods used to establish backwards pores and results of analysis on their properties and suitability for DNA sequencing.

11.3.1 Establishment of backwards pores

Backwards pores have been observed to occur as a fraction of insertions obtained in the regular process of isolating individual MspA insertions. This is therefore an acceptable, but inefficient method for obtaining backwards pores.

MspA's preference for forwards-orientation insertion may be because of the hydrophilic residues on the vestibule region of the pore. In order to insert in the backwards orientation, these residues need to make their way to the *trans* side of the bilayer.

Indeed, insertion of backwards pores can be promoted by disrupting the bilayer or placing MspA on the *trans* side of the bilayer. Several methods inspired by this hypothesis have been seen to be effective in improving the rate of backwards pore insertions.

Repeated bubbling

By, repeatedly disrupting and re-forming the bilayer by using a pipette to blow a bubble over the aperture, it is possible to increase the rate of backwards pore insertions to approximately 50% of all insertions. When forwards pores are inserted, the bilayer is simply zapped and reformed until an insertion in the desired orientation

appears. This method is simple and requires no additional materials preparation or experimental techniques beyond those already used in running a nanopore setup. As such, this is the most common method used at the University of Washington nanopore laboratory for obtaining backwards pore insertions.

There are two primary drawbacks of this method. One is the stress it places on the bilayer. After a sufficiently long period of bubbling, bilayers become less stable. Fortunately, this only means that the experimenter must switch to a freshly primed puck and aperture, and the net effect is simply a slight increase in the materials preparation workload. The other drawback is the time it takes to obtain an insertion: bubbling is generally not as effective as directly refluxing MspA-filled buffer onto the bilayer.

Pre-mixed MspA-lipid paint

The success rate of the repeated bubbling method can be enhanced by not adding MspA to the front well but instead saturating the lipid paint with MspA protein prior to bilayer establishment. Since the MspA is localized within the bilayer in this method, rather than on one side or another of the bilayer, the. This method is also highly conservative of MspA, requiring much less protein than the amounts typically used in nanopore experiments.

To make pre-mixed MspA paint,

1. Prepare lipid paint on a slide as described in §3.2.

2. Immediately before adding MspA to the lipid paint, dilute MspA from concentrated stock in DI water to a concentration of 1 microgram/milliliter. Use only water with no detergent, as detergent destabilizes bilayers.

3. Place a single 1 microliter drop of diluted MspA on each lipid pile.

4. Cover the slide to prevent dust from contaminating the paint, and dry the paint in a vacuum chamber for 15 minutes, until water is completely evaporated.

After mixing the paint, it should be used to establish a bilayer in the usual way. Insertions are obtained by repeated bubbling as described in the previous section. It is best to use the paint immediately, and mix fresh paint every day that an experiment using pre-mixed paint is needed.

Reversed setup

The most laborious but most surefire method of obtaining a backwards pore is to reverse the setup. In other words, MspA is added to the *trans* side of the bilayer. With the experimental apparatus described in §3.1, the experimenter only has direct physical access to one side of the bilayer. To isolate a single pore, the perfusion of MspA out of the buffer and away from the bilayer is a critical step. Therefore, the MspA should still be added to the front well.

To make sure that the pore orientation is backwards relative to the enzyme, DNA, and applied voltage, this means that the setup should be operated at a reversed polarity, where the front well is at a positive bias compared to the back well. Additionally, the U-tube must be filled with a solution containing DNA and enzyme. Obviously, this must occur prior to the formation of the bilayer. To accomplish this setup, the procedure is as follows:

1. Mix at least 20 microliters total volume of DNA, enzyme, and any reagents and cofactors at final experimental concentrations.

2. Using a syringe, fill U-tube with enzyme-filled buffer until buffer comes out of the aperture. Fill the remainder of the back well and the front well with clean buffer.

3. Applying a voltage at opposite polarity from an ordinary experiment, establish a bilayer.

4. Add MspA to front well and seek an insertion as in the standard experiment.

5. Run the experiment with all voltages at opposite polarity.

This method produces a high rate of backwards pore insertions, because no changes have been made to the method of insertion; only the means of introduction of reagents to the already-inserted pore has changed. However, it suffers from a significant limitation: the handling of the experimental targets is heavily prescribed. Nothing can be added to or removed from the back well containing the enzyme and DNA after the experiment has begun. This restricts application of this method in cases where the enzyme is sensitive or has short-lived activity at room temperature, or in cases where the salt concentration preferred by the enzyme makes establishing a bilayer difficult.

11.4 Sequencing with variant orientations

Using the backwards pore, we found that enzyme controlled DNA translocation produces ion current traces with distinct levels, different from the levels observed for the same DNA sequence in the forwards pore (figure 2b). We obtained nanopore reads of long genomic DNA from the Phi X 174 bacteriophage. We obtained a total of 50 reads from 17 different backwards pores, and used these reads to construct a 4-mer map for the backwards pore (figure 11.4(a)) using the method described in §9.2.

The choice of $k = 4$ for the k-mer map is so that the forwards and backwards pores may be compared directly. Whether the backwards pore actually may be modeled with a smaller k is irrelevant, as the 50 reads contained enough data to estimate the 256 ion currents in the 4-mer map sufficiently well.

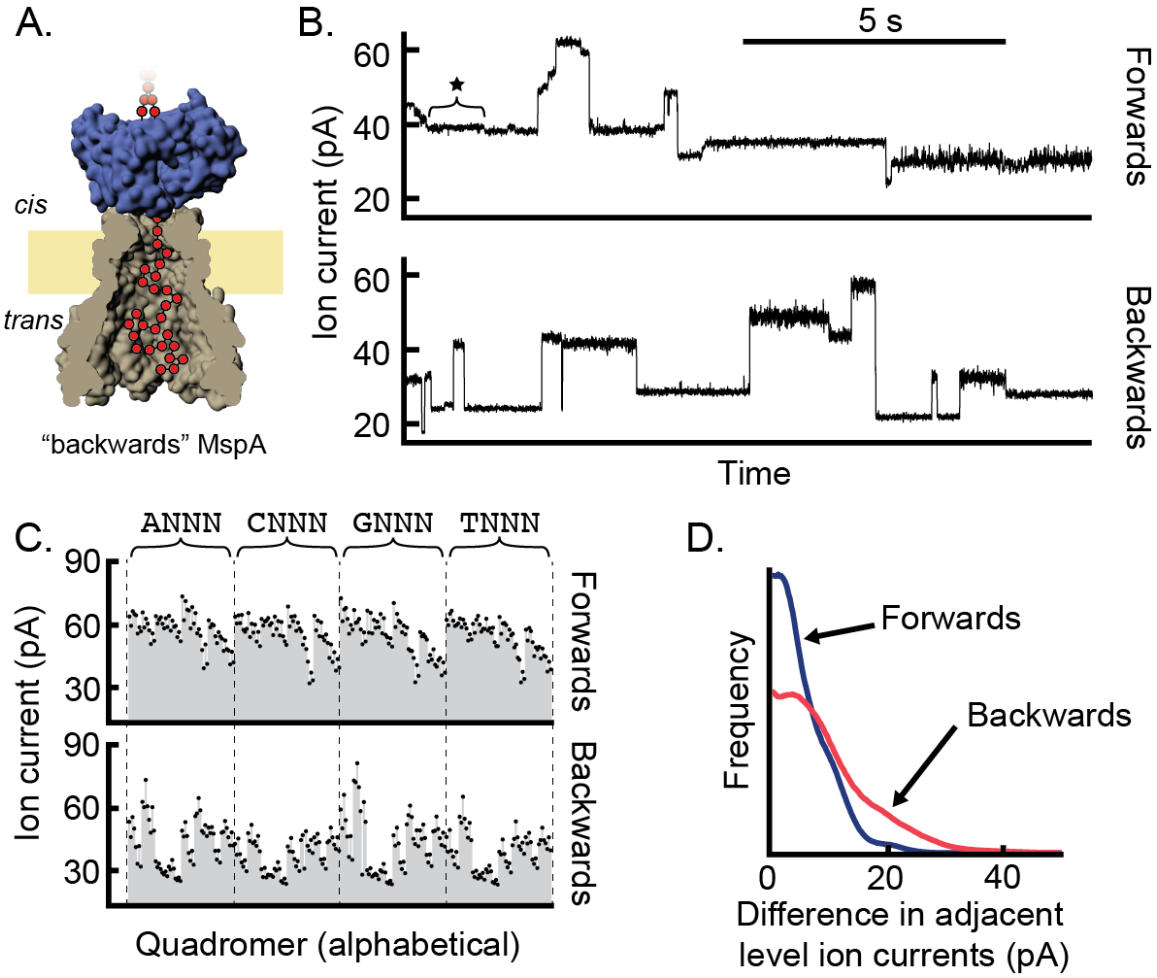

Figure 11.3: **Forwards and backwards MspA data comparison.** (a) Cartoon of backwards MspA experimental configuration. (b) Typical data from forwards (top) and backwards (bottom) MspA nanopore experiments. (c) Forwards (top) and backwards (bottom) 4-mer maps. 4-mers are shown in alphabetical order. (d) Forwards (blue) and backwards (red) ion current differences between adjacent ion current states.

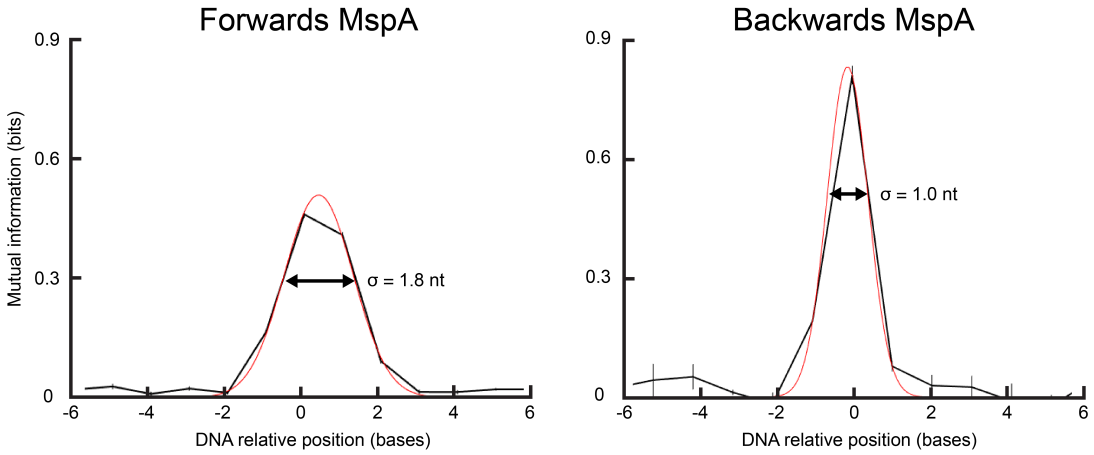

Figure 11.4: **Forwards and backwards pore readhead width.** Mutual information between DNA base and the observed ion current is plotted against position of that base relative to the constriction of the pore. The backwards pore shows a narrower response to DNA base, consistent with the hypothesis of §**??**. Plots were generated from all long PhiX174 reads.

In the backwards 4-mer map, we observe that a C blocking the constriction causes the lowest ion current levels, consistently under 15% of the open pore current, while T or G blocking the constriction cause states in the middle of the ion current range, and A blockages cause the largest ion currents. This is different from the forwards pore, in which T blockages cause the lowest ion currents and C blockages produce ion currents in the middle range. Additionally, we observe that the backwards pore is most sensitive to a single base in the second position within a 4-mer; for example, the C in ACGT. Comparatively, the forwards pore has approximately equal sensitivity to the central two bases (figure 11.4).

To test whether the backwards pore addresses the error mode caused by indistinguishable adjacent states in the forwards pore, we aligned the measured ion current sequences to predicted ion current sequences for Phi X 174, following the methods of

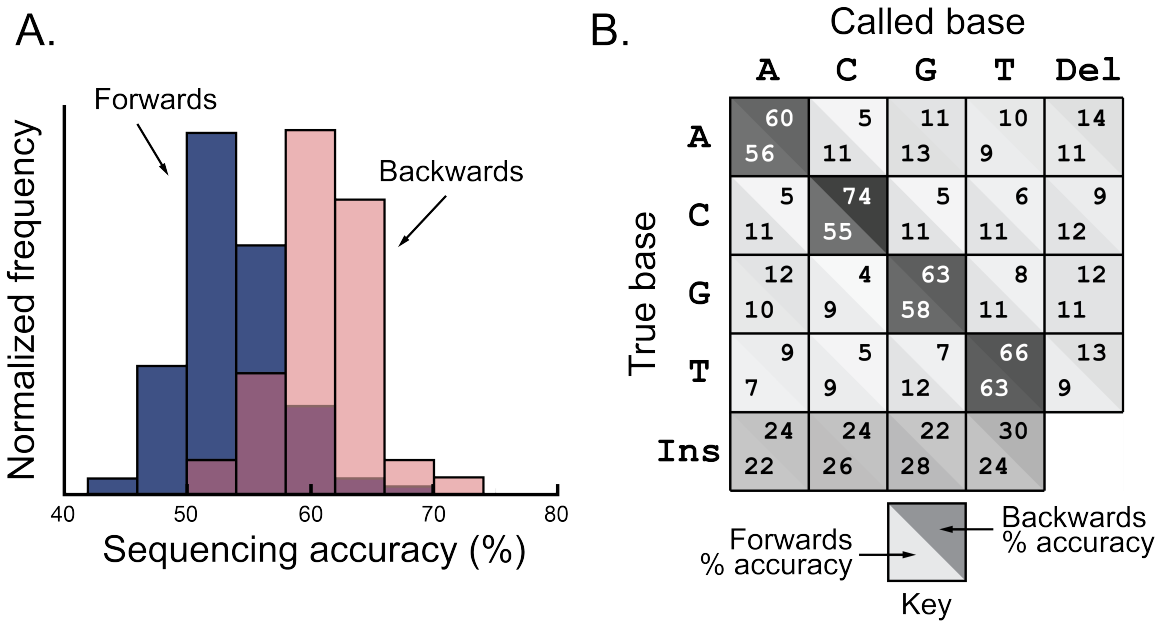

Figure 11.5: **Forwards and backwards sequencing results.**

Laszlo 2014. Alignments of 128 reads of Phi X 174 genomic DNA measured with the forwards pore match to 72.3 0.3% of the predicted levels between the beginning and end of each read. In alignments of 50 reads from the backwards pore, this improves to 79.1 0.3% of the predicted states. This indicates that more of the expected states were found and successfully aligned to the sequence.

As predicted, we observed that the magnitude of ion current differences between adjacent 4-mers is larger in the backwards pore. Figure 11.4(b) shows the distribution of expected ion current differences between adjacent 4-mers. The backwards pore map has a larger median difference between adjacent 4-mer currents than the forwards pore map by 1.9 pA, indicating a larger signal-to-noise ratio for backwards pore experiments. This increases the likelihood of identifying and correctly aligning ion current states.

Using the map built from the sense reads of Phi X 174 DNA, we sequenced the reads of the antisense DNA using a simple hidden Markov model. Median single-read sequencing accuracy calculated as # matches / (# mismatches + # insertions + # deletions + # matches) is 55% for the forwards pore, and improves to 63% with the backwards pore (figure 11.4(a)). The very low ion currents caused by C blockages lead to much higher accuracy in identifying Cs. (figure 11.4(b)).

11.5 SPRNT with variant orientations

The capability of understanding nanopore signals with any orientation of DNA and pore relative to the applied voltage greatly improves the flexibility of the SPRNT technique. In fact, much of the impetus for developing the k-mer maps of alternative orientations has come out of necessity for SPRNT experiments.

For example, being able to feed DNA from either end enables the application of either an opposing or assisting force to the motor enzyme controlling the DNA. In some cases[1] it has been much easier to obtain enzyme-controlled reads of DNA in one orientation than another. Being able to understand nanopore data regardless of these restrictions enables the study of enzymes for which this is the case.

Additionally, in some SPRNT experiments, the pore orientation affects interaction rate, especially in the troubleshooting phase. If the length of DNA available to interact with the pore is too short to feed completely through the vestibule and constriction of the forwards pore, it may still be able to interact with the backwards pore, whose constriction is closer to the side the enzyme is on. Differences in interactions with the forwards and backwards pores are accordingly useful as a diagnostic tool when troubleshooting DNA construct design for SPRNT.

Finally, because the backwards pore has different characteristic ion currents from

[1]Such as for *E. coli* RNA polymerase and for Pcra X mutant helicase.

the forwards pore, variation of pore orientation can be a useful tool when the DNA being studied cannot be freely changed. This is the case, for example, in studies of sequence dependent enzyme behavior, where the DNA sequence is prescribed by the needs of the experimenter. For some sequences, the nanopore may not provide an ion current profile with adequately sized steps to confidently detect every enzyme step. In these cases, using an alternate orientation of MspA provides a different sequence of currents that may have more distinguishable ion current transitions.

Chapter 12
VARIABLE-VOLTAGE NANOPORE EXPERIMENTS

In this chapter, I show how replacing the constant bias voltage with a time-varying voltage substantially reduces the impact of both enzyme missteps and k-mer map degeneracies in nanopore sequencing. in §12.1, I provide an overview of the variable voltage experiment adapted from the main text of a manuscript under review at the time of the writing of this thesis. In §12.2 I provide detailed information about the methods developed for the acquisition and analysis of the data required by the variable voltage method. Work on variable voltage sequencing was performed in close collaboration with Matthew Noakes, and comprises the main text and supplement of a paper published in Nature Biotechnology in 2019[58].

12.1 Overview of variable-voltage sequencing

Nanopore sequencing is limited by a low single-passage de novo sequencing accuracy compared with other established sequencing platforms[60][61]. Improved accuracy can be achieved by combining multiple high-error nanopore reads to produce a consensus sequences with fewer errors[62]. However, this approach trades throughput for accuracy, and is still limited by systematic errors. Additionally, some nanopore applications, such as pathogen detection at low concentrations, or metagenomics studies, ideally need to be able to identify a single molecule of DNA with only one read. Therefore, the path toward fully realized nanopore sequencers requires improvement of the baseline single-passage accuracy.

Most of the single-passage sequencing errors can be attributed to two error modes:

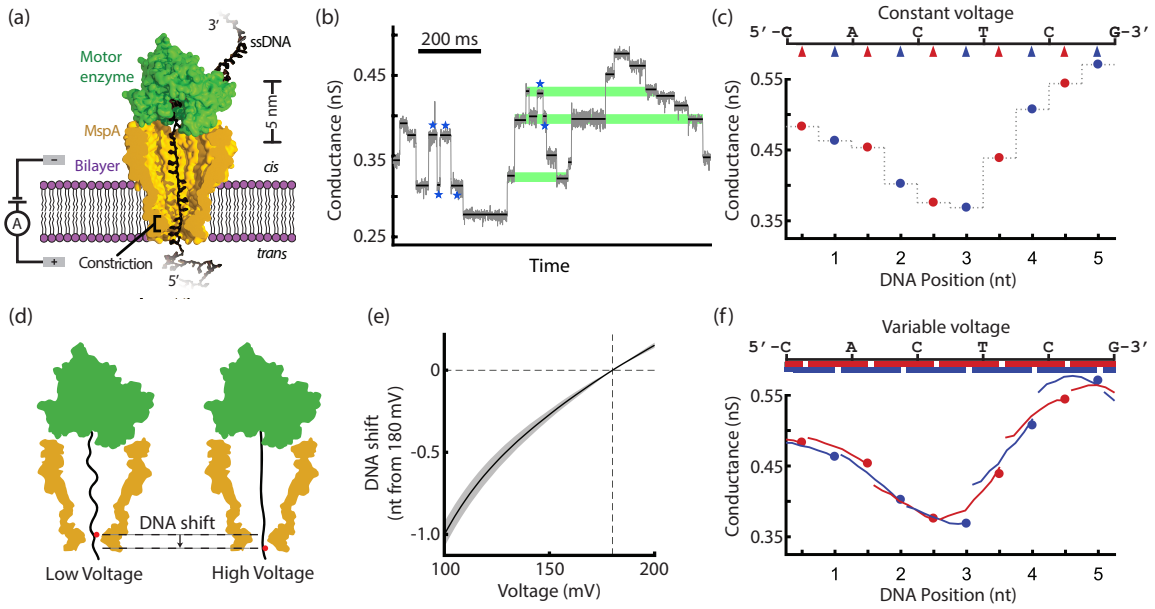

Figure 12.1: **Basic principles of variable-voltage nanopore sequencing.** (a) A lone MspA nanopore (gold) inserts into a lipid bilayer (purple) separating two chambers, cis and trans. A Hel308 helicase motor enzyme (green) controls the motion of ssDNA (black) through the pore while we apply a voltage across the bilayer and measure the conductance through the pore. (b) A segment of constant-voltage nanopore data (gray data, downsampled to 500 Hz) shows several common sequencing error modes. The overlaid black lines show the mean conductance at each enzyme step. Blue stars mark the locations of enzyme missteps. Green bars highlight indistinguishable conductance states generated by distinct DNA sequences. (c) In constant-voltage sequencing, we sample the conductance of the DNA-blockaded pore at discrete locations (red and blue arrows) along the DNA sequence (top). The resulting signal is a time-ordered series of mean conductance values, one for each enzyme step (red and blue points; gray line to guide the eye). Red points mark odd-numbered half-nucleotide steps by the Hel308 helicase and blue points mark even-numbered steps. (d) Applying different voltages changes the pulling force on the DNA. Higher forces cause the DNA to extend further, shifting the DNA within the constriction of the pore and changing the DNA bases affecting the conductance. This cartoon shows a DNA base (red dot) changing position when the applied voltage is increased. (e) The range of applied voltages in our variable-voltage experiments (100 mV to 200 mV) shifts the DNA position in the pore by more than a full nucleotide. The position shift is relative to the DNA position at 180 mV. The shaded region shows the standard deviation of the position shift measurement. (f) Variable-voltage sequencing samples the conductance of the DNA-blockaded pore continuously along the DNA sequence (top). Red and blue bars (top) show the overlapping ranges along the DNA molecule probed during the voltage swing at odd (red) and even (blue) enzyme steps. In the resulting signal, each enzyme step is characterized by a conductance vs. DNA position curve (red and blue curves), rather than by a single mean conductance as in constant-voltage sequencing (red and blue points).

distinct sequences with indistinguishable conductance signals and irregular steps by the motor enzyme (figure 12.1 (b)). To decode an observed signal, the base-calling algorithm must use a model of the blockaded pore conductance that maps observed conductance values to the most likely DNA sequence[24]. The conductance through the nanopore is influenced by several nucleotides near the constriction of the pore[31][63], resulting in a complex map of conductance to the underlying sequence. In many instances, different sequence segments generate statistically indistinguishable conductance values, thereby leading to error-prone base calls (figure 12.1 (b), green bars).

Irregular stepping by the DNA-controlling motor enzyme can also introduce sequencing errors. Ideally, the DNA-controlling enzyme would move DNA unidirectionally through the pore in discrete steps of uniform length. However, the stochastic stepping of enzymes frequently diverges from this ideal behavior[64]. In addition to regular forward steps, "skips" can occur when multiple forward steps occur in quick succession, too fast to electronically resolve the intermediate step or steps. Additionally, "backsteps" can occur when the enzyme backtracks to a previously observed position along the DNA (figure 12.1 (b), blue stars). The existence of irregular enzyme steps means that the observed time order of conductance states does not necessarily match the base order in the DNA, and no part of the nanopore signal clearly distinguishes these steps from regular processive behavior. This problem makes finding the true sequence from the observed conductance states difficult, and causes sequencing errors.

We hypothesized that replacing the constant bias voltage with a time-varying voltage would substantially reduce the impact of both of these error modes. In our sequencing experiments, we used Mycobacterium smegmatis porin A (MspA) as our nanopore. MspA has a single narrow constriction region that is ideally suited to resolve nucleotide-long enzyme steps along single-stranded DNA (ssDNA)[65][59]. We

used the Hel308 DNA helicase enzyme from Thermococcus gammatolerans EJ3 (hereafter referred to as Hel308) as the motor enzyme to control DNA translocation through the pore. Hel308 has been observed to take two steps per nucleotide as it translocates along ssDNA, with each step approximately a half nucleotide in length[48]. These half-nucleotide steps provide two conductance measurements per nucleotide (figure 12.1(c)).

Positive voltage applied across the nanopore generates a force on the DNA threaded through the pore. Varying the magnitude of this voltage changes the force pulling on the DNA. The force stretches the section of DNA between the DNA-binding sites within Hel308 and the high field region at the nanopore's constriction[48]. Increasing the applied voltage elongates the DNA and shifts the relative position of the DNA in the constriction (figure 12.1 (d)). A voltage change from 100 mV to 200 mV repositions the DNA in the pore by slightly more than a full nucleotide (figure 12.1 (e)). Thus, the applied voltage serves as a fine control over the DNA position in the pore.

The fine DNA position control using the variable-voltage complements the discrete stepping of the motor enzyme. We combine the enzyme and voltage control methods by replacing the constant applied voltage with a 200 Hz, symmetric triangle waveform voltage from 100 to 200 mV. The positive overall bias is necessary to keep the DNA-enzyme complex held on top of the pore. The 200 Hz triangle wave frequency goes through several cycles for each Hel308 step (average rate 20 steps/s in our sequencing conditions). While the motor enzyme steps along the entire length of the DNA, the changing voltage repositions the DNA incrementally within each enzyme step. Together, the enzyme steps and the variable-voltage sample the effect of the DNA on the pore's conductance nearly continuously along the DNA (figure 12.1 (f)).

In the constant-voltage signal, the pore conductance is probed only at a single DNA position at each enzyme step. Each step is thus only characterized by a sin-

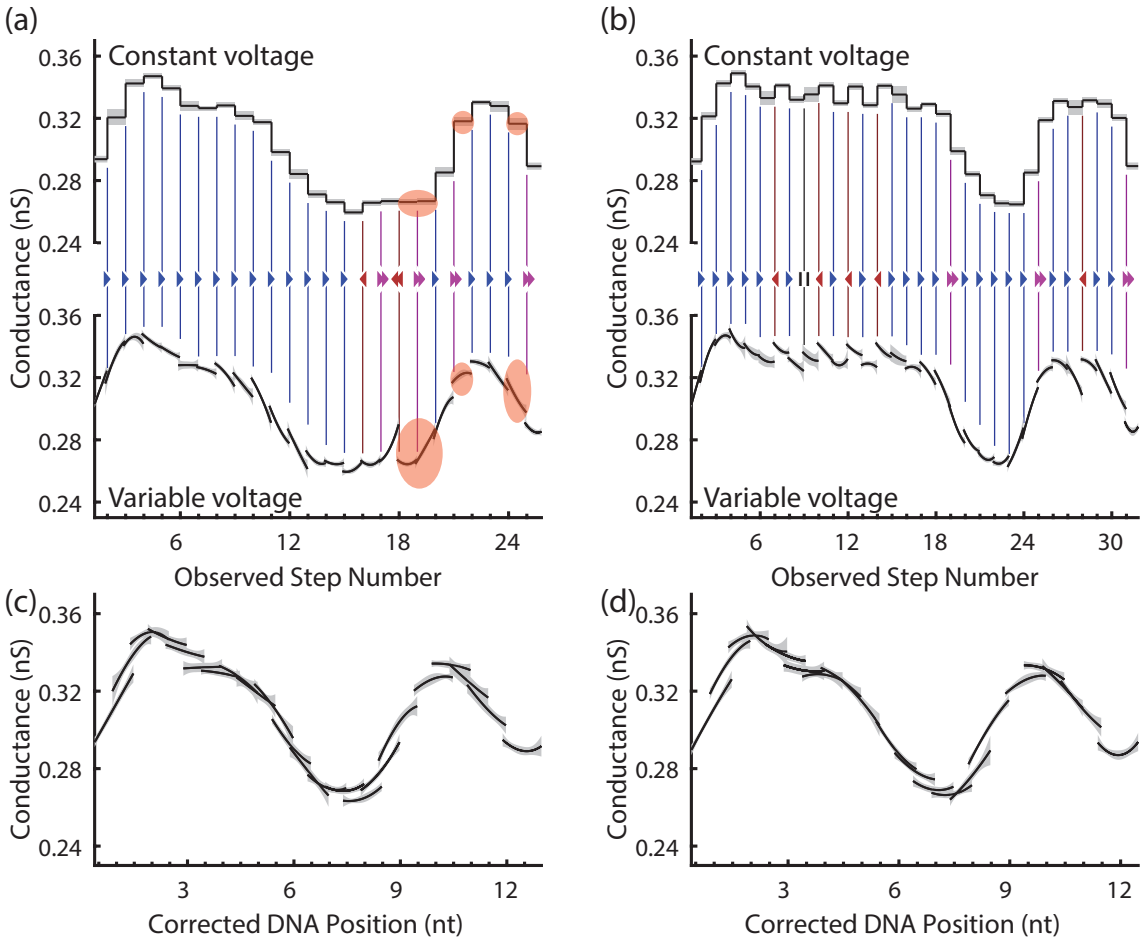

Figure 12.2: **Error correction in variable-voltage sequencing.** (a) and (b) compare the measured conductance vs. position curve segments of two typical variable-voltage reads (bottom) of the same DNA sequence to a reproduction of the corresponding constant-voltage reads (top), created by extracting the conductance of these segments at 180 mV only. Nearly identical conductances at 180 mV are easily distinguishable when the full conductance vs. position curves are compared (states highlighted in orange). Enzyme stepping errors are inferred from discontinuities in the curve and identification of repeated curve segments. These are marked with blue right arrows (forward steps), magenta double right arrows (skips), red left arrows (backsteps), and black pause symbols (hold steps). Shaded regions show the standard deviation of each conductance measurement. (c) and (d) show these reads after correction of the identified enzyme stepping errors. Although the uncorrected reads look dissimilar, the corrected reads are almost identical and will be decoded into the same DNA sequence.

gle value: the conductance at that DNA position (figure 12.1 (a) and (b); upper). Variable-voltage sequencing instead probes the conductance continuously over a 1 nucleotide long range at each enzyme step, characterizing each step by a conductance vs. position curve (figure 12.1 (a) and (b); lower). These curve segments provide additional identifying information as to the generating DNA sequence compared to the mean conductance alone. Two sequences with nearly identical conductance values in the constant-voltage mode can be distinguished based on the shape of the curves generated by the variable-voltage (figure 12.1 (a), orange highlights).

The variable-voltage signal also provides information about the correct ordering of the measured enzyme steps and can be used to infer the existence of steps too fast to observe. The ability to identify enzyme missteps is enabled by the variable-voltage technique's continuous sampling of the conductance through the pore as a function of DNA position. In variable-voltage sequencing, at each consecutive Hel308 half-nucleotide step, the full-nucleotide stretch caused by the voltage sweep samples the conductance at many of the same DNA positions as the previous and next Hel308 steps. Therefore, each measured segment of the conductance vs. position curve will be overlapping and continuous with the segments measured at adjacent Hel308 steps. If two consecutively measured segments are not overlapping and continuous, a non-uniform step such as a backstep or skip must have occurred. The degree of overlap between consecutive measurements can therefore be used to identify and correct enzyme missteps (figure 12.1 (a) and (b); colored arrows). A probabilistic support vector machine informed by the shapes of the curves immediately preceding and following each enzyme step is used on the variable-voltage signal to identify and eliminate misorderings caused by enzyme missteps and reestablish the order most representative of the generating DNA sequence. The resulting corrected signal (figure 12.1 (c) and (d)) is free of enzyme missteps and is more easily decoded into the correct

DNA sequence. Measurements of DNA positions that go completely unobserved due to enzyme skips cannot be filled in at this stage. However, the overlap information can be used to label the probable locations of enzyme skips in the final signal to be sequenced. This information tells the sequencer that one or more bases must be inserted at this location, reducing the detrimental impact of enzyme skips relative to constant-voltage nanopore sequencing.

To objectively evaluate the extent to which the variable-voltage method improves single-passage sequencing accuracy over the constant-voltage method, we tested both sequencing methods on the same target DNA sequence, using the same enzyme (Hel308) and nanopore (MspA). In both cases, we used a hidden Markov model (HMM) to decode the generating DNA sequence for the observed signal. For both constant- and variable-voltage sequencing, we used a model mapping each unique 6 base sequence segment (6-mer) to an associated conductance signal. We generated this model empirically by measuring the signal of known DNA sequences (ΦX-174 and lambda phage DNA, as well as synthetic oligos) using our variable-voltage sequencing conditions. For constant-voltage sequencing, we extracted a constant-voltage 6-mer model from the variable-voltage model to ensure that any systematic model errors affecting the sequencing accuracy of one method affected both methods equally.

We used the pET28a vector as the target DNA sequence because it provided a non-synthetic DNA testing ground for the two methods separate from the sequences that were used in constructing the 6-mer model. We fragmented the pET28a vector using a selection of restriction enzymes, allowing us to attach the necessary sequencing adapters and increasing the likelihood of reading sequences at all locations on the pET28a vector given the limited processivity of Hel308 (1000 nt). The variable-voltage method does not reduce the nanopore sequencer's ability to sequence long (multiple kilobasde) reads.

We obtained reads of plasmid pET28a fragments using both constant-voltage (31 reads, n = 9368 bases across all reads) and variable-voltage (97 reads, n = 17309 bases across all reads) methods. Enzyme steps in the constant- and variable-voltage conductance signals were detected using a custom change-point detection algorithm, segmenting the data into distinct conductance states. In the variable-voltage experiments, the capacitive charging currents from the bilayer were removed from each state using custom software. We used the overlap information between successive conductance states to identify and correct enzyme mis-steps in the variable-voltage reads, then both sets of reads were calibrated and sequenced. For both the constant- and variable-voltage sequencing results, we determined the ground truth sequence for each read by aligning the called sequence to the pET28a reference sequence. Based on the alignment, we calculated the per-base sequencing accuracy e as

$$e = \frac{\#\text{ matches}}{\#\text{ matches} + \#\text{ mismatches} + \#\text{ insertions} + \#\text{ deletions}}. \tag{12.1}$$

The uncertainty in per-base sequencing accuracy is calculated using binomial errors as

$$\delta e = \sqrt{\frac{e(1-e)}{N}}, \tag{12.2}$$

where N is the number of bases sequenced.

12.1.1 Variable-voltage sequencing results

Relative to the constant-voltage reads, the variable-voltage reads have fewer base calling errors (miscalls, deletions, and insertions, figure 12.1.1 (a) and (b)). The average per-base accuracy in the variable-voltage reads is 79.3 0.3% (SEM) for single passages of a single-stranded DNA molecule. This represents a substantial improvement

compared with nanopore sequencing using constant-voltage, which had an average accuracy of 62.7 0.5% (SEM) for the same DNA sample. Our constant-voltage sequencing accuracies are similar to single-passage, unpolished 1D reads reported for the Oxford Nanopore Technologies' MinION device[66][67]. To contextualize the relative accuracy of the two methods, we compared the distribution of observed per-read accuracies with the accuracy distribution for random sequences of the same lengths aligned against the pET28a reference sequence. The sequencing accuracies of these random sequences is about 58% (figure 12.1.1 (c) and (d)); this random base call accuracy is so high (that is, much larger than 25%) because of the freedom provided to the alignment algorithm to call insertions, deletions or mismatches. Whereas the constant-voltage read accuracies only barely outperform the accuracies of randomly generated sequences (figure 12.1.1 (c)), the variable-voltage read accuracies are substantially higher than the distribution of random accuracies (figure 12.1.1 (d)). We conclude that the variable-voltage method recovers significantly more information from the target DNA and thereby substantially increases the base calling accuracy.

12.1.2 Outlook for variable-voltage sequencing

Improved single-read accuracy should enable fewer reads to be assembled into a high-accuracy consensus sequence, thereby reducing sequencing time and cost. Additionally, variable-voltage sequencing overcomes systematic errors, such as sequence-dependent enzyme mis-steps[64] and indistinguishable signals, that persist even when the information from many reads is combined. Variable-voltage reads can be more confidently identified with only single-read coverage. This capability is necessary for nanopore sequencing applications in which high coverage is not an option, such as metagenomics studies or pathogen detection at low concentrations.

The additional information provided by the variable-voltage signal is complemen-

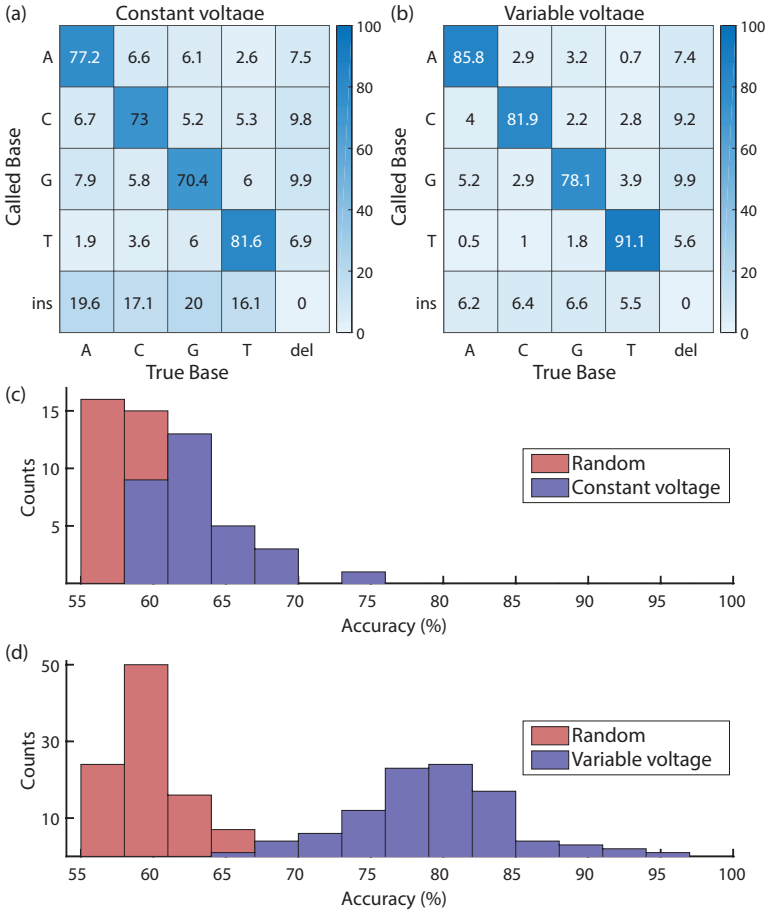

Figure 12.3: **Performance using constant- and variable-voltage sequencing.** Confusion matrices for sequencing using constant-voltage (a) and variable-voltage (b) show a reduction in mismatch, insertion and deletion errors across all bases with the use of variable-voltage. Histograms of single-passage accuracies for 31 constant-voltage reads (9368 total bases; 2203 called As, 2188 called Cs, 2166 called Gs, 2144 called Ts, 667 called gaps) (c) and 97 variable-voltage reads (17309 total bases; 4021 called As, 4073 called Cs, 3813 called Gs, 4081 called Ts, 1321 called gaps) (d) show a significant improvement in the distribution of read accuracies using variable-voltage. Single-passage sequencing accuracies are plotted in blue, with the distribution of accuracies for randomly generated sequences of the same lengths plotted in red. While constant-voltage nanopore sequencing is only a few percent above random base calling, the variable-voltage method yields a substantial improvement.

tary to other nanopore sequencing improvements, including more processive and predictable motor enzymes, more sophisticated base calling algorithms[68][69][70], reading both sense and antisense of the target DNA strand in "2D" techniques[60] (or the similar "1D squared" technique) or polishing reads with a consensus of passages of different DNA molecules[71][72]. Existing nanopore sequencers already consist of hundreds or thousands of parallel nanopores which are separately addressable with distinct driving voltages, so the variable-voltage method requires little re-engineering of the nanopore sequencing device other than the application of a waveform in place of a constant voltage. Consequently, our method could be used to improve sequencing accuracy of most existing platforms. The performance of variable-voltage nanopore sequencing will further improve as larger data sets are used to train both the model that maps conductance curves to DNA bases and the enzyme mis-step classifier.

We anticipate that incorporating our variable-voltage method into nanopore sequencing platforms will enable wide scale improvement of all nanopore-based DNA sequencing applications, including species identification, epigenetic mapping and higher accuracy de novo genome sequencing at lower coverage.

12.2 Methods for variable voltage sequencing

This section is intended to provide a complete guide to the interested person as to the data collection and analysis methods necessary to fully implement the variable-voltage nanopore DNA sequencing technique described in §12.1.

§12.2.1 provides an overview of the experimental conditions and equipment used in the experiments described in §12.1.

§12.2.2 covers how we measured the elongation of DNA within MspA in response to applied voltage. We discuss how position shift was calculated, and compare the force-extension results against a known model of DNA elasticity.

§12.2.3-12.2.5 provide details as to what data was collected to generate the 6-mer model used sequence both the constant- and variable-voltage data, as well as how that data was curated and analyzed to create the final working model. These sections include information concerning how the initial 6-mer model was generated using previous work, how genomic DNA was prepared for nanopore experiments, how the resulting data set was built into a 6-mer model, and how the constant-voltage model was extracted from the variable-voltage model.

In §12.2.6-refsec:si6, we detail the methods and algorithms used to process, analyze, and ultimately sequence the raw nanopore sequencing data. These notes outline the entire data analysis pipeline necessary to sequence variable-voltage nanopore data, and describe the attendant algorithms. These include the algorithms used to detect enzyme steps, remove the capacitive signal from the raw data, extract features for sequencing, correct enzyme missteps, and finally base call the resulting data.

The relevant kinetic properties of the Hel308 DNA helicase are covered in §12.2.12-12.2.13. These sections examine the activity of Hel308 in the buffer conditions and under the applied voltages in our constant- and variable-voltage sequencing experiments.

Finally, §12.3.1-12.4.1 provide details on the final sequencing verification experiment, and overall experimental notes and statistics. Specifically, we discuss how the DNA was prepared for the verification experiment, as well as how the resulting data was analyzed.

12.2.1 Experimental conditions

Proteins

The same mutant MspA protein was used in all 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/D134R27. All sequencing experiments used the Hel308 helicase enzyme from Thermococcus gammatolerans EJ3 (accession number WP-015858487.1). Hel308 was expressed in E. coli using standard techniques. All proteins were stored at -20 C until immediately before use.

DNA Sequences and Constructs

Short DNA oligonucleotides were synthesized and purified using column purification methods at Stanford University Protein and Nucleic Acid Facility. The ΦX-174 DNA (NCBI reference sequence NC-001422.1) was obtained from New England Biolabs. The lambda phage DNA (GenBank J02459.1) was obtained from Promega. The pET-28a DNA was obtained from collaborators who used it as an expression vector for another DNA sequence not used in this work. The complete DNA sequences for ΦX-174, lambda, and pET28a can be found at http://dx.doi.org/10.6084/m9.figshare.7140896.v1.

All experiments were conducted with the DNA threaded through the pore 5' end first. DNA constructs for Hel308 experiments consisted of a template read strand and a cholesterol-tagged blocking strand. A negatively-charged terminal phosphate was attached to the 5' end of the template strand, increasing the capture rate of that end by MspA. The cholesterol tag at the 5' end of the blocking strand anchors the DNA constructs into the bilayer, increasing the local concentration near the pore and increasing the capture rate.

Nanopore Experiments

All experiments were conducted as described in §3. Briefly, experiments were established with a device made from Teflon that contains two 50 μL chambers (cis and trans). The two chambers are connected by a Teflon heat-shrink "u-tube", 30 μL in

volume. The cis side of the u-tube narrows into a horizontal 20 μm aperture. Both chambers and the u-tube were filled with the operating buffers. The cis chamber was connected to ground via an Ag/AgCl electrode, while the trans-side Ag/AgCl electrode was connected to an Axopatch 200B integrating patch clamp amplifier (Axon Instruments) that also supplied the positive driving voltage. A lipid bilayer was formed 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. Following bilayer formation, M2-NNN-MspA was added to the cis chamber to a final concentration of 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 chamber buffer was perfused out and replaced with MspA-free buffer to prevent the insertion of additional pores. The Hel308 motor enzyme was added to the cis chamber to a final concentration of 50 nM, and DNA was added to a final concentration of 5 nM.

Hel308 is used as a translocase, rather than a helicase, in the sequencing experiments presented here, similar to previously described experiments[48][64]. Briefly, Hel308 loads onto the overhanging 3' end of the template DNA strand at the single-stranded/double-stranded junction. The 5' end of the template strand is captured by the pore, and the blocking strand is sheared off as the template strand is pulled through the pore. Hel308 is too large to fit through MspA, and arrests the template strand translocation once the duplexed blocking strand has been completely sheared away. Hel308 proceeds as a translocase from 3' to 5' along the template strand, incrementally pulling the DNA out of the pore towards cis.

Operating Buffers

All experiments were conducted using symmetric cis and trans buffer conditions of 400 mM KCl with 10 mM HEPES at pH 8.00 0.05. The cis buffer additionally contained 1 mM EDTA, 1 mM DTT, 10 mM MgCl2, and 100 μM ATP. ATP-containing buffer was re-perfused into cis approximately once per hour to prevent depletion of ATP and accumulation of ADP. Experiments were performed at 37 C.

Data Acquisition and Analysis

Experiments were controlled and data were acquired with custom acquisition software written in LabView (National Instruments, version 2018) at a sampling rate of 50 kHz. The ionic current signal was low pass filtered at 10 kHz in the patch clamp amplifier. Ionic current traces were analyzed using custom programs written in Matlab (the Mathworks, version 2018a).

Reads were filtered using a custom compression filter to eliminate transient fluctuations in ionic current unrelated to translocating DNA sequence. Enzyme-controlled DNA translocation events were detected with a thresholding algorithm as described in previous work[24]. For constant-voltage experiments, the open pore ionic current value was determined for the data, and an event was called whenever the ionic current drops below 75% of the open pore value. The event end was 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 ionic current less than 10% or greater than 70% of the open pore value) were automatically discarded. Remaining events were classified by-eye based to select events with a large number of enzyme steps. The same thresholding method was used for event detection in the variable-voltage data, with the sole difference being that the variable-voltage data was first downsampled to 200 Hz, thus removing the periodic characteristics of the signal.

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

Statistics

A complete accounting of the number of reads collected on each DNA strand can be found in §12.4.1.

In figure 12.1 (e), the uncertainty in the position shift as a function of voltage was determined using a bootstrapping method. The overall position shift was determined via analysis of the consensus signal of 18 variable-voltage reads of the same DNA sequence, as described in §12.2.2. Using 10 unique subsets drawn from these 18 reads, we conducted identical analyses of the position shift as a function of voltage. The reported uncertainty (shaded region around the shift curve) is the standard deviation of these bootstrapped measurements.

In figure 12.1, the uncertainty around each conductance measurement (shaded regions) was determined as follows. For variable-voltage measurements (figure 12.1 (a) and (b) lower panels and figure 12.1 (c) and (d)) we determined the covariance of the 3 principal component coefficients characterizing each segment by taking the covariance over the independent measurements of these 3 coefficients collected during each half-cycle of the voltage through the duration of that enzyme step (more in §12.2.4). We converted the associated covariance of each mean conductance curve to a standard deviation around the mean by taking 100 random draws from a multivariate normal distribution with matching mean and covariance, then taking the standard deviation of these 100 random curves at each DNA position. For the constant-voltage

measurements in the upper panels of figure 12.1 (a) and (b), the shaded regions represent the standard deviation around the mean conductance extracted from the variable-voltage data at each enzyme step at the DNA position corresponding to the constant-voltage operating value of 180 mV.

For the determination of constant-voltage sequencing accuracy, we measured the average identity rate over all 9368 bases sequenced with this method over 31 separate reads. The uncertainty in the overall accuracy was determined using a binomial error model as discussed in the main text. The average variable-voltage sequencing accuracy, as well as its associated uncertainty, were calculated in the same fashion, using all 17309 bases sequenced over 97 separate reads. The confusion matrices in figure 12.1.1 (c) and (d) broke down the sequencing results by base identity. For constant-voltage sequencing, the 9368 total calls broke down into 2203 As, 2188 Cs, 2166 Gs, 2144 Ts, and 667 gaps. For variable-voltage sequencing, the 17309 total calls broke down into 4021 As, 4073 Cs, 3813 Gs, 4081 Ts, and 1321 gaps.

12.2.2 Elongation of DNA in MspA

Stretching Measurement

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 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[43] of the elastic response of ssDNA 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

$$x = L_c(\coth(\frac{Fb}{k_BT}) - \frac{k_BT}{Fb})(1 + \frac{F}{S}), \tag{12.3}$$

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 $Fb >> k_BT$, so the coth term can be well-approximated as identically equal to 1. With this approximation, the force-extension relation simplifies to

$$x = L_c(1 - \frac{k_BT}{Fb})(1 + \frac{F}{S}) \tag{12.4}$$

The Kuhn length of ssDNA is known to depend upon salt concentration. From Bosco (2014)[73] we expect a Kuhn length of around 1.50 nm for the 400 mM KCl conditions in our variable-voltage experiments. We also from this publication take a reasonable value of the stretching modulus S to be 800 pN.

Following the analysis in Derrington (2015)[48], we observe that in our system the end-to-end extension x is fixed as the distance between the 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 $F = \alpha V$ (α some proportionality constant) gives

$$x = L_c(1 - \frac{k_BT}{\alpha Vb})(1 + \frac{\alpha V}{S}) \tag{12.5}$$

Changing the applied voltage from V to βV will change the contour length of the

DNA within the pore from L_c to ωL_c:

$$x = \omega L_c(1 - \frac{k_B T}{\beta \alpha V b})(1 + \frac{\beta \alpha V}{S}) \tag{12.6}$$

Here, the elongation ratio ω is the ratio between the contour length of DNA in the pore at the two voltages V and βV. Solving equations 12.5 and 12.6 for ω gives us a model predicting the elongation ratio ω as a function of the voltage ratio β as

$$\omega_{model} = \beta \langle \frac{(b\alpha V - k_B T)(S + \alpha V)}{(b\beta\alpha V - k_B T)(S + \beta\alpha V)} \rangle. \tag{12.7}$$

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

$$\omega_{meas}(\beta) = \frac{N_{ref} + \delta(\beta)}{N_{ref}}, \tag{12.8}$$

where δ is the measured position shift from 180 mV (figure 12.4) and N_{ref} is the number of nucleotides between the last hold point within the enzyme and the constriction at the reference voltage of 180 mV. Position shift is calculated as described in 12.2.2.

From Bhattacharya (2016) [40], we estimate $N_{ref} = 12$ nt. Fitting equation 12.7 to our data shows that a single parameter fit with $\alpha = 1.47 \pm 0.08 \frac{e^-}{nm}$ describes the data well (figure 12.4). 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 α parameter corresponds to a force of $\sim$42 pN at 180 mV. This force estimate has larger uncertainties than the uncertainty in α as the estimate is criti-

cally 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 vestibule, voltage dependence fo the position of the enzyme relative to MspA, and voltage-induced deformation of the enzyme or the pore.

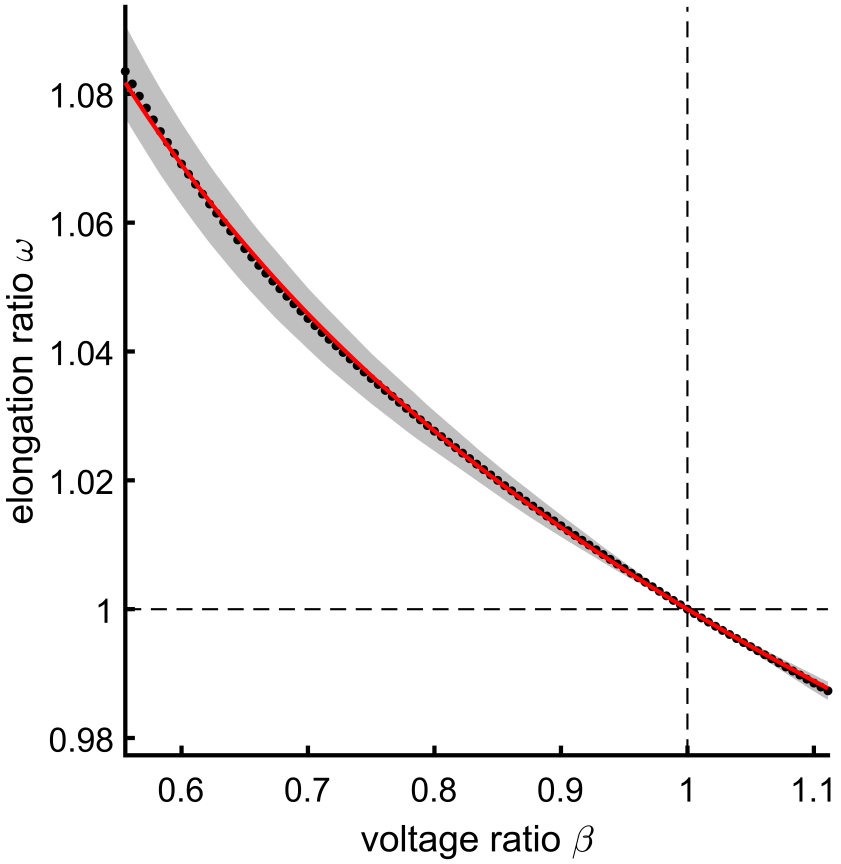

Figure 12.4: **DNA stretching in MspA.** The fractional DNA elongation relative to DNA position at 180 mV (ω) is plotted (black points) relative to the fractional applied voltage relative to 180 mV (β). The gray shading shows the 1 standard deviation interval. The uncertainty was determined using 10 bootstraps, each consisting of a random distinct subset of the 18 read consensus used for the stretching measurement. The DNA stretch was calculated individually for each bootstrapped consensus, and the overall standard deviation in the stretch was taken as the standard deviation over these bootstrapped measurements. The red line shows the ex-FJC fit with $\alpha = 1.47 \pm 0.8 \frac{e^-}{nm}$.

Position Shift Calculation

The position shift calculation was done using the consensus conductance measurements for a known DNA sequence (stretching read strand, table 12.2). The consensus conductances were determined from 18 separate measurements of the DNA sequence using standard variable-voltage sequence conditions (table 12.3).

The position shift between two voltages V_1 and V_2 is determined as the shift that best places the conductance profile measurements from each of the two voltages along a single spline. The shift is calculated between consecutive voltages (e.g. 101 mV to 102 mV) because the conductance profile changes over large changes in voltages. As the DNA is more elongated at higher voltages, the conductance measured at each DNA position is a function of a smaller number of nucleotides. Consequently, features of the conductance profile that are blurred out at low voltages due to averaging can become apparent at higher voltages. This effect is accounted for during our normalization procedure, discussed in 12.2.10.

The shift best placing the conductance measurements from the two voltages along a single spline is calculated as follows: After first normalizing the conductances (12.2.10), 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 ϕ, we calculate a match score $\mathcal{M}$ by taking the sum-square difference between the two splines for the given shift, and dividing out by the total number of compared points:

$$\mathcal{M}(\phi) = \frac{\sum_{i=1}^{N_{pts}} (spG_1^{(i)} - spG_2^{(i+\phi)})^2}{N_{pts} - \phi} \tag{12.9}$$

The shift ϕ_0 giving the best (smallest) match score gives us the shift that makes

the two splines most similar. This ϕ_0 is taken as the position shift between V_1 and V_2. The uncertainty $\delta\phi$ was determined by measuring the shift using subsets of the entire dataset. From the longer measured sequence, we extracted 9 distinct subsets, and measured the shift function for each.We took $\delta\phi$ as the standard deviation of these separate measurements.

12.2.3 6-mer map constrtuction

General Considerations

To sequence the variable-voltage reads, we determine the DNA sequence most likely to have generated the observed series of conductance states. In order to decode this DNA sequence, we require a map relating the conductance signal and the DNA sequence in the pore. The nanopore signal is modeled as being generated by the k nucleotides (i.e. the k-mer, with k an integer) nearest the constriction of the pore. This model is described by a map of the 4^k possible k-mers to the conductances typically observed when they are in the pore. The k-mer model has been previously validated as an effective model for nanopore signal prediction[31].

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. To better model this effect, we expanded the 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 $\boldsymbol{p}$ and their covariance $\boldsymbol{\Sigma^p}$ (§12.2.4).

Previous work by our lab used the Φ29 DNA polymerase (DNAP) as the motor protein controlling the DNA, which steps in full nucleotide increments. This work 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[48]. As the 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 the model represent 8192 total states, two for each k-mer.

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 conductance of the translocating DNA 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 phi29 DNAP 4-mer model (figure 12.5).

Sampling this curve at the appropriate DNA positions gives an estimate of the variable-voltage 6-mer states. The first of the two Hel308 helicase states is known to have the same DNA registration within the pore as Φ29 DNAP[48]. The constant-voltage model was derived from experiments run at a bias of $180mV$, and we know from the DNA force-extension curve (§12.2.2) that the variable-voltage sweep stretches the DNA $\sim +0.1nt$ from the $180mV$ position at its highest voltage and $\sim -0.9nt$ from the $180mV$ position at its lowest voltage. So, to predict the state 1 conductance vs. position curve, we sample the interpolating curve at equally-spaced points from $-0.9nt$ to $+0.10nt$ (figure 12.5 (a)). 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 $0.65nt$ (figure 12.5 (b)).

The amplitudes of the 3 principle components $\boldsymbol{p}$ for each 6-mer state can now be calculated from the predicted $G - V$ curve (§12.2.4). 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.

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, then use the measured signals of this known DNA to update the initial 6-mer model.

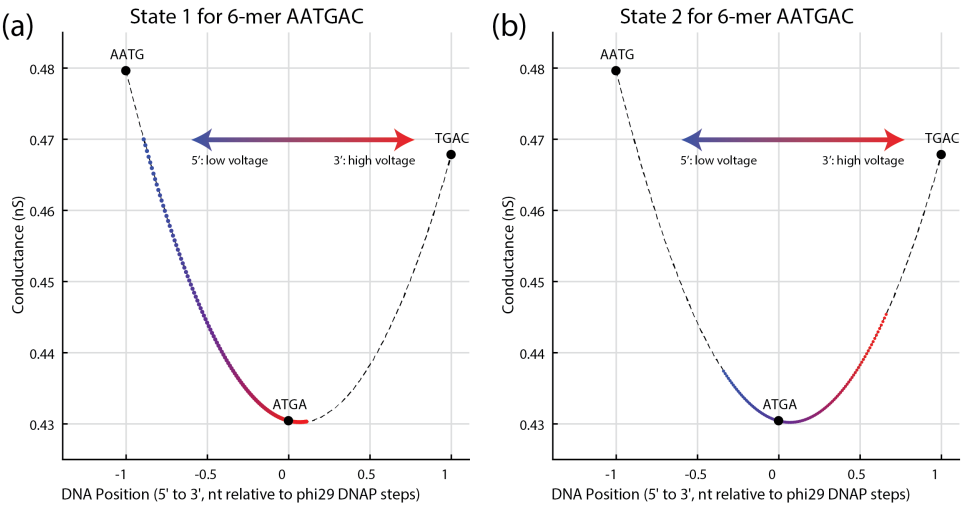

Figure 12.5: **Method of generating variable-voltage Hel308 helicase 6-mer predictions from the constant-voltage Φ29 DNAP 4-mer model.** Black points show the constant-voltage 4-mer model predictions for the 3 4-mers comprising the 6-mer of interest. Black dashed line shows the quadratic fit to the 4-mer 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 Φ29 DNAP full-nucleotide steps. The conductance value of the Hel308 state 1 prediction coincides with the Φ29 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.

ΦX174 We first measured the 5386 bp ΦX174 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). ΦX174 was prepared in 20 μg batches. For each batch, a mixture of

   (a) 40 μL of ΦX174 DNA at 500 $\frac{ng}{\mu L}$

   (b) 5 μL of 10x CutSmart (New England Biolabs) Buffer

   (c) 1 μL of PstI-HF restriction enzyme at 100 $\frac{Units}{\mu L}$

   (d) 1 μL of AvaII restriction enzyme at 10 $\frac{Units}{\mu L}$

   (e) 3 μL 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 (figure 12.6 (a) and (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 μL of molecular biology grade water.

3. Two DNA adapters are attached to each of the fragments enabling reading by the nanopore (figure 12.6 (c) and (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 12.1). The threading adapter is formed at high concentration by mixing equal volumes of 12.5 μM threading strand and cholesterol blocker, then annealing to yield a 12.5 μM 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 12.1). The loading adapter is formed at high concentration by mixing equal volumes 12.5 μM loading strand and loading blocker, then annealing to yield a 12.5 μM 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 μL of 100 nM $\Phi X174$ DNA

   (b) 6 μL of 10x T4 ligase buffer (New England Biolabs)

   (c) 2 μL of 12.5 μM threading adapters (to give 5:1 ratio of adapters to target sticky ends)

   (d) 2 μL of 12.5 μM loading adpaters (to give 5:1 ratio of adapters to target sticky ends)

   (e) 1 μL of molecular biology grade water

   (f) 1 μL 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 (figure 12.6 (e)) were purified from the remaining un-ligated adapters and the heat-inactivated ligase on a DNA Clean and Concentrator column, and eluted into 50 μL 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.

λ 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 λ 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, 12.2.13), 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 λ 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 simply for easy of use, as these required only

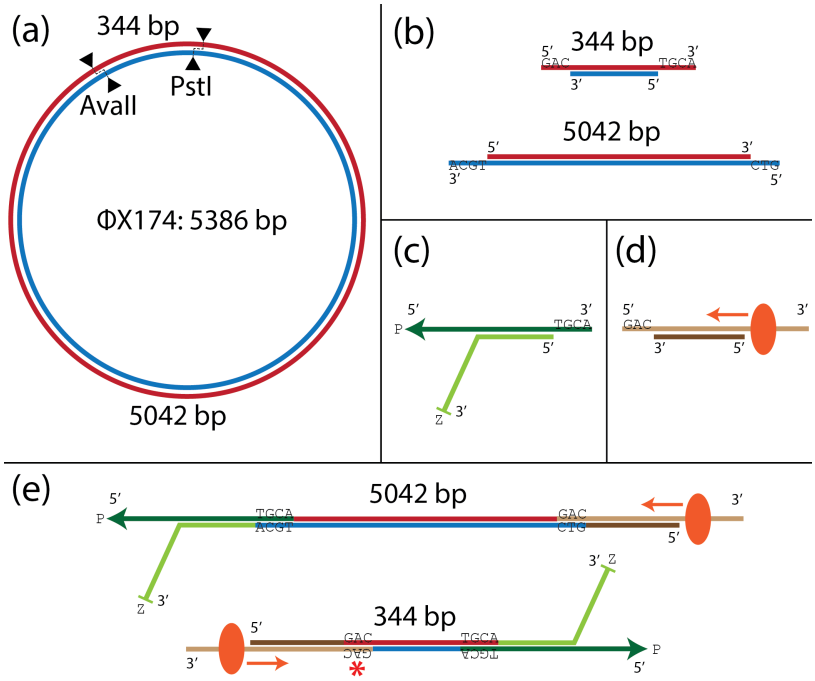

Figure 12.6: **Preparation of ΦX 174 for variable-voltage sequencing.** (a) The circular 5386 bp genome is cut twice using the AvaII and PstI restriction enzymes. (b) Restriction results in two fragments of 344 and 5042 bp, with sticky ends of size 3 and 4 nt. (c) The threading adapter consists of a 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 fo 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.

a centrifuge, and not the Covaris sonicator instrument. For both shearing methods, the library preparation proceeded as follows:

1. The full length λ DNA (Promega) was fragmented using either Blue miniTUBEs (in 20 μg batches) or gTUBEs (in 30 μg batches) (figure 12.7 (a) and (b)).

   For miniTUBE fragmentation, 20 μg of λ DNA was suspended in Tris EDTA buffered at pH 8.0 to a total volume of 200 μL. DNA was then fragmented using the Covaris M220 Focused-ultrasonicator, using the recommended settings for product fragments of ∼3000 bp in length.

   For gTUBE fragmentation, 30 μg of λ DNA was suspended in molecular biology grade water to a total volume of 150 μL. The gTUBE was then centrifuged on an Eppendorf 5417R centrifuge for 30 seconds at 12400 rpm (corresponding to 16200 g), resulting in fragments of ∼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 (figure 12.7 (c)). Using the NEBNext end repair module (New England Biolabs), a mixture of

   (a) 5 μg fragmented DNA

   (b) 10 μL of NEBNext 10x End Repair Reaction Buffer

   (c) 5 μL of NEBNext End Repair Enzyme Mix

   (d) Molecular biology grade water to total volume of 100 μL

   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 (figure 12.7 (d)). A mixture of

   (a) 5 μg of λ DNA

   (b) 5 μL of 10x NEBNext dA-Tailing Reaction Buffer

   (c) 3 μL of Klenow Fragment (3' $\rightarrow$ 5' exo$^-$)

   (d) Molecular biology grade water to a total reaction volume of 50 μL

   was incubated at 37°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 λ 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 (figure 12.7 (e)), equimolar parts of the λ threading strand and the λ cholesterol blocker (table 12.1) were mixed and annealed to a final concentration of 10 μM. Similarly, for the loading adapter (figure 12.7 (f)), equal molar parts of the λ loading strand and the λ loading blocker were mixed and annealed to a final concentration of 10 μM.

5. Adapters were ligated to the dA-tailed λ DNA fragments using T4 DNA ligase (New England Biolabs). A mixture of

   (a) 10 μg of λ DNA fragments

   (b) 3 μL of 10 μM threading adapters (for a $\sim$ 10:1 adapter to dA-tail end ratio)

   (c) 3 μL of 10 μM loading adapters (for a $\sim$ 10:1 adapter to dA-tail end ratio)

(d) 15 μL of 10X Ligation Buffer

(e) 7.5 μL T4 DNA Ligase

(f) Molecular biology grade water up to a total reaction volume of 150 μL

was incubated at 22°C for 125 minutes, then heat inactivated at 65°C for 10 minutes. The ligation products (figure 12.7 (g)) 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 λ 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.

Building the Empirical 6*-mer Model from Genomic Reads*

Having measured a total of 309410 enzyme steps along genomic DNA tracks (120867 in λ, 188543 in $\Phi X174$) 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 ($\Phi X174$ or λ).

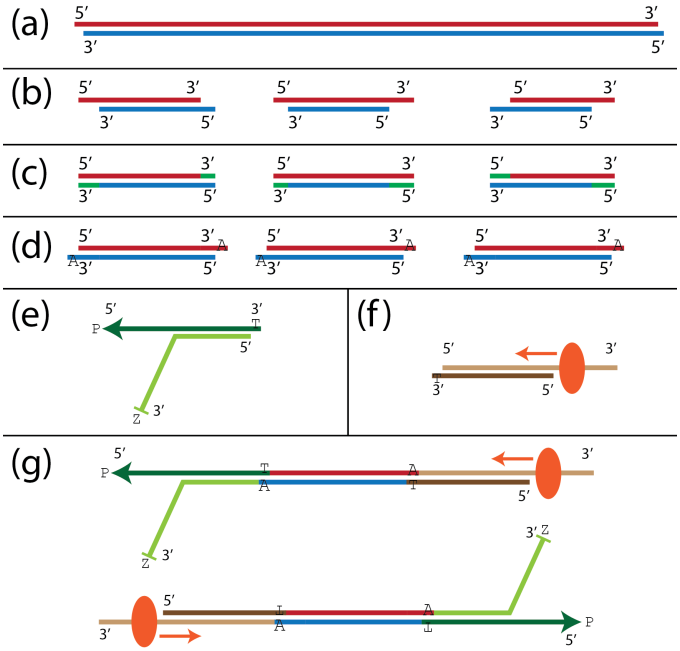

Figure 12.7: **λ DNA fragmentation.** (a) Full length double stranded λ DNA, 48502 bp. (b) The genomic DNA is sheared into random fragments of average length 3 kb (miniTUBE) or 6 kb (gTUBE). (c) The random 3' and 5' overhangs generated through the shearing are repaired (green segments) using the NEBNext 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 λ 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 λ 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.

For the first construction of the empirical model, the predicted signal is given by the initial model described previously.

Each read of genomic DNA is aligned to the predicted signal for its reference sequence using the BCJR alignment algorithm[51]. The alignment maps the $G - V$ curve at each measured conductance 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 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 (figure 12.8). 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.

Filling in Unmeasured 6*-mers*

After constructing the empirical 6-mer model from long reads of $\Phi X174$ and λ 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[74] to construct a minimal

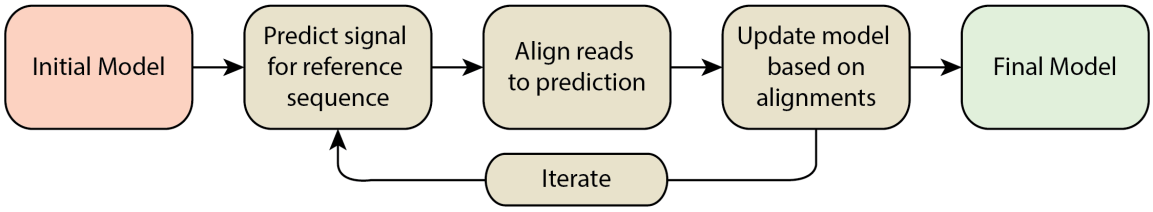

Figure 12.8: **Iterative map construction flow chart.**

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 12.1, figure 12.9). We 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 expectation maximization approach: 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.

12.2.4 Variable voltage state features

Following change point detection (12.2.7) and capacitance compensation (12.2.8), 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

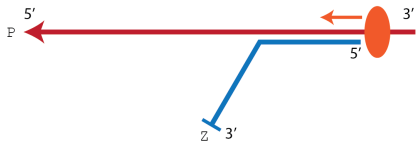

Figure 12.9: **Fill-in strands for 6-mer map.** DNA constructs for the fill-in data set are composed of two partially-complementary short oligos. The fill-in template (one of a possible 6) in red has a 5' terminal phosphate (P, arrowhead) facilitating threading of this end into the pore. The 5' phosphate 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, cross-bar). 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.

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, $\boldsymbol{g}$. 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 $\boldsymbol{g}$ 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 (figure 12.10).

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

$$\boldsymbol{p}_i = \langle \boldsymbol{\pi}_1; \boldsymbol{\pi}_2; \boldsymbol{\pi}_3 \rangle^T * \boldsymbol{g}_i, \tag{12.10}$$

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 (figure 12.11).

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 $\boldsymbol{g}_i^j$ of the state's conductance feature vector $\boldsymbol{g}_i$, 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 $\boldsymbol{p}_i^j$ as distinct measurements of the overall principal component feature vector $\boldsymbol{p}_i$. Given t half-cycle measurements, we can estimate the covariance in the state's conductance ($\boldsymbol{\Sigma}_i^{\boldsymbol{g}}$) and principal component features ($\boldsymbol{\Sigma}_i^{\boldsymbol{p}}$) as

$$\boldsymbol{\Sigma}_i^{\boldsymbol{g}} = \mathbb{E}_{j \in 1:t} \langle (\boldsymbol{g}_i^j - \mathbb{E}_{j \in 1:t} \langle \boldsymbol{g}_i^j \rangle)(\boldsymbol{g}_i^j - \mathbb{E}_{j \in 1:t} \rangle \boldsymbol{g}_i^j \rangle)^T \rangle \tag{12.11}$$

and

$$\boldsymbol{\Sigma}_i^{\boldsymbol{p}} = \rangle_{j \in 1:t} \langle (\boldsymbol{p}_i^j - \rangle_{j \in 1:t} \langle \boldsymbol{p}_i^j \rangle)(\boldsymbol{p}_i^j - \rangle_{j \in 1:t} \langle \boldsymbol{p}_i^j \rangle)^T \rangle. \tag{12.12}$$

The estimators $\boldsymbol{\Sigma}_i^{\boldsymbol{g},\boldsymbol{p}}$ 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, $\boldsymbol{\Sigma}_i^{\boldsymbol{g},\boldsymbol{p}}$ has $\frac{d}{2} * (d-1)$ independent entries, where d is the dimensionality of the associated $\boldsymbol{g}$ or $\boldsymbol{p}$ feature vector. So, in order to get a good

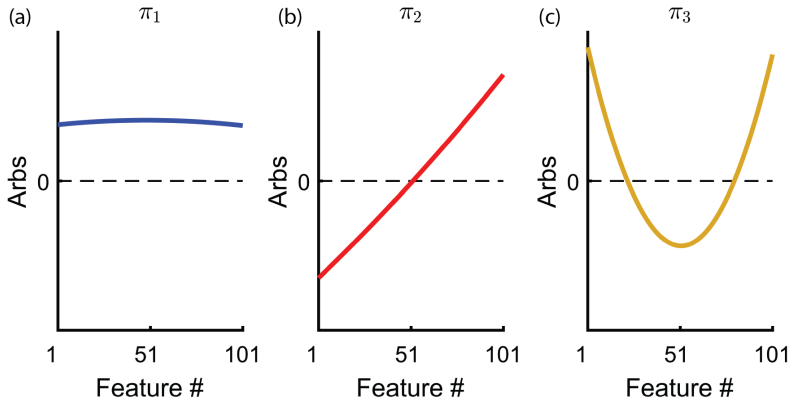

Figure 12.10: **Principal component vectors for feature extraction.** (a), (b), and (c) show the first, second, and third principal component vectors for the variable voltage data, respectively. Linear combinations of these three vectors can describe all observed conductance vs. DNA position states. The three vectors roughly represent an offset (a), slope (b), and curvature (c) and thus primarily describe the states as quadratic curves.

estimate of the covariance of the conductance features $\boldsymbol{g}_i$ 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 $\boldsymbol{p}_i$ 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 (12.2.12). 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.

12.2.5 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

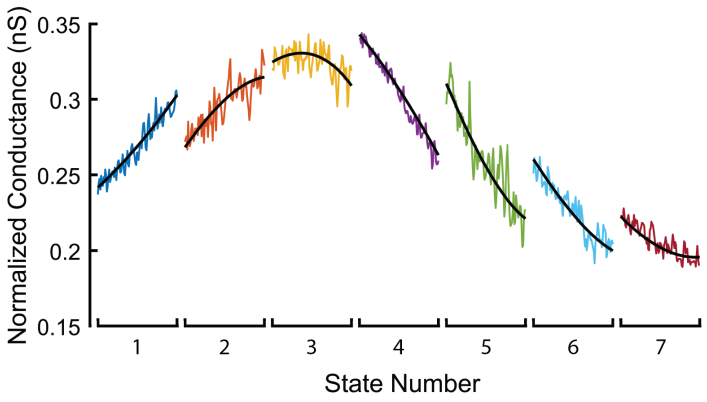

Figure 12.11: **Principal component description of conductance states.** Linear combinations of the 3 principal components (black curves) satisfactorily describe the 101-dimensional conductance states (colored curves). The description preserves the state shape while discarding parameters describing only noise.

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 methods equally.

The constant-voltage model is extracted from the variable-voltage model as shown in figure 12.12. 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 value of the variable-voltage conductance curve at that same point.

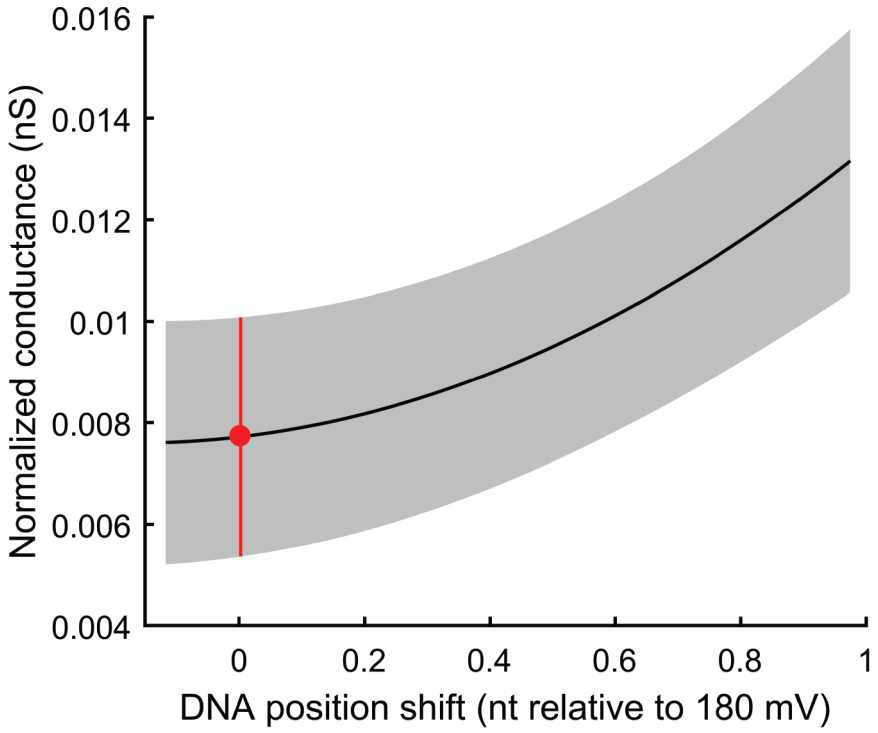

Figure 12.12: **Constant voltage model extraction.** The constant-voltage model value for a given 6-mer (e.g. *ATGAGA*) is taken as the point (red) in the variable-voltage conductance curve for that 6-mer in the variable-voltage model (black) corresponding to 180 mV (0 nt shifted relative to 180 mV). The uncertainty (red line) is taken as the variation in the variable-voltage model prediction (gray shading) at the 180 mV point.

12.2.6 Flicker removal 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 conductance than the state they start from, and always return to the state they started in. These flicker states cannot be mapped to any predicted conductance 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 (§12.2.9) 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 conductance measurement x_n in a read with a one-sample t-test against the data surrounding it:

$$t_n = \frac{x_n - \mu_{[n-k,n+k]}}{\sigma_{[n-k,n+k]}}, \tag{12.13}$$

where

$$\mu_{[n-k,n+k]} = \frac{1}{2k}\left[\left(\sum_{m=n-k}^{n+k} x_m\right) - x_n\right] \tag{12.14}$$

and

$$\sigma^2_{[n-k,n+k]} = \frac{1}{2k}\left[\left(\sum_{m=n-k}^{n+k} x_m^2\right) - x_n^2\right] - \mu^2_{[n-k,n+k]} \tag{12.15}$$

are the mean and variance of the data k points to the left and right of the point being

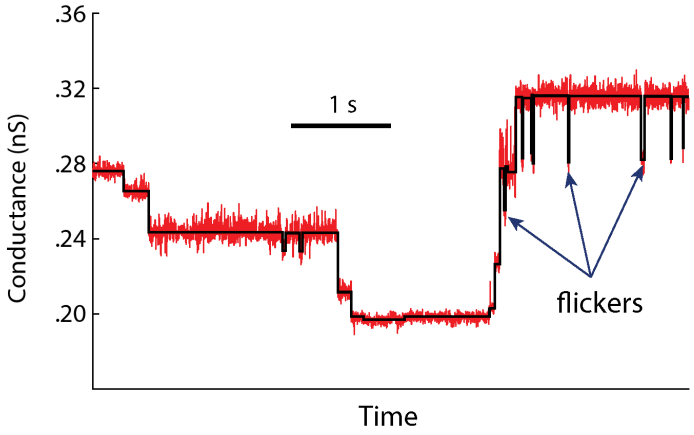

Figure 12.13: **Example of flicker states.** The raw ionic current trace (downsampled to 5 kHz) for a Hel308-controlled DNA translocation event is shown in red, with the states found by the change point detection algorithm overlaid in black. The arrows identify several flickers–transient decreases in the ionic current that are not cannot be mapped to any sequence state.

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. In 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 in constant-voltage experiments.

12.2.7 Change point detection

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 (figure 12.14). 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 (§12.2.4), along with their associated covariance.

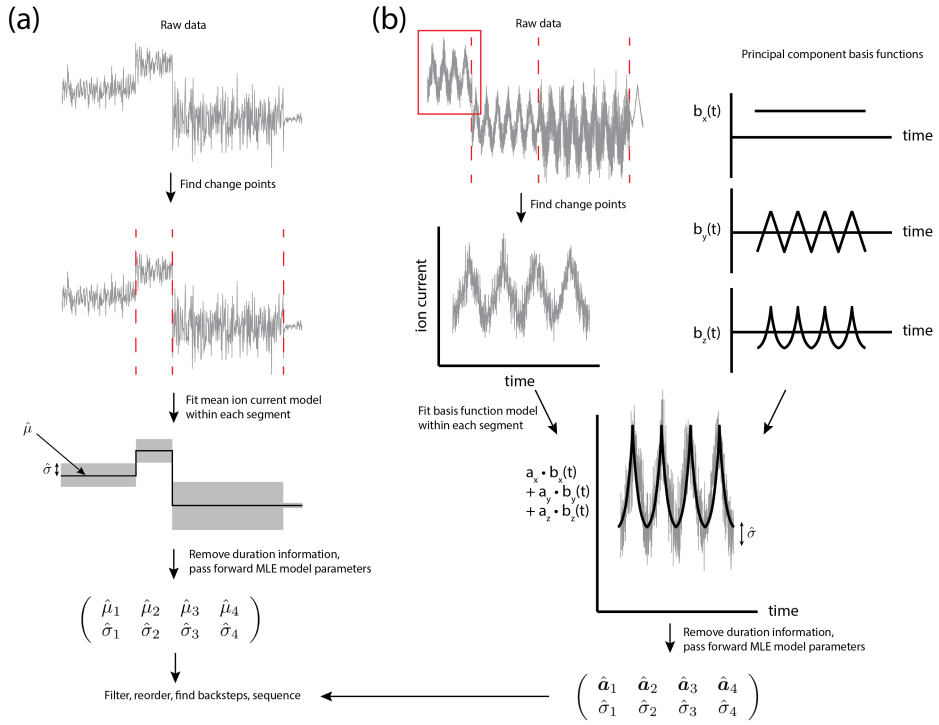

Figure 12.14: **Schematic of change point detection aglorithm.** (a) For constant-voltage experiments, change point detection involves finding transitions between states of different ionic current means and variances. (b) For variable-voltage experiments, we are looking for changes in the parameters of a time-dependent model describing the data. The model we use is a sum of basis functions which are the principal components of the nanopore signal, parameterized by the amplitude of each basis function.

The data is partitioned into enzyme steps using a change point detection algorithm

(algorithm 12.7). 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 θ_T) or by two models (θ_L, θ_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 modeled the variable-voltage data by the five largest principal components of the periodic functions that represent the raw data of each enzyme state (figure 12.15). 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. 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[55][56] to determine the appropriate penalty.

Algorithm 12.7 Change point detection algorithm

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

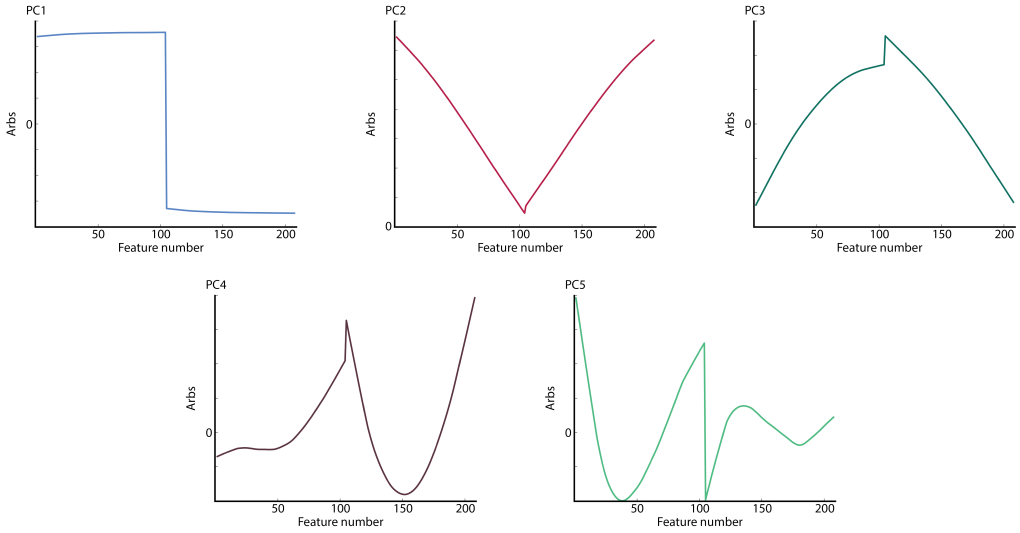

Figure 12.15: **Principal components for change point detection.** The five principal component vectors used to model the variable-voltage time series date are shown. Linear combinations of these five vectors can describe the observed data.

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 $\{x_1, x_2, \ldots, x_N\}$, $x \in \mathbb{R}^k$, choose a model consisting of some number of change points $\{t_a, t_b, \ldots\}$, and a different set of parameters $\{\theta_a, \theta_b, \ldots\}$ describing the data between each change point. Our change point detection algorithm (algorithm 12.7) 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 θ with normally distributed noise σ. Under these assumptions, the probability density of obtaining a measurement $x(t)$ at a time t given a choice of parameters θ for that time is

$$p\left(x(t), t \,|\, \theta\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f(t;\theta)-x(t))^2}{2\sigma^2}} \tag{12.16}$$

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

$$p\left(x \,|\, \theta\right) = \prod_{t=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f(\theta)_t - x_t)^2}{2\sigma^2}} \tag{12.17}$$

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

$$\log \mathcal{L} = -\frac{1}{2} \sum_{t=1}^{N} \log 2\pi\sigma^2 + \frac{(f(\theta)_t - x_t)^2}{\sigma^2} \tag{12.18}$$

For change point detection, we are interested in the relative likelihood between using two different sets of parameters θ_L and θ_R to model the data to the left and right of a possible change point, versus using one set of parameters θ_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

$$\log \mathcal{L} = -\frac{1}{2}\Bigg[\sum_{t=L}^{L+N_L-1} \log 2\pi\sigma_L^2 + \frac{(f(\theta_L)_t - x_t)^2}{\sigma_L^2} + \sum_{t=L+N_L}^{N_T} \log 2\pi\sigma_R^2 + \frac{(f(\theta_R)_t - x_t)^2}{\sigma_R^2} - \sum_{t=L}^{L+N_T-1} \log 2\pi\sigma_T^2 + \frac{(f(\theta_T)_t - x_t)^2}{\sigma_T^2} \Bigg] \tag{12.19}$$

If our maximum likelihood estimate for θ 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:

$$\log \hat{\mathcal{L}} = -\frac{1}{2}\left[N_L \log 2\pi\hat{\sigma}_L^2 + N_L \frac{\hat{\sigma_L}^2}{\hat{\sigma}_L^2} + N_R \log 2\pi\hat{\sigma}_R^2 + N_R \frac{\hat{\sigma}_R^2}{\hat{\sigma}_R^2} - N_T \log 2\pi\hat{\sigma}_T^2 - N_T \frac{\hat{\sigma}_T^2}{\hat{\sigma}_T^2}\right] \tag{12.20}$$

$$= -\frac{1}{2}\left[N_R \log \hat{\sigma}_R^2 + N_L \log \hat{\sigma}_L^2 - N_T \log \hat{\sigma}_T^2\right] \tag{12.21}$$

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[55][56] 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

$$\text{CPIC} = -\log \hat{\mathcal{L}} + p(N, d) = \frac{N_R}{2}\log \hat{\sigma}_R^2 + \frac{N_L}{2}\log \hat{\sigma}_L^2 - \frac{N_T}{2}\log \hat{\sigma}_T^2 + p_d(N_T) \tag{12.22}$$

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

$$\text{CPIC}(\lambda) = \frac{N_R}{2} \log \hat{\sigma}_R^2 + \frac{N_L}{2} \log \hat{\sigma}_L^2 - \frac{N_T}{2} \log \hat{\sigma}_T^2 + \lambda p_d(N_T) \tag{12.23}$$

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

$$\hat{\mu} = \frac{1}{R - L + 1} \sum_{t=L}^{R} x_t \tag{12.24}$$

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

$$\hat{\mu} = \frac{X_R - X_L}{R - L + 1}. \tag{12.25}$$

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,

$$\hat{\sigma}^2 = \frac{1}{R - L + 1} \sum_{t=L}^{R} (x_t - \hat{\mu})^2 = \left[\frac{1}{R - L + 1} \sum_{t=L}^{R} x_t^2 \right] - \hat{\mu}^2 \tag{12.26}$$

Again defining and pre-calculating the cumulate sum

$$X_t^2 = \sum_{s=1}^{t} x_s^2, \tag{12.27}$$

we quickly calculate the MLE variance as

$$\hat{\sigma}^2 = \frac{X_R^2 - X_L^2}{R - L + 1} - \hat{\mu}^2. \tag{12.28}$$

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 θ_i,

$$f(\theta)_t = \sum_{i=1}^{p} \theta_i b_{it}. \tag{12.29}$$

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

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{t=L}^{R} \left(x_t - \sum_{i=1}^{p} \theta_i b_{it} \right)^2 \tag{12.30}$$

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

$$2 \sum_{t=L}^{R} \left(x_t - \sum_{i=1}^{p} \hat{\theta}_i b_{it} \right) b_{jt} = 0 \tag{12.31}$$

$$\sum_{t=L}^{R} x_t b_{jt} = \sum_{i=1}^{p} \hat{\theta}_i \sum_{t=L}^{R} b_{it} b_{jt} \tag{12.32}$$

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

$$B_{ijt} = \sum_{s=1}^{t} b_{it} b_{jt} \quad \text{(note: } B_{ijt} \text{ is symmetric in } i \text{ and } j.) \tag{12.33}$$

$$c_{it} = \sum_{s=1}^{t} x_t b_{it} \tag{12.34}$$

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

j, the expression for $\hat{\theta}$ becomes

$$[\mathbf{c}_R - \mathbf{c}_L]^T = \hat{\boldsymbol{\theta}}^T [B_R - B_L]^T \tag{12.35}$$

$$[\mathbf{c}_R - \mathbf{c}_L] = [B_R - B_L] \hat{\boldsymbol{\theta}} \tag{12.36}$$

$$\hat{\boldsymbol{\theta}} = [B_R - B_L]^{-1} [\mathbf{c}_R - \mathbf{c}_L] \tag{12.37}$$

We can now calculate σ on that domain to be used in the CPIC calculation. The sum of squared errors is

$$\hat{\sigma}^2 = \frac{1}{R - L + 1} \sum_{t=L}^{R} \left[x_t - \sum_{i=1}^{p} \hat{\theta}_i b_{it} \right]^2 . \tag{12.38}$$

Expanding the squared term,

$$\hat{\sigma}^2 = \frac{1}{R - L + 1} \sum_{t=L}^{R} \left[x_t^2 - 2 \sum_{i=1}^{p} \hat{\theta}_i x_t b_{it} + \sum_{i,j=1}^{p} \hat{\theta}_i b_{it} b_{jt} \hat{\theta}_j \right] . \tag{12.39}$$

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

$$\hat{\sigma}^2 = \frac{1}{R - L + 1} \left[\sum_{t=L}^{R} x_t^2 - 2 \sum_{i,j=1}^{p} [B_R - B_L]^{-1}_{ij} [c_R - c_L]_j \sum_{t=L}^{R} x_t b_{it} + \sum_{i,j,k,l=1}^{p} [B_R - B_L]^{-1}_{ik} [c_R - c_L]_k \left(\sum_{t=L}^{R} b_{it} b_{jt} \right) [B_R - B_L]^{-1}_{jl} [c_R - c_L]_l \right] . \tag{12.40}$$

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

cumulate expressions,

$$\hat{\sigma}^2 = \frac{1}{R-L+1}\left[X_R^2 - X_L^2 - 2\sum_{i,j=1}^{p}\left[B_R - B_L\right]_{ij}^{-1}\left[c_R - c_L\right]_j\left[c_R - c_L\right]_i + \sum_{i,j,k,l=1}^{p}\left[B_R - B_L\right]_{ik}^{-1}\left[c_R - c_L\right]_k\left[B_R - B_L\right]_{ij}\left[B_R - B_L\right]_{jl}^{-1}\left[c_R - c_L\right]_l\right]. \tag{12.41}$$

$$\hat{\sigma}^2 = \frac{1}{R-L+1}\left[X_R^2 - X_L^2 - 2\sum_{i,j=1}^{p}\left[B_R - B_L\right]_{ij}^{-1}\left[c_R - c_L\right]_j\left[c_R - c_L\right]_i + \sum_{i,j=1}^{p}\left[B_R - B_L\right]_{ik}^{-1}\left[c_R - c_L\right]_j\left[c_R - c_L\right]_i\right]. \tag{12.42}$$

$$\hat{\sigma}^2 = \frac{1}{R-L+1}\left[X_R^2 - X_L^2 - \sum_{i,j=1}^{p}\left[c_R - c_L\right]_i\left[B_R - B_L\right]_{ij}^{-1}\left[c_R - c_L\right]_j\right] \tag{12.43}$$

Or, in vector notation,

$$\hat{\sigma}^2 = \frac{1}{R-L+1}\left[X_R^2 - X_L^2 - \left[\mathbf{c}_R - \mathbf{c}_L\right]\left[B_R - B_L\right]^{-1}\left[\mathbf{c}_R - \mathbf{c}_L\right]\right] \tag{12.44}$$

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.

12.2.8 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 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 (figure 12.16 (a) and (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 $I_{down}(V)$ (figure 12.16 (c)).

4. Taking the difference between the two $I-V$ curves, we get the asymmetry between the sweeps, $H(V) = I_{down}(V) - I_{up}(V)$ (figure 12.16 (d)).

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 $H(V)$ (figure 12.16 (e) and (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^{cc}_{up} and I^{cc}_{down}:

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

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

The corrected curves show no residual hysteresis, and the spikes around the turnarounds have been eliminated (figure 12.16 (g)).

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 (figure 12.16 (h)).

12.2.9 Enzyme 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, the recombination filter, and the reorder filter. Each stage of state filtering aims to eliminate a specific error mode common to the data.

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

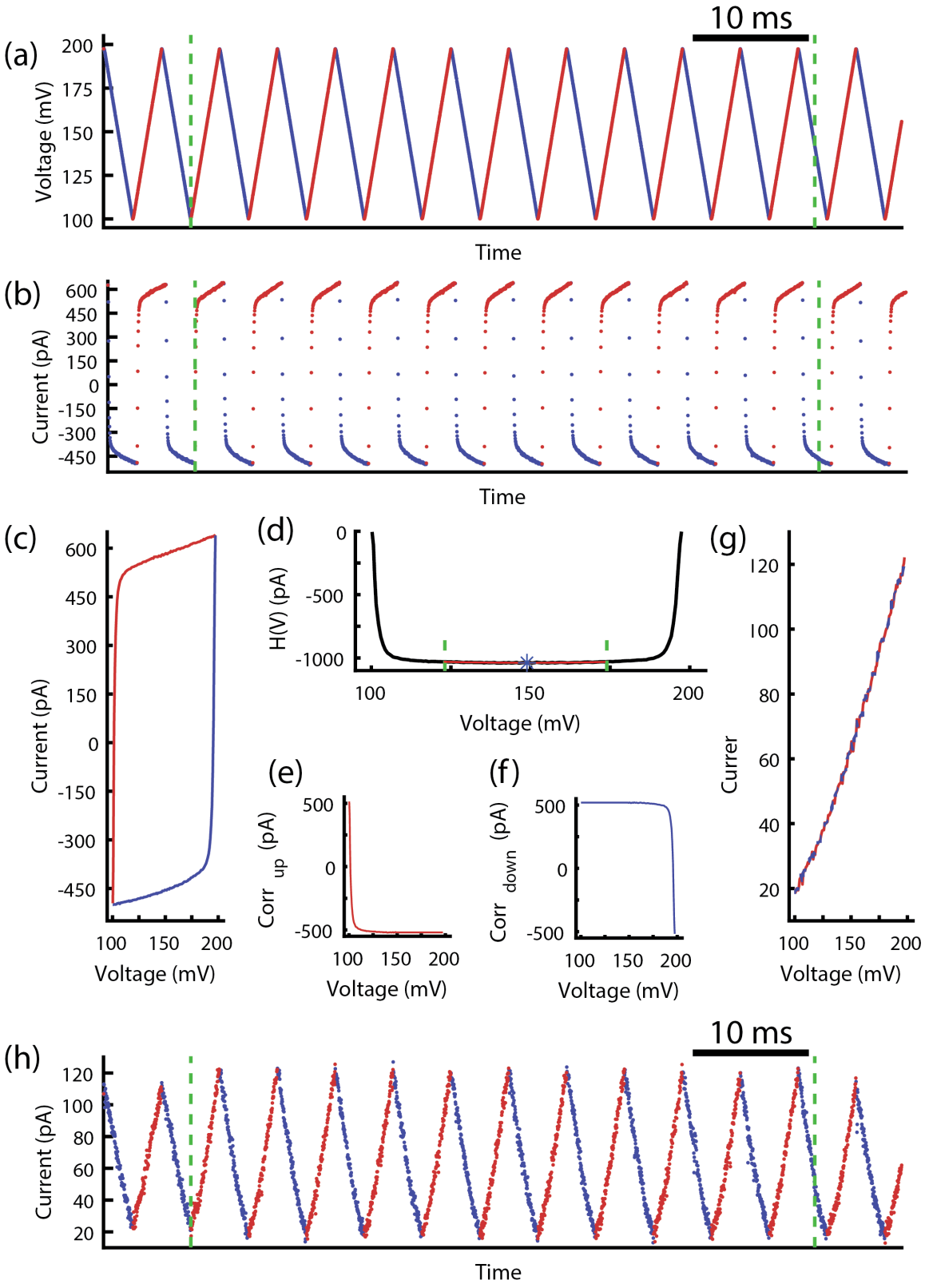
(a)
10 ms
Voltage (mV)
200
175
150
125
100
Time
(b)
Current (pA)
600
450
300
150
0
-150
-300
-450
Time
(c)
Current (pA)
600
450
300
150
0
-150
-300
-450
100 150 200
Voltage (mV)
(d)
H(V) (pA)
0
-500
-1000
100 150 200
Voltage (mV)
(e)
Corr up (pA)
500
0
-500
100 150 200
Voltage (mV)
(f)
Corr down (pA)
500
0
-500
100 150 200
Voltage (mV)
(g)
Currer
120
100
80
60
40
20
100 150 200
Voltage (mV)
(h)
10 ms
Current (pA)
120
100
80
60
40
20
Time

Figure 12.16: **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.

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 12.2.6, 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 (12.2.7) 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 12.8). The process is iterated because P^i_{bad}, 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^i_{bad} 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 12.8 Removal Filter

assume we start with N states $\{\boldsymbol{x}^i\}_{i\in 1:N}$
$stop \leftarrow false$
while $\sim stop$ **do**
    $anyremoved \leftarrow false$
    calculate the SVM feature vectors $\{\boldsymbol{\xi}^i\}_{i\in 2:N-1}$ **from** the states $\{\boldsymbol{x}^i\}_{i\in 1:N}$
    calculate the bad state probabilities $\{P^i_{bad}\}_{i\in 2:N-1}$ **from** the SVM feature vectors $\{\boldsymbol{\xi}^i\}_{i\in 2:N-1}$
    for $j \in 2:N-1$ **do**
        if $P^i_{bad} > T_{thresh}$ **then**
            remove $\boldsymbol{x}^j$ from $\{\boldsymbol{x}^i\}$ ▷ Remove states above the removal threshold
            $anyremoved \leftarrow true$ ▷ Stop iterating if nothing is removed
        end if
    end for
    if $\sim anyremoved$ **then**
        $stop \leftarrow true$
    end if
end while
remove $\boldsymbol{x}^1$ and $\boldsymbol{x}^N$ from $\{\boldsymbol{x}^i\}$ ▷ 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 (12.2.4) 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 (12.2.3). 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 (12.2.3) 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 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, 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 (figure 12.17 (a)). 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 $\mathbb{S}$ (figure 12.17 (b)). These data are then fit by the logit function

$$f(\mathbb{S}|\alpha,\beta) = \frac{1}{1-e^{-(\alpha\mathbb{S}+\beta)}} \tag{12.45}$$

using a global likelihood maximization fit (figure 12.17).

Together, the SVM and the fit logit function give us a way to calculate P^i_{bad} 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^i_{bad} for the state.

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

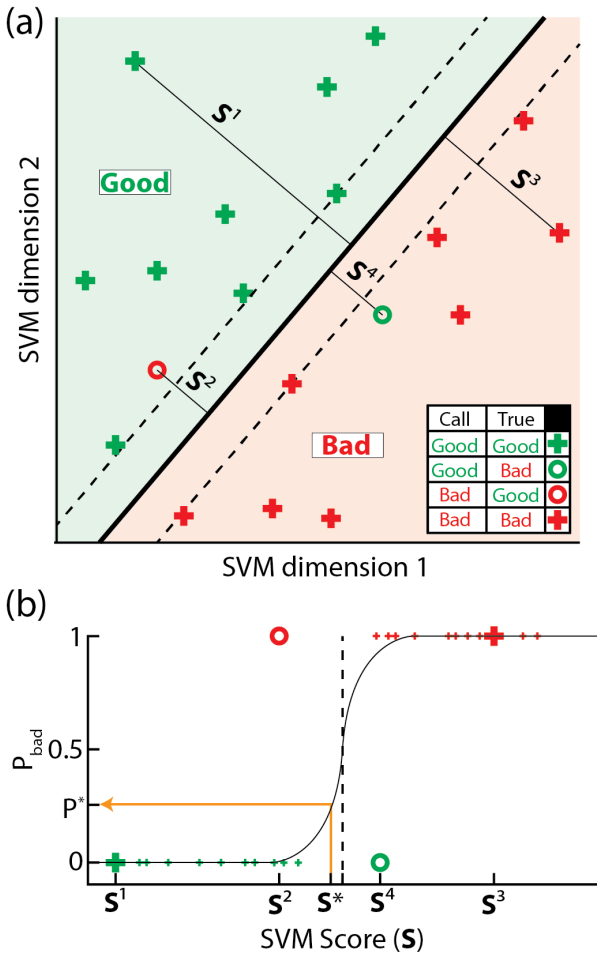

Figure 12.17: **Converting SVM outputs to P_{bad} probabilities.** (a) In this classifier (contrived data), points occupying the space (shaded green) above the decision boundary (solid black line) are classified as good while those below (shaded red) are classified as bad. Points marked with a "plus" 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 $\mathbb{S}^i$, 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 $\mathbb{S}$. The dashed black vertical line is at $\mathbb{S} = 0$, 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 $\mathbb{S}$ are converted into probabilities that the state is bad using a logit function. During removal filtering, an unknown state is assigned a score $\mathbb{S}^*$ by the SVM. This score is then converted into a probability it is bad (P^*_{bad}) using the logit function (orange arrow).

match to themselves.We conduct a Needleman-Wunsch-style alignment of the states $\{\boldsymbol{x}^i\}$ with themselves, $\mathbb{A}(\{\boldsymbol{x}^i\}, \{\boldsymbol{x}^i\})$ (algorithm 12.9). In this alignment, alignment of a state i to itself $\mathbb{A}(\boldsymbol{x}^i, \boldsymbol{x}^i)$ can be thought of as establishing $\boldsymbol{x}^i$ as a unique, previously unobserved state. Conversely, alignment of a state i to a different (previous) state j, $\mathbb{A}(\boldsymbol{x}^i, \boldsymbol{x}^j), i \neq j$ 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 (figure 12.18). 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 (§12.2.4)).

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. 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), $\boldsymbol{S}_{SK}$, $\boldsymbol{S}_{SB}$, and $\boldsymbol{S}_{SH}$. These three SVMs are all trained on labeled transitions generated from the ΦX-174 data used to build the 6-mer model. In the same manner as was described above for the good/bad SVM classifier, 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, α and β) 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.

$\boldsymbol{S}_{SK}$ differentiates between steps and skips (88.7% correct on the validation set), $\boldsymbol{S}_{SB}$ differentiates between steps and backsteps (98.2% correct on the validation set), and $\boldsymbol{S}_{SH}$ 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

$$\frac{P_S^{mn}}{P_K^{mn}} = \frac{\text{logit}(\mathbb{S}_{SK}, \alpha_{SK}, \beta_{SK})}{1 - \text{logit}(\mathbb{S}_{SK}, \alpha_{SK}, \beta_{SK})}, \tag{12.46}$$

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

$$P_S^{mn} + P_B^{mn} + P_H^{mn} + P_K^{mn} = 1 \tag{12.47}$$

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

$$P_{B3} = P_B^3. \tag{12.48}$$

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

$$P_{B+} = P_B \tag{12.49}$$

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 x $(L + 3)$ 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 figure 12.19. Using this self-alignment method to identify repeated states, we conduct recombination filtering as described in algorithm 12.9.

Algorithm 12.9 Recombination filter

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

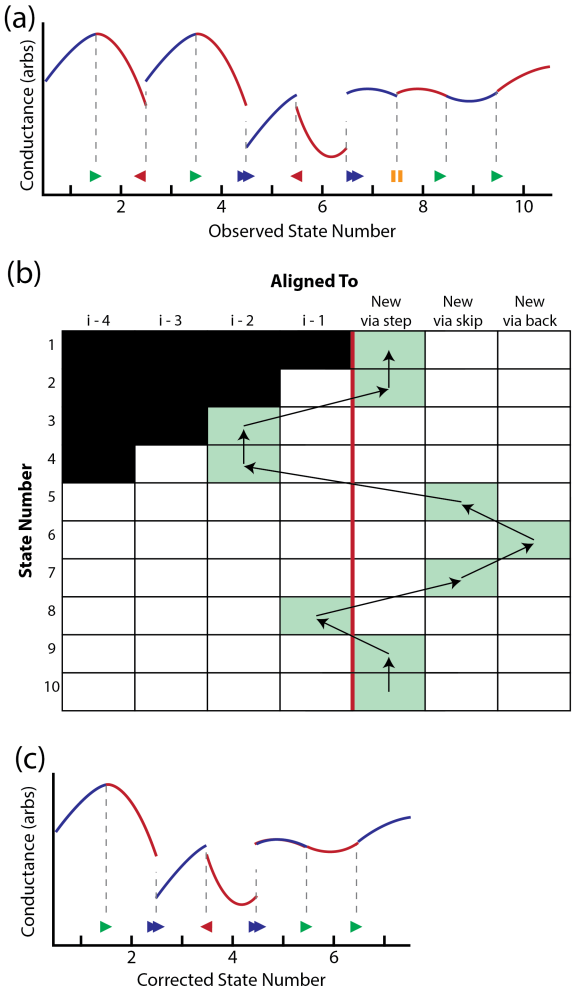

Figure 12.18: **Self-alignment procedure for recombination filter.** (a) The recombination filter seeks to find repeated instances of the same k-mer conductance state in a sequencing read. Shown here is a toy example of a sequencing read with various missteps. Toy data is modeled on a single-nucleotide-stepping 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.

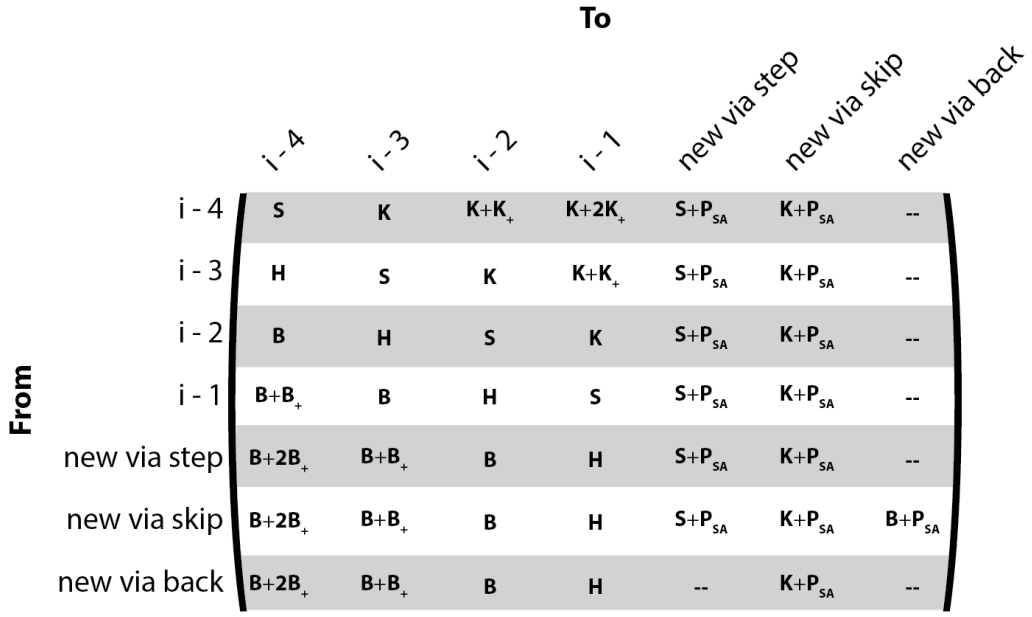

Figure 12.19: **Self-alignment transition penalties.** During self-alignment, the transition from a starting point in the alignment (rows) into a final point in the alignment matrix (columns) takes an additive penalty. Alignments of a state to a previously measured state take a penalty equal to the log 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.

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 figure 12.18c) 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 associ-

ated logit functions (as in the recombination filter) 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: $\boldsymbol{S}_{SK}$ to decide between steps and skips, and $\boldsymbol{S}_{SB}$ 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 $N-1$ x 3 matrix $\mathcal{P}$:

$$\mathcal{P} = \begin{bmatrix} P_S^1 & P_K^1 & P_B^1 \\ P_S^2 & P_K^2 & P_B^2 \\ \cdots & \cdots & \cdots \\ P_S^{N-1} & P_K^{N-1} & P_B^{N-1} \end{bmatrix} \tag{12.50}$$

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

$$\mathbb{P} = \log(\mathcal{P}) \tag{12.51}$$

Now with the step-type log-probabilities calculated, we use a dynamic programming algorithm (algorithm 12.10) 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 $\mathbb{L}$:

$$\mathbb{L} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \tag{12.52}$$

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 12.10 Reordering filter

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

12.2.10 Conductance normalization

Following capacitance compensation (12.2.8), the response to the changing voltage in the variable-voltage signal retains a nuisance component in addition to the DNA-position-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 conductance when blockaded by a charged molecule, is mostly additive with the DNA-dependent 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 DNA-position-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 $\not{g}_i(V)$ 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:

$$\not{g}_i(V) = g_i(V) - \frac{1}{N}\sum_{j=1}^{N} g_j(V) \tag{12.53}$$

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 (figure 12.20). That is, at high voltage, states with normalized conductances well above the mean tend to 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 $\not{g}_i$ a slope m_i. These slopes are then linearly fit to the voltage means of the normalized conductances,

$$\langle \not{g}_i \rangle = \frac{1}{N_V} \sum_{j=1}^{N_V} \not{g}_i(V_j), \tag{12.54}$$

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

$$\not{g}_i(V) = g_i(V) - \frac{1}{N} \sum_{j=1}^{N} g_j(V) - \alpha V \left(g_i(V) - \frac{1}{N} \sum_{j=1}^{N} g_j(V) \right) \tag{12.55}$$

12.2.11 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

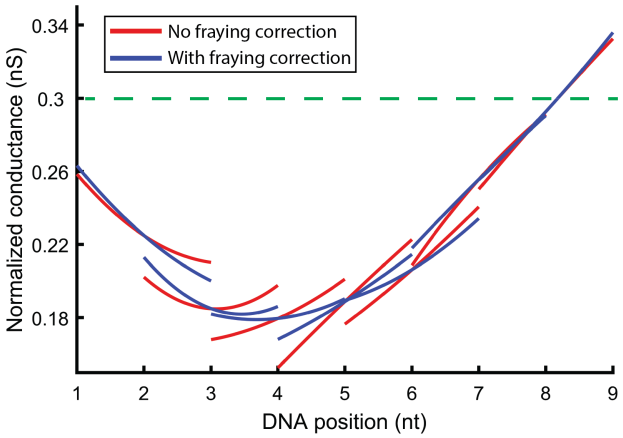

Figure 12.20: **Fraying correction.** The linear fray correction accounts for the exaggerated effects of a few bases on the conductance at high voltage. The initial mean-only normalization (red) demonstrates systematic discontinuities around peaks (not shown) and troughs (shown here) where the high-voltage 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).

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, $AAAACA$. Our full adaptation and calculation of the measured-to-model alignment is described in detail below.

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 principal component amplitudes and the associated uncertainty (for measured states) or covariance (for model states) covariance matrix. The measured state is written as $\boldsymbol{x}_n$ with uncertainty matrix $\Sigma_{\boldsymbol{x}_n}$, and the reference state is written as $\boldsymbol{y}_j$ with covariance matrix $\Sigma_{\boldsymbol{y}_j}$. The match score between these states is given by

$$S_{nj} = \frac{1}{\sqrt{(2\pi)^d \frac{|\Sigma_{\boldsymbol{x}_n}^{-1}||\Sigma_{\boldsymbol{y}_j}^{-1}|}{|\Sigma_{\boldsymbol{x}_n}^{-1}+\Sigma_{\boldsymbol{y}_j}^{-1}|}}} \exp\left[-\frac{1}{2}\left(\Sigma_{\boldsymbol{x}_n}^{-1}\boldsymbol{x}_n - \Sigma_{\boldsymbol{y}_j}^{-1}\boldsymbol{y}_j\right)^T \left(\Sigma_{\boldsymbol{x}_n}^{-1} + \Sigma_{\boldsymbol{y}_j}^{-1}\right)\left(\Sigma_{\boldsymbol{x}_n}^{-1}\boldsymbol{x}_n - \Sigma_{\boldsymbol{y}_j}^{-1}\boldsymbol{y}_j\right).\right] \tag{12.56}$$

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

$$s_{nj} = \log S_{nj} = \frac{1}{2}\Bigg[d\log 2\pi + \log|\Sigma_{\boldsymbol{x}_n}^{-1}| + \log|\Sigma_{\boldsymbol{y}_j}^{-1}| - \log|\Sigma_{\boldsymbol{x}_n}^{-1} + \Sigma_{\boldsymbol{y}_j}^{-1}| \\ - \left(\Sigma_{\boldsymbol{x}_n}^{-1}\boldsymbol{x}_n - \Sigma_{\boldsymbol{y}_j}^{-1}\boldsymbol{y}_j\right)^T \left(\Sigma_{\boldsymbol{x}_n}^{-1} + \Sigma_{\boldsymbol{y}_j}^{-1}\right) \left(\Sigma_{\boldsymbol{x}_n}^{-1}\boldsymbol{x}_n - \Sigma_{\boldsymbol{y}_j}^{-1}\boldsymbol{y}_j\right).\Bigg]. \tag{12.57}$$

Hel308 Backstep Kinetics

We use the known backstep kinetics of the Hel308 enzyme to inform our sequencing. Specifically, previous work 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)[64]. Consequently, measured states determined to have backstepped during enzyme step correction 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[64]. The overall probability that a state will backstep is also estimated from these 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

$$P_{ind|\sim b} = \frac{\frac{1}{2} - P_b * P_{ind|b}}{P_{ind|\sim b}}. \tag{12.58}$$

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:

$$P_{dep|b} = 1 - P_{ind|b} \tag{12.59}$$

$$P_{dep|\sim b} = 1 - P_{ind|\sim b} \tag{12.60}$$

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

$$S_{ij} \leftarrow S_{ij} + p_{ind|b} \text{ if } j \text{ is odd} \tag{12.61}$$

and

$$S_{ij} \leftarrow S_{ij} + p_{dep|b} \text{ if } j \text{ is even} \tag{12.62}$$

Likewise, for every measured state i where we did not observe a backstep, we update the row $S_{i:}$ as

$$S_{ij} \leftarrow S_{ij} + p_{ind|\sim b} \text{ if } j \text{ is odd} \tag{12.63}$$

and

$$S_{ij} \leftarrow S_{ij} + p_{dep|\sim b} \text{ if } j \text{ is even} \tag{12.64}$$

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

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 §12.2.9) 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 ($\boldsymbol{S}_1$ and $\boldsymbol{S}_2$), trained on labeled examples of steps and variously sized skips from the ΦX-174 data collected during the 6-mer model construction. The scores $\mathbb{S}_i$ output by the SVMs $\boldsymbol{S}_i$ are converted into probabilities using the same logit procedure as described previously in 12.2.9.

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

$$P_{mn}^{(1)} = \text{logit}(\mathbb{S}_1, \alpha_1, \beta_1), \tag{12.65}$$

with the logit function as defined previously and using logit parameters α_1 and β_1 determined from global likelihood maximization over a labeled validation set (§12.2.9).

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

$$P_{mn}^{(2|\sim 1)} = \text{logit}(\mathbb{S}_2, \alpha_2, \beta_2). \tag{12.66}$$

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

$$P_{mn}^{(2)} = P_{mn}^{(2|\sim 1)} * P_{mn}^{(\sim 1)} = \text{logit}(\mathbb{S}_2, \alpha_2, \beta_2) * \big(1 - \text{logit}(\mathbb{S}_1, \alpha_1, \beta_1)\big). \tag{12.67}$$

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

$$P_{mn}^{(k)} = P_{mn}^{(k-1)} * P_{mn}^{(+)}. \tag{12.68}$$

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

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:

$$T_{mn,ij} = P\big(\substack{\text{state } m \text{ is a measurement} \\ \text{of true map state } i} \;\big|\; \substack{\text{state } n \text{ is a measurement} \\ \text{of true map state } j}\big) \tag{12.69}$$

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,

$$\tau_{ij1} = P\big(\substack{\text{state } t \text{ is a measurement} \\ \text{of true map state } i} \;\big|\; \substack{\text{state } t+1 \text{ is a measurement}, \\ \text{of true map state } j} \;\; \text{step size} = 1\big) \tag{12.70}$$

$$= \begin{cases} 1 & i \text{ is a "pre" state and } j \text{ the corresponding "post" state} \\ 1/4 & i \text{ is a "post" state and } j \text{ a succeeding 6-mer} \\ 0 & \text{otherwise} \end{cases} \tag{12.71}$$

where we define two 6-mers as "successive" when they share 5 nucleotides shifted by one position, e.g. A<u>CGTAC</u> could be succeeded by <u>CGTAC</u>T. We then define a similar matrix for larger sizes of step, which is calculated by taking powers of the single half-step matrix:

$$\tau_{ijk} = P\big(\substack{\text{state } t \text{ is a measurement} \\ \text{of true map state } i} \;\big|\; \substack{\text{state } t+1 \text{ is a measurement}, \\ \text{of true map state } j} \;\; \text{step size} = k\big) = (\tau_{ij1})^k. \tag{12.72}$$

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:

$$T_{mn,ij} = P^{(1)}_{mn}\tau_{ij1} + \sum_{k=2}^{12} P^{(2)}_{mn}\left(P^{(+)}_{mn}\right)^{k-2}\tau_{ijk}. \tag{12.73}$$

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

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 $n-1$:

$$\mathbb{A}_{1j} = s_{1j} + \log\left(1 - P_1^{(\text{bad})}\right) \tag{12.74}$$

$$\mathbb{A}_{nj} = \log \sum_{k=1}^{8192} \sum_{m=1}^{n-1} \exp\left\{ s_{nj} + t_{mn,kj} + h_{mk} + \log\left(1 - P_n^{(\text{bad})}\right) + \sum_{l=m+1}^{n-1} \log P_l^{(\text{bad})} \right\}, \quad n > 1, \tag{12.75}$$

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, $P^{(bad)}$ 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 to assign a unique $P^{(bad)}$ 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[51]. We take two additional steps to increase speed. Firstly, using the approximation

$$\log \sum_i e^{a_i} \approx \arg\max_i a_i, \tag{12.76}$$

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:

$$\mathbb{A}_{nj} = \max_{k,m} \left\{ s_{nj} + t_{mn,kj} + h_{mk} + \log\left(1 - P_n^{(\text{bad})}\right) + \sum_{l=m+1}^{n-1} \log P_l^{(\text{bad})} \right\}, \quad n > 1. \tag{12.77}$$

We also record a traceback array,

$$\mathbb{B}_{nj} = \arg\max_{k,m} \left\{ s_{nj} + t_{mn,kj} + h_{mk} + \log\left(1 - P_n^{(\text{bad})}\right) + \sum_{l=m+1}^{n-1} \log P_l^{(\text{bad})} \right\}, \quad n > 1. \tag{12.78}$$

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[53] approximating the results of the MAP algorithm[51].

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 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.

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,

$$(n_{\text{final}},\ a_{\text{final}}) = \underset{(n,\ j)}{\arg\max}\ \mathbb{A}_{nj}, \tag{12.79}$$

$$(n_i, a_i) = \mathbb{B}_{n_{i+1}, a_{i+1}}. \tag{12.80}$$

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.

12.2.12 Hel308 step durations

The distribution of step durations for Hel308 in our sequencing experimental conditions is shown in figure 12.21. We need 3 complete voltage cycles (15 ms) to accurately estimate the covariance of the 3 principal components for a state (12.2.4), 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 (§12.2.8) increases with increasing rate of change in the voltage. If the capacitive current becomes too large, it could rail the amplifier (rail is ± 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 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.

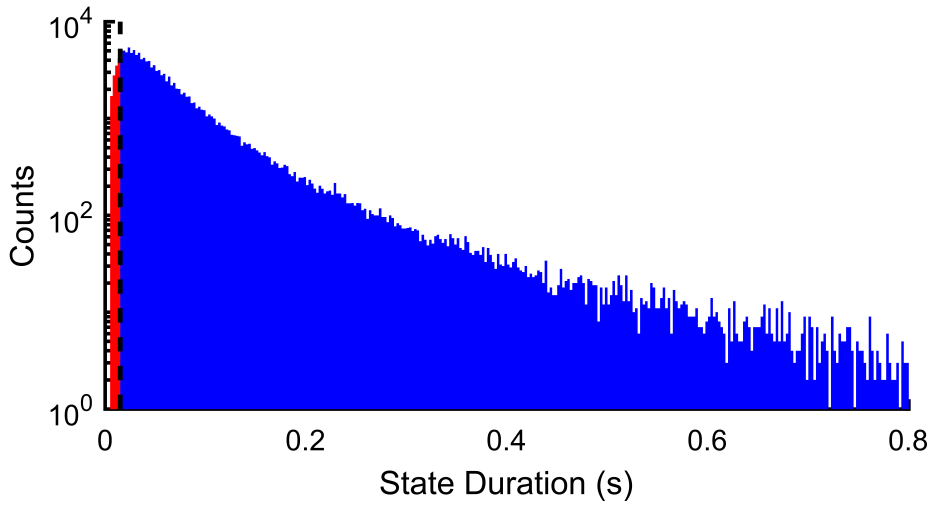

Figure 12.21: **Hel308 step durations in variable-voltage sequencing conditions.** Marked in red are states too short to accurately estimate the covariance of the 3 principal components. States of sufficient length for this estimation are marked in blue.

12.2.13 Hel308 processivity

12.3 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[64].

We observed Hel308's processivity in both the constant- and variable-voltage conditions by looking at the read lengths obtained on our ΦX-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 ΦX-174 reference, we investigated 49 constant-voltage reads and 50 variable-voltage reads of the long fragment, all starting at the same location in the genome (at the AvaII cut site). Hel308 shows little ability to unwind dsDNA 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 ΦX-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 (figure 12.22). 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 $l_p = 999 \pm 174$ nt for the constant-voltatge data and $l_p = 933 \pm 132$ nt for the variable-voltage data. The processivity in the two conditions is identical within statistical uncertainty, meaning the change from constant to variable voltage has no effect on read length.

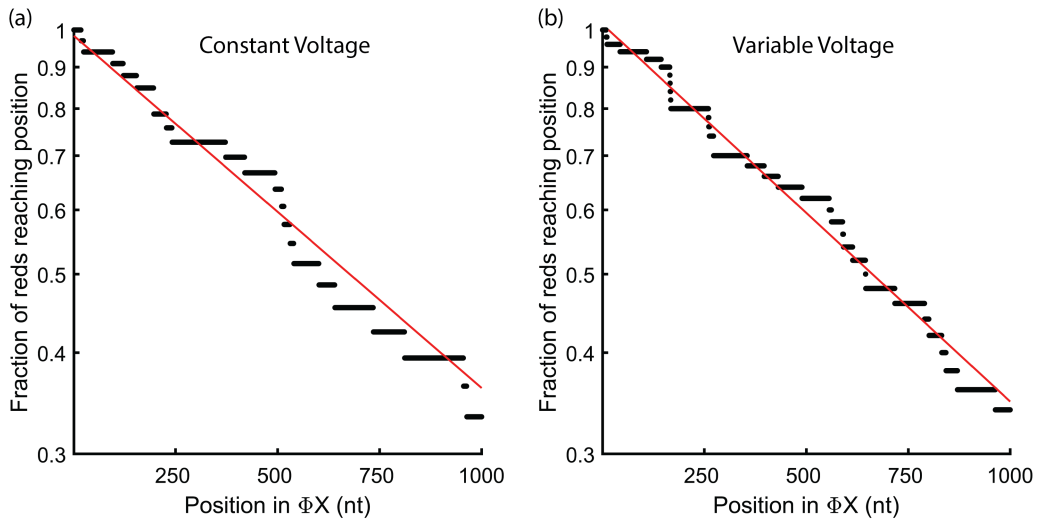

Figure 12.22: **Hel308 processivity in constant voltage (a) and variable voltage (b) sequencing conditions.** The fraction of reads (all starting from the same cut site in ΦX-174) reaching a given position in the genome is plotted as black dots on a logarithmic y-scale. We see an exponential fall-off 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 $f(l) = e^{\frac{l}{l_p}}$. The best fit for constant voltage is given by $l_p = 999$ nt, the best fit for variable voltage is given by $l_p = 933$ nt. The processivity in the two conditions is identical within statistical uncertainty.

12.3.1 Sequencing verification experiments

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 (12.3), 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 ∼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. These fragments 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 μL of 10000 $\frac{U}{mL}$ NspI and 7 μL of 5000 $\frac{U}{mL}$ Sau3aI per 17.5 μg 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 12.2)

   (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 (§12.2.3).

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, 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 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 cut-

sites. 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.

12.3.2 DNA sequences

12.4 DNA Sequences

Tables 12.1 and 12.2 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 λ phage (Promega) and ΦX-174 genomes (New England Biolabs) as well as the pET-28a vector (from collaborators).

DNA Construct Name	Sequence (5'->3')
ΦX174 Experiments	1 6 11 16 21 26 31 36 41 46 51
ΦX174 threading strand	PXAAA AAAAC CTTCC XCCTT CCCAT CATCA TCAGA TCTCA_CGCGG TGCA
ΦX174 cholesterol blocker	PCCGC GTGAG ATCTG AAAAA TTTAA ACCCA AAXZ
ΦX174 loading strand	PGACC CGCCA AGTAC AAGTA AGCCT ACGCC TACGG TTTTT TTTTT TTTTT TTTTT
ΦX174 loading blocker	CCGTA GGCGT AGGCT TACTT GTACT TGGCG G
λ-phage Experiments	1 6 11 16 21 26 31 36 41 46 51
λ threading strand	PTACT ACTAC TACTA CTACX XTTTT GAGCC TCTCA CTATC GCATT CTCAT GCAGG T
λ cholesterol blocker	PCCTG CATGA GAATG CGATA GTGAG ATCGT AGCCQ QQQZ
λ loading strand	PGGAC GTACT CTTAC GCTAT CACTC TTCGT AGCC
λ loading blocker	AGAGT GATAG CGTAA GAGTA CGTCC T
Missing 6-mer Experiments	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 XTTTT TTGGC GCTTC ATACA GCCGC GCCGG CGAGA TTTTG GCGAG ACAGG CACGC GCGAG CCCAA TCTAT TTTCA ATCTA CGTAT ACTAG GGGGT TCTAG TACTT TTTCT CACTA TCGCA TTCTC ATGCA GGTCG TAGCC
Fill-in template 2	PTACT ACTAC TACTA CTACX XTTTT CTAGT ACACT AGACT AGTCC CTACT ACGAT TTTTC TACGA TTAGG GCCCT ATCTA ATCTA GAGTT TTTCT AGAGT AGGGA CCCCG GGACT CCCTT GTATT TTTCT CACTA TCGCA TTCTC ATGCA GGTCG TAGCC
Fill-in template 3	PTACT ACTAC TACTA CTACX XTTTT CCTTG TAGAT CCTAT ACGGA CGGGG TCTCT TTTTG GTCTC TAGCG CTCGA ATGTG TCGAC ACCTT TTTGA CACCT CAGAG ACCTA GCTAG GCTAG TGTTT TTTCT CACTA TCGCA TTCTC ATGCA GGTCG TAGCC
Fill-in template 4	PTACT ACTAC TACTA CTACX XTTTT CTAGT GTACA CCTCG GACCG GTGCC CTCGA TTTTC CCTCG AGAGG ACCAT GCTAG CCCCC CGCTT TTTCC CCGCT ATACA AGTAC CCGAG TTAGA ACTTT TTTCT CACTA TCGCA TTCTC ATGCA GGTCG TAGCC
Fill-in template 5	PTACT ACTAC TACTA CTACX XTTTT TAGAA CTAGG ATAGG GTGGG GCACA TACCT TTTTC ATACC TAGGT CCGAA TCGAT CTTAG CCTAT TTTTA GCCTA AGGGT AGACG TGATT GGGCC TACTT TTTCT CACTA TCGCA TTCTC ATGCA GGTCG TAGCC
Fill-in template 6	PTACT ACTAC TACTA CTACX XTTTT CATAC GTAGC ATTTT TCATA GGCCC CATTT TTCAT CCCGC GCATT TTTCA TCCTA CGCAT TTTTT CTCAC TATCG CATTC TCATG CAGGT CGTAG CC
Fill-in cholesterol blocker	CCTGC ATGAG AATGC GATAG TGAGA QQQQZ
Legend: P = phosphate, Q = 18 carbon spacer, X = abasic site, Z = cholesterol tag	

Table 12.1: Table of short DNA sequences used for constructing the variable-voltage 6-mer model.

DNA Construct Name	Sequence (5'->3')
pET28a Sequencing Experiments	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 GCTTT 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 GCTTT TCATG
NspI cholesterol blocker	PAAAA GCAAT AATTA GTAAA GTACT GCACT TCAAT QQQQZ
NspI loading strand	PATTG AAGTG CAGTA CTTTA CTAAT TATTG CTTTT TCTGA GCC
NspI loading blocker	AAAAG CAATA ATTAG TAAAG TACTG CACTT CAATC ATG
DNA Stretching Experiments	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
Stretching read strand	PTACT ACTAC TACTA CTACX XTTTT TTCTG CAGTG CAGTT TGCTT TGCGA AATGA AACTG CAGTT TGCGA AATTG CAGTT TGCGA AAGTT TGCTG CACGA AATTT TGCTG CAGGA AATTT TTTTC TCACT ATCGC ATTCT CATGC AGGTC CGAGC C
Stretching cholesterol blocker	CCTGC ATGAG AATGC GATAG TGAGA QQQQZ
Legend: P = phosphate, Q = 18 carbon spacer, X = abasic site, Z = cholesterol tag	

Table 12.2: Table of short DNA sequences used for measuring DNA stretching and for validating variable-voltage sequencing performance.

12.4.1 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 table12.3.

Experiment Description	Enzyme control mechanism	DNA Sequence	Number of Pores	Number of Events
6-mer model building, SVM training	Hel308	ΦX174	19	155
6-mer model building	Hel308	λ 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	97
Constant-voltage sequencing	Hel308	pET28a	21	31
DNA stretching	Hel308	Stretching strand	2	18

Table 12.3: 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.

Part V

AFTERWORD

Nanopores, in their broad biotechnology and fundamental science applications, are possessed of a rich and yet understandable physics. Treating these devices as physical systems that can be fundamentally understood rather than as black boxes, the nanopore lab at the University of Washington has improved the functionality and predictability of nanopores, and found fertile ground for their incorporation into new inventions and methods.

Interpreting the noisy and abstract signal that is the output of the nanopore device is challenging, and in these liminal situations where information is present yet obscured, the precise nature of the question a scientist chooses to ask of the data cascades down and affects every result that scientist reports.

My hope is that the way we have found it productive to think about and ultimately answer these questions has been laid plain by this thesis, and that these ideas may be taken up by the larger nanopore community. I also hope that this is not the final word on the subject, and that our field's conception of the physics of these systems continues to evolve.

BIBLIOGRAPHY

[1] B. Alberts. *Molecular Biology of the Cell*. W. W. Norton and Company, sixth edition, Nov 2014.

[2] Molecular models of dna. Website: https://en.wikipedia.org/wiki/Molecular_models_of_DNA, Feb 2019.

[3] J. D. Watson. A structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, April 1953.

[4] T. H. Bestor. Notes on the role of dynamic dna methylation in mammalian development. *PNAS*, 112(22):6796–6799, Jun 2015.

[5] J. R. Edwards. Dna methylation and dna methyltransferases. *Epigenetics and Chromatin*, 10(23), May 2017.

[6] R. Holliday. Dna modification mechanisms and gene activity during development. *Science*, 187(4173):226–232, Jan 1975.

[7] J. Shendure. Dna sequencing at 40: past, present and future. *Nature*, 550:345–353, Oct 2017.

[8] J. Shendure. Next-generation dna sequencing. *Nat Biotech*, 26:1135–1145, Oct 2008.

[9] S. Goodwin. Coming of age: ten years of next-generation sequencing technologies. *Nat Reviews*, 17:333–351, May 2016.

[10] National Human Genome Research Institute. The cost of sequencing a human genome. Website, Jul 2016.

[11] S. Brenner, editor. *Encyclopedia of Genetics*. Elsevier Science Inc., 2001.

[12] A. P. J. de Joning. Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genet*, 7(12), Dec 2011.

[13] E. Meaburn. Next generation sequencing in epigenetics: insights and challenges. *Semin Cell Dev Biol*, 23(2):192–199, Apr 2012.

[14] J. J. Kasianowicz. Analytical applications for pore-forming proteins. *Biochimica et Biophysica Acta*, 1858:593–606, Oct 2016.

[15] D. Branton. The potential and challenges of nanopore sequencing. *Nat Biotech*, 26:1146–1153, Oct 2008.

[16] E. V. Koovin. *Sequence - Evolution - Function: Computational Approaches in Comparative Genomics*. Kluwer Academic, 2003.

[17] C. H. Reynolds, editor. *Drug Design: Structure- and Ligand-Based Approaches*. Cambridge University Press, 1 edition, 2010.

[18] R. A. Robinson. *Electrolyte Solutions*. Dover, 2012.

[19] G. F. Schneider. Dna translocation through graphene nanopores. *Nano Letters*, 10(8):3163–3167, July 2010.

[20] N. A. W. Bell. Dna origami nanopores. *Nano Lett*, 12(1):512–517, Dec 2012.

[21] S. Hernández-Ainsa. Dna origami nanopores: developments, challenges and perspectives. *Nanoscale*, 6:14121–14132, 2014.

[22] F. Haque. Solid-state and biological nanopore for real-time sensing of single chemical and sequencing of dna. *Nano Today*, 8(1):56–74, Feb 2014.

[23] L. Liu. Fabrication of nanopores with ultrashort single-walled carbon nanotubes inserted in a lipid bilayer. *Nat Protocols*, 10:1670–1678, Oct 2015.

[24] A. H. Laszlo. Decoding long nanopore sequencing reads of natural dna. *Nat Biotech*, 32(8):829–33, Jun 2014.

[25] E. N. Ervin. Creating a single sensing zone within an alpha-hemolysin pore via site directed mutagenesis. *Bionanoscience*, 4(1):78–84, Mar 2014.

[26] D. Stoddart. Single-nucleotide discrimination in immobilized dna oligonucleotides with a biological nanopore. *PNAS*, 106(19):7702–7707, May 2009.

[27] G. M. Cherf. Automated forward and reverse ratcheting of dna in a nanopore at five angstrom precision. *Nat Biotech*, 30(4):344–348, Apr 2012.

[28] C. Heinz. The core of the tetrameric mycobacterial proin mspa is an extremely stable beta-sheet domain. *J Biol Chem*, 278(10):8678–8685, Mar 2003.

[29] J. Karow. Stratos genomics working on raising funds to move sequencing-by-expansion technology to market. Website, Feb 2014.

[30] I. M. Derrington. Nanopore dna sequencing with mspa. *PNAS*, 107(37):16060–16065, Sep 2010.

[31] E. A. Manrao. Reading dna at single-nucleotide resolution with a mutant mspa nanopore and phi29 dna polymerase. *Nat Biotech*, 30(4):349–353, Mar 2012.

[32] D. R. Garalde. Highly parallel direct rna sequencing on an array of nanopores. *Nat Methods*, 15:201–206, Jan 2018.

[33] A. H. Laszlo. Detection and mapping of 5-methylcytosine and 5-hydroxymethylcytosine with nanopore mspa. *PNAS*, 110(47):18904–18909, Nov 2013.

[34] J. Schreiber. Error rates for nanopore discrimination among cytosine, methylcytosine, and hydroxymethylcytosine along individual dna strands. *PNAS*, 110(47):18910–18915, Nov 2013.

[35] A. C. Rand. Mappoing dna methylation with high throughput nanopore sequencing. *Nat Methods*, 14(4):411–413, Nov 2017.

[36] J. M. Craig. Direct detection of unnatural dna nucleotides dnam and d5sics using the mspa nanopore. *PloS One*, Nov 2015.

[37] L. Restrepo-Pérez. Paving the way to single-molecule protein sequencing. *Nat Nanotechnol*, 13(9):786–796, Sep 2018.

[38] D. A. Malyshev. A semi-synthetic organism with an expanded genetic alphabet. *Nature*, 509(7500):385, 2014.

[39] A. H. Laszlo. Mspa nanopore as a single-molecule tool: From sequencing to sprnt. *Methods*, 105:75–89, Aug 2016.

[40] S. Bhattacharya. Water mediates recognition of dna sequence via ionic current blockade in a biological nanopore. *ACS Nano*, 10(4):4644–4651, Apr 2016.

[41] E. W. Frey. Understanding the physics of dna using nanoscale single-molecule manipulation. *Front Phys*, 7(5):576–581, Oct 2012.

[42] T. Li. Brownian motion at short time scales. *Ann. Phys. (Berlin)*, 525(4):281–295, Jan 2013.

[43] S. B. Smith. Direct mechanical measurements of the elasticity of single dna molecules by using magnetic beads. *Science*, 158(5085):1122–1126, Nov 1992.

[44] W. S. Chen. Direct observation of multiple pathways of single-stranded dna stretching. *Phys Rev Lett*, 105(21), Nov 2010.

[45] P. J. Flory. Spatial configuration of macromolecular chains. Nobel prize lecture, Dec 1974.

[46] A. Fiasconaro. Exact analytical solution of the extensible freely jointed chain model. ArXiv paper., May 2018.

[47] C. Ortiz. A review of elasticity models for extension of single polymer chains. Course notes, MIT 3.052.

[48] I. M. Derrington. Subangstrom single-molecule measurements of motor proteins using a nanopore. *Nat Biotech*, 33(10):1073–1075, Oct 2015.

[49] D. B. McIntosh. Sequence-dependent elasticity and electrostatics of single-stranded dna: signatures of base-stacking. *Biophysical Journal*, 3(4):659–666, Feb 2014.

[50] J. E. Hall. Access resistance of a small circular pore. *J Gen Physiol*, 66(4):531–532, Oct 1975.

[51] L. R. Bahl. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans Inform Theory*, 20(2):284–287, Mar 1974.

[52] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, Mar 1973.

[53] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans Inform Theory*, IT(13):260–269, Apr 1967.

[54] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[55] P. A. Wiggins. An information-based approach to change-point analysis with applications to biophysics and cell biology. *Biophysical Journal*, 109(2):346–354, Jul 2015.

[56] C. H. LaMont. The development of an information criterion for change-point analysis. *Neural Computation*, 28(3):594–612, Mar 2016.

[57] C. Lee. Multiple sequence alignment using partial order graphs. *Bioinformatics*, 16(3):452–464, Mar 2002.

[58] M. T. Noakes. Increasing the accuracy of nanopore dna sequencing using a time-varying cross membrane voltage. *Nat Biotech*, 2019.

[59] E. A. Manrao. Nucleotide discrimination with dna immobilized in the mspa nanopore. *PloS One*, 6, 2011.

[60] M. Jain. Minion analysis and reference consortium: Phase 2 data release and analysis of r9.0 chemistry. *F1000Research*, 6:760, 2017.

[61] F. J. Rang. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome Biol*, 19, 2018.

[62] N. J. Loman. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat Methods*, 12:733–735, 2015.

[63] S. Bhattacharya. Molecular dynamics study of mspa arginine mutants predicts slow dna translocations and ion current blockades indicative of dna sequence. *ACS Nano*, 6:6960–6968, 2012.

[64] J. M. Craig. Revealing dynamics of helicase translocation on single-stranded dna using high-resolution nanopore tweezers. *PNAS*, 114(45):11932–11937, Oct 2017.

[65] T. Z. Butler. Single-molecule dna detection with an engineered mspa protein nanopore. *PNAS*, 105(52):20647–20652, Dec 2008.

[66] R. Krishnakumar. Systematic and stochastic influences on the performance of the minion nanopore sequencer across a range of nucleotide bias. *Sci Rep*, 8(3159), 2018.

[67] E. Rames. Evaluation of minion nanopore sequencing for rapid enterovirus genotyping. *Virus Res*, 252:8–12, 2018.

[68] V. Boža. Deepnano: Deep recurrent neural networks for base calling in minion nanopore reads. *PloS One*, 12, 2017.

[69] H. Teng. Chiron: Translating nanopore raw signal directly into nucleotide sequence using deep learning. bioRxiv, 2018.

[70] M. David. Nanocall: an open source basecaller for oxford nanopore sequencing data. *Bioinformatics*, 33:49–55, 2017.

[71] R. Vaser. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res*, 2017.

[72] J. Simpson. Signal-level algorithms for minion data. contribute to jts/nanopolish development by creating an account on github.

[73] A. Bosco. Elastic properties and secondary structure formation of single-stranded dna at monovalent and divalent salt conditions. *Nucleic Acids Res*, 42(3):2064–2074, Feb 2014.

[74] N. G. de Bruijn. A combinatorial problem. *Koninklijke Nederlandse Akademie V. Wetenschappen*, 49:758–764, 1946.