UNIVERSITY OF CALIFORNIA

SANTA CRUZ

EXPERIMENTS AND STATISTICAL ANALYSIS WITH DNA IN BIOLOGICAL AND SOLID-STATE NANOPORES

A thesis submitted in partial satisfaction
of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

RAJ MAITRA

June 2013

The Thesis of RAJ MAITRA
is approved:

Professor William B. Dunbar, Chair

Professor Hongyun Wang

Professor Gabriel Elkaim

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures **vi**

List of Tables **xi**

Abstract **xii**

Dedication **xiii**

Acknowledgments **xiv**

1 Introduction **1**

1.1 Biological Nanopore 1

1.2 Protein Complexes on the Nanopore 2

1.3 Solid-state Nanopores 4

2 Validation and Residual Analysis of a Nanopore State Space Model **7**

2.1 Introduction 7

2.2 Methods 9

2.2.1 Experimental Conditions 9

2.2.2 Model for the Nanopore System 9

2.2.3 Model Fitting and Validation 9

2.3 Results/Discussion 12

2.3.1 Model Fitting 12

2.3.2 Correlation of Residuals 12

2.4 Conclusion 14

2.5 Acknowledgements 15

3 Characterizing the Precision of a Novel Amplifier with Captured DNA Polymers **16**

3.1 Introduction 16

3.2 Methods 18

3.3 Results/Discussion 19

3.4 Conclusion 21
3.5 Acknowledgements 21

4 Preliminary Modeling of DNA Motion through a Dual Solid State Pores **24**

4.1 Introduction 24
4.2 Methods 26
4.2.1 Model 26
4.2.2 Numerical Methods 29
4.3 Results/Discussion 31
4.4 Conclusion 32
4.5 Acknowledgements 32

5 Preliminary Experiments with Solid-State Nanopores and Nucleosome Species **34**

5.1 Introduction 34
5.2 Methods 35
5.2.1 Holders 35
5.2.2 DNA Species 36
5.2.3 Setup 38
5.2.4 Experiment 38
5.3 Discussion/Results 40
5.4 Conclusion 41
5.5 Acknowledgements 42

6 Conclusion **45**

References **47**

Appendix **55**

6.1 Holder Protocols 55
6.1.1 Holder #1, UCSB 55
6.1.2 Holder #2, Stanford 55

6.1.3 Holder #3, UCSC 56
6.2 Miscellaneous Protocols 56
6.2.1 Protocol - RecA bound to λDNA 56
6.2.2 Protocol - Biopore Preliminary Protocol 58
6.2.3 Protocol - Biopore Experiment 59
6.2.4 Protocol - Polyacrylamide Gel Setup 60
6.3 DNA Translocation Model Simulation Code 61

List of Figures

1 Insertion of an α-hemolysin channel into a cell membrane. (i) shows zero ionic current through the cell membrane (0 pA) and an AHL diffusing toward the cell membrane until (ii) the pore inserts itself and the current rises to what is called open channel current. Capture of single-stranded DNA into an AHL pore (iii) is possible, however, full passage through the pore is hindered by double-stranded DNA (iv). Forcing the double-strand through will eventually break the hydrogen bonds bewteen base-pairs resulting in two single strands. . 1

2 Cross-sectional illustration of AHL and MspA biological nanopores. AHL has a limiting aperture of 1.5 nm that only allows passage of ssDNA. However this region of sensitivity is longer in length than MspA. MspA's tapering structure focuses sensitivity at the *trans* end. The bottom focal point is ~ 0.5 nm. Ionic current is affect mostly by 3-4 nucleotides within this space. 3

3 Circuit diagram of the nanopore system, including a simplified amplifier circuit. In (i), the parameters are: voltage V_p on the output of the headstage; stray capacitance C_p between ground and the electrodes, electrode wires and electrolyte; electrolyte access resistance R_a; transmembrane voltage V_m; membrane capacitance C_m; and, channel resistance $R_c = 1/G_c$. In (ii), the parameters are: feedback resistance to the amplifier, R_f; the command voltage, V_{CMD}; and the output current I_{out}. 8

4 Example of transients from real data (dotted) and grey-box (grey), black-box(black) models due to a voltage step. Each model is fifth-order and the data is 5kHz. The grey-box model had a fit of of 95.71% while the black-box model had a t of 97.43%. 13

5 Autocorrelation of the residuals. The residual autocorrelation coefficients(from lag 0 to 25) for the five-state greybox model are shown in (a), while (b) shows the autocorrelation coefficients for GB-5 after adding the noise matrix K into the model. After accounting for process noise with K, the autocorrelation coefficients for lagged datasets reduces significantly. 15

6 Time-lapsed capturing of DNA polymer bound to streptavidin. In step (i), a α-hemolysin channel is diffusing throughout the *cis*-chamber above the lipid bilayer. At this state there is no passage of potassium and chloride ions through the membrane, thus no ionic current. Between 1 and 2 seconds, the channel inserts (ii) and at a command voltage of 120 mV and 1 M KCl, the resistance of the channel is 1 GΩ and the current is 120 pA. After adding the DNA+streptavidin complex to the *cis* chamber (iii), a complex will enter the pore. After holding for 2.5 seconds (iv), the voltage is reversed (v) and the complex is kicked out to return to an openchannel (vi). The mean of the event (iv) is taken and that is one entity in a population of these events. 19

7 Polymer constructs used to map conductance sensitivity for Axopatch 200B and Nanoclamp. (a) shows the sequences of the all-DNA homopolymer and 1ab(7-10). They are four different constructs with a 5' biotinylated tail in which the protein streptavidin binds to. The **X** denotes the position of the abasic residue. (b) provides the structure of an abasic residue, which is the absence of the nucleobase from the deoxy-ribose suger along the phosphate-backbone. (c) provides the nucleotide position map of streptavidin-bound polymer captured within an AHL channel (shown as a cross-sectional cutout). 20

8 Axopatch current sensitivity map. (a) shows raw mean ionic current for each event captured on a single AHL channel. Single-abasic homopolymers (1ab(7-9)), were added one at a time with a line delineating each addition. Here, we can see a difference ($\sim$ 1.4 pA) in current level between a standard DNA and a 1ab(7) template although the Nanoclamp didnt show the difference between these two templates. This is because the Axopatch 200B has smaller input noise than the CMOS CMS noise (= 4.07 pA RMS). (b) shows a the constructed map from taking the mean of each single population. The positions contribute most to pore resistance are positions 8 and 9. This result agrees with our measurements using the Nanoclamp. Here, we separately measured a current level for position 10 after perfusing the previous experiment shown. 22

9 Nanoclamp current sensitivity map. (a) shows the raw mean ionic current for each event captured on an AHL channel. Unlike the Axopatch 200B, the Nanoclamp was unable to see any current changes in 1ab(7) with respect to the all-DNA template. Polymers 1ab(9-10) showed the most significant current changes from the all-DNA template. (b) is the constructed map of current change with respect to position in an AHL pore. The map follows the same trend as the Axopatch 200B's map. (c) Histograms of the event populations. Each population distribution overlaps due to the small current changes and wide variance. This is obviously due to less resolution from the Nanoclamp. 23

10 Illustration of dual-pore setup. Two solid-state nanopores are aligned one on top of the other. Three electrodes are used, the third being a common ground bewteen the other two. Voltages could possibly trap a strand of DNA in both pores(as seen). A slight voltage differential after capture could possibly cause a translocation rate slow enough for sequencing. 25

11 The x and y coordinates are used to calculate length, L, bewteen the points, as well as the angle θ between points; from these variables, the x and y-components to the spring force can be calculated. 29

12 Example simulation of DNA translocation. Command voltage was set to 1000 mV since the time step was very low, dt = $10^{-5}\mu$s. The length of the strand was 1000 base-pairs. Frames are shown 100 time steps at a time to show the general motion of the strand through the pore. Initial and end conformations are shown in the left panel. . . . 33

13 Holder #1 is a 2-part teflon-based holder. Each part fit together like a lego connection; the top piece has an extruding cube shape with a pin hole that fits into a cubed well in bottom piece, aligning the top pin hole to a bottom pin hole. Two polydimethylsiloxane (PDMS) layers act as the gaskets, sandwiching the silicon membrane and providing liquid tight seals. 35

14 Holder #2 is made of teflon and comes in two parts. There are two holes on the sides of each piece to allow two bolts to thread through and clamp the two pieces together. In doing this, the PDMS gaskets sandwich the SiN membrane. The wells have very small holes to the membrane allowing for a low volume of buffer in each well. 36

15 Holder #3 has two parts. There are two metal prongs on one piece that align the two pieces. A large bolt screws the two pieces together, sandwiching the SiN membrane between two gaskets. 37

16 Three types of molecular complexes used in solid-state nanopore experiments. i) illustrates double-stranded λDNA which is 2.2 nm in width and roughly 48,000 base pairs. ii) shows RecA proteins forming complexes with λDNA. The protein wraps itself around the phosphate-backbone nonspecifically [1], as seen in the illustration. The RecA+DNA complex is 10-11 nm in width. Nucleosomes iii) are roughly the same width (10-11 nm). The species of molecule we used on the experiments had an array of 17 nucleosomes (multiple histones bound to DNA) on a 3 Kbp strand of DNA. 38

17 Transmission electron microscope images of the pores drilled by the Schmidt Lab (a-c) and the Garajan lab (d). Images are not as definite (a-c) in order to reduce dimensional changes caused by the microscope. 39

18 IV-curve analysis for pores 1-3. The experimentally determined resistances are smaller than the expected value for the pores if they were cylindrical. So the proposed dimensions of the drilled pores (defined by the TEM images) may have changed during imaging, or the inner structure of the pores are smaller that the opening diameter. 42

19 IV-curve analysis for pore # 4. The experimentally determined resistance is a lot smaller than the expected value for this pore as well, however, with this pore, translocations were detected. 43

20 Varying event behavior of nucleosomes on pore #1. There is obvious interaction between the sample of nucleosomes with the solid-state nanopore. From these results, we cannot be certain whether these are complete translocations through the pore. It is possible that the complex is getting stuck in the pore and popping out but this needs to be verified and tested. 44

List of Tables

1 Results of 4-state and 5-state grey-box models, and black-box models, fit to α-hemolysin nanopore. 13

2 Results of PEM fitting† of 6-state model (**??**) to Axon PATCH-1U data in whole-cell mode. 14

3 Results of PEM fitting of 4-state (**??**) and 5-state (**??**) models to Axon PATCH-1U in patch mode. 14

4 Parameters for each equation are listed as well as their value used in simulation. Dimensions in the simulation were standardized to nm, mV and μs. 30

5 Parameters for each equation are listed as well as their value used in simulation. Dimensions in the simulation were standardized to nm, mV and μs. 32

6 We have various pores that have been drilled from 3 different labs. The two providers in this table are presented in the paper. 37

7 Measured resistances on each solid-state nanopore versus the theoretical resistances. 41

EXPERIMENTS AND STATISTICAL ANALYSIS WITH DNA IN BIOLOGICAL AND SOLID-STATE NANOPORES

Raj Maitra

Abstract

Nanopores are a powerful tool for probing the structure of single molecules, such as a strand of DNA. Acquiring as much information from the output signal generated by a nanopore is essential and ultimately dependent on several factors: offline estimation of pore parameters must be based on a good model; the amplifier that produces the ionic current signal from the pore must be low noise; nanopore experiments must be designed intelligently to properly test hypotheses; and—most importantly—the pore must be the right size. This thesis discusses these topics in four chapters. Chapter 1 discusses the design and validation of a nanopore circuit model used for conductance estimation. Chapter 2 presents research results characterizing noise and signal resolution of a novel patch-clamp amplifier and its implications on next-generation nanopore platforms. Chapter 3 discusses preliminary research on modeling the motion of double-stranded DNA for future experiments with dual solid-state nanopores. Finally, chapter 4, is preliminary research performed with the solid-state nanopore and discussing various pore dimensions. Ultimately, the reader will be presented with research that covers the biochemical and computational ends of nanopore technology and how they are used to obtain as much information as possible from a single current output.

Dedication

This is dedicated to the miniscule parts of this universe that make up something bigger.

Acknowledgements

I would like to acknowledge my principal investigator, Bill Dunbar, who I have been grateful to work with these past 3 years. To Chris O'Donnell, Shea Ellerson, Daniel Geralde, Darrel Deo and Asma Uz-Zaman for lab support throughout these past years. Don Wiberg for introducing me to Bill. Robin Abu-Shumays, for introducing me to the wet-lab and guiding me through biological nanopore experiments. To the 201/217 regulars (John Vivian, Max Cherf, Andrew Smith, Doug Marks, Jeff Nivala, Joey Dahl, Jacob Schreiber and Zak Wescoe), thanks for keeping me sane. Finally, I would like to acknowledge my Minnesota clan. They'll be in my DNA until it mutates.

1 Introduction

A nanopore is a nanometer sized hole in a membrane that separates two compartments (*cis* and *trans*) of ionic solution. There are two classes of nanopores: biological and solid-state. A biological nanopore is a protein pore that inserts into a lipid bilayer, also known as a synthetic cell membrane[14]. Electrodes submerged in each compartment supply a command voltage V_{CMD} across the bilayer. Upon insertion of the nanopore into the lipid bilayer, a patch-clamp amplifier generates ionic current through the protein pore. Figure (1) illustrates channel insertion with a standard α-hemolysin (AHL) protein pore.

1.1 Biological Nanopore

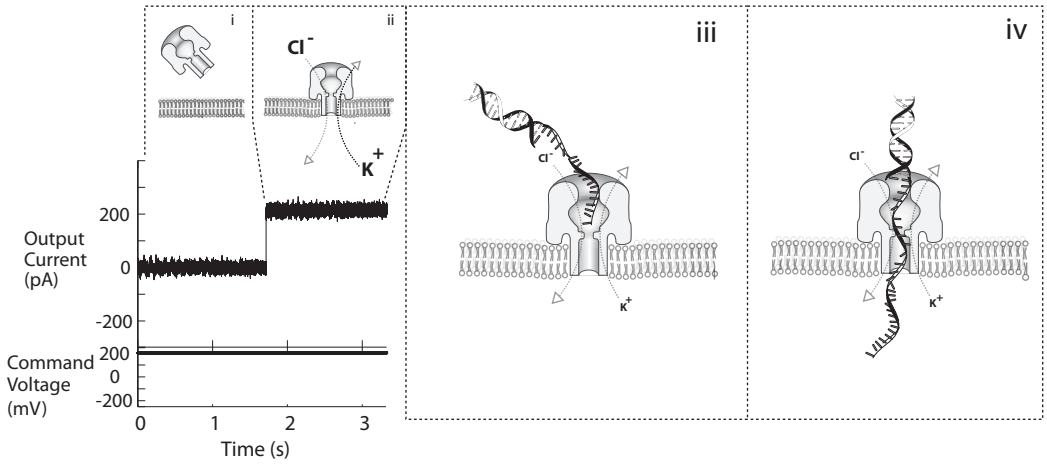


Figure 1: Insertion of an α-hemolysin channel into a cell membrane. (i) shows zero ionic current through the cell membrane (0 pA) and an AHL diffusing toward the cell membrane until (ii) the pore inserts itself and the current rises to what is called open channel current. Capture of single-stranded DNA into an AHL pore (iii) is possible, however, full passage through the pore is hindered by double-stranded DNA (iv). Forcing the double-strand through will eventually break the hydrogen bonds bewteen base-pairs resulting in two single strands.

There are many types of biological nanopores. The most common is α-hemolysin. It is a protein native to the bacterium Staphylococcus aureus and is a primary virulence factor in S. aureas pneumonia [46]. AHL will spontaneously insert into a foreign cell's lipid bilayer, disrupting the cell's electrochemical gradient. The

protein channel's transmembrane β-barrel provides the mechanistic capability to puncture a lipid bilayer [50]. AHL's structure is shown in Figure (2). AHL's barrel is approximately 4 nm in length with a 2.2 nm inner diameter, which necks at the top to the limiting aperture, the most narrow portion of the pore with a diameter of approximately 1.5 nm. Proximal to the limiting aperture, on the *cis*-side, is a vestibule (2.6 nm maximum diameter). The limiting aperture and barrel (stem) are just wide enough to accommodate ssDNA, while the vestibule can accommodate duplex DNA.

Although AHL is the standard protein used for nanopore experimentation, it is unable to detect sequence of a translocating DNA strand[25, 44, 45]. To overcome this problem researchers have engineered AHL to improve its sensitivity for DNA analysis and sequencing applications [27, 12, 5, 51]. Oxford Nanopore Technologies (ONT) and researchers at the University of Oxford (led by Hagan Bayley) engineered AHL by covalently attaching a molecular adapter in the lower stem that binds free nucleotide[12]. The mutant pore is capable of discriminating the four nucleoside 5-monophosphate molecules with high accuracy.

A group from the University of Washington (led by Jens Gundlach) have engineered an alternative pore, Mycobacterium smegmatis porin A (MspA). It has a funnel-like shape, focusing sensitivity of the measured current to the bottom (*trans*-side) of the channel, Figure (2)[16, 9]. The limiting aperture has a diameter of 1.2 nm with a height of $\sim$ 0.5 nm. Current through MspA is weighted on fewer nucleotides near this region ($\sim$ 3-4) [40] versus the standard AHL pore which bears great weight on $\sim$ 7-9 [32]. Presently in academia, MspA is the most promising biological pore for sequencing.

1.2 Protein Complexes on the Nanopore

Capturing enzymes (catalytic proteins) when bound to a DNA strand has allowed researchers to monitor enzyme kinetics as well as slow the speed at which DNA translocates through the pore [10]. Examples of enzymes that have been examined on AHL include the bacteriophage T7 DNA polymerase, the Klenow fragment of E.

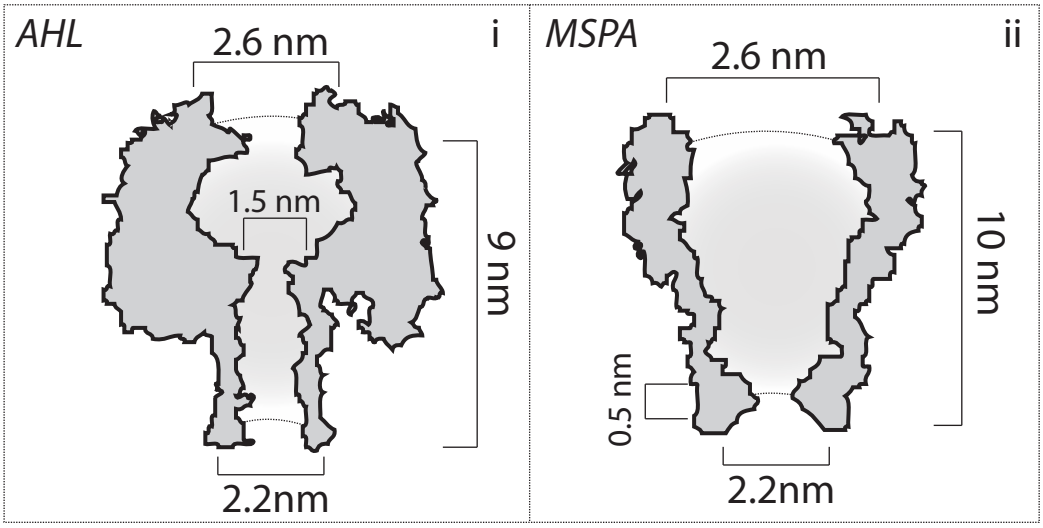


Figure 2: Cross-sectional illustration of AHL and MspA biological nanopores. AHL has a limiting aperture of 1.5 nm that only allows passage of ssDNA. However this region of sensitivity is longer in length than MspA. MspA's tapering structure focuses sensitivity at the *trans* end. The bottom focal point is $\sim$ 0.5 nm. Ionic current is affect mostly by 3-4 nucleotides within this space.

coli [42, 23, 11, 13], Φ29 [35] DNA polymerases. In order to sequence DNA with an enzyme that ratchets DNA through nanopore one nucleotide at a time, there are several requirements that an enzyme must meet: first, the complex must be stable enough to withstand the constant voltage force during each catalytic cycle; second, the force generated by the enzyme to translocate along DNA must be large enough to overcome the competing voltage force on the DNA that holds the complex on the nanopore; third, the command voltage must be high enough to generate enough signal-to-noise (S/N) in the measured current to accurately monitor DNA stepwise translocation; lastly, the motor must process the DNA strand at a slow enough rate, such that transitions between base-specific current levels are visible. A group from University of California, Santa Cruz (led by Mark Akeson) achieved these requirements with the bacteriophage Φ29 DNA polymerase. They designed a blocking oligo that hinders procession of Φ29 along a DNA strand while in bulk phase[10]. Once the complex is captured, the blocking oligo is stripped off and Φ29 begins to process the strand at 400 ms/nt (2.5 nt/s), a rate that is suitable for sequencing[10]. The Φ29 mechanism was coupled with MspA, by Akeson and Gundlach, and they have demonstrated that it is possible to produce sequence dependent

current traces[10, 39]. Still, there are challenges to optimize this experiment: first, significant computational effort is necessary to deconvolute a signal that corresponds to a combination of 3-4 nucleotides; second, distinguishing the number of bases in a homopolymer region is nondeterministic due to the exponential dwell time; third, the Φ29 motor is imperfect and can randomly backup along a strand as well as jump forward.

In the commercial sector, Oxford Nanopore Technologies has engineered several devices that claim single-nucleotide reading capabilities with a biological nanopore and protein motor. The company introduced new nanopore-based sequencing devices: the GridION and MinION. Both claim high-throughput DNA sequencing with significantly longer reads than modern sequencing platforms. The GridION platform has an array of 2000 individual nanopores at present. On each nanopore the claimed read length is 5100 kb at a rate of 150 Mb per hour for up to 6 h. With 20 GridIONs working in parallel, the company claims that the human genome could be sequenced in 15 min [18]. The MinION is a smaller USB powered sequencer with 512 nanopores of similar specifications. Although their platforms use biological nanopores, the company has invested interest in solid-state-based nanopore sequencing through the solid-state nanopore research at Harvard University.

1.3 Solid-state Nanopores

Solid-state nanopores are nanosized holes formed by drilling or etching the opening in a solid-state substrate[15] and in comparison to biological pores, solid-state pores are chemically and mechanically more stable [55, 57, 56, 58, 29, 48]. Many are silicon based, such as silicon oxide ($Si0_2$) or silicon nitride (SiN), but recently, researchers from University of Illinois, Urbana-Champaign (led by Rashid Bashir) have employed aluminum oxide (Al_2O_3) with graphene for solid-state DNA analysis [57]. Al_2O_3 has improved mechanical and electrical properties compared to silicon based substrates[57]. However, many researchers are turning toward graphene as a substrate for nanopore platforms. Its single-atomic layer thickness is ideal for sequencing purposes.

Solid-state nanopores can vary in size depending on the fabrication technique. In 2001, Gene Golovchecko et al (Harvard) demonstrated nanometer precision pores drilled in silicon substrates using ion beam milling[33]. Multiple methods for drilling nanopores are now routine: chemical etching, ion beam sculpting, e-beam drilling, atomic layer deposition and feedback-controlled sputtering[15]. At present, pore diameters can range from tens of nanometers to as small as 1 nm. The thickness of the substrate can vary from 5 nm to $>$ 10 nm. Although the opening of the pore can be measured by electron microscopy, the internal dimensions of the pore are unknown; thus, two pores that have approximately the same size opening, can behave differently when conducting ionic current[49].

As previously mentioned, there is great interest in graphene as a potential solid-state sequencing platform. A single sheet of graphene is only one atom thick (0.34 nm), which is comparable to the height of a nucleotide. It has been proposed that due to graphene's thin membrane it should be possible to distinguish nucleotides. Alternatively, graphene has potential to supply tunneling current across a nanopore. In numerical simulations, it has been showed that nucleobase detection is achievable by transverse conductance shifts due to an ssDNA strand passing through a graphene nanogap [43]. Practically, graphene has been shown to exhibit higher frequency noise than silicon based pores[48, 41]. Other researchers have attempted to insulate graphene membranes with other materials (e.g. TiO_2 [41]) but there is still more work to be done in order to realize the potential of graphene nanopores.

Two seperate methods have been employed as a means to detect nucleotides passing through a pore: embedded electrodes (within a pore) and chemically functionalizing the inside of a nanopore [8, 34]. At Osaka University, tunnneling currents across a solid state pore have been shown to reduce the translocation rate of DNA [53, 54]. At Arizona State University, Stuart Lindsay and colleagues have functionalized silicon surfaces with a detection molecule (a benzamide-based molecule:4(5)-substituted 1-H-imidazole-2-carboxamide) that hydrogen-bonds to each DNA base in different orientations [34]. Functionalized transverse electrodes within a nanopore will interact with the passing bases and cause a tunneling current unique to the base. The

detection S/N is considerably higher than appears to be possible with ionic-current based sensing through biological nanopores. This method leads the field in detecting bases with a solid-state nanopore.

Translocation rate control methods have not yet been shown with solid-state nanopores. Coupling Φ29 with a solid-state pore would be ideal and has been tested [17], but it is not clear whether enzyme functionality is maintained on an imprecise surface. Hybrid nanopores (biological pores caught within solid-state pores)[26] have potential to bridge embedded functionality in a solid-state pore (electrodes, chemical modifications) and the Φ29 mechanism for controlled DNA translocation, but this has not been explored. Alternatively, our lab has designed a dual pore (explained in more detail in Chapter 3) which aligns two pores in series, commanded by two seperate voltage sources. Simultaneouse capture of DNA in both pores could allow active voltage control of DNA motion, possibly slowing the translocation rate through one pore or even holding the strand at rest in both pores.

Overall, the nanopore shows great promise as a next-generation sequencing platform and it is likely that nanopores will be be ubiquitous throughout biotechnology.

2 Validation and Residual Analysis of a Nanopore State Space Model

2.1 Introduction

The natural behavior of the nanopore system is circuit-like[24]. In the nanopore system a lipid bilayer or cell membrane separates two electrochemical solutions impeding any movement of ions between solutions. In essence, it acts as a charged capacitor with infinite resistance[2, 7, 59]. Applying a polarity-changing voltage across the membrane will force positive and negative ions to charge and discharge on the lipid membrane like a capacitor. Also, the capacitance of the lipid membrane is proportional to the surface area of the membrane itself. Inserting a protein channel, such as α-hemolysin, into a lipid membrane, allows ions to pass through the membrane via nanopore. Voltage applied across the membrane will drive ions through the pore at a rate proportional to the resistance of the pore.

A pore inserted into a lipid membrane can be modeled as a resistor (protein channel, R_c) and capacitor (lipid membrane, C_m) in parallel. When larger charged molecules flow through the protein channel, the impedence of the channel increases briefly due to the volume of the larger molecule and its chemical interaction with the pore[47, 2, 7, 59].

In addition to R_c and C_m, there is access resistance R_a at the electrodes. This is due to the redox reaction between electrons in the wire and potassium and chloride ions in aqueous solution. There is also parasitic capacitance C_p between ground and the electrodes. Lastly, the whole circuit model requires the axopatch amplifier modeled as an op-amp with resistive feedback. Figure (3) illustrates the full circuit model with respect to the nanopore system.

Due to the capacitive elements in the nanopore system, (lipid bilayer, parasitic capacititance) voltage changes with an amplifier cause transient responses in current, ultimately masking the ionic current through the nanopore. If oscillatory voltage control were used to move a DNA strand up and down through a pore (like rereading a strand), conductance of the pore ($G_c = R_c^{-1}$) would have to be estimated due to

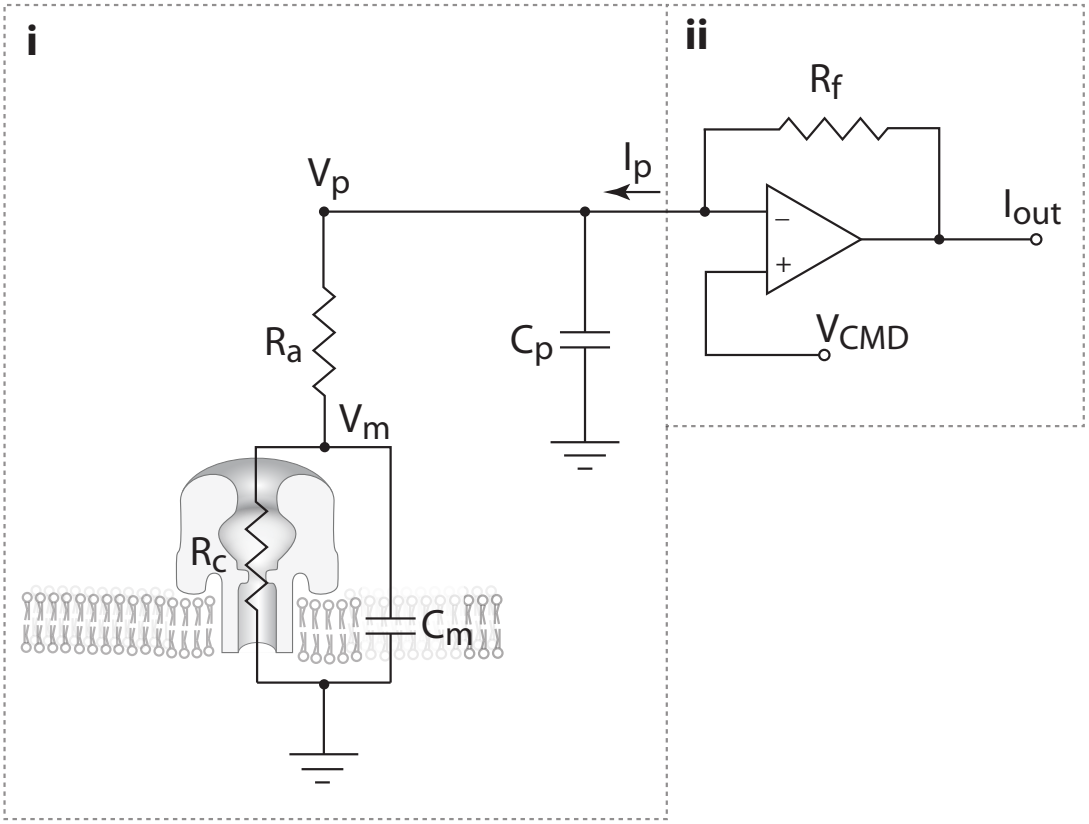


Figure 3: Circuit diagram of the nanopore system, including a simplified amplifier circuit. In (i), the parameters are: voltage V_p on the output of the headstage; stray capacitance C_p between ground and the electrodes, electrode wires and electrolyte; electrolyte access resistance R_a; transmembrane voltage V_m; membrane capacitance C_m; and, channel resistance $R_c = 1/G_c$. In (ii), the parameters are: feedback resistance to the amplifier, R_f; the command voltage, V_{CMD}; and the output current I_{out}.

capacitance excitation. We used this circuit to design three state space models for conductance estimation. Each model was simulated and compared to real nanopore data for system identification. Overall, each model accurately captured the nanopore system dynamics and subsequently they were used in an extended kalman filter for conductance estimation.

2.2 Methods

2.2.1 Experimental Conditions

A single α-hemolysin nanopore inserted in a lipid bilayer was used to acquire transient current traces for system identification. Data was amplified and filtered at 1 and 5 kHz bandwidth with an Axopatch 200B patch-clamp amplifier. The data was recorded at 100-250 kHz sampling frequency with a Digidata 1440A analog-to-digital converter. Transient measurements were taken at 23^o C in 0.3 M KCl, 15 mM K-HEPES, 1 mM EDTA and pH 8 $\pm$ 0.05.

2.2.2 Model for the Nanopore System

The proposed models of the nanopore system are based on the electrical circuit representation presented in figure 3. From this circuit, three models have been derived by Dr. Daniel R. Geralde and Dr. Gang Wang from the Dunbar Laboratory. The first, is a six-state model in the following state-space form:

$$\begin{aligned} \dot{\mathrm{x}} &= \mathrm{Ax} + \mathrm{Bu} \\ \mathrm{y} &= \mathrm{Cx} \end{aligned}$$

Where x_1 is equal to the current $\mathrm{I_p}$ from the amplifier. x_2 is the derivative of $\mathrm{I_p}$ or x_1. This is the same for all subsequent states,

2.2.3 Model Fitting and Validation

We used the System Identification Toolbox in MATLAB to evauluate how well the four-state, five-state and six-state models compared to real nanopore data and a circuit which is mathematically equivalent, the Axon PATCH-1U. The state-space models were defined as grey-box models with the previously defined structure and six unknown parameters ($\mathrm{C_p}$, $\mathrm{C_m}$, $\mathrm{G_a}$, $\mathrm{G_c}$ for six-state, $\mathrm{C_\Sigma}$, $\mathrm{G_c}$ for four and five) that were estimated using the prediction-error-minimization method (PEM) in MATLAB to optimally fit the measured data. PEM was a preliminary method used for parameter estimation. We used this method to gauge how close we were getting to

the actual values of our parameters, however, after we had confidence in our model, we designed an extended kalman filter (EKF) to estimate our parameters. Using an EKF versus PEM for parameter estimation is more appropriate and mathematically optimal. For the purpose of system identification we used PEM to get a general sense of how well parameter estimation will be before applying an EKF.

To compare to each grey-box model, we generated arbitrary black-box models of 3^{rd} to 9^{th} order in MATLAB. We tested the fitness of the three grey-box models and used the black-box models of orders 3-9 to determine whether models of different order could produce comparable or better fits.

Each model was fit to 21 different step-response outputs from either α-hemolysin or the Axon PATCH-1U model circuit in 'patch' (for state models four and five) or 'whole-cell' mode (six-state model). Step-responses were taken at both 1 and 5 kHz bandwidth. 5 mV voltage steps were used so as to keep the output response within the linear range of the Axopatch 200B amplifier and not allow saturation. This allowed for accurate parameter estimation with PEM, for both conductance and capacitive parameters. Since the structure of α-hemolysin is asymmetric (its *cis* opening of 25Å and *trans* opening of 15Å), the pore acts as a partial rectifier and conductance of the pore is dependent on the sign of the voltage input. With a salt concentration of 0.3 M KCl, the theoretical value of G_c is ~ 0.333 nS for positive voltages and ~ 0.2 nS for the negative. For this project, we estimated G_c for both negative and positive inputs, however, only estimates with the positive inputs are presented in the results section. The access conductance, G_a, is constant and taken to be 100 nS. Theoretical capacitance values for C_p and C_m are 2-3 pF and 3-5 pF respectively and combining these two parameters we have C_Σ 6-7 pF.

The function call *pem(data, grey_model)* returns the grey-box model with estimated values for the stipulated parameters in the grey_model iddata object. The function call *pem(data, i, 'nk', 1)* returns a black box model of order 'i' and delay 'nk' = 1. The resultant models were then simulated and compared to real data with the MATLAB function *compare(data, pem_model)*, returning a fit percentage. Model and parameter fits were done on a seperate set of data than the data we validated

with.

Percentage fit was determined by a normalized root-mean-square (RMS) in MATLAB. The fit function returns a percentage and is defined as the following:

$$\text{fit} = 100\left(1 - \frac{\text{norm}(\hat{\text{y}} - \text{y})}{\text{norm}\left(\text{y} - \text{mean}(\text{y})\right)}\right)\%$$

where $\hat{\text{y}}$ is the sampled model output and y is the sampled measurements. If there is a fit of 100% then the modeled output fits the measured exactly. There is possibility for negative fit percentages.

In addition to calculating the fit percentage of a model to real data, we analyzed the autocorrelation of the residuals as a function of lag. That is, the correlation coefficients were calculated with the residual vector $\text{r} = \text{y} - \hat{\text{y}}$ and the residual vector lagged from 0 to N times:

$$\text{R}(\text{r}_\text{t}, \text{r}_{\text{t}+\delta}) = \sum_{\delta=0}^{\text{N}} \frac{\text{E}[(\text{r}_\text{t})(\text{r}_{\text{t}+\delta})]}{\sigma_\text{t}\sigma_{\text{t}+\delta}} \tag{1}$$

where R is the correlation coefficient, r_t is the residual vector,

To compensate for colored noise found in residual autocorrelations, for each model we estimated a noise distribution matrix K. The noise distribution matrix K allows us to model the noise in our system as white noise. More specifically, K defines how that noise is introduced into each state. We call this type of noise, process noise. Modeling it can possibly give us a better overall model for parameter estimation. The final state space form with K looks like,

$$\dot{\text{x}} = \text{Ax} + \text{Bu} + \text{K}\omega$$

where our state-space form has the added noise distribution matrix K that defines how a white-noise variable ω is introduced into each state.

2.3 Results/Discussion

2.3.1 Model Fitting

Resulting fits to real nanopore data with grey-box models four and five are presented in table (1) and to Axon PATCH-1U circuit in 'patch' mode in table (3). Fitting results for the six-state model with the Axon PATCH-1U in 'whole-cell' mode are presented in table (2).

For 5kHz BW data from the nanopore system, both the four-state and five-state models fit $\sim 93 - 95\%$ or better. At 1kHz BW, both models fit the data at higher percentages ($\sim 96\%$), which may be due to less noise present at lower bandwidth. A 5th order blackbox model showed the highest fit for nanopore data at 5kHz BW with an average fit of $96.49\% \pm 0.13$. At 1kHz BW, the 11th order blackbox model had the highest average fit of $97.31\% \pm 0.45$. The average precentage fit decreased for blackbox models beyond 5th order with 5 kHz data and beyond 11th order with 1 kHz data. The grey-box models gave comparable fit percentages to the blackbox fits. To illustrate the closeness in output response simulated versus actual output response, figure (4) overlays the transient step-responses of fifth order greybox/blackbox models to 5 kHz data.

Fit results for the six-state model varied more than the four-state and five-state models. At 5 kHz BW, the six-state model had an average fit of $96.07 \pm 0.21\%$. At 1 kHz, the fit percentage dropped dramatically to 80.87 ± 0.19. Estimates for six-state model parameters were comparable to the true values.

2.3.2 Correlation of Residuals

The autocorrelation was calculated with a lag from 0 to 25. The plot in figure (5A) shows that the autocorrelation between lagged datasets is significant, suggesting that there is colored noise in the nanopore system or there are unmodeled dynamics. Adding the estimated noise matrix, K, to the four-state and five-state model—depicted in figure(5B)—reduced lagged autocorrelations near zero, suggesting, also, that there is colored noise in the system and it can be described with K. It is also important to point out that adding a noise matrix, K, slightly reduced the fit

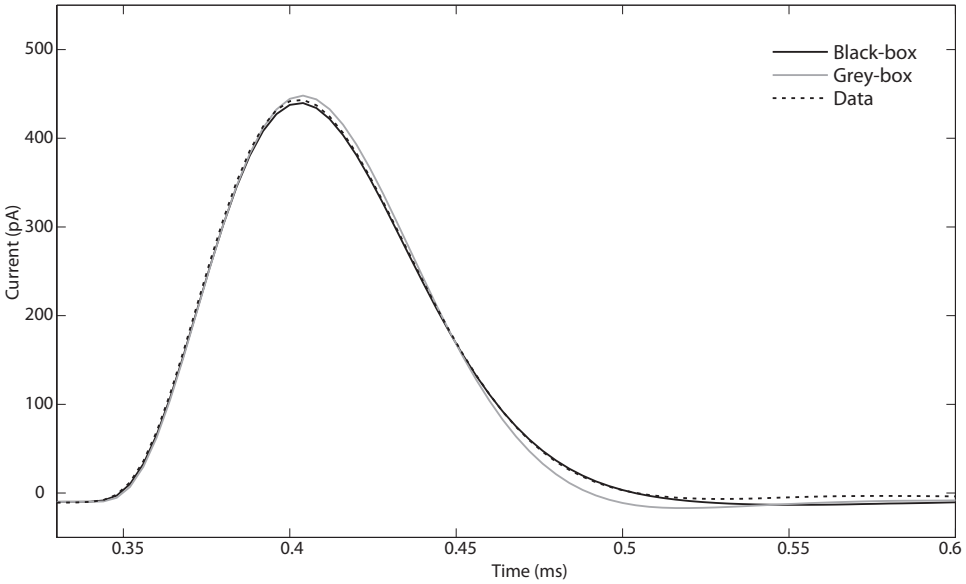


Figure 4: Example of transients from real data (dotted) and grey-box (grey), black-box(black) models due to a voltage step. Each model is fifth-order and the data is 5kHz. The grey-box model had a fit of of 95.71% while the black-box model had a t of 97.43%.

Table 1: Results of 4-state and 5-state grey-box models, and black-box models, fit to α-hemolysin nanopore.

Model Type-Order[*]	$\mathbf{C_\Sigma}$ **± s.d.**[†] **(pF)**	$\mathbf{G_c}$ **± s.d.**[†] **(nS)**	**Model Fit %**[†]	**Bessel** f_c **(kHz)**
GB-4[‡]	6.39± 0.02	0.333± 0.001	92.81± 0.32	5
GB-4	6.23± 0.19	0.334±0.004	91.94±1.05	5
GB-5[‡]	6.41±0.02	0.332±0.001	94.98±0.15	5
GB-5	6.19±0.08	0.335±0.002	93.85±0.65	5
BB-5	—	—	96.49 ± 0.13	5
GB-4[‡]	6.64±0.03	0.331±0.002	96.03±0.33	1
GB-4	6.73±0.028	0.329±0.002	95.78±0.33	1
GB-5[‡]	6.64±0.03	0.331±0.002	96.10±0.30	1
GB-5	6.78±0.09	0.328±0.002	95.49±0.57	1
BB-11	—	—	97.31±0.45	1
True values[§]	6-7	0.333		

[*] GB - grey box, BB - black box, and number is the state-space model order.

[†] Using 21 separate step-repsonse data sets, the mean and standard deviations were computed from 21 estimates that were fit by the PEM method (parameters), and 21 fitness calculations using (1).

[‡] The noise model was not estimated in these results.

[§] The "true values" $G_c \sim 0.333$ nS and $C_\Sigma \sim$ 6-7 pF (a range) are discussed in the main text.

percentages.

Table 2: Results of PEM fitting† of 6-state model (**??**) to Axon PATCH-1U data in whole-cell mode.

$\mathbf{C}_p \pm$ **s.d.*** **(pF)**	$\mathbf{G}_c \pm$ **s.d.*** **(nS)**	$\mathbf{C}_m \pm$ **s.d.*** **(pF)**	$\mathbf{G}_a \pm$ **s.d.*** **(nS)**	**Model Fit %***	**Bessel** f_c **(kHz)**
5.24±0.05	1.89±0.01	33.94±0.25	89.67±0.63	96.07±0.21	5
4.092±0.004	1.9968±0.0005	33.10±0.01	99.25±0.04	80.87±0.19	1
4-6	2	33	100	(true values)	

* Using 21 separate step-repsonse data sets, the mean and standard deviations were computed from 21 estimates that were fit by the PEM method (parameters), and 21 fitness calculations using (1).
† The noise model was not estimated in these results.

Table 3: Results of PEM fitting of 4-state (**??**) and 5-state (**??**) models to Axon PATCH-1U in patch mode.

Model Type-Order	$\mathbf{C}_\Sigma \pm$ **s.d.*** **(pF)**	$\mathbf{G}_c \pm$ **s.d.*** **(nS)**	**Model Fit %**	**Bessel** f_c **(kHz)**
GB-4†	6.05±0.02	0.100±0.001	94.88±0.03	5
GB-4	6.00±0.02	0.101±0.001	94.80±0.03	5
GB-5†	6.07±0.02	0.100±0.001	95.69±0.35	5
GB-5	5.95±0.36	0.098±0.014	93.94±4.84	5
BB-5	—	—	91.92±0.19	5
GB-4†	6.18±0.03	0.100±0.002	96.89±0.55	1
GB-4	6.26±0.03	0.100±0.002	96.66±0.55	1
GB-5†	6.05±0.21	0.100±0.002	95.21±2.29	1
GB-5	6.24±0.06	0.100±0.002	96.38±0.45	1
BB-12	—	—	96.15±0.34	1
True values‡	5	0.1		

* Mean and standard deviations were computed from 21 estimates that were fit by the PEM method, using 21 separate step-repsonse measurements.
† The noise model was not estimated in these results.
‡ The values reported for the elements in the patch-mode circuit.

2.4 Conclusion

Overall, the four-state and five-state model are good models for estimating the parameters G_c and C_Σ for the nanopore system. We can accurately estimate these parameters with real nanopore data. Subsequent work has applied these models in an extended kalman filter and, under voltage-varying conditions, conductance-changes can be monitored in real time.

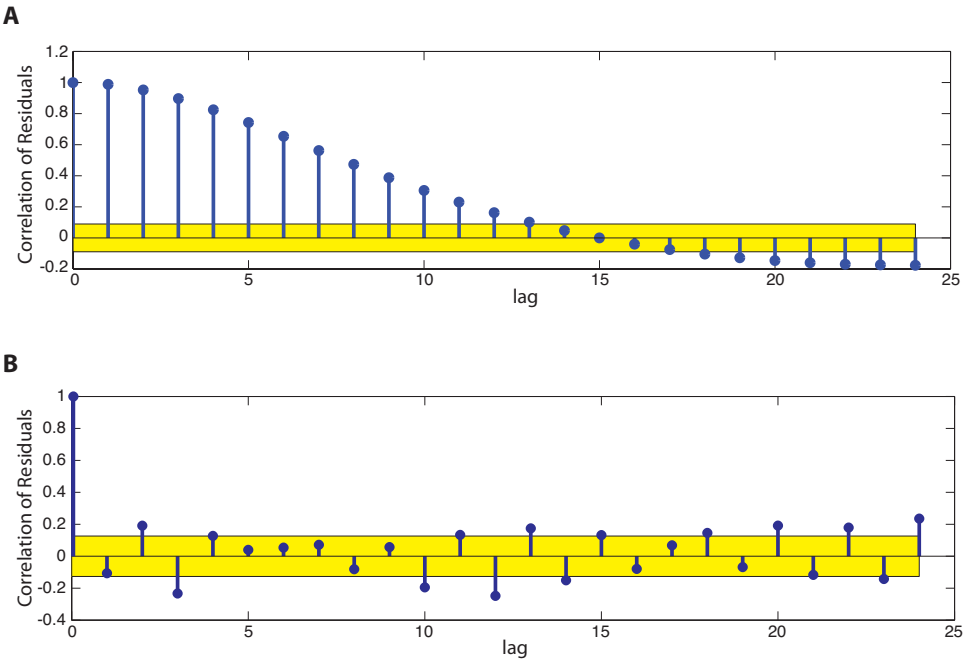


Figure 5: Autocorrelation of the residuals. The residual autocorrelation coefficients(from lag 0 to 25) for the five-state greybox model are shown in (a), while (b) shows the autocorrelation coefficients for GB-5 after adding the noise matrix K into the model. After accounting for process noise with K, the autocorrelation coefficients for lagged datasets reduces significantly.

2.5 Acknowledgements

I would like to acknowledge Daniel Geralde, Gang Wang and William Dunbar for deriving the circuit model. Daniel Geralde also performed the biological nanopore experiments that provided real data for the system identification process. Chris O'Donnell designed the extended kalman filters from the state-space models. All fit percentages and tables were created and coded by me with the help and guidance of Daniel Geralde. Also, autocorrelations and noise analysis were done by me.

3 Characterizing the Precision of a Novel Amplifier with Captured DNA Polymers

3.1 Introduction

The following chapter describes excerpts from work published in *Sensors and Actuators B: Chemical*, in which I was 2nd author [32].

In the previous chapter, the importance of estimating G_c conductance under varying voltage conditions is discussed. However, estimating parameters in our state equations is highly dependent on low-noise, high-resolution signal amplification, without which, estimation is no longer a possibility. Modern sequencing platforms that use nanopore technology involve integrated CMOS amplifiers with low-noise, low-power and high resolution to recover nucleotide sequence from translocating DNA. As nanopores are applied to new on-chip platforms, it has become more important to engineer small-scale amplifiers that can resolve the nanoscale signals, especially for downstream estimation.

In this chapter we present experiments done to characterize the noise and resolution of a novel patch-clamp amplifier (the Nanoclamp) fabricated in a 0.35 μm CMOS process. To compare how our Nanoclamp fairs against the industrial standard, Axopatch 200B, we designed an experiment that produced ionic current changes comparable to a DNA sequencing process.

Detecing individual nucleotides on a strand of DNA involves three things: a mechanism to slow the DNA as it translocates through the pore, a highly discriminitive pore structure, and an amplifier precise enough to distinguish several picoampere changes[38]. The first two have been realized recently by coupling a DNA strand to a Φ29 polymerase in an intelligent manner that only allows procession of the protein along the polymer after the complex is captured on the nanopore[36, 10]. After capture on the pore, the Φ29 polymerase ratchets the DNA strand through the pore at 40 nucleotides per second, slow enough for modern data-acquisition sampling frequencies to capture all current changes. The last component to DNA sequencing on the nanopore, is a highly precise amplifier. The standard Axopatch 200B

offers great resolution—for the experiment previously described—however, in order to apply nanopore assays practically, they need to be scalable and portable.

A low-noise, low-power, high-resolution amplifier offers the versatility needed for modern micro-electro-mechanical (MEM) platforms to be mass-marketed and practical for user implementation[3, 30, 6, 37, 31]. After the initial construction of our lab's first amplifier (the Nanoclamp), we aimed to quantitatively characterize its ability to discern pA changes comparable to those seen in nanopore sequencing methods[10, 40].

Since engineered pores used for sequencing methods are proprietary, our experiment tested each amplifier's ability to distinguish a single nucleotide modification at different positions within a standard α-hemolysin pore. α-hemolysin (AHL) is asymetric along the length of the pore. It has an opening of 25Å on the *cis* end, pinches into its limiting aperture of 15Å near its midpoint before tapering outward to the *trans* opening of 22Å. The limiting aperture allows the passing of only single-stranded DNA (ssDNA) which has a width of 10 Å versus double-stranded DNA (dsDNA) which has double the width. The ability to detect single-nucleotide modifications at different positions in AHL is much more difficult than in an engineered pore like MspA. Engineered pores are designed in such a way that only a minimal number of nucleotides (2-3) of a polymer captured in the pore affect the conductance, versus AHL, where 8-10 nucleotides affect conductance almost equally. Since AHL can hold 20 nucleotides within the pore at a time, we used 20 different 40mer homopolymers (all cytosine) with a single abasic residue (nucleotide without the nucleobase, illustrated in Figure (7)) at positions varying from 1-20 in the pore, Figure (7). Each polymer had a biotin marker on the 5' end of the strand allowing for a streptavidin protein to bind and act as a stopper when the polymer goes into the pore, resulting in the single abasic residue held in its position.

Results from the experiment were used to define a map of AHL conductance-sensitivity with respect to location within the pore as well as to characterize of the Nanoclamp's signal resolution.

3.2 Methods

Experiments were performed on a single α-hemolysin channel empedded in a lipid bilayer. Each experiment was done in 1 M KCl buffer with 10 mM HEPES, 1 mM EDTA, a pH of 8.00 ± 0.05 and at a temperature of 23^{o}C. The 40-mer 5'-biotinylated oligonucleotides were mixed in a 1:1 ratio with Streptavidin (SA) to produce SA/DNA complexes at a final concentration of 1 μM in the cis chamber.

We examined 15 different 40-mer ssDNA oligonucleotides on the Axopatch 200B and from those fifteen, we examined the five strands that affected the pore conductance the most on the Nanoclamp. Each 40-mer is a homopolymer (all cytosine), with a single-abasic (1',2'-dideoxy) residue at a prescribed position (one template is all DNA). The 5' end of each ssDNA is biotinylated. Each oligonucleotide is mixed in a one-to-one ratio with Streptavidin—a protein secreted by *Streptomyces Avidinii* bacteria—to form a 5' biotin-streptavidin linkage. We show the sequence of 5 different homopolymers used on the Nanoclamp in Figure (7). Polymer+streptavidin complexes were forced into the pore with a VCMD of 120 mV, held for 2.5 seconds and subsequently ejected at -50 mV; we illustrate this mechanism in Figure (6). The logic was executed by a finite-state machine on an FPGA. Capture by the 3' end resulted in an immobilized complex, which was used to examine the impeded current amplitude for each for the five oligonucleotides.

The sensitivity of each amplifier was characterized by mapping the delta-current amplitude of a particular abasic strand (1ab(x), where x is position 1-15) and its difference from an all-DNA template. For example, the polymer, 1ab(8), denotes that the abasic residue is located at position eight on the strand and in AHL. Both the all-DNA+streptavidin template and the 1ab(8)+streptavidin were placed into the *cis*-chamber of the experiment. After hundreds of captures, two populations can bee seen by taking the mean of each event and plotting them. The baseline current is taken to be the lower current that pertains to the all-DNA template. Delta-current was determined by the following:

$$\Delta \mathrm{I}_{\mathrm{1ab(x)}} = \mathrm{mean}(\mathrm{I}_{\mathrm{1ab(x)}}) - \mathrm{mean}(\mathrm{I}_{\mathrm{allDNA}}) \tag{2}$$

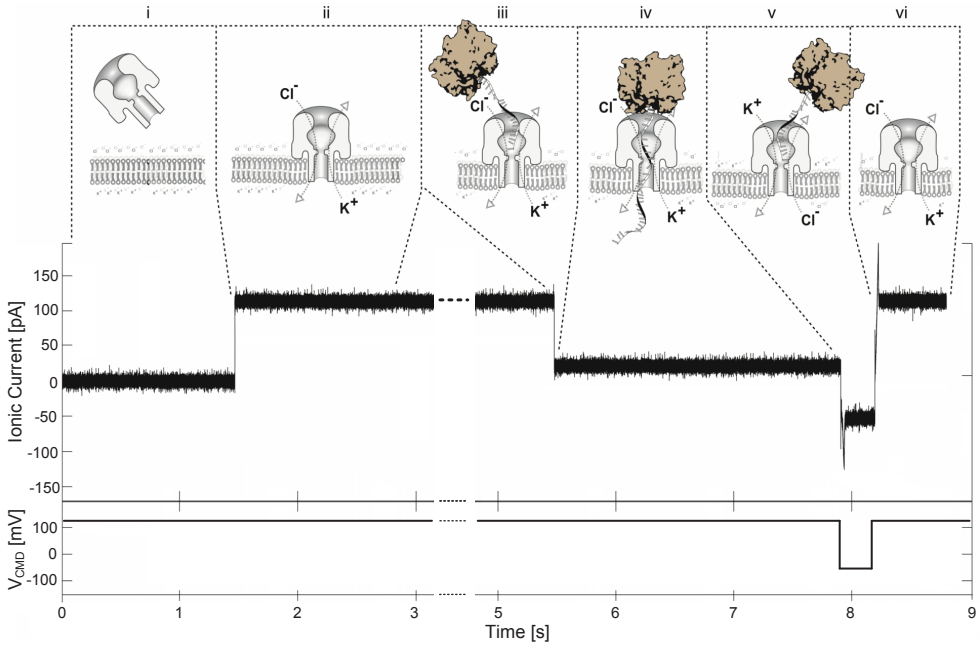


Figure 6: Time-lapsed capturing of DNA polymer bound to streptavidin. In step (i), a α-hemolysin channel is diffusing throughout the *cis*-chamber above the lipid bilayer. At this state there is no passage of potassium and chloride ions through the membrane, thus no ionic current. Between 1 and 2 seconds, the channel inserts (ii) and at a command voltage of 120 mV and 1 M KCl, the resistance of the channel is 1 GΩ and the current is 120 pA. After adding the DNA+streptavidin complex to the *cis* chamber (iii), a complex will enter the pore. After holding for 2.5 seconds (iv), the voltage is reversed (v) and the complex is kicked out to return to an openchannel (vi). The mean of the event (iv) is taken and that is one entity in a population of these events.

3.3 Results/Discussion

Current sensitive maps for each amplifier were determined and they are presented in Figures (8-9). As shown in previous studies [62], abasic residues in the wider vestibule do not measurably affect the current. With the Axopatch 200B, we measured that there is no statistically significant current difference when templates 1ab(1-6) are compared to all DNA. Templates of 1ab(x) with x = 11-15 have the abasic residue positioned in the β-barrel area and generate a relatively uniform amplitude that is slightly higher than that of the all DNA template (Figure (9)). The β-barrel space has higher sensitivity than the vestibule area. However, it does not have enough sensitivity to discern single-abasic residue displacements within the im-

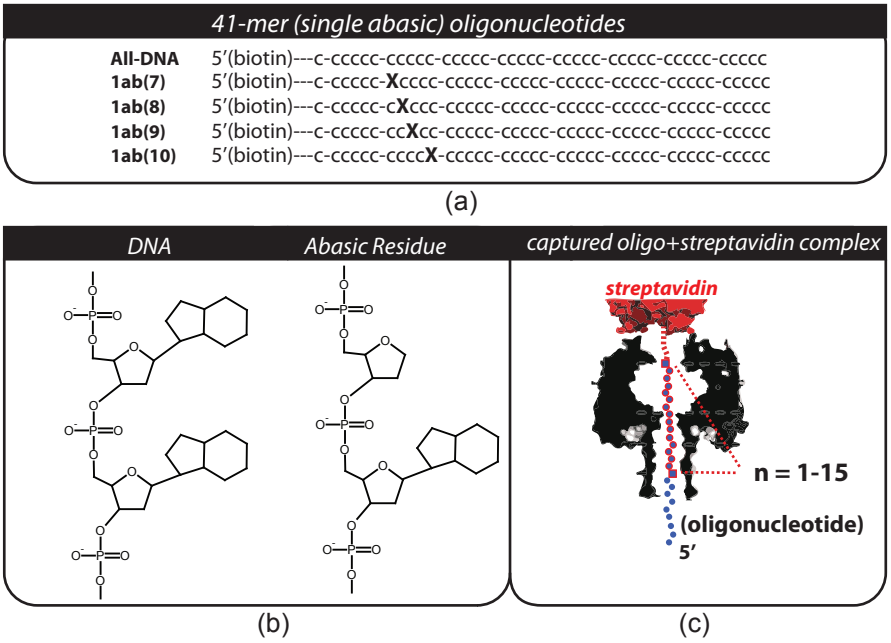


Figure 7: Polymer constructs used to map conductance sensitivity for Axopatch 200B and Nanoclamp. (a) shows the sequences of the all-DNA homopolymer and 1ab(7-10). They are four different constructs with a 5' biotinylated tail in which the protein streptavidin binds to. The **X** denotes the position of the abasic residue. (b) provides the structure of an abasic residue, which is the absence of the nucleobase from the deoxy-ribose suger along the phosphate-backbone. (c) provides the nucleotide position map of streptavidin-bound polymer captured within an AHL channel (shown as a cross-sectional cutout).

mobilized DNA strand. At positions corresponding to 1ab(8), 1ab(9) and 1ab(10), we span the most limiting region within AHL. As explained in the introduction, the aperture has a diameter of ∼1.5 nm. Consistent with other studies [62, 10], our experimental results show that the highest average current levels are shown for those positions. At position 8, the Nanoclamp read an average current amplitude of 1ab(8) templates is approximately 25.44 ± 0.92 pA, which is 3.58 pA higher than the baseline DNA template and also has a resistance of 4.72 GΩ, Figure (**??**). At the position 9, template of 1ab(9) has the highest current level on average measured as 26.69 ± 0.867 pA (∼4.50 GΩ). From position 10, the current amplitude starts decreasing. The average current of 24.73 ± 1.06 pA (∼4.85 GΩ) for 1ab(10) is lower

than the current levels at positions 8 and 9, and slightly higher than the baseline current. This is because position 10 is close to the β-barrel area.

3.4 Conclusion

In conclusion, the nanoclamp is able to resolve several picamperes of current changes due to angstrom shifts in a captured polymer. Detecting current changes at this magnitude is comparable to nanopore sequencing applications. In future work, CMS input-referred noise current for the Nanoclamp will be reduced with the objective of achieving at most $\sim$ 1 pA RMS. Although the Axopatch fairs better than the Nanoclamp for lower noise at recording bandwidths, only the Nanoclamp amplifier can scale for multi-channel nanopore implementations, which will be necessary for efficient genomic sequencing.

3.5 Acknowledgements

Experiments were designed by William Dunbar and me. Experiments were performed by Jungsuk Kim and me. The Nanoclamp amplifier was designed by Jungsuk Kim with the guidance of Ken Pedrotti.

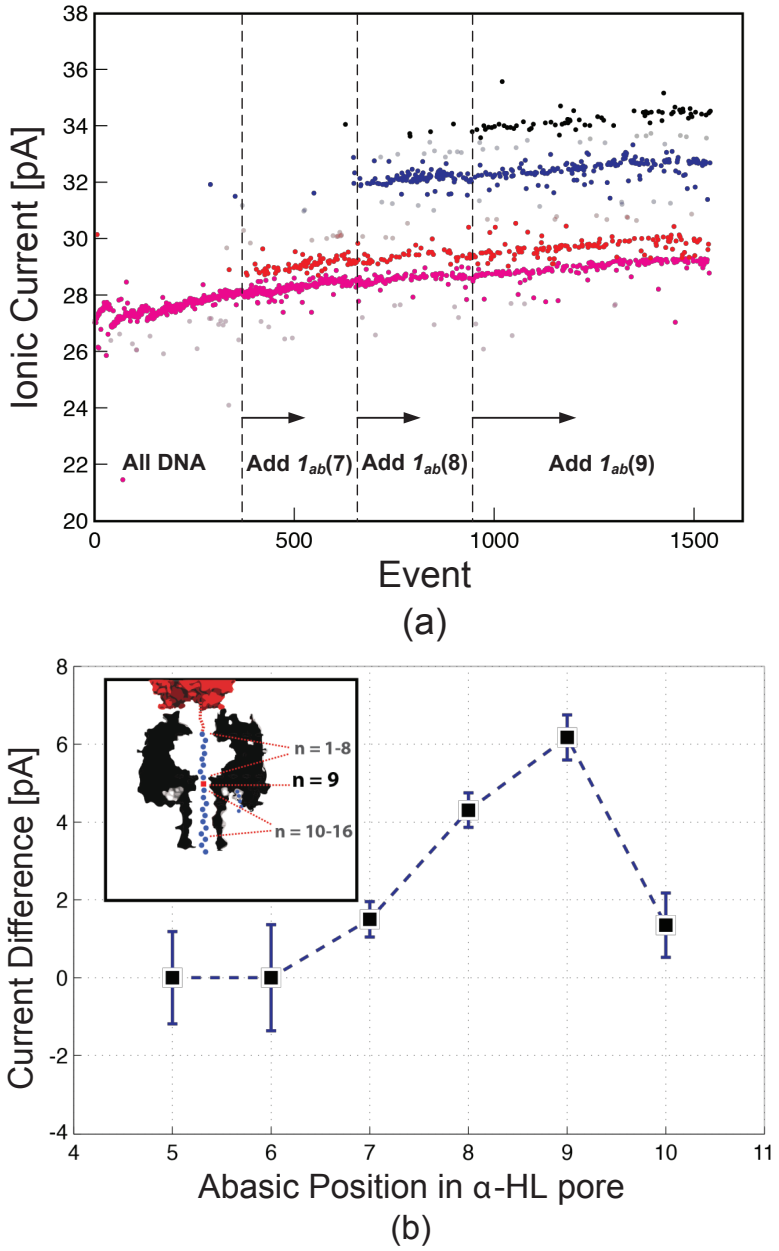


Figure 8: Axopatch current sensitivity map. (a) shows raw mean ionic current for each event captured on a single AHL channel. Single-abasic homopolymers (1ab(7-9)), were added one at a time with a line delineating each addition. Here, we can see a difference ($\sim$ 1.4 pA) in current level between a standard DNA and a 1ab(7) template although the Nanoclamp didnt show the difference between these two templates. This is because the Axopatch 200B has smaller input noise than the CMOS CMS noise (= 4.07 pA RMS). (b) shows a the constructed map from taking the mean of each single population. The positions contribute most to pore resistance are positions 8 and 9. This result agrees with our measurements using the Nanoclamp. Here, we separately measured a current level for position 10 after perfusing the previous experiment shown.

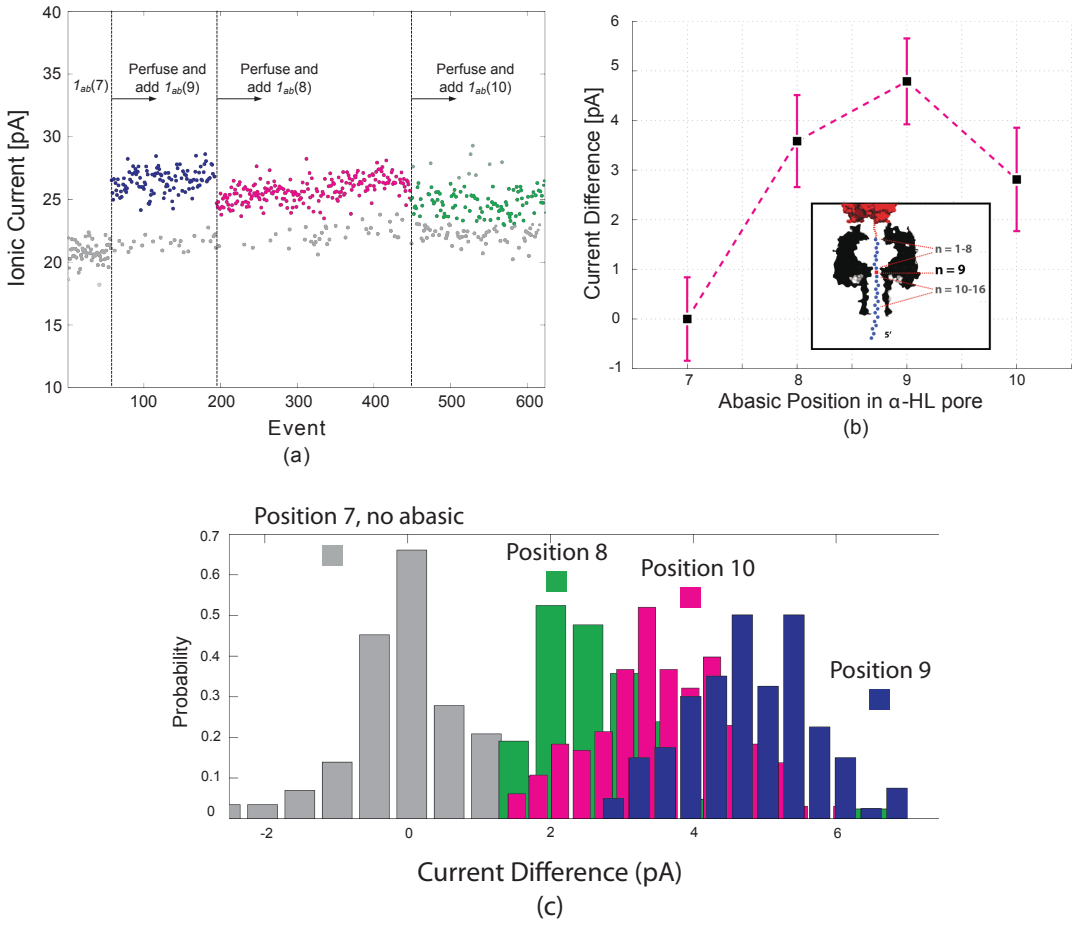


Figure 9: Nanoclamp current sensitivity map. (a) shows the raw mean ionic current for each event captured on an AHL channel. Unlike the Axopatch 200B, the Nanoclamp was unable to see any current changes in 1ab(7) with respect to the all-DNA template. Polymers 1ab(9-10) showed the most significant current changes from the all-DNA template. (b) is the constructed map of current change with respect to position in an AHL pore. The map follows the same trend as the Axopatch 200B's map. (c) Histograms of the event populations. Each population distribution overlaps due to the small current changes and wide variance. This is obviously due to less resolution from the Nanoclamp.

4 Preliminary Modeling of DNA Motion through a Dual Solid State Pores

4.1 Introduction

Work presented in this chapter is preliminary to a future project in process now. The chapter serves as a introduction to this future work and attempts to illustrate the process of designing the Langevin equation used for simulating a DNA strand translocating through a nanopore. Code for the model (MATLAB) is supplied in the appendix.

In the introduction, we discuss the potential of solid-state nanopores as a future platform for DNA sequencing. Presently, there is no mechanism in which DNA is slowly processed through a solid-state nanopore. The $\Phi 29$ molecular motor has been used to slow the movement of DNA through a biological nanopore, but this has not been demonstrated with solid-state nanopores. An alternative method to control the movement of DNA through a nanopore has been proposed by this lab (the Dunbar Biocontrols Laboratory) in which two solid-state nanopores are used in series.

The dual pore method positions two pores in such a way that the openings to both pores are aligned one on top of the other, Figure (10). To precisely control the motion of DNA with this method, a strand must first be captured in both pores; this could be possible by capturing DNA in one pore and as it translocates, the leading end of the strand is captured again in the second pore. Upon capture in both pores, a DNA strand could possibly be held static by applying equal and opposite voltage forces or slowly pulled through the pores by creating a small force differential between the opposing voltages.

To verify whether the dual-pore method is possible, preliminary modeling was done as a stepping stone to a more involved model with DNA initially caught between two pores. From these models we hope to design an optimal input controller for slowing DNA translocation through a solid-state nanopore.

Published models concerning the behavior of DNA translocation through a nanopore are described with Langevin dynamics([61, 4, 19, 60]); our model was developed sim-

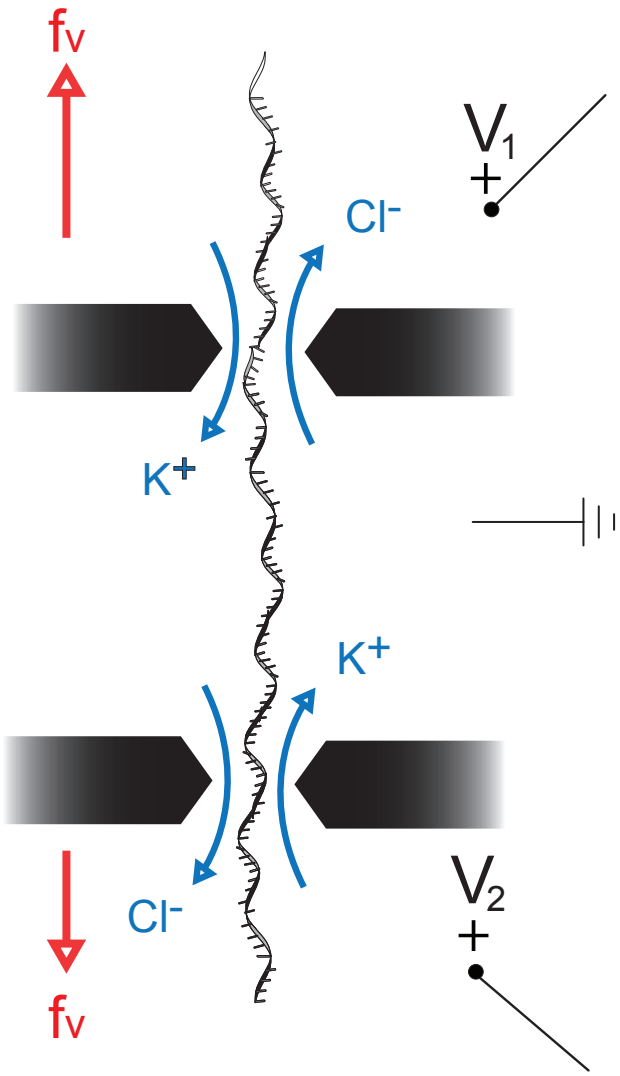


Figure 10: Illustration of dual-pore setup. Two solid-state nanopores are aligned one on top of the other. Three electrodes are used, the third being a common ground bewteen the other two. Voltages could possibly trap a strand of DNA in both pores(as seen). A slight voltage differential after capture could possibly cause a translocation rate slow enough for sequencing.

ilarly.

4.2 Methods

4.2.1 Model

The motion of a particle is governed by the Langevin equation[60, 28, 52]:

$$m\frac{d^2X(t)}{dt^2} = -\gamma\frac{dX(t)}{dt} - f_\omega(t) \tag{3}$$

where X(t) is the position coordinates of a particle at time t, m is the mass of the particle, γ is the drag on the particle and $f_\omega(t)$ is Brownian motion. For a charged particle under influence of a voltage force, we add to the model another force term and the Langevin equation becomes

$$m\frac{d^2X(t)}{dt^2} = f_v(t) - \gamma\frac{dX(t)}{dt} - f_\omega(t) \tag{4}$$

where $f_v(t)$ is the voltage force on the particle. If we consider a chain of particles bonded together, a force acting upon one particle will influence the motion of adjacent particles. We can model the bond behavior between particles as a spring, and with this additional term the langevin equation becomes

$$m\frac{d^2X(t)}{dt^2} = f_v(t) - f_k(t) - \gamma\frac{dX(t)}{dt} - f_\omega(t) \tag{5}$$

where $f_k(t)$ is the spring force bewteen particles. Since the particles in our model represent DNA, it is assumed that the mass m of the particles is very small and has little effect on their motion. This allows us to simplify the Langevin equation by assuming m = 0. The first half of the equation disappears and we solve for dX(t)/dt to get

$$\frac{dX(t)}{dt} = \frac{1}{\gamma}[f_v(t) - f_k(t) - f_\omega(t)] \tag{6}$$

The simplified model is now 1^{st} order and it can be used to describe the behavior of a DNA strand translocating through a nanopore by making the following assumptions: i) the DNA strand is composed of N points, with a rest length between each point

of R_L; ii) the DNA is initially captured within a nanopore; iii) the DNA strand is in 2-dimensional space. The average mass of a mononucleotide is considered to be 5.43×10^{-10} pg and the distance between stacked base pairs is approximately 0.34 nm. Assuming a rest length $R_L = 10$ nm, the mass of a single point could be approximated to be $m(R_L/(0.68\text{ nm})) = 8.3796 \times 10^{-9}$ pg. DNA's mass is very insignificant and it is appropriately simplified to 0.

After defining the general structure of the Langevin equation, the dynamics of each variable must be specified. The voltage force is assumed to affect only the points within the nanopore at time t. Outside of the pore, it is assumed that there is no voltage force affecting the strand. This is not accurate in the experimental setting, since diffusion is not the only force dictating the capture rate of a strand. It is also assumed that the voltage force acts only in the x direction along the nanopore. The voltage force $f_v(t)$ is modeled as,

$$f_v(t) = \lambda \frac{eV}{d} \tag{7}$$

where λ is the linear charge density, e is the elementary charge, V is the command voltage and d is the depth of the pore.

If we assume that DNA is like a cylinder, then viscous drag acts upon a single point on the DNA strand in two different ways: i) parallel drag and ii) perpendicular drag. The viscous drag constant γ can be modelled in terms of two different equations if we assume DNA is like a cylinder with segments of length R_L,

$$\gamma_{||} = \frac{2\pi\eta R_L}{\ln(\frac{R_L}{R}) + 0.2} \tag{8}$$

$$\gamma_{\perp} = \frac{4\pi\eta R_L}{\ln(\frac{R_L}{R}) + 0.84} \tag{9}$$

where η is the viscosity of the medium, R is the radius of the DNA strand, and R_L is the rest length between points. Equation (8) is the parallel drag on a cylinder of length R_L and radius R. The derived Langevin equation is simplified to only one

drag constant and γ is considered to be the average beween $\gamma_{||}$ and $\gamma_{\perp}$.

The Brownian motion term $f_{\omega}(t)$ is a function of the diffusion coefficient D and the derivative of a Weiner process dW. A Brownian particle's trajectory is directly proportional to its diffusion coefficient D and governed by the Einstein relation (Stokes-Einstein equation)[60],

$$D = \frac{k_b T}{\gamma} \tag{10}$$

where k_b is Boltzmann's constant and T is the temperature in Kelvin. The magnitude of a Brownian particle's displacement is defined by the equation,

$$f_{\omega}(t) = \frac{dW(t)}{dt}\sqrt{2D} \tag{11}$$

where $dW \sim N(0,dt)$. Due to its stochastic nature, dW/dt has to be interpreted as a difference equation in the form,

$$dX(t+dt) = X(t) + ...dW(t)\sqrt{2D} \tag{12}$$

Rearranging Eq. (11) to solve for X(t) yields,

$$dX(t) = ... + W(t)\sqrt{2Ddt} \tag{13}$$

where $W(t) \sim N(0,1)$. Determining the correct time step dt for simulation is important for numerical stability. As dt increases, the dX(t) from the Brownian force increase proportionally.

The spring force, $f_k(t)$, is dependent on the position of the point on the strand. The first and last points on the strand are affected by only one spring; points inbetween the the first and last are affected by two springs. Our model is considered to be in 2-Dimensional Cartesian space in which directional changes in the x-component and y-component are accounted for. Figure (11) illustrates the geometric interpretation of a DNA strand modeled as points and diagrams each parameter that affects the spring force on a given point. The spring constant depends on the angle between

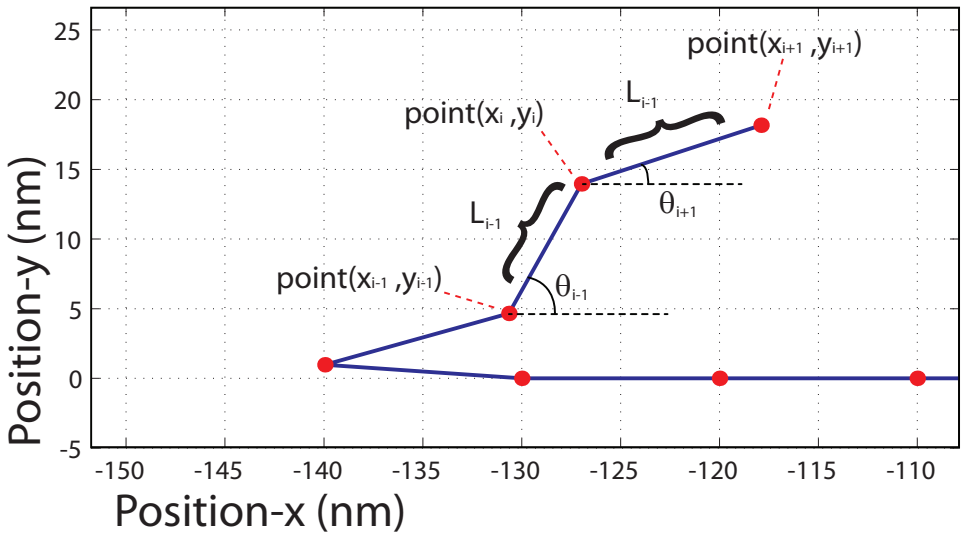


Figure 11: The x and y coordinates are used to calculate length, L, bewteen the points, as well as the angle θ between points; from these variables, the x and y-components to the spring force can be calculated.

the points θ, the length of the spring L and the rest length R_L. The spring force in the x-direction is given by

$$f_k^i = k(L - R_L)\cos(\theta) \tag{14}$$

and the spring force in the y-direction is given by,

$$f_k^i = k(L - R_L)\sin(\theta) \tag{15}$$

Each variable definition was used to program the model in Equation (6). Parameter values are provided in Table (4).

4.2.2 Numerical Methods

We use both the Euler and Runge-Kutta numerical methods to solve our stochastic differential equation. The Euler method is a 1^{st} order numerical procedure to solving ordinary differential equations and can be understood by looking back at Equation

Parameter	Symbol	Value/Dimensions
linear charge density	λ	0.03 C nm^{-1}
elementary charge	e	0.3 C
command voltage	V	100 mV
pore depth	d	30 nm
rest length	R_L	10 nm
DNA radius	R	2.2 nm
viscosity	η	10^{-3} Pa μs
boltzmann constant	k_b	$1.38 \times 10^{-23} m^2 s^{-2} K^{-1} kg$
temperature	T	298 K
spring constant	k	1000 pN/m

Table 4: Parameters for each equation are listed as well as their value used in simulation. Dimensions in the simulation were standardized to nm, mV and μs.

(6). If we multiply dt on both sides, Equation (6) becomes

$$dX(t) = \frac{1}{\gamma}[f_v(t) - f_k(t) - f_\omega(t)]dt \quad (16)$$

$$X(t+1) = X(t) + dX(t) \quad (17)$$

where solving for dX in a for-loop and adding back to X will give the position of the DNA strand in the next time step. A MATLAB example is provided,

```
for i = 1:length(X)
    %calculte fv(x), fw(x), fk
    %...
    %deterimine dX
    dX(i,1) = (fv(1)*dt - fk(1)*s_theta*dt + fomega*(min(max(randn,
        -4), 4)))/gamma;%x-component
    dX(i,2) = (fv(2)*dt - fk(2)*c_theta*dt + fomega*(min(max(randn,
        -4), 4)))/gamma;%y-component
end
%add back to the current position to get the position of the strand at
     t+1
X = X + dX;
```

The code provided is just a small piece of a larger for-loop that breaks up the spring forces into either the first point, middle points, or the last point on the strand. The

physics is different for each. This example is just to illustrate simple implementation of the Euler method. The dX variable calculated from the for loop would be added to the position X to find the position at the next time iteration. Fourth order Runge-Kutta is a more accurate method of solving differential equations and it takes the form

$$\begin{aligned} X(t+1) &= X(t) + \frac{1}{6}dt\left(k_1 + \frac{1}{2}k_2 + \frac{1}{2}k_3 + k_4\right) && (18)\\ k_1 &= X(t) && (19)\\ k_2 &= X\left(t + \frac{1}{2}dt, X(t) + \frac{dt}{2}k_1\right) && (20)\\ k_3 &= X\left(t + \frac{1}{2}dt, X(t) + \frac{dt}{2}k_2\right) && (21)\\ k_4 &= X\left(t + dt, X(t) + k_3 dt\right) && (22) \end{aligned}$$

Runge-Kutta breaks a dt-step into 4 incremental steps (k), averaging the increments together. Code for this implementation is provided in the appendix.

4.3 Results/Discussion

Parameter values in Table (4) were used to calculate the approximate forces and constants. Each variable definition was used in simulation (code supplied in Appendix). The average values are reported in Table (5). The voltage force generated from a 100 mV command voltage is $\sim$ 9.4 pN per point on a DNA strand. This is consistent with values calculated in [28]. Changing λ will directly affect the amount of voltage force applied to each point, but the linear density used is comparable to [28]. The variance of the Brownian force is very large, so in simulation it is necessary to set a larger time step dt or use a larger drag force to settle the Brownian term. A large time step may force the simulation to become numerically unstable. The large diffusion constant D ($\sim 79.1\text{nm}^2\ \mu\text{s}^{-1}$) contributes to the large Brownian force. A time step of $dt = 10^{-5}\mu$s gives a $dX = \sqrt{2Ddt} = 0.0399$. In simulation, if the voltage force is $f_v(t)dt$ or 9.4dt, Brownian motion will drown out the affects of voltage. Parameters must be tuned to allow voltage force to rule within the pore. As the model gets more developed, more accurate values for some of these parameters will be used

to accurately describe the behavior of DNA translocation. The code written outputs

Variable/Constant	Symbol	Calculated/Average Value
Voltage	$f_v(t)$	$\sim$ 9.4 pN
Brownian	$f_\omega(t)$	$\sim$N(-400, 400) pN
Drag	γ	0.0513 pN μs nm^{-1}
Diffusion Coefficient	D	79.70 nm^2 μs^{-1}

Table 5: Parameters for each equation are listed as well as their value used in simulation. Dimensions in the simulation were standardized to nm, mV and μs.

an 2-D animation of a strand translocating through a nanopore. Boundary conditions are accounted for by elastic collision equations. Equations from this chapter are used to simulate the physics on the strand. A frame-by-frame illustration of the actual plots in MATLAB are provided, Figure(12).

4.4 Conclusion

In conclusion, there is much work to be done on this project. Parameters in the Langevin model must better approximate molecular bahavior within a nanopore. A stable time-step must be chosen. There is still need to apply it to a dual pore model. So far, the code provided in the Appendix generates animations of variable-sized strands passing through a nanopore but with littel accuracy compared to more rigorous models. Overall, there is much work to be done on this project.

4.5 Acknowledgements

The model was designed by both William Dunbar and I, with the guidance of Hongyun Wang. Coding and animations were done by me.

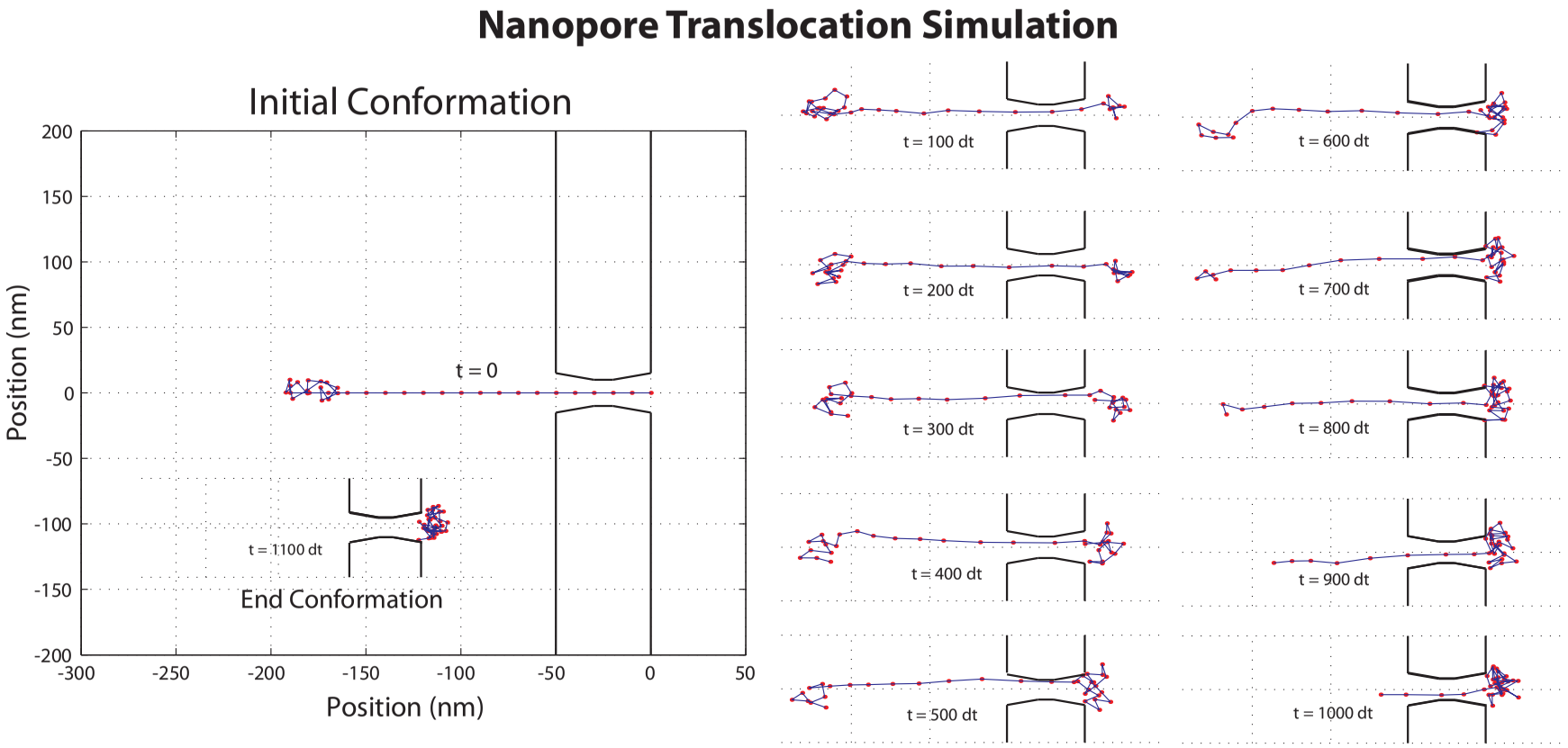


Figure 12: Example simulation of DNA translocation. Command voltage was set to 1000 mV since the time step was very low, dt = $10^{-5}\mu$s. The length of the strand was 1000 base-pairs. Frames are shown 100 time steps at a time to show the general motion of the strand through the pore. Initial and end conformations are shown in the left panel.

5 Preliminary Experiments with Solid-State Nanopores and Nucleosome Species

5.1 Introduction

Biological nanopores are discussed throughout the first and second chapters. However, for potential nanopore MEMS (micro-electromechanical) devices, biological nanopores are not as ideal as solid-state nanopores. Biological nanopores require expensive reagents, that is, the protein pore and lipid bilayer. Most lipids used in academic research involve the spontaneous formation of the bilayer, which is non-trivial, and the stability of the bilayer varies from experiment to experiment. Some labs have successfully engineered chemical procedures to synthetically construct lipid bilayers that are more durable, offering stable performance in MEMS devices. For each experiment, a single protein pore must spontaneously insert into the bilayer, a process that involves feedback control and perfusion to minimize the chance of a second insertion. Solid-state pores are ready for experimentation after drilling. A successful experiment is dependent on i) a properly setup holder—a chassis to isolate the silicon membrane between two liquid compartments such that an ionic connection is made only through the pore— and ii) a pore that is fully drilled through with the right size and resolution necessary for the experiment.

In this chapter, preliminary experiments with solid-state nanopores are explained with various types of holders. Insight is provided into the experimentation with solid-state nanopores: general power spectrum trends for a solid-state nanopore on our setup are shown; IV-curve analysis is presented and discussed; and current traces of nucleosome+DNA complexes are presented and discussed as well.

The aim of the work done in this chapter was to develop solid-state experiments into our lab until routine. Protocols for each holder were optimized as well protocols to treat SiN membranes. A large amount of data has been recorded over many pores, however, only a select few are presented.

5.2 Methods

5.2.1 Holders

We tested multiple silicon nitride (SiN) pores of varying sizes, specifically, nanopores ranging from 70 to $\sim$ 5 nm in diameter and 10 to 30 nm in pore length. Each of these pores has been tested on one (or multiple) of three holders; a holder is a chassis that securely positions the silicon membrane between two liquid compartments in such a way that ionic-liquid connection between the two compartments can only be made through the nanopore. The three holders used are from three different labs: Luke Theogarajan's Lab in UCSB (#1), Vincent Tabbard-Cosa's lab in University of Ottawa (#2) and our own Dunbar holder from UCSC(#3). Each holder requires a different protocol (supplied in Appendix) but practically they all accomplish the same thing.

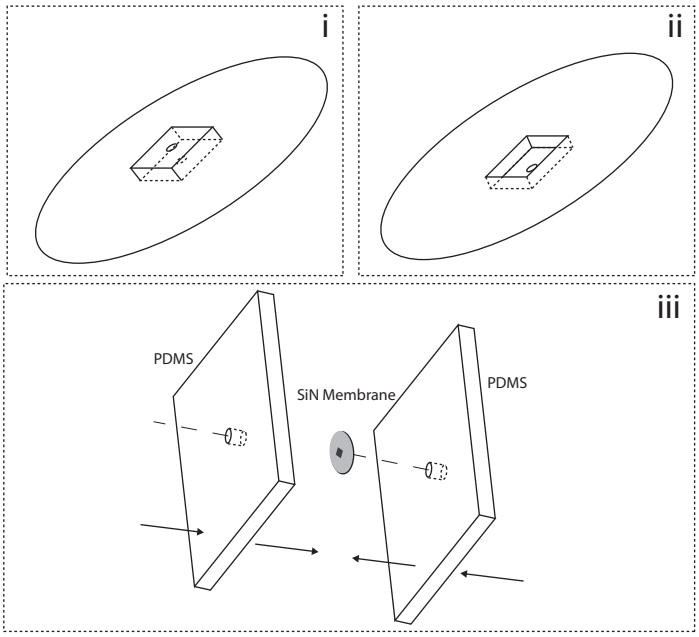


Figure 13: Holder #1 is a 2-part teflon-based holder. Each part fit together like a lego connection; the top piece has an extruding cube shape with a pin hole that fits into a cubed well in bottom piece, aligning the top pin hole to a bottom pin hole. Two polydimethylsiloxane (PDMS) layers act as the gaskets, sandwiching the silicon membrane and providing liquid tight seals.

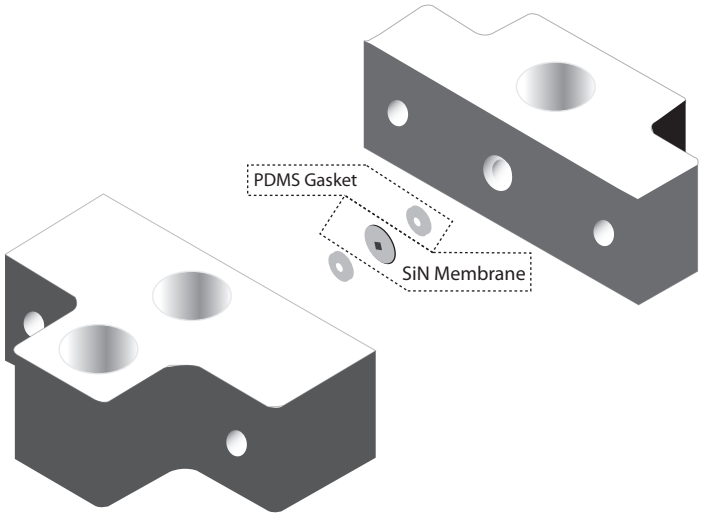


Figure 14: Holder #2 is made of teflon and comes in two parts. There are two holes on the sides of each piece to allow two bolts to thread through and clamp the two pieces together. In doing this, the PDMS gaskets sandwich the SiN membrane. The wells have very small holes to the membrane allowing for a low volume of buffer in each well.

5.2.2 DNA Species

Three types of molecules were tested on each pore:

1. λDNA: genomic DNA (48 Kbp) isolated from bacteriophage lambda
2. λDNA+RecA: a protein complex essential in DNA repair and recombination
3. dsDNA+Histones (3 Kbp nucleosomes): the complex in which chromosomes are packaged in

Figure (16) illustrates these three complexes and gives relative dimensions. The sample of λDNA was purchased from New England Biolabs. For each λDNA sample used in a Nanopore experiment, the final concentration in the *cis* well was 1 μM. The RecA protein was purchased from New Englabd Biolabs as well. λDNA+RecA complexes were formed with 1.2 nm λDNA, 6.5 μM RecA, 1.5 mM ATPγs (NEB), 10x RecA Buffer (NEB) [21]. The mixture was incubated at 37^{o} C for 1 hour and

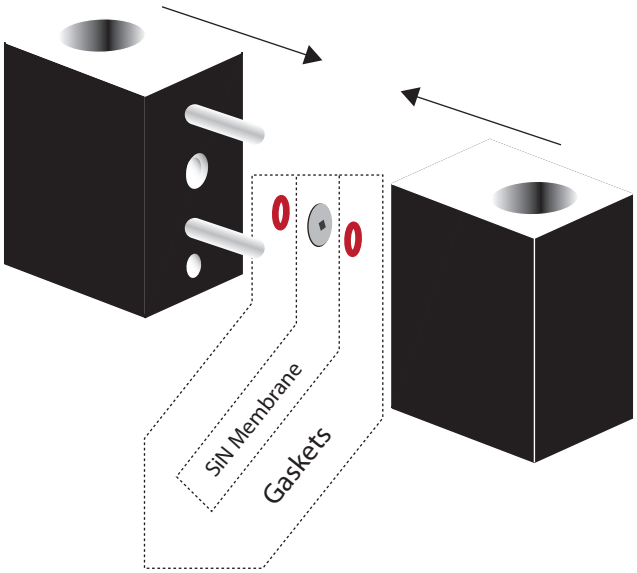


Figure 15: Holder #3 has two parts. There are two metal prongs on one piece that align the two pieces. A large bolt screws the two pieces together, sandwiching the SiN membrane between two gaskets.

Provider	Diameter	Height
Yuzvinsky and Schmidt, UCSC	10 nm, 15 nm, 20 nm	10, 30 nm
Yemeni and Garajan, UCSB	19 nm	30 nm

Table 6: We have various pores that have been drilled from 3 different labs. The two providers in this table are presented in the paper.

kept on ice during the experiment. A protocol is supplied in the Appendix.

Nucleosome samples were provided by Michael Poirer at OSU. The molecule contains a 17mer array of nucleosomes on a 3 Kbp strand of DNA. The complex was diluted to $\sim$ 1 nM in the *cis* well, which is relatively small compared to the standard 1 uM concentration, but species concentrations as low as 0.1 nM have been shown to have a capture rate of 1 event every 600 ms [22]. This was done at 1 nM due to the amount supplied by the Poirer Lab.

Data presented in this chapter was captured on four different pores. Electron microscope images of each pore are shown in Figure (17).

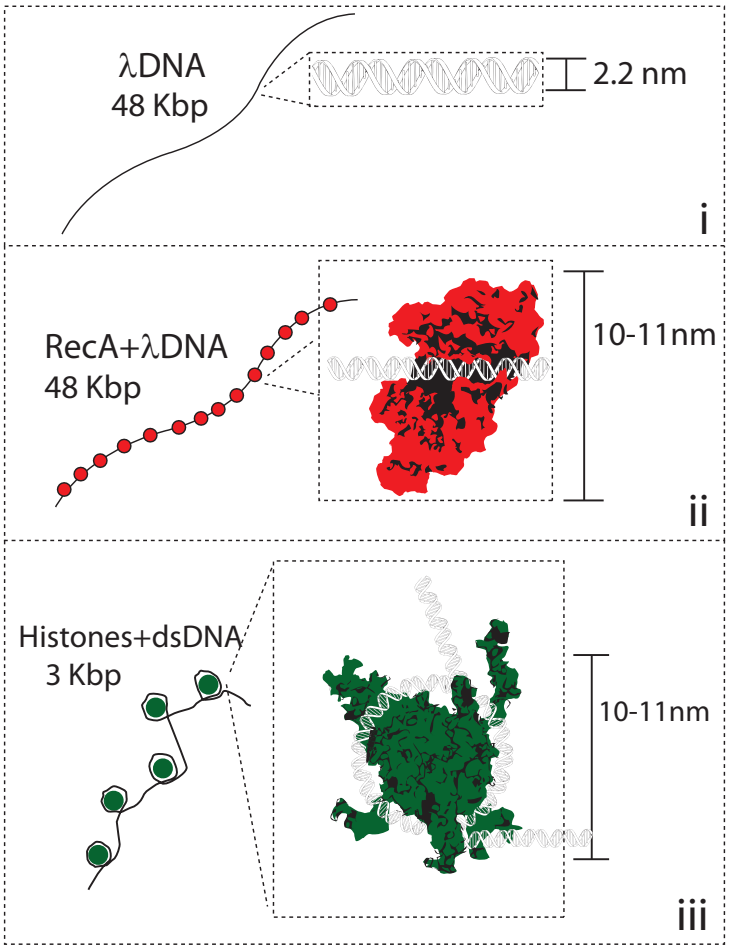


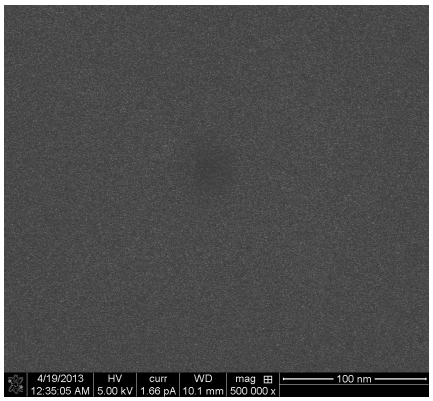
Figure 16: Three types of molecular complexes used in solid-state nanopore experiments. i) illustrates double-stranded λDNA which is 2.2 nm in width and roughly 48,000 base pairs. ii) shows RecA proteins forming complexes with λDNA. The protein wraps itself around the phosphate-backbone nonspecifically [1], as seen in the illustration. The RecA+DNA complex is 10-11 nm in width. Nucleosomes iii) are roughly the same width (10-11 nm). The species of molecule we used on the experiments had an array of 17 nucleosomes (multiple histones bound to DNA) on a 3 Kbp strand of DNA.

5.2.3 Setup

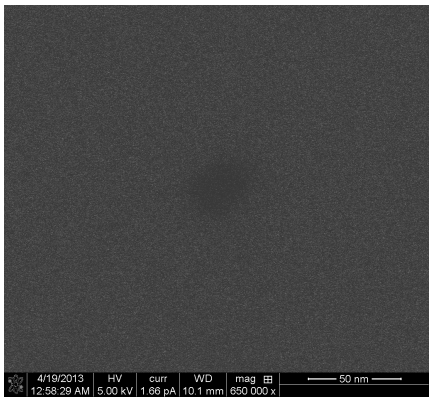
Data was amplified by an Axopatch 200B at 1 kHz BW and acquired by a Digidata 1400A at a sampling frequency of 100 kHz. The ionic buffer had a salt concentration of 1 M LiCl (for the experiments shown) and a pH of 8.

5.2.4 Experiment

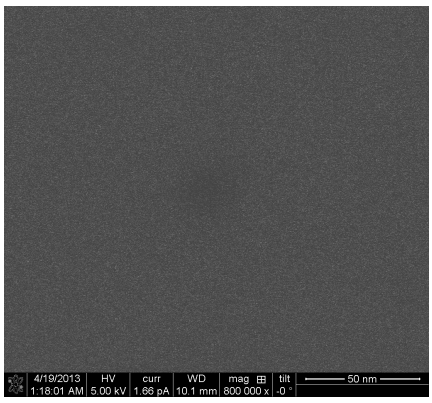
For each pores that was setup, a specific procedure was followed:



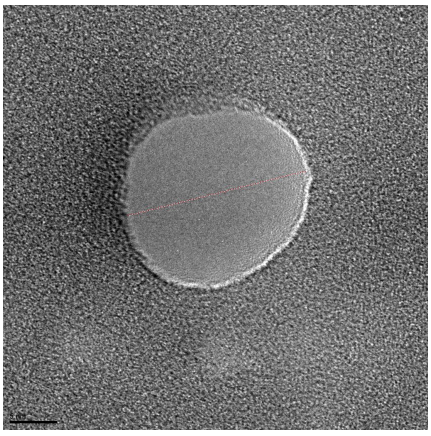
(a) Pore 1 is $\sim$ 20 nm diameter x 10nm length



(b) Pore 2 is $\sim$ 15 nm diameter x 10nm length



(c) Pore 3 is $\sim$ 10 nm diameter x 10nm length



(d) Pore 4 is $\sim$ 19 nm diameter x 30 nm length

Figure 17: Transmission electron microscope images of the pores drilled by the Schmidt Lab (a-c) and the Garajan lab (d). Images are not as definite (a-c) in order to reduce dimensional changes caused by the microscope.

1. 3 minutes of open channel current was acquired for power spectrum analysis
2. Several iterations of voltage titration from 250 to -250mV with a 25mv steps and 1 second hold for IV-curve analysis
3. 10 minutes of 100 mV hold with added DNA sample to capture events

5.3 Discussion/Results

Three pores were tested from a set of pores drilled by Yuzvinsky and Schmidt (Holger Schmidt Lab). Figure (17) shows transmission electron microscope images of the pores. The empirical resistances were calculated from voltage titrations done on each pore and the IV curves are very linear (Figure (18)). However, the measured resistance for each pore versus the theoretical resistance, considering just their measured dimensions, are very off. The equation used to calculate a pores theoretical resistance (published in [20]),

$$G = \frac{\pi \mathrm{d}^2}{4\mathrm{L}}((\mu_{\mathrm{K}} + \mu_{\mathrm{Cl}})\mathrm{ne} + \mu_{\mathrm{K}}\frac{4\sigma}{\mathrm{d}}) \quad (23)$$

$$\mathrm{n} = (10^3\mathrm{M})6.022 \times 10^{23} \quad (24)$$

$$\mathrm{R} = \mathrm{G}^{-1} \quad (25)$$

where d is the diameter of the pore, L is the length of the pore, μ_{K} is the electrophoretic mobility of potassium (7.616×10^{-8} m^2/Vs), μ_{Cl} is the electrophoretic mobility of chlorine (7.909×10^{-8} m^2/Vs), n is the number density of ions, e is the elementary charge, σ is the surface charge density of SiN which is taken to be the same as SiO_2, 0.06 C/m^2, and M is the salt concentration of the ionic buffer.

Table (7) compares measured pore resistance and theoretical pore resistance. There is large variability when making solid-state nanopores. To attain small pore diameters is non-trivial; membranes are very sensitive and over drilling a pore may produce a complete pore (with openings on both ends of the membrane), but the pore may be too large. It is possible that pores 1-3 were not completely drilled through or that one opening to the pore is consistent with the TEM image but the other opening is significantly smaller.

Pore	Measured	Theoretical
1	52.781 MΩ	3.527 MΩ
2	46.942 MΩ	6.065 MΩ
3	57.897 MΩ	12.810 MΩ
4	53.236 MΩ	3.887 MΩ

Table 7: Measured resistances on each solid-state nanopore versus the theoretical resistances.

During the experiment, 3 different samples were tested on each pore and individual times, λDNA, λDNA+RecA and Nucleosome complexes. None of the samples were seen translocating or interacting with the solid-state nanopores.

Experiments on the 19 nm pore (from Garajan) had a measured resistance of 53.236 MΩ which is far from the theoretical resistance of 3.887 MΩ (Table (7), Figure(19)), however, translocations were deteced with a sample of nucleosomes when the voltage was reversed, that is, we had negatively charged nucleosome complexes in the postive trans well, Figure (20).

We observed various types of events: sharp spikes, broad event dwell times with a clear amplitude and blockades that had to be "kicked" out with a reverse voltage. Figure (20) examples each of these various types. After analyzing the data, it is clear the nucleosome complexes were not formed, that is, DNA was not wrapped around histones to form the nucleosome complexes. After a discussion with Michael Poirer's Lab, we concluded that the salt concentration was too high; the nucleosome complexes are stable at salt concentrations < 0.5 M, so at 1 M LiCl, it is likely no nucleosomes were formed. Since complexes were likely not formed, and a reverse polarity was used to detect translocations, we hypothesize that positively charged histones were translocating and clogging the nanopore, not DNA. However we need to further test this.

5.4 Conclusion

Overall, we aim to accurately map protein complexes along a DNA strand. Results from this experiment are preliminary steps to using the solid-state nanopore for this purpose. Future experiments will be done to verify these results as well as to

optimize the experiment mapping considerations.

5.5 Acknowledgements

Pores for all experiments (not just the ones presented) were provided by Nanopore Solutions, Luke Garajan's Laboratory (UCSB) and Holger Schmidt's Laboratory (UCSC). Nucleosomes were provided by Michael Poirer at OSU. Holders were provided by Nanopore Solutions, Garajan Lab, Vincent Tabbard's Lab at UBC, and William Dunbar's Lab. Experiments were designed and performed by me.

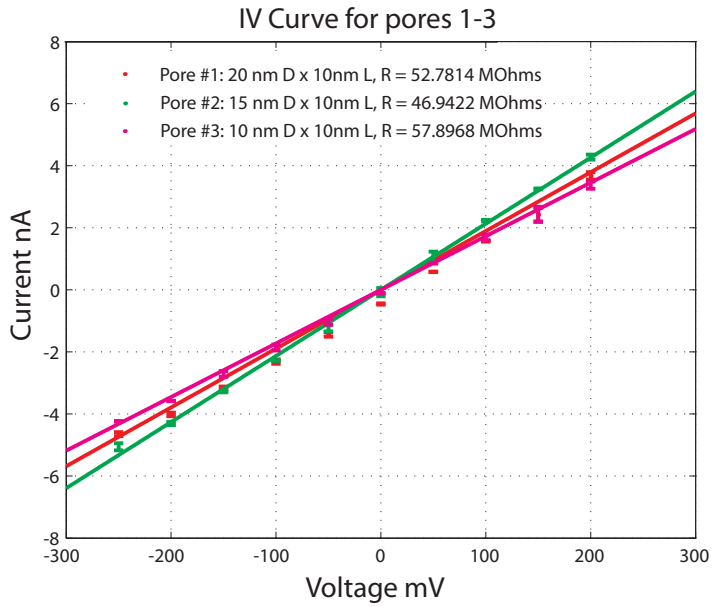


Figure 18: IV-curve analysis for pores 1-3. The experimentally determined resistances are smaller than the expected value for the pores if they were cylindrical. So the proposed dimensions of the drilled pores (defined by the TEM images) may have changed during imaging, or the inner structure of the pores are smaller that the opening diameter.

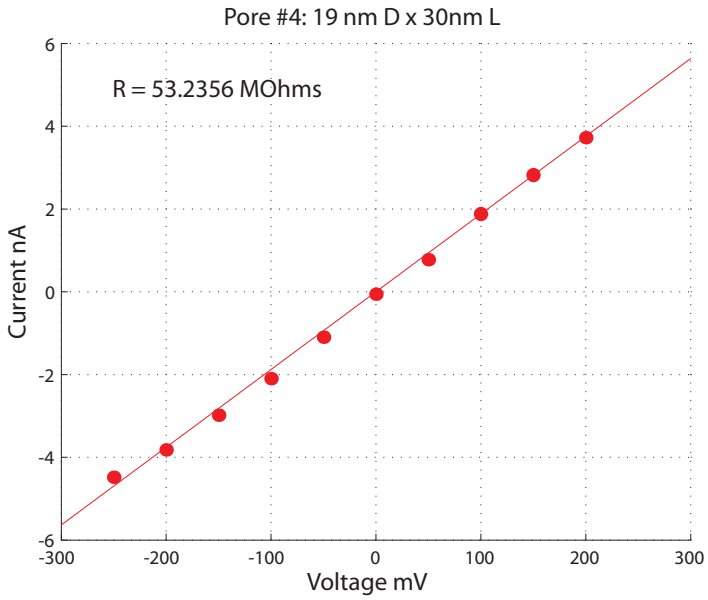


Figure 19: IV-curve analysis for pore # 4. The experimentally determined resistance is a lot smaller than the expected value for this pore as well, however, with this pore, translocations were detected.

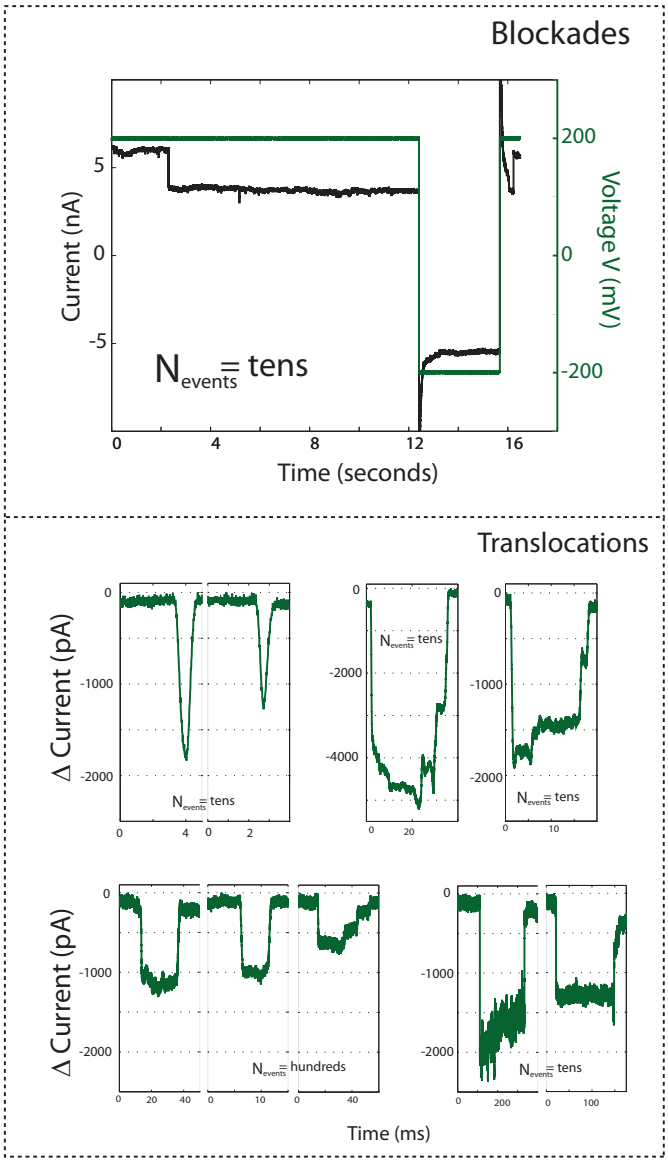


Figure 20: Varying event behavior of nucleosomes on pore #1. There is obvious interaction between the sample of nucleosomes with the solid-state nanopore. From these results, we cannot be certain whether these are complete translocations through the pore. It is possible that the complex is getting stuck in the pore and popping out but this needs to be verified and tested.

6 Conclusion

Nanopores are a remarkable tool for detecting molecular structure. In concept, it is simplistic but its ramifications in the biotech industry are major. Optimizing a nanopore platform for DNA sequencing can increase read lengths and reduce cost of reagents compared to other modern platforms. The research in this thesis has aimed to optimize several areas within the nanopore technology in order to realize its sequencing potential.

In chapter 2, a circuit model of the nanopore is defined and shown to accurately measure the system behavior of a nanopore. This work was published [24]. Subsequent work with the circuit model in state-space form can accurately estimate conductance G_c changes. In an ideal dual-pore setup, it would be possible to apply an estimator like this to detect fast conductance changes as a DNA strand is oscillating through a pore, potentially resequencing a single strand of DNA.

In chapter 3, a novel amplifier was tested for its ability to distinguish current changes comparable to those in a nanopore-sequencing experiment. This work was published [32]. The novel amplifier is comparable to the precision of the standard Axopatch 200B amplifier, and due to its scalability, we can multiplex the amplifier on a platform for generic high-throughput nanopore detection in the future.

In chapter 4, a preliminary model for DNA translocation through a nanopore is designed based on the Langevin equation. Simulation scripts are written and presented. In the future, the model can be developed further and eventually applied to a dual-pore setup to verify experimental results.

Finally, chapter 5 discusses preliminary experiments with the solid-state nanopore and optimizing experimental conditions to see nucleosome complexes translocating through a pore. We showed that we are in the beginning steps to detecting interaction between the pore and molecular complexes. In future experiments, we hope to regularly fabricate pores and test novel biomolecular complexes on them.

Nanopores have potential to become a ubiquitous biomolecular assaying tool as we continue to engineer it. We can properly estimate physical attributes to the nanopore system and possibly in real time; new practical platforms are being developed with

newly designed amplifiers that are scalable and low-noise; novel nanopore setups, such as the dual pore, could possibly take over as the ideal translocation rate-control method. From these achievements and future research, it is undoubtable that nanopores will continue to progress as a modern tool for molecular biology.

References

[1] Kenji Adzuma. No sliding during homology search by reca protein. *J. Bio. Chem.*, 273:31565–31573, November 1998.

[2] M Akeson, D Branton, J J Kasianowicz, E Brandin, and D W Deamer. Microsecond time-scale discrimination among polycytidylic acid, polyadenylic acid, and polyuridylic acid as homopolymers or as segments within single RNA molecules. *Biophys J*, 77(6):3227–33, Dec 1999.

[3] J. Xu-T.K. Vanderlick E. Culurciello B. Goldstein, D. Kim. Cmos low current measurement system for biomedical applications. *IEEE Transaction on Biomedical Circuits and Systems*, 6:111119, 2012.

[4] D. P. Hoogerheide B. Lu, F. Albertorio and J. A. Golovchenko. Origins and consequences of velocity fluctuations during dna passage through a nanopore. *Biophys. Jou.*, 101:70–79, July 2011.

[5] A. Banerjee, E. Mikhailova, S. Cheley, L. Q. Gu, M. Montoya, Y. Nagaoka, E. Gouaux, and H. Bayley. Molecular bases of cyclodextrin adapter interactions with engineered protein nanopores. *Proc. Natl. Acad. Sci. U.S.A.*, 107(18):8165–8170, May 2010.

[6] S.J. Chen Z. Zhu D.E. Kotecki R.L. Smith S.D. Collins B.C. Gierhart, D.G. Howitt. Nanopore with transverse nanoelectrodes for electrical characterization and sequencing of dna. *Sensors and Actuators B: Chemical*, 132:593600, 2008.

[7] Seico Benner, Roger J A Chen, Noah A Wilson, Robin Abu-Shumays, Nicholas Hurt, Kate R Lieberman, David W Deamer, William B Dunbar, and Mark Akeson. Sequence-specific detection of individual DNA polymerase complexes in real time using a nanopore. *Nat Nanotechnol*, 2(11):718–724, Nov 2007.

[8] D. Branton, D. W. Deamer, A. Marziali, H. Bayley, S. A. Benner, T. Butler, M. D. Ventra, S. Garaj, A. Hibbs, X. Huang, S. B. Jovanovich, P. S. Krstic,

S. Lindsay, X. S. Ling, C. H. Mastrangelo, A. Meller, J. S. Oliver, Y. V. Pershin, J. M. Ramsey, R. Riehn, G. V. Soni, V. Tabard-Cossa, M. Wanunu, M. Wiggin, and J. A. Schloss. The potential and challenges of nanopore sequencing. *Nat. Biotechnol.*, 26(10):1–8, Sep 2008.

[9] T. Z. Butler, M. Pavlenok, I. M. Derrington, M. Niederweis, and J. H. Gundlach. Single-molecule DNA detection with an engineered MspA protein nanopore. *Proc. Natl. Acad. Sci.*, 105(52):20647–20652, 2008.

[10] G. M. Cherf, K. R. Lieberman, H. Rashid, C. E. Lam, K. Karplus, and M. Akeson. Automated forward and reverse ratcheting of DNA in a nanopore at 5-Å precision. *Nat. Biotech.*, 2012.

[11] J. Chu, M. González-López, S. L. Cockroft, M. Amorin, and M. R. Ghadiri. Real-time monitoring of DNA polymerase function and stepwise single-nucleotide DNA strand translocation through a protein nanopore. *Angew. Chem.*, 122(52):10304–10307, Nov 2010.

[12] J. Clarke, H. C. Wu, L. Jayasinghe, A. Patel, S. Reid, and H. Bayley. Continuous base identification for single-molecule nanopore DNA sequencing. *Nat. Nanotech.*, 4(4):265–270, 2009.

[13] S.L. Cockroft, J. Chu, M. Amorin, and M. R. Ghadiri. A single-molecule nanopore device detects DNA polymerase activity with single-nucleotide resolution. *J. Amer. Chem. Soc.*, 130(3):818–820, 2008.

[14] D.W. Deamer and D. Branton. Characterization of nucleic acids by nanopore analysis. *Acc. Chem. Res.*, 35:817–825, 2002.

[15] C. Dekker. Solid-state nanopores. *Nat. Nanotech.*, 2:209–215, 2007.

[16] I. M. Derrington, T. Z. Butler, M. D. Collins, E. Manrao, M. Pavlenok, M. Niederweis, and J. H. Gundlach. Nanopore DNA sequencing with MspA. *Proc. Natl. Acad. Sci. U.S.A.*, 107(37):16060–5, Sep 2010.

[17] B. Dorvel, G. Sigalov, Q. Zhao, J. Comer, V. Dimitrov, U. Mirsaidov, A. Aksimentiev, and G. Timp. Analyzing the forces binding a restriction endonuclease to DNA using a synthetic nanopore. *Nuc. Acids Res.*, 37(12):4170–79, 2009.

[18] M. Eisenstein. Oxford nanopore announcement sets sequencing sector abuzz. *Nat. Biotech.*, 30:295–296, 2012.

[19] A. J. Storm et al. Fast dna translocation through a solid-state nanopore. *Nano Lett.*, 5:11931197, February 2005.

[20] R. M. M. Smeets et. al. Salt dependence of ion transport and dna translocation through solid-state nanopores. *Nano Letters*, 6(1):80–95, October.

[21] R. M. M. Smeets et al. Translocation of reca-coated double-stranded dna through solid-state nanopores. *Nano Letters*, 9(9):3089–3095, October.

[22] Singer et al. Electronic barcoding of a viral gene at the single-molecule level. *Nano Letters*, 12:17221728, Feb 2012.

[23] D. Garalde, C. Simon, J. Dahl, H. Wang, M. Akeson, and K. R. Lieberman. Distinct complexes of DNA polymerase I (Klenow fragment) for base and sugar discrimination during nucleotide substrate selection. *J. Biol. Chem.*, 286(16):14480–14492, 2011.

[24] D. R. Garalde, C. R. O'Donnell, R. D. Maitra, D. M. Wiberg, G. Wang, and W. B. Dunbar. Modeling the biological nanopore instrument for biomolecular state estimation. *IEEE Trans. on Contr. Syst. Tech.*, PP:1, Dec 2012.

[25] B. Gyarfas, F. Olasagasti, S. Benner, D. Garalde, K. R. Lieberman, and M. Akeson. Mapping the position of DNA polymerase-bound DNA templates in a nanopore at 5 Å resolution. *ACS Nano.*, 3(6):1457–1466, 2009.

[26] A. R. Hall, A. Scott, D. Rotem, K. K. Mehta, H. Bayley, and C. Dekker. Hybrid pore formation by directed insertion of α-haemolysin into solid-state nanopores. *Nat. Nanotech.*, 5(12):874–7, Dec 2010.

[27] S. Howorka, S. Cheley, and H. Bayley. Sequence-specific detection of individual DNA strands using engineered nanopores. *Nat. Biotech.*, 19:636–9, 2001.

[28] T. Ala-Nissila I. Huopaniemi, K. Luo and S. Ying. Langevin dynamics simulations of polymer translocation through nanopores. *J. Chem. Phys.*, 125, August 2006.

[29] S. M. Iqbal, D. Akin, and R. Bashir. Solid-state nanopore channels with DNA selectivity. *Nat. Nanotech.*, 2:243–248, 2007.

[30] C.A. Merchant-M. Drndic K.L. Shepard J.K. Rosenstein, M. Wanunu. Integrated nanopore sensing platform with sub-microsecond temporal resolution. *Nature Methods*, 9:487492, 2012.

[31] J. Kim, R. D. Maitra, K. Pedrotti, and W. B. Dunbar. A patch-clamp asic for nanopore-based dna analysis. *IEEE Transactions on Biomedical Circuits and Systems*, PP:1, July 2012.

[32] J. Kim, R. D. Maitra, K. Pedrotti, and W. B. Dunbar. Detecting single-abasic residues within a dna strand immobilized in a biological nanopore using an integrated cmos sensor. *Sensors and Actuators B: Chemical*, 177:10751082, February 2013.

[33] J. Li, D. Stein, C. McMullan, D. Branton, M.J. Aziz, and J.A. Golovchenko. Ion-beam sculpting at nanometre length scales. *Nature*, 412:166–9, 2001.

[34] F. Liang, S. Li, S. Lindsay, and P. Zhang. Synthesis, Physicochemical Properties, and Hydrogen Bonding of 4(5)-Substituted 1-H-Imidazole-2-carboxamide, a Potential Universal Reader for DNA Sequencing by Recognition Tunneling. *Chem. Eur. J.*, 18(19):5998–6007, 2012.

[35] K. R. Lieberman, G. M. Cherf, M. J. Doody, F. Olasagasti, Y. Kolodji, and M. Akeson. Processive replication of single DNA molecules in a nanopore catalyzed by phi29 DNA polymerase. *J. Am. Chem. Soc.*, 132(50):17961–72, 2010.

[36] Kate R Lieberman, Gerald M Cherf, Michael J Doody, Felix Olasagasti, Yvette Kolodji, and Mark Akeson. Processive replication of single DNA molecules in a nanopore catalyzed by phi29 DNA polymerase. *J Am Chem Soc*, 132(50):17961–72, Dec 2010.

[37] M. Rossi-M. Crescentini G. DAvino A. Baschirotto M. Tartagni M. Bennati, F. Thei. A sub-pa $\delta\sigma$ current amplifier for single-molecule nanosensors. *IEEE International Solid-State Circuits Conference*, page 348349, 2009.

[38] R. Maitra, J. Kim, and W. Dunbar. Recent advances in nanopore sequencing. *Electrophoresis, Special Issue: Next Generation Sequencing and Genotyping*, 33:34183428, December 2012.

[39] E. A. Manrao, I. M. Derrington, A. H. Laszlo, K. W. Langford, M. K. Hopper, N. Gillgren, M. Pavlenok, M. Niederweis, and J. H. Gundlach. Reading DNA at single-nucleotide resolution with a mutant mspa nanopore and phi29 DNA polymerase. *Nat. Biotech.*, 2012.

[40] E. A. Manrao, I. M. Derrington, M. Pavlenok, M. Niederweis, and J. Gundlach. Nucleotide discrimination with DNA immobilized in the MspA nanopore. *PloS one*, 6(10):e25723–9, 2011.

[41] Christopher A Merchant, Ken Healy, Meni Wanunu, Vishva Ray, Neil Peterman, John Bartel, Michael D Fischbein, Kimberly Venta, Zhengtang Luo, A T Charlie Johnson, and Marija Drndic. Dna translocation through graphene nanopores. *Nano. Lett.*, 10:2915–2921, 2010.

[42] F. Olasagasti, K. R. Lieberman, S. Benner, G. M. Cherf, J. M. Dahl, D. W. Deamer, and M. Akeson. Replication of individual DNA molecules under electronic control using a protein nanopore. *Nat. Nanotech.*, 5(11):798–806, 2010.

[43] H. W. C. Postma. Rapid sequencing of individual dna molecules in graphene nanogaps. *ACS Nano. Lette.*, 10:420–425, 2010.

[44] R. F. Purnell, K. K. Mehta, and J. J. Schmidt. Nucleotide identification and orientation discrimination of DNA homopolymers immobilized in a protein nanopore. *Nano. Lett.*, 8(9):3029–3034, 2008.

[45] R. F. Purnell and J. J. Schmidt. Discrimination of single base substitutions in a DNA strand immobilized in a biological nanopore. *ACS Nano.*, 3(9):2533–2538, 2009.

[46] B. E. Ragle and J. B. Wardenburg. Anti-alpha-hemolysin monoclonal antibodies mediate protection against staphylococcus aureus pneumonia. *Infect. Immun.*, 77(7):2712–2718, 2009.

[47] Bert Sakmann and Erwin Neher, editors. *Single-Channel Recording*. Plenum Press, 1995.

[48] Grégory F Schneider, Stefan W Kowalczyk, Victor E Calado, Grégory Pandraud, Henny W Zandbergen, Lieven M K Vandersypen, and Cees Dekker. DNA translocation through graphene nanopores. *Nano. Lett.*, 10(8):3163–7, Aug 2010.

[49] R. M. M. Smeets, U. F. Keyser, N. H. Dekker, and C. Dekker. Noise in solid-state nanopores. *Proc. Natl. Acad. Sci.*, 105(2):417–421, 2008.

[50] L. Song, M. Hobaugh, C. Shustak, S. Cheley, H. Bayley, and J. E. Gouaux. Structure of staphylococcal α-hemolysin, a heptameric transmembrane pore. *Science*, 274(5294):1859–1866, 1996.

[51] D. Stoddart, A. J. Heron, J. Klingelhoefer, E. Mikhailova, G. Maglia, and H. Bayley. Nucleobase recognition in ssDNA at the central constriction of the alpha-hemolysin pore. *Nano. Lett.*, 10(9):3633–7, Sep 2010.

[52] G. Oster T. Elston, H. Wang. Energy transduction in atp synthase. *Nature*, 391:510–513, January 1998.

[53] M. Tsutsui, Y. He, M. Furuhashi, S. Rahong, M. Taniguchi, and T. Kawai. Transverse electric field dragging of DNA in a nanochannel. *Sci. Rep.*, 2(394), Jan 2012.

[54] M. Tsutsui, M. Taniguchi, K. Yokota, and T. Kawai. Identifying single nucleotides by tunnelling current. *Nat. Nanotech.*, 5:286–290, Jan 2010.

[55] B. M. Venkatesan and R. Bashir. Nanopore sensors for nucleic acid analysis. *Nat. Nanotech.*, 6:615–624, 2011.

[56] B. M. Venkatesan, D. Estrada, S. Banerjee, X. Jin, V. E. Dorgan, M. Bae, N. R. Aluru, E. Pop, and R. Bashir. Stacked Graphene-Al_2O_3 Nanopore Sensors for Sensitive Detection of DNA and DNA-Protein Complexes. *ACS Nano.*, 6(1):441–450, 2012.

[57] B. M. Venkatesan, A. B. Shah, J. Zuo, and R. Bashir. DNA Sensing Using Nanocrystalline Surface-Enhanced Al2O3 Nanopore Sensors. *Adv. Mater.*, 20(8):1266–1275, 2010.

[58] Bala Murali Venkatesan, Brian Dorvel, Sukru Yemenicioglu, Nicholas Watkins, Ivan Petrov, and Rashid Bashir. Highly sensitive, mechanically stable nanopore sensors for dna analysis. *Adv. Mater.*, 21:2771–2776, 2009.

[59] W Vercoutere, S Winters-Hilt, H Olsen, D Deamer, D Haussler, and M Akeson. Rapid discrimination among individual DNA hairpin molecules at single-nucleotide resolution using an ion channel. *Nat Biotechnol*, 19(3):248–52, Mar 2001.

[60] H. Wang. Stokes efficiency of molecular motors with inertia. *Applied Mathematics Letters*, 22:7983, February 2009.

[61] H. Wang and G. Oster. Ratchets, power strokes, and molecular motors. *Appl. Phys. A.*, 75:315323, February 2002.

[62] N. Wilson, R. Abu-Shumays, B. Gyarfas, H. Wang, K. R. Lieberman, M. Akeson, and W. B. Dunbar. Electronic control of DNA polymerase binding and unbinding to single DNA molecules. *ACS Nano.*, 3(4):995–1003, 2009.

Appendix

6.1 Holder Protocols

6.1.1 Holder #1, UCSB

6.1.2 Holder #2, Stanford

1. Clean the teflon holder
   (a) Disassemble the teflon holder and boil in $HN0_3$ for 5-10 minutes
2. Clean the silicon washers
   (a) In a small beaker, fill to 5-10 mL a 1:1 ratio of H_20 to EtOH
   (b) Place two of the washers within the solution and let bathe for 10-20 minutes
   (c) Pull the washers from the bath and let dry on a clean glass slide
3. Clean the solid-state nanopore chips
   (a) Place whatever chips are to be cleaned in a 10 mL solution of $HN0_3$ and heat to a boil for 5-10 minutes
   (b) After boil, add 10 mL of H_20 and let sit for 5 minutes
   (c) Take the chips out of the solution and place onto a PDMS gel bed or glass slip temporarily
   (d) Pour out water/acid mixture and add just deionized H_20
      i. At this point, the chip should be treated with pirahna solution at 90^oC for 10-30 minutes
   (e) Place chips into the water for 5 minutes
   (f) Repeat the last three steps but fill with EtOH this time
4. Assemble the apparatus
   (a) Place one of the silicon washers in the teflon holder (where the indentation is)

(b) Place one of the solid-state chips on it with the window facing inward toward the silicon gel-ring (make sure they line up)

(c) Place the second silicon ring ontop of the silicon chip

(d) Place the second teflon piece ontop of the first and assemble with the screws

(e) Fill both wells with 50-100 μL of EtOH

(f) Place the entire apparatus in a vacuum for 5-10 minutes or until it seems there are no more bubbles surfacing

(g) Perfuse each will with salt buffer 4 times

(h) place electrodes and check for conductance

6.1.3 Holder #3, UCSC

6.2 Miscellaneous Protocols

6.2.1 Protocol - RecA bound to λDNA

- mix into a 1.5 mL microcentrifuge tube the following reagents to a final volume of 10-50 μL and to the required concentrations:

  1. 1.2 nM λDNA
  2. 6.5 μM RecA
  3. 1.5 mM ATPγs
  4. 10x RecA buffer
  5. H_20 to the final volume

- heat at 37^o C for 1 hour

- keep on ice, pipette into the nanopore *cis* chamber

Notes This protocol was based on the mixture in the Dekker paper: *Translocation of RecA-Coated Double-Stranded DNA through Solid-State Nanopores, Nano Lett. (2008) vol. 9 (9) pp. 3089-3095.* The specimens were used on a 20 nm SiO_2

nanopore. The final concentrations for the reactants when in the *cis* chamber are the following:

λDNA	1.2 nM
RecA	6.5 μM
ATPγs	1.5 mM

The RecA protein (New England Biolabs) has a molecular mass of 38.796 kDa or 38,796 g/mol. The sample purchased has a concentration of 2 mg/mL:

$$2\text{mg/mL}(38,796\text{g/mol})^{-1} = 51.55\mu\text{M} \tag{26}$$

We use the 10x RecA Buffer from New England Biolabs. It is 10x so the final volume of the reaction should be ten times the amount of 10x buffer added:

$$\text{V}_{\text{final}} = 10 \times \text{V}_{\text{10xbuffer}} \tag{27}$$

So for a final volume of 25μL the amount of 10x buffer to add is 2.5 μL.

The λDNA (New England Biolabs) purchased contained 1.25 mL and 500 μg/μL. The length of the DNA is 48.502 kbp. The average molecular weight of two nucleotides is 607.4 g/mol. The approximate molecular weight of the dsDNA:

$$500\mu\text{g}/\mu\text{L}[(48,502\text{bp})(607.4\text{g/mol}) + 157.9]^{-1} = 16.97\text{nM} \tag{28}$$

The ATPγs (Roche) is an analogue of ATP that contains a radiactive element. The substrated hinders the catalytic step in RecA's mechanism. The substrate comes in a powder form. This is for 1 mg used. The molecular weight of the substrate is 547 g/mol.

$$1\text{mg}(547\text{g/mol})^{-1}(365.6\mu\text{L}) = 5\text{mM} \tag{29}$$

6.2.2 Protocol - Biopore Preliminary Protocol

Lipid Preparation

1. Pull chloroform-suspended lipid from fridge and let thaw
2. Prepare 2 vials of ethanol (for Gilmont Pipette rinsing)
3. When lipid has thawed:
   (a) Pull ethanol with Gilmont Pipette and push out (rinsing)
   (b) Gilmont Pipette - 25 μL lipid
      i. 20μL - Vial (coating for aperture)
      ii. 5μL - Coverslips (lipid balls)
4. Dry vial in dessicator overnight or for 1 hour
5. Cover the coverslip and let sit on the benctop to dry

Aperture Preperation

1. Boil aperture in Nitric Acid (HNO_3) or some Acidic Solvent for cleaning
   (a) Rinse with H_2O 3 times
2. Prepare 2 vials of ethanol (for Gilmont Pipette rinsing)
3. Gilmont Pipette with ethanol
4. Take rinsed aperture to bench, pull H_2O, ethanol, hexane(not completely necessary) through the U-tube in that order
5. Place aperture in stage

General Station (Lipid, Buffer, Electrodes, Perfusion) Preparation

1. Resuspend dried lipid (within vial) in hexane ($\sim 20\mu L$)
2. Make Buffer (*this varies between experiments*)
   (a) pH 8

(b) 0.5 - 1M KCl (Ionic Strength)

(c) 10 mM HEPES (Buffer Strength)

(d) 1mM EDTA (Chelating Agent)

3. Fill Perfusion Device with buffer

4. Rinse electrodes with water, bathe in 1:10 bleach to water mix for 10 minutes

6.2.3 Protocol - Biopore Experiment

Bilayer Creation and Channel Insertion

1. Pull reconstituted lipid in hexane through aperture

   (a) pipette 1 μL onto the aperture (cis well)

   (b) from the trans side, connect a syringe and pull air through the U-Tube 4 times

   (c) repeat 4 times or until aperture is reltatively sticky and ready (judgement call)

2. Fill U-tube with buffer

3. Create lipid ball

4. Form bilayer

5. Pipette 0.5-1μL of 1:20 α-hemolysin (variable amount, depending on preference)

6. Wait for channel insertion

7. Perfuse $\sim 3mL$ of buffer to reduce the chance of a second insertion

8. If bilayer breaks, return to step 1 or wherever seems necessary

6.2.4 Protocol - Polyacrylamide Gel Setup

Gel Preperation

1. Make the 11% page gel, (in a 500 mL beaker)
   (a) 42g Urea
   (b) 28.7 mL of 40% acryl/bis (in the 4^o C fridge)
   (c) 10 mL 10x TBE
   (d) Bring to 100 mL with H_20
2. Sit mixture on hotplate, wrap with parafilm and place a stir rod in with speed up to 4

Glass Plate Preperation

1. Wash both glass plates with tap water, dH_20, milliQ, EtOH
2. In fume hood, use sigmacote on one side of the glass plates (300 μL and spread with kim wipe)
3. Assemble the glass plates with the spaces, comb and clips (to keep the assembly together)

Pouring Gel

1. Grab APS and TEMED from fridge
2. With a p200, pipette 65 μL of TEMED into the stirring gel mixture
3. With a p1000, pipette 650 μL of APS into the mix
4. Immediately take up 60 mL of the mix with a 60mL syringe and pour into the plates until the entire thing is full
5. Place the comb

Sample Preparation

1. Heat the heat block to a temperature between 80-100 C

2. Take urea loading buffer (8M urea, 20mM EDTA, 5mM tris pH 7):

   (a) 4.8 g urea

   (b) 400 μL 0.5M EDTA

   (c) 50 μL 1M tris pH 7

   (d) bring to 10 mL ddH_20

3. Add 130 μL of urea load to each sample

4. Denature at 90^o C for 3 min

Load Gel

1. Take polymerized gel and wash the wells with deionized water

2. Place vertically in holder

3. Fill baths with TBE 1X buffer

4. Take syringe, pull some TBE and blow away bubbles from bottom and urea from the top

5. Pipette 10 μL of dye in middle well

6. Run for 10 minutes, then pause

7. Add DNA to wells

8. Run for 10 minutes, then pause

9. Add metal plate to glass to help dissipate heat

6.3 DNA Translocation Model Simulation Code

```
%% 2nd Order Simulation of DNA translocation
%Raj Maitra
%Notes:
%Dekker, Fast Translocation through solid-state nanopore
%http://pubs.acs.org/doi/full/10.1021/nl048030d
```

```
%drag force ~ 0.3 pN
%voltage force ~ 9.4 pN

close all
clc
clear all
figure(1)

% box and nanopore boundary
boundary = [800  400;
            -400 -400]; %box boundary (so particles don't leave view)
pore_outer_y = [-15 15]; %pore boundary
pore_outer_x = [-50 0];
pore_inner_y = [-10 10];
pore_inner_x = [-30 -20];
pore_dx = 500;
pore1 = [pore_outer_y; %pore1 boundary
         pore_outer_x;
         pore_inner_y;
         pore_inner_x];
pore2 = [pore_outer_y; %pore2 boundary separated by 500 nm
         pore_outer_x+pore_dx;
         pore_inner_y;
         pore_inner_x+pore_dx];
R = diff(pore_inner_y)*10^-9; %nm
d = diff(pore_outer_x)*10^-9; %nm


%Universal Parameters
dt = 0.00001; %time step
kb = 1.3806503*10^(-23)*10^12*10^9; %boltzmann constant pN*nm/K
T = 296.15; %kelvin

%Mass
m = 5; %pg

%DNA Strand Length
```

```
43 a = 0.34*10^-9; %nm
44 Nbp = 1000; %number of base pairs
45 R_l = 10*10^-9; %spring (bond) rest length
46 N=(Nbp*a)/R_l;   %number of Kuhn sections
47 L=(Nbp*a)*10^9;    %maximum DNA Length (nm)
48
49 %Spring Force
50 k = 1000; %pN/nm spring constant of bonds
51
52 %Voltage Force
53 V = 2000*10^-3; %mV
54 ld = R_l/(2*a); %linear density
55 e = 1.6*10^-19; %coulombs
56 f_v = ld*(e*V/d)*10^12; %voltage force pN: 4*pi*(10^-3 Pa s)*(10*10^-9
       m)/(ln((10^-9 m)/(2*10^-9 m))+0.84) to pN*us/nm
57
58 %Drag Force
59 r = 1*10^-9; %nm
60 eta = 10^-3; %viscosity Pa s
61 gamma = (4*pi*eta*R_l)/(log(R_l/(2*r))+0.84)*10^9; %pN*us/nm
62
63 %Diffusion
64 D = kb*T/(gamma); %Diffusion constant nm^2/us
65
66 %Brownian Force
67 omega = sqrt(2*D*dt); %brownian motion
68
69 %Store Parameters
70 parameters = [f_v gamma omega k R_l*10^9 m];
71
72 %Initial Settings
73 position = initialize_dna(N,R_l*10^9)';
74 velocity = zeros(size(position));
75 t = 0;
76 j = 0;
77 order = 1;
78 animate_translocate(position,boundary,pore1,pore2)
```

```
79 pause
80
81 %Main Loop
82 while true%position(end,1) < 0
83     [position, velocity] = numerical_solve(position, velocity, dt,
           boundary, pore1, pore2, parameters,order);
84     t = t+dt; %total time elapsed
85     j = j+1;
86
87     if ~rem(j,100)
88         animate_translocate(position,boundary,pore1,pore2)%plot
               position at time t
89         pause%pause for a brief moment
90     end
91 end
92
93 display(['A ' num2str(Nbp) ' bp strand of DNA translocated through the
       pore in ' num2str(t) ' us.'])
```

code/main_script_translocation_simulation.m

```
1 %% Numerical Methods
2 % Runge Kutta
3 % Euler
4 function [positionf, velocityf] = numerical_solve(position, velocity,
      dt, boundary, pore1, pore2, parameters, order)
5     if order == 2
6         %step 1
7         u = position;
8         v = velocity;
9         du1 = v;
10        dv1 = dv_rk(u, v, parameters, pore1, pore2);
11
12        %step 2
13        u = u + ( dt / 2.0 ) * du1;
14        v = v + ( dt / 2.0 ) * dv1;
15        du2 = v;
16        dv2 = dv_rk(u, v, parameters, pore1, pore2);
```

```
17
18          %step 3
19          u = u + ( dt / 2.0 ) * du2;
20          v = v + ( dt / 2.0 ) * dv2;
21          du3 = v;
22          dv3 = dv_rk(u, v, parameters, pore1, pore2);
23
24          %step 4
25          u = u + dt * du3;
26          v = v + dt * dv3;
27          du4 = v;
28          dv4 = dv_rk(u, v, parameters, pore1, pore2);
29
30          %now combine the derivative estimates and compute new state
31          positionf = position + ( dt / 6 ) * ( du1 + du2*2 + du3*2 +
                du4 );
32          velocityf = velocity + ( dt / 6 ) * ( dv1 + dv2*2 + dv3*2 +
                dv4 );
33
34
35      elseif order == 1
36          runge = 0;
37          euler = 1;
38          if runge
39              %step 1
40              u = position;
41              du1 = du_rk(u, parameters, dt, pore1);
42
43              %step 2
44              u = u + ( dt / 2.0 ) * du1;
45              du2 = du_rk(u, parameters, dt, pore1);
46
47              %step 3
48              u = u + ( dt / 2.0 ) * du2;
49              du3 = du_rk(u, parameters, dt, pore1);
50
51              %step 4
```

```
52              u = u + dt * du3;
53              du4 = du_rk(u, parameters, dt, pore1);
54
55              dxy = ( dt / 6 )*( du1 + du2*2 + du3*2 + du4 );
56          end
57
58          if euler
59               u = position;
60               dxy = du_euler(u, parameters, dt, pore1, pore2);
61          end
62
63          %now combine the derivative estimates and compute new state
64          positionf = position + dxy;
65          velocityf = velocity;
66      end
67
68      [positionf velocityf] = collision_detect(position, positionf,
            velocityf, boundary, pore1);
69      [positionf velocityf] = collision_detect(position, positionf,
            velocityf, boundary, pore2);
70  end
```

code/numerical_solve.m

```
1  function dxy = du_euler(u, parameters, dt, pore1, pore2)
2      %parameters = [f_v gamma_p gamma_o omega];
3      %molecule physical parameters
4      f_v = parameters(1);
5      gamma = parameters(2);
6      omega = parameters(3);
7      k = parameters(4);
8      R_l = parameters(5);
9
10     pore_outer_y = pore1(1,:);
11     pore_outer_x = pore1(2,:);
12     pore_inner_y = pore1(3,:);
13
14     pore2_outer_y = pore2(1,:);
```

```
15      pore2_outer_x = pore2(2,:);
16      pore2_inner_y = pore2(3,:);
17
18
19      for i = 1:length(u)
20          %initialize force voltage
21          fx = 0;
22          fy = 0;
23
24          %if particle is within the electric field, add force
25          %inner pore
26          if u(i,1) < max(pore_outer_x) && u(i,1) >= min(pore_outer_x)
                && u(i,2) > min(pore_inner_y) && u(i,2) < max(pore_inner_y
                )
27              fx = f_v; %convert to pN
28          end
29
30
31          %if just a point particle
32          if length(u(:,1)) == 1
33              dxy(i,1) = (fx*dt)/gamma;
34              dxy(i,2) = (fy*dt)/gamma;
35              break
36          end
37
38          %spring length and angle
39          if i < length(u)
40              L2 = sqrt((u(i,1)-u(i+1,1))^2+(u(i,2)-u(i+1,2))^2);
41              c_theta2 = (u(i,2)-u(i+1,2))/L2;
42              s_theta2 = (u(i,1)-u(i+1,1))/L2;
43              K2 = (k)*(L2-R_l);
44          end
45
46          %if particle past the first, then add the pulling force of the
47          %previos particle
48          if i > 1
49              L1 = sqrt((u(i-1,1)-u(i,1))^2+(u(i-1,2)-u(i,2))^2);
```

```
50              c_theta1 = (u(i-1,2)-u(i,2))/L1;
51              s_theta1 = (u(i-1,1)-u(i,1))/L1;
52              K1 = (k)*(L1-R_l);
53          end
54
55
56          %deterimine dxy
57          if i == 1
58              %particle i == 1
59              dxy(i,1) = (fx*dt - K2*s_theta2*dt + omega*(min(max(randn,
                    -4), 4)))/gamma;
60              dxy(i,2) = (fy*dt - K2*c_theta2*dt + omega*(min(max(randn,
                    -4), 4)))/gamma;
61          elseif i == length(u)
62              %particle i == end
63              dxy(i,1) = (fx*dt + K1*s_theta1*dt + omega*(min(max(randn,
                    -4), 4)))/gamma;
64              dxy(i,2) = (fy*dt + K1*c_theta1*dt + omega*(min(max(randn,
                    -4), 4)))/gamma;
65          else
66              %particle 1 < i < end
67              dxy(i,1) = (fx*dt*100 + K1*s_theta1*dt - K2*s_theta2*dt +
                    omega*(min(max(randn, -4), 4)))/gamma;
68              dxy(i,2) = (fy*dt*100 + K1*c_theta1*dt - K2*c_theta2*dt +
                    omega*(min(max(randn, -4), 4)))/gamma;
69          end
70      end
71  end
```

code/du_euler.m

```
1  function [dna] = initialize_dna(N,kuhn)
2  %Generates initial position of dna
3
4    x=zeros(1,N);   % place to store x locations
5    y=zeros(1,N);   % place to store y locations
6
7    x(1)=0.0;           % initial x location
```

```
8   y(1)=0.0;            % initial y location
9   bound = 200;                      % boundary for where DNA can't pass
10
11  for i=1:N           % take N steps
12    theta = 360*rand;
13    %First three joint
14    if i*kuhn < bound
15        x(i+1)=x(i)+kuhn*cosd(180);
16        y(i+1)=y(i)+kuhn*sind(180);
17        continue
18    end
19    %[x,y] = add_angle(theta,x,y,kuhn,i);
20    if theta > 90 && theta < 180
21        theta = 180-theta;
22        %setting the angle to any number within the range
23            x(i+1)=x(i)-kuhn*cosd(theta);
24            y(i+1)=y(i)+kuhn*sind(theta);
25    elseif theta > 180 && theta < 270
26        theta = 270-theta;
27        %setting the angle to any number within the range
28            x(i+1)=x(i)-kuhn*cosd(theta);
29            y(i+1)=y(i)-kuhn*sind(theta);
30    elseif theta > 270 && theta < 360
31        theta = 360-theta;
32        %setting the angle to any number within the range
33            x(i+1)=x(i)+kuhn*cosd(theta);
34            y(i+1)=y(i)-kuhn*sind(theta);
35    else
36        %setting the angle to any number within the range
37            x(i+1)=x(i)+kuhn*cosd(theta);
38            y(i+1)=y(i)+kuhn*sind(theta);
39    end
40    if sqrt((x(i+1)+bound)^2+(y(i+1))^2) > 100 %if the strand goes
          past make the angle 180
41        if x(i+1) < -bound
42            if y(i+1) > 0
43                x(i+1)= x(i)+kuhn*cosd(theta);
```

```
44                  y(i+1)= y(i)-kuhn*sind(theta);
45              else
46                  x(i+1)= x(i)+kuhn*cosd(theta);
47                  y(i+1)= y(i)+kuhn*sind(theta);
48              end
49          else
50              if y(i+1) > 0
51                  x(i+1)= x(i)-kuhn*cosd(theta);
52                  y(i+1)= y(i)-kuhn*sind(theta);
53              else
54                  x(i+1)= x(i)-kuhn*cosd(theta);
55                  y(i+1)= y(i)+kuhn*sind(theta);
56              end
57          end
58      end
59  end
60
61
62  dna = [x;y];
63  plot(x,y);
64  hold on
65  plot(x,y,'.r');
66  xlabel('position (nm)')
67  ylabel('position (nm)')
68
69  %draw the pore
70  line([-100 -100],[10 100],'Color','k','LineWidth',2)
71  line([0 0],[10 100],'Color','k','LineWidth',2)
72  line([-80 -20],[5 5],'Color','k','LineWidth',2)
73  line([-80 -100],[5 10],'Color','k','LineWidth',2)
74  line([-20 0],[5 10],'Color','k','LineWidth',2)
75
76  line([-100 -100],[-10 -100],'Color','k','LineWidth',2)
77  line([0 0],[-10 -100],'Color','k','LineWidth',2)
78  line([-80 -20],[-5 -5],'Color','k','LineWidth',2)
79  line([-80 -100],[-5 -10],'Color','k','LineWidth',2)
80  line([-20 0],[-5 -10],'Color','k','LineWidth',2)
```

```
81   grid on
```

code/initialize_dna.m

```
1 function   [positionf velocityf] = collision_detect(position,
      positionf, velocityf, boundary, pore)
2
3     pore_outer_y = pore(1,:);
4     pore_outer_x = pore(2,:);
5     pore_inner_y = pore(3,:);
6     pore_inner_x = pore(4,:);
7
8
9     for i = 1:length(position(:,1))
10
11        %if collided with x boundary\
12        collide = find(positionf(:,1) < min(boundary(1:2,1)));
13        if collide
14            collide = positionf(:,1) < min(boundary(1:2,1));
15            new_position = min(boundary(1:2,1))-(positionf(collide,1)
                  - min(boundary(1:2,1)));
16            positionf(collide,1) = new_position;
17            velocityf(collide,1) = -velocityf(collide,1);
18        end
19
20        collide = find(positionf(:,1) > max(boundary(1:2,1)));
21        if collide
22            collide = positionf(:,1) > max(boundary(1:2,1));
23            new_position = max(boundary(1:2,1))-(positionf(collide,1)
                  - max(boundary(1:2,1)));
24            positionf(collide,1) = new_position;
25            velocityf(collide,1) = -velocityf(collide,1);
26        end
27
28        %if collided with y boundary
29        collide = find(positionf(:,2) < min(boundary(1:2,2)));
30        if collide
31            collide = positionf(:,2) < min(boundary(1:2,2));
```

```
32              new_position = min(boundary(1:2,2))-(positionf(collide,2)
                    - min(boundary(1:2,2)));
33              positionf(collide,2) = new_position;
34              velocityf(collide,2) = -velocityf(collide,2);
35          end
36
37          collide = find(positionf(:,2) > max(boundary(1:2,2)));
38          if collide
39              collide = positionf(:,2) > max(boundary(1:2,2));
40              new_position = max(boundary(1:2,2))-(positionf(collide,2)
                    - max(boundary(1:2,2)));
41              positionf(collide,2) = new_position;
42              velocityf(collide,2) = -velocityf(collide,2);
43          end
44
45          %if collided with nanopore
46          %if past left side of pore
47          for i = 1:length(position(:,1))
48              if positionf(i,1) > min(pore_outer_x) && positionf(i,1) <
                    max(pore_outer_x) && positionf(i,2) < min(pore_inner_y
                    )
49                  %we know it collided with the lower pore part
50                  %if collided with left side of pore
51                  if positionf(i,2) < min(pore_outer_y)
52                      if position(i,1) < min(pore_outer_x)
53                          new_position = min(pore_outer_x)-(positionf(i
                                ,1) - min(pore_outer_x));
54                          positionf(i,1) = new_position;
55                          velocityf(i,1) = -velocityf(i,1);
56                      %if collided with right side of pore
57                      elseif position(i,1) > max(pore_outer_x)
58                          new_position = max(pore_outer_x)-(positionf(i
                                ,1) - max(pore_outer_x));
59                          positionf(i,1) = new_position;
60                          velocityf(i,1) = -velocityf(i,1);
61                      end
62
```

```
63                  %if collided with angled bottom left side of pore
64                  elseif positionf(i,1) < min(pore_inner_x) && positionf
                        (i,2) < -(pore_outer_y(1,2)-(1/4)*(positionf(i,1)-
                        pore_outer_x(1,1)))
65                      dy = abs(positionf(i,2)-((1/4)*(positionf(i,1)-
                            pore_outer_x(1,1))-pore_outer_y(1,2)));
66                      dx = abs(positionf(i,1)-((4)*(positionf(i,2)+
                            pore_outer_y(1,2))+pore_outer_x(1,1)));
67                      theta = 2*(180+26.565);
68                      new_positionY = ((1/4)*(positionf(i,1)-
                            pore_outer_x(1,1))-pore_outer_y(1,2)) + dy;
69                      new_positionX = ((4)*(positionf(i,2)+pore_outer_y
                            (1,2))+pore_outer_x(1,1)) + dx;
70                      positionf(i,2) = new_positionY; %update y
71                      positionf(i,1) = new_positionX; %update x
72                      newvelocityX = velocityf(i,1)*cosd(theta)+
                            velocityf(i,2)*sind(theta);
73                      newvelocityY = velocityf(i,1)*sind(theta)-
                            velocityf(i,2)*cosd(theta);
74                      velocityf(i,1) = newvelocityX;
75                      velocityf(i,2) = newvelocityY;
76
77
78                  %if collided with angled bottom right side of pore
79                  elseif positionf(i,1) > max(pore_inner_x) && positionf
                        (i,2) < (pore_inner_y(1,1)-(1/4)*(positionf(i,1)-
                        pore_inner_x(1,2)))
80                      dy = abs(positionf(i,2)-(-pore_inner_y(1,2)-(1/4)
                            *(positionf(i,1)-pore_inner_x(1,2))));
81                      dx = abs(positionf(i,1)+(-pore_inner_x(1,2)+(4)*(
                            positionf(i,2)+pore_inner_y(1,2))));
82                      theta = 2*(180-26.565);
83                      new_positionY = (-pore_inner_y(1,2)-(1/4)*(
                            positionf(i,1)-pore_inner_x(1,2))) + dy;
84                      new_positionX = -(-pore_inner_x(1,2)+(4)*(
                            positionf(i,2)+pore_inner_y(1,2))) - dx;
85                      positionf(i,2) = new_positionY; %update y
```

```
86                      positionf(i,1) = new_positionX; %update x
87                      newvelocityX = velocityf(i,1)*cosd(theta)+
                            velocityf(i,2)*sind(theta);
88                      newvelocityY = velocityf(i,1)*sind(theta)-
                            velocityf(i,2)*cosd(theta);
89                      velocityf(i,1) = newvelocityX;
90                      velocityf(i,2) = newvelocityY;
91                  %if collided with inside bottom side of pore
92                  elseif position(i,2) > min(pore_inner_y) && position(i
                        ,1) < pore_inner_x(1,2) && position(i,1) >
                        pore_inner_x(1,1)
93                      new_position = min(pore_inner_y)-(positionf(i,2) -
                             min(pore_inner_y));
94                      positionf(i,2) = new_position;
95                      velocityf(i,2) = -velocityf(i,2);
96                  end
97              elseif positionf(i,1) > min(pore_outer_x) && positionf(i
                    ,1) < max(pore_outer_x) && positionf(i,2) > max(
                    pore_inner_y)
98                  %we know it collided with the top pore part
99                  %if collided with left side of pore
100                 if positionf(i,2) > max(pore_outer_y)
101                     if position(i,1) < min(pore_outer_x)
102                         new_position = min(pore_outer_x)-(positionf(i
                                ,1) - min(pore_outer_x));
103                         positionf(i,1) = new_position;
104                         velocityf(i,1) = -velocityf(i,1);
105                     %if collided with right side of pore
106                     elseif position(i,1) > max(pore_outer_x)
107                         new_position = max(pore_outer_x)-(positionf(i
                                ,1) - max(pore_outer_x));
108                         positionf(i,1) = new_position;
109                         velocityf(i,1) = -velocityf(i,1);
110                     end
111
112
113
```

```
114                 %if collided with angled top left side of pore
115                 elseif positionf(i,1) < min(pore_inner_x) && positionf
                        (i,2) > (pore_outer_y(1,2)-(1/4)*(positionf(i,1)-
                        pore_outer_x(1,1)))
116                     dy = abs(positionf(i,2)-(pore_outer_y(1,2)-(1/4)*(
                            positionf(i,1)-pore_outer_x(1,1))));
117                     dx = abs(positionf(i,1)+(-pore_outer_x(1,1)+(4)*(
                            positionf(i,2)-pore_outer_y(1,2))));
118                     theta = 2*(180-26.565);
119                     new_positionY = (pore_outer_y(1,2)-(1/4)*(
                            positionf(i,1)-pore_outer_x(1,1))) - dy;
120                     new_positionX = -(-pore_outer_x(1,1)+(4)*(
                            positionf(i,2)-pore_outer_y(1,2))) + dx;
121                     positionf(i,2) = new_positionY; %update y
122                     positionf(i,1) = new_positionX; %update x
123                     newvelocityX = velocityf(i,1)*cosd(theta)+
                            velocityf(i,2)*sind(theta);
124                     newvelocityY = velocityf(i,1)*sind(theta)-
                            velocityf(i,2)*cosd(theta);
125                     velocityf(i,1) = newvelocityX;
126                     velocityf(i,2) = newvelocityY;
127
128
129
130                 %if collided with angled top right side of pore
131                 elseif positionf(i,1) > max(pore_inner_x) && positionf
                        (i,2) > (pore_inner_y(1,2)+(1/4)*(positionf(i,1)-
                        pore_inner_x(1,2)))
132                     dx = abs(positionf(i,1)-(pore_inner_x(1,2)+(4)*(
                            positionf(i,2)-pore_inner_y(1,2))));
133                     dy = abs(positionf(i,2)-(pore_inner_y(1,2)+(1/4)*(
                            positionf(i,1)-pore_inner_x(1,2))));
134                     theta = 2*(180+26.565);
135                     new_positionY = (pore_inner_y(1,2)+(1/4)*(
                            positionf(i,1)-pore_inner_x(1,2))) - dy;
136                     new_positionX = (pore_inner_x(1,2)+(4)*(positionf(
                            i,2)-pore_inner_y(1,2))) - dx;
```

```
137                     positionf(i,2) = new_positionY; %update y
138                     positionf(i,1) = new_positionX; %update x
139                     newvelocityX = velocityf(i,1)*cosd(theta)+
                            velocityf(i,2)*sind(theta);
140                     newvelocityY = velocityf(i,1)*sind(theta)-
                            velocityf(i,2)*cosd(theta);
141                     velocityf(i,1) = newvelocityX;
142                     velocityf(i,2) = newvelocityY;
143                 %if collided with inside top side of pore
144                 elseif position(i,2) < max(pore_inner_y) && position(i
                        ,1) < pore_inner_x(1,2) && position(i,1) >
                        pore_inner_x(1,1)
145                     new_position = max(pore_inner_y)-(positionf(i,2) -
                             max(pore_inner_y));
146                     positionf(i,2) = new_position;
147                     velocityf(i,2) = -velocityf(i,2);
148                 end
149             end
150         end
151     end
```

code/collision_detect.m

```
1 function  animate_translocate(position,boundary,pore1,pore2)
2     %initialize drawing window
3     figure(1)
4     clf
5
6       %subplot(3,2,[1 3]);
7         %[pvolumex pvolumey] = circle(position,r);
8         plot(position(:,1),position(:,2), 'r.')
9         hold on
10        plot(position(:,1),position(:,2),'b')
11        %plot(position(:,1),position(:,2),'b','LineWidth',2)
12
13        %hold on
14        %plot(pvolumex,pvolumey,'b.')
15        grid on
```

```
16          axis([min(boundary(:,1)) max(boundary(:,1)) min(boundary(:,2))
                 max(boundary(:,2))])
17          %axis([-125 15 -25 20]);
18          %axis([position(1,1)-20 position(1,1)+20 position(1,2)+20
                position(1,2)+20]);
19          xlabel('Position (nm)')
20          ylabel('Position (nm)')
21          title('Nanopore Translocation Simulation')
22          axis([-300 50 -200 200])
23          pore_color = 'k';
24
25          %draw the pore1
26          pore_outer_y = pore1(1,:);
27          pore_outer_x = pore1(2,:);
28          pore_inner_y = pore1(3,:);
29          pore_inner_x = pore1(4,:);
30          width =1;
31          line([pore_outer_x(1,1) pore_outer_x(1,1)],[pore_outer_y(1,2)
                max(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
32          line([pore_outer_x(1,2) pore_outer_x(1,2)],[pore_outer_y(1,2)
                max(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
33          line(pore_inner_x,[pore_inner_y(1,2) pore_inner_y(1,2)],'Color
                ',pore_color,'LineWidth',width)
34          line([pore_inner_x(1,1) pore_outer_x(1,1)],[pore_inner_y(1,2)
                pore_outer_y(1,2)],'Color',pore_color,'LineWidth',width)
35          line([pore_inner_x(1,2) pore_outer_x(1,2)],[pore_inner_y(1,2)
                pore_outer_y(1,2)],'Color',pore_color,'LineWidth',width)
36
37          line([pore_outer_x(1,1) pore_outer_x(1,1)],[pore_outer_y(1,1)
                min(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
38          line([pore_outer_x(1,2) pore_outer_x(1,2)],[pore_outer_y(1,1)
                min(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
```

```
39          line(pore_inner_x,[pore_inner_y(1,1) pore_inner_y(1,1)],'Color
                ',pore_color,'LineWidth',width)
40          line([pore_inner_x(1,1) pore_outer_x(1,1)],[pore_inner_y(1,1)
                pore_outer_y(1,1)],'Color',pore_color,'LineWidth',width)
41          line([pore_inner_x(1,2) pore_outer_x(1,2)],[pore_inner_y(1,1)
                pore_outer_y(1,1)],'Color',pore_color,'LineWidth',width)
42
43          %draw the pore2
44          pore_outer_y = pore2(1,:);
45          pore_outer_x = pore2(2,:);
46          pore_inner_y = pore2(3,:);
47          pore_inner_x = pore2(4,:);
48          width =1;
49          line([pore_outer_x(1,1) pore_outer_x(1,1)],[pore_outer_y(1,2)
                max(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
50          line([pore_outer_x(1,2) pore_outer_x(1,2)],[pore_outer_y(1,2)
                max(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
51          line(pore_inner_x,[pore_inner_y(1,2) pore_inner_y(1,2)],'Color
                ',pore_color,'LineWidth',width)
52          line([pore_inner_x(1,1) pore_outer_x(1,1)],[pore_inner_y(1,2)
                pore_outer_y(1,2)],'Color',pore_color,'LineWidth',width)
53          line([pore_inner_x(1,2) pore_outer_x(1,2)],[pore_inner_y(1,2)
                pore_outer_y(1,2)],'Color',pore_color,'LineWidth',width)
54
55          line([pore_outer_x(1,1) pore_outer_x(1,1)],[pore_outer_y(1,1)
                min(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
56          line([pore_outer_x(1,2) pore_outer_x(1,2)],[pore_outer_y(1,1)
                min(boundary(1:2,1))],'Color',pore_color,'LineWidth',width
                )
57          line(pore_inner_x,[pore_inner_y(1,1) pore_inner_y(1,1)],'Color
                ',pore_color,'LineWidth',width)
58          line([pore_inner_x(1,1) pore_outer_x(1,1)],[pore_inner_y(1,1)
                pore_outer_y(1,1)],'Color',pore_color,'LineWidth',width)
```

```
59          line([pore_inner_x(1,2) pore_outer_x(1,2)],[pore_inner_y(1,1)
                pore_outer_y(1,1)],'Color',pore_color,'LineWidth',width)
60
61 end
```

code/animate_translocate.m