NANOPORE-BASED READOUT OF ENCODED DNA NANOSTRUCTURES

Niklas Alexander Ermann

Cavendish Laboratory, Department of Physics
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

Trinity Hall

May 2020

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Niklas Alexander Ermann
May 2020

Abstract

Nanopore-Based Readout of Encoded DNA Nanostructures

Niklas Alexander Ermann

Nanopore sensing is a powerful technique for the detection and study of single molecules. The underlying resistive pulse principle relies on molecules modulating an ionic current as they pass through a nanoscale constriction between two reservoirs. The technique has recently been extended to the detection of modifications attached along a single molecule composed of a double strand of DNA. As the strand threads through the nanopore, attachments produce secondary peaks in the already reduced current. The ability to encode information in modification sequences and self-assemble designed strands through DNA nanotechnology opens up promising possibilities ranging from highly parallel single-molecule sensing to DNA data storage. This work investigates experimental methods and analysis techniques to improve the readout of information contained in the position and type of modifications attached along the strand.

Experimentally, it is found that fluid flows opposing the movement of DNA strongly promote single-file, unfolded entry into conical glass nanopores. The effect can be induced both with flows due to externally applied pressure and electro-osmotic flows (EOF) along the pore walls. The ability to prevent folded entry into the pore is crucial for decoding of attached modifications as it facilitates the identification and classification of secondary current peaks.

In the investigation of analysis methods, Bayesian inference is assessed as a tool to extract information from the current signal. While useful to detect a strand's folding state, it is found that deep learning-based approaches exhibit superior accuracy where enough training data is available. Two convolutional neural network (CNN) architectures are presented, the first one targeted at the classification of previously known modification sequences and the second one for the identification of modifications at arbitrary positions.

The last chapter shows that the deep learning approach is capable of classifying translocation events in real time. A probability calculation reveals how many events are required for a measurement to conclusively show the presence of a certain molecule. Re-measuring molecules by driving them in and out of the nanopore is investigated as a method to improve modification decoding. The thesis concludes with a critical discussion of the presented techniques and their contribution to the development of sensing and data storage applications built on nanopore-based readout of structural modifications on DNA strands.

Für meine Eltern

Acknowledgements

A PhD takes a while, and once the thesis is written you sit down and wonder how you have made it to the end. Like with any long journey, you realise most of the credit goes to the people who have helped you along the way. To stick with the travelling metaphor, without their support you might make it to Aktau, but good luck getting on the ferry. A single page can never be enough to thank everyone who has helped me get on the ferry. Yet it seems preferable to thank some and hope for the understanding of those omitted rather than say nothing at all.

My thanks go to my supervisor Prof Ulrich Keyser for his support and advice over the last three and a half years. Dr Kaikai Chen and Dr Jinglin Kong introduced me to the wonderful world of nanopore fabrication and chip making. With Dr Karolis Misiunas I discovered deep learning and a shared enthusiasm for the latest tech fad. Nikita Hanikel taught me chemistry, Vivian Wang how to build a dataset. Dr Nicole Weckman offered sensor advice and a sympathetic ear for a scientist's troubles. Dr Jinbo Zhu trusted me with his unpublished data and tirelessly answered all of my questions. Office 1.92 has been a constant source of support and amusement, thank you Michael, Diana, Jinbo, Marcus, Karolis and all temporary inhabitants. The last three years would not have been the same without table football; Alex, Jeff, Jehangir, Kareem, Karolis, Michael and Vlad may your reflexes grow slow and your wrists weak. It has been a joy to be part of the KeyserLab Class of 2020, thank you Jeff, Michael, Mustafa, Raj and Will for making it a memorable journey. Science is a dish best served with financial support, I thank Trinity Hall, the Cambridge Trust and the EPSRC for PhD funding.

Last but not least I am eternally grateful to my friends, who keep my feet on the ground and my head pointing in the right direction. And to my family, without whom I would not have made it this far.

Publications

Some results in this thesis have appeared in the following publications:

Ermann, N., Hanikel, N., Wang, V., Chen, K., Weckman, N. E. & Keyser, U. F. Promoting Single-File DNA Translocations through Nanopores Using Electro-Osmotic Flow. *The Journal of Chemical Physics* **149,** 163311 (2018)

Ermann, N., Chen, K. & Keyser, U. F. Bayesian Inference for Nanopore Data Analysis. arXiv: 1904.01040 (2019)

Chen, K., Kong, J., Zhu, J., Ermann, N., Predki, P. & Keyser, U. F. Digital Data Storage Using DNA Nanostructures and Solid-State Nanopores. *Nano Letters* **19,** 1210–1215 (2019)

Misiunas, K., Ermann, N. & Keyser, U. F. QuipuNet: Convolutional Neural Network for Single-Molecule Nanopore Sensing. *Nano Letters* **18,** 4040–4045 (2018)

Not included in this thesis:

Wang, V.*, Ermann, N.* & Keyser, U. F. Current Enhancement in Solid-State Nanopores Depends on Three-Dimensional DNA Structure. *Nano Letters* **19**, 5661–5666 (2019)
* equal contributions

Weckman, N. E., Ermann, N., Gutierrez, R., Chen, K., Graham, J., Tivony, R., Heron, A. & Keyser, U. F. Multiplexed DNA Identification Using Site Specific dCas9 Barcodes and Nanopore Sensing. *ACS Sensors* **4,** 2065–2072 (2019)

Table of contents

Chapter 1

Introduction

In 1953, the US Patent Office granted Wallace H. Coulter a patent for his "invention [relating] to means for detecting and counting particles suspended in a fluid medium" [6]. His idea was to set up an ion flow through a small constriction which would be modulated by passing particles. The number of modulations in the recorded current would be a direct measure for the number of particles. The Coulter counter is widely used in the quantification of physiological particles such as blood cells and the underlying principle has become known as resistive pulse sensing [7].

One year earlier, Alan L. Hodgkin and Andrew F. Huxley had published their findings on trans-membrane ionic currents in giant squid axons [8–10]. Their technique, termed voltage clamping, allowed the measurement of small variations in current while maintaining a constant voltage across a cellular membrane. Their work is now considered seminal in the study of signal propagation along neurons. Thirty years later, Hladky & Haydon [11] demonstrated that the voltage clamp could be extended to detect the formation of individual pores in an artificial lipid membrane due to gramicidin dimers. Building on this technique, Erwin Neher and Bert Sakmann detected the opening state of a single ion channel in a cellular membrane [12]. Referred to as patch clamping, the improved approach relied on sealing a small membrane region with the tip of a glass pipette. Ionic currents on the scale of picoamperes could be detected through an individual nanometre-sized channel. Both Hodgkin & Huxley and Neher & Sakmann would eventually receive the Nobel Prize for their discoveries.

The scene seemed set for a combination of patch clamping and resistive pulse sensing: Neher and Sakmann had shown that a picoamp current could be set up and measured across nanoscale pores, while the Coulter counter demonstrated the detection of particles by current blockades. Yet it would take another twenty-odd years until the idea of nanopore sensing truly took hold. In 1996, Kasianowicz *et al.* [13] showed the detection of single-stranded

DNA (ssDNA) passing through the biological α-hemolysin channel. The translocating DNA molecules produced characteristic drops in the current trace which could be related to their length. While previous studies had inserted polymers into biological channels, they had done so to characterise the channel itself based on conductance and noise changes [14–16]. Kasianowicz *et al.* [13] returned to Coulter's original resistive pulse sensing idea and demonstrated that the same principle could be applied to detect molecules on the nanoscale.

The authors recognised the technique's revolutionary potential for DNA sequencing: if the current signal from a translocating DNA molecule could be correlated with individual bases, the sequencing of arbitrarily long strands could be achieved in a comparatively simple system. After years of development and commercialisation, this goal has been reached by Oxford Nanopore Technologies' MinION sequencer [17–20]. It must be noted that an enzymatic motor at the entrance to a modified biological pore threads the DNA strand through the pore in order to slow down the translocation speed [21], a significant alteration to the simple original idea of driving molecules through a pore electrophoretically. Nevertheless, the MinION device has become a successful sequencing platform used in a wide range of fields.

While the promise of long-read sequencing has been a key driver for the development of nanopore sensing, the flexibility of the resistive pulse method has spawned research into various sensing platforms and target molecules. They are unified by the underlying sensing principle of detecting changes in an ionic current as envisioned by Wallace H. Coulter for much larger molecules. As shown in Figure 1.1, a nanometre-sized constriction separates two reservoirs filled with electrolyte solution. Applying a voltage across the pore sets up an ionic current through the pore. The electric field electrophoretically drives charged analytes through the constriction, each passage produces a characteristic change in the current. A passage is commonly called a translocation. The current signal depends on the size, charge and surface properties of the analyte.

Based on this idea a range of nanopore sensing platforms have emerged. The biological channels used in sequencing and Kasianowicz et al.'s proof-of-concept have been complemented by pores in solid-state membranes. Early attempts relied on inserting gold nanotubules into openings in filter membranes to detect analyte concentrations from changes in conductivity [22, 23]. While nanoscale openings could be fabricated successfully, the technique did not allow addressing individual pores for single-molecule detection. Sensing in a single solid-state pore was first achieved by Li *et al.* [24], who fabricated a 5 nm pore in a Si_3N_4 membrane with a focused ion beam and observed translocations of double-stranded DNA (dsDNA). Storm *et al.* [25] simplified the process by drilling pores with electron-beam lithography. Fabrication in a transmission electron microscope (TEM) became the standard

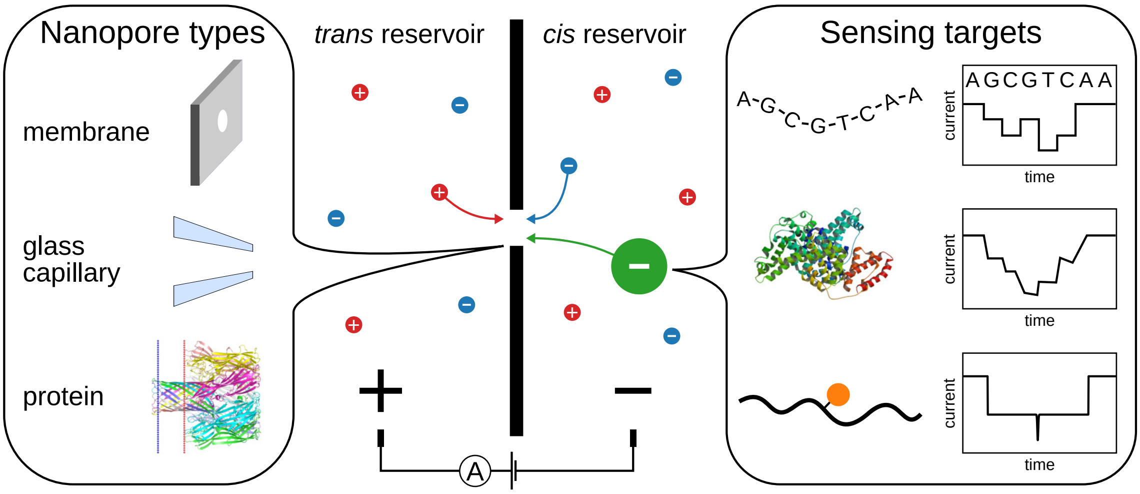


Fig. 1.1 Nanopore sensing. Applying a voltage across a constriction separating two electrolyte-filled reservoirs sets up a ion current, the passage of analytes (green circle) produces characteristic current drops. Solid-state pores consist of a hole in a silicon-based 2D membrane (upper left) or a nanoscale opening at the tip of a laser-pulled glass capillary (middle left). Biological nanopores are found in membrane channels such as α-hemolysin* (lower left). DNA sequencing has long been a central goal of nanopore detection, based on characteristic current levels for each base in the pore (upper right). The flexibility of the technique, however, allows a much wider sensing scope, for example protein characterisation (middle right) or the localisation of attachments along DNA strands (lower right).

*Alpha-hemolysin (https://opm.phar.umich.edu/proteins/28) from the OPM database, Lomize group, College of Pharmacy, University of Michigan. Licensed under CC BY 3.0 (https://creativecommons.org/licenses/by/3.0/)

for pores in silicon-based membranes until Kwok *et al.* [26] reported that a high voltage can induce dielectric breakdown, offering a direct method of producing nanometre-sized openings [27, 28]. An even simpler alternative to 2D membranes is the use of glass capillaries as pioneered by Steinbock *et al.* [29]. Using electrophysiology tools intended for the fabrication of glass pipettes for patch clamping, a glass cylinder with a sub-millimetre inner diameter is heated in a central spot and pulled apart to produce two separate pieces with conical tips at the separation point. Through careful tuning of the pulling parameters, the opening at the tip can be shrunk to diameters down to 5 nm [3]. This nanopore fabrication method is inexpensive and simple compared to membrane pores, but offers less control over the pore geometry. Another intriguing possibility is the insertion of self-assembled DNA nanostructures into a solid-state pore to create hybrid channels [30–32]. Although the approach offers nanoscale precision in the design of the DNA structure, the increased noise associated with fluctuations of the non-rigid DNA pore has to date prevented widespread use in sensing applications.

Similarly to the emergence of various nanopore types, the scope of detectable analytes has continuously expanded over the last two decades. The general nature of resistive pulse sensing means that the only requirements for a molecular target are that it can pass through the pore and modulate the ion current in doing so. The initial proof-of-concepts with single- and double-stranded DNA were complemented by the direct detection of translocating proteins. Han *et al.* [33] observed the passage of bovine serum albumin (BSA) molecules through a 55 nm membrane pore. Further work extended experiments to different protein species and pore types [34, 35]. In a study introducing location-specific detection of attachments along a double strand of DNA, Kowalczyk *et al.* [36] identified the binding sites of RecA proteins, building on previous work limited to detecting the presence of RecA [37]. The position of binding sites was revealed by secondary reductions in current below the dsDNA level, see the bottom right current trace in Figure 1.1 for an illustration. The idea of relating secondary current drops to the location of attachments opened up a range of sensing possibilities. Firstly, bound structures can mark certain base pair sequences as a less complex alternative to full sequencing. This has been demonstrated through the binding of peptide nucleic acids (PNAs) [38], biotin-transferring methyltransferases [39, 40] and dCas9 proteins [5, 41]. Secondly, DNA can act as a 'carrier' to move analytes through the pore, allowing specific detection through targeted binding sites even if direct sensing is not possible, e.g. due to a low protein charge or small exclusion volume [42–46]. Thirdly, the use of DNA nanotechnology allows the attachment of rationally designed structures at arbitrary positions along the DNA strand. Pioneered by Rothemund [47] as 'DNA origami', the technique relies on the predictability of Watson-Crick base-pairing to self-assemble structures on the nanoscale [48].

DNA nanotechnology offers great flexibility in the design of DNA carriers for nanopore sensing. Plesa *et al.* [49] used the method to attach a protrusion to a double strand of DNA to study its translocation velocity based on the position of a secondary current drop as envisioned by Wanunu [50]. A crucial insight is that the position of attachments encodes information in the DNA structure. Nanopores then allow rapid readout of the contained modification sequence. Bell & Keyser [51] used this idea to devise an encoding scheme comprised of 5 attachments in one half of a DNA carrier, Figure 1.2 illustrates the approach. A series of dumbbell DNA attachments represents a '1' bit, the absence of any modifications indicates a '0' bit. The two outer positions are always present and serve as markers to frame the information-encoding region. This leaves 3 bits which can be switched on or off, resulting in $2^3 = 8$ possible 'barcodes'. The idealised current traces in Figure 1.2 show the expected readout from three example sequences containing the '000', '110' and '101' barcodes. An important advantage of this encoding scheme is its scalability. Each additional modification doubles the number of distinct DNA carriers, allowing the design of large carrier libraries. In their proof-of-concept for 3 changeable bits, Bell & Keyser [51] demonstrated the technique's potential as a single-molecule sensing platform for the detection of a large number of targets. They uniquely associated each barcode with a binding site for a specific analyte in the second half of the carrier. The current trace during translocation through a nanopore then contained a peak sequence revealing the barcode as well as an additional peak indicating the presence or absence of an analyte. Due to the unique mapping between barcode and binding site, this allows the simultaneous detection of many molecular species at the single-molecule level.

Encoding information in DNA structure for readout with nanopores has since formed the basis for applications beyond sensing. Chen *et al.* [3] demonstrated a data storage scheme based on 56 differently sized DNA overhangs along a DNA carrier, increasing the information density to roughly one bit every 25 nm. This was later extended to a read/write system with basic encryption of the stored data [52]. The crucial requirement for both the information storage and sensing applications is the ability to infer the position and type of attachments from the nanopore current signal. Readout of the information-encoding modification sequence is only possible if secondary current peaks can be decoded reliably. The success of this decoding depends both on the quality of the current signal and on the subsequent data processing. This work investigates the two paths, presenting experimental methods as well as analysis techniques to improve the readout of modifications sequences along a DNA strand. Chapter 3 discusses the effect of fluid flows in glass capillary nanopores on the conformation of a translocating strand and how experimental parameters affect those flows. Chapter 4 presents Bayesian inference as an analysis method for nanopore data. Chapter 5 investigates deep learning methods to decode DNA structure, presenting two

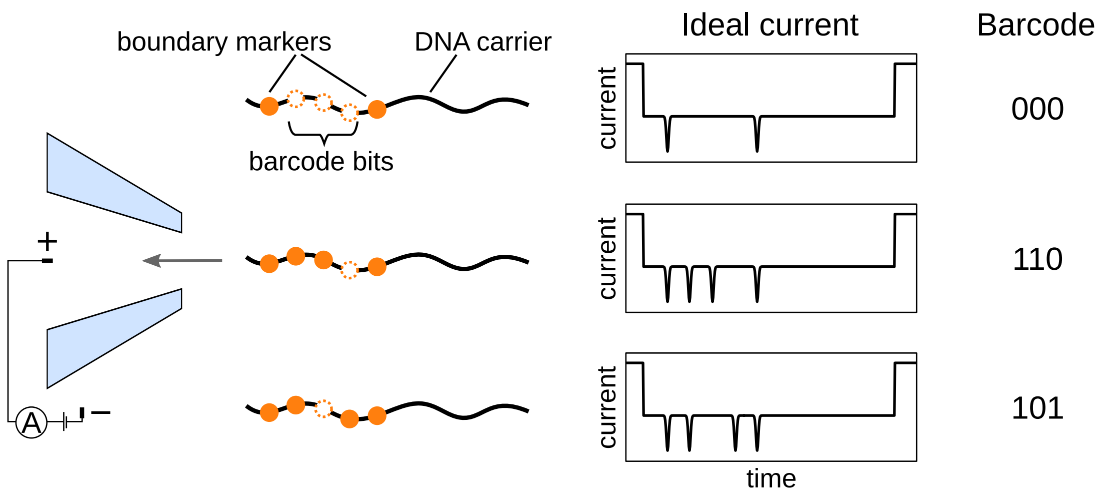


Fig. 1.2 Encoding information in DNA structure. In the scheme introduced by Bell & Keyser [51], the presence and absence of attachments along a DNA carrier strand represent a '1' and '0' bit respectively. Translocation through a nanopore produces the idealised current traces shown in the second column, with each peak corresponding to a modification attached to the strand. Three changeable positions allow the creation of $2^3 = 8$ distinct DNA carriers or 'barcodes', each additional attachment doubles this number. By uniquely associating each barcode with an analyte binding site in the second half of the strand, the authors demonstrated the feasibility of this approach as a sensing platform for a highly scalable number of targets.

convolutional neural network (CNN) implementations for the processing of current signals. Chapter 6 outlines the use of these analysis methods in real-time single-molecule detection. The thesis concludes with a summary of the results and a discussion of the missing pieces for the use of nanopore-based decoding in sensing and storage applications. Before moving on to results, the next chapter outlines the experimental methods and materials used in this work.

Chapter 2

Materials & Methods

2.1 Nanopores

The nanopores used in this work are fabricated from commercially available glass capillaries with an inner diameter of 0.2 mm and an outer diameter of 0.5 mm and a filament running along the inner wall (Sutter Instrument, CA, US). One capillary at a time is fastened into a pipette puller (P-2000, Sutter Instrument, CA, US), whose laser heats up a central spot while simultaneously pulling on either end. The glass in the laser focus melts and becomes malleable while the pulling continues, until the capillary eventually splits into two parts. The malleability of the glass at the central breaking point means that the resulting ends are conically shaped, narrowing towards a fine tip. With the correct pulling parameters, a nanoscale hole can be maintained at the tip during the breaking process, constituting a nanopore. While the tip becomes more likely to melt into a closed surface for smaller holes, this fabrication method has been shown to produce pore diameters down to 5 nm [3].

In addition to fabrication considerations, the desired pore diameter depends on the sensing goal. Pores much larger than the target molecule fail to resolve detailed features, since the signal-to-noise ratio decreases with a smaller relative current change due to a molecule in the pore. Pores approaching the size of the target, on the other hand, are more prone to clogging and reduce the capture rate of target molecules. For the DNA sensing experiments in this work, I found pore diameters between 10 nm and 15 nm to exhibit desirable characteristics in terms of resolution, lifetime and fabrication success. The pore size was determined by measuring a current-voltage curve prior to adding the measurement sample. The resulting resistance value can be plugged into analytical models to obtain an estimate of the diameter. Scanning electron microscope (SEM) images have revealed a double-taper geometry at the

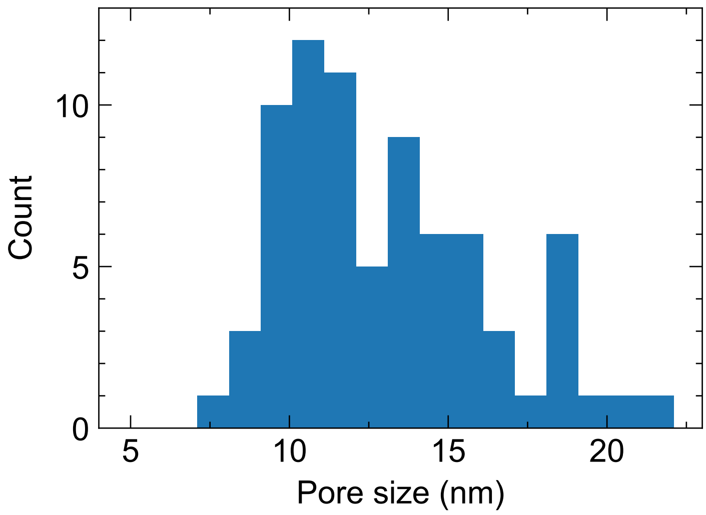


Fig. 2.1 Diameters of nanopores used in this work, determined from current-voltage curves and a resistance model. The pipette puller settings in table 2.1 produced a distribution of pore sizes with the majority of diameters in the target range between 10 nm and 15 nm.

Heat	Filament	Velocity	Delay	Pull
480	0	25	170	200

Table 2.1 Pipette puller (P-2000, Sutter Instrument, CA, US) settings to produce nanopores with a diameter between 10 nm and 15 nm.

tip of nanopores pulled from glass capillaries, with a steeper taper angle just before the opening [42]. The resistance model for this non-trivial shape is discussed in appendix A.

The pulling process produces a distribution of diameters and Figure 2.1 shows a histogram of all the pores used in this work. While there is considerable spread, most diameters lie in the target range between 10 nm and 15 nm. Table 2.1 shows the pipette puller parameters used to fabricate the pores.

2.2 *Experimental setups*

To be used for resistive pulse sensing, the glass capillaries with a nanopore at one end must be positioned so that each end sits in a separate reservoir, with the capillary as the only connection between the two chambers. I used two different experimental setups in this work, as described in the following.

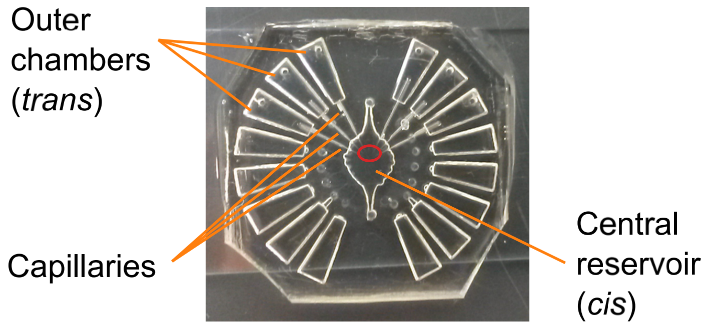


Fig. 2.2 Polydimethylsiloxane (PDMS) chip with glass nanopores. The pores sit at the tip of glass capillaries in the middle of the central (*cis*) reservoir (region marked by red circle). The chip holds up to 16 capillaries which connect the radially placed outer (*trans*) chambers with the central reservoir. In the chip shown here only the upper 6 channels are fitted with capillaries. Electrodes are connected through the holes in the PDMS layer.

2.2.1 Chip setup

In an approach introduced by Bell *et al.* [53], up to 16 capillaries are placed in a radial pattern in a polydimethylsiloxane (PDMS) chip. The nanopore tips sit in a central reservoir while 16 separate outer chambers house the capillaries' blunt ends. Figure 2.2 shows an example chip with the upper 6 channels occupied. The PDMS is prepared in its liquid state by mixing the base with the curing agent (Sylgard 184 silicone elastomer kit, The Dow Chemical Company, MI, US) at a 10:1 ratio and pouring it into a 3D-printed mould. Baking in a 60 °C oven for 2 h solidifies the PDMS and raw chips can be cut out of the mould. After removing rough edges, 1 mm holes are punched into the reservoirs for fluid access, as well as each channel connecting the central reservoir with the outer chambers. The channel holes allow flowing liquid PDMS into the channel to create a tight seal once the capillaries have been positioned. The capillaries are cut to fit into the chip and placed in the channels with the nanopore tip in the central reservoir. A glass slide serves as the base to which we attach the PDMS chip, it is cleaned with ethanol and dried under a nitrogen stream before use. Both the glass slide and the PDMS are placed in a plasma cleaner (Femto, Diener electronic, Germany) for 10 s and subsequently pressed together which firmly bonds the PDMS to the glass. Since the capillaries loosely sit in the channels, fluid can still move between the reservoirs outside the nanopores. To create a tight seal we flow liquid PDMS into the previously punched channel holes and place the bonded PDMS/glass chip onto a hot plate for 10 min at 120 °C for curing. Once prepared and covered with scotch tape the chips can be stored for later use.

To prepare a chip for measurement it is placed in a plasma cleaner for 2 min to make the inner surfaces hydrophilic. Measurement buffer is pipetted into all outer chambers and the central reservoir immediately after plasma cleaning. In addition to their role as a fluid inlet, the reservoir holes serve as access points for silver/silver-chloride (Ag/AgCl) electrodes to create an electrical connection across the nanopores. At the anode (+), dissociated chloride ions (Cl^-) react with Ag, producing AgCl and releasing an electron (e^-) into the wire. At the cathode (-), electrons react with AgCl to release (Cl^-) into the electrolyte while leaving behind Ag. The two reactions can be summed up as

$$Ag + Cl^- \rightarrow AgCl + e^-$$
$$AgCl + e^- \rightarrow Ag + Cl^-$$

at the anode and cathode respectively. It is apparent that the reactions are reversible, which allows voltage switching. In addition, Ag/AgCl electrodes exhibit transfer kinetics fast enough to produce an Ohmic current-voltage response in the experimentally required voltage range of ± 1 V [50]. This means that the system can be viewed as an equivalent electrical circuit, without constraints imposed by the electrochemistry of charge transfer at the electrodes. The electrodes are prepared from 1 mm silver wire stored in a 10 % solution of sodium hypochlorite and replaced regularly.

Once the reservoirs have been filled with measurement solution and the electrodes inserted into the holes, each nanopore is tested by recording a current-voltage curve between -600 mV and 600 mV. The electrode in the outer chamber is moved between chambers to address different pores while the electrode in the central reservoir acts as the ground. A pore is found to be suitable for measurement if the current-voltage data indicates a diameter in the desired range and the root-mean-square (RMS) noise is below about 15 pA. Common failure modes are broken tips which exhibit low resistance or pores with air trapped in the channel or partially closed during the pulling process, which greatly increases the resistance. For diameters between 10 nm and 15 nm at least half of manufactured pores tend to be usable. Once the functioning pores have been identified, the solution in the central reservoir is pipetted out and replaced with sample solution. For negatively charged DNA, the measurement can then begin by applying a positive voltage to the outer chamber, which drives the molecules into the pore. The outer chamber connected to the blunt end of the capillary is referred to as the *trans* reservoir, the central chamber containing the nanopore tips is called the *cis* reservoir.

For pore diameters between 10 nm and 15 nm and a salt concentration of 4 M, a driving voltage of 600 mV produces an ion current between 10 nA and 15 nA. DNA translocations take on the order of milliseconds, meaning substructure during a translocation event appears in a frequency range of around 10 kHz. The current amplifier must therefore be capable

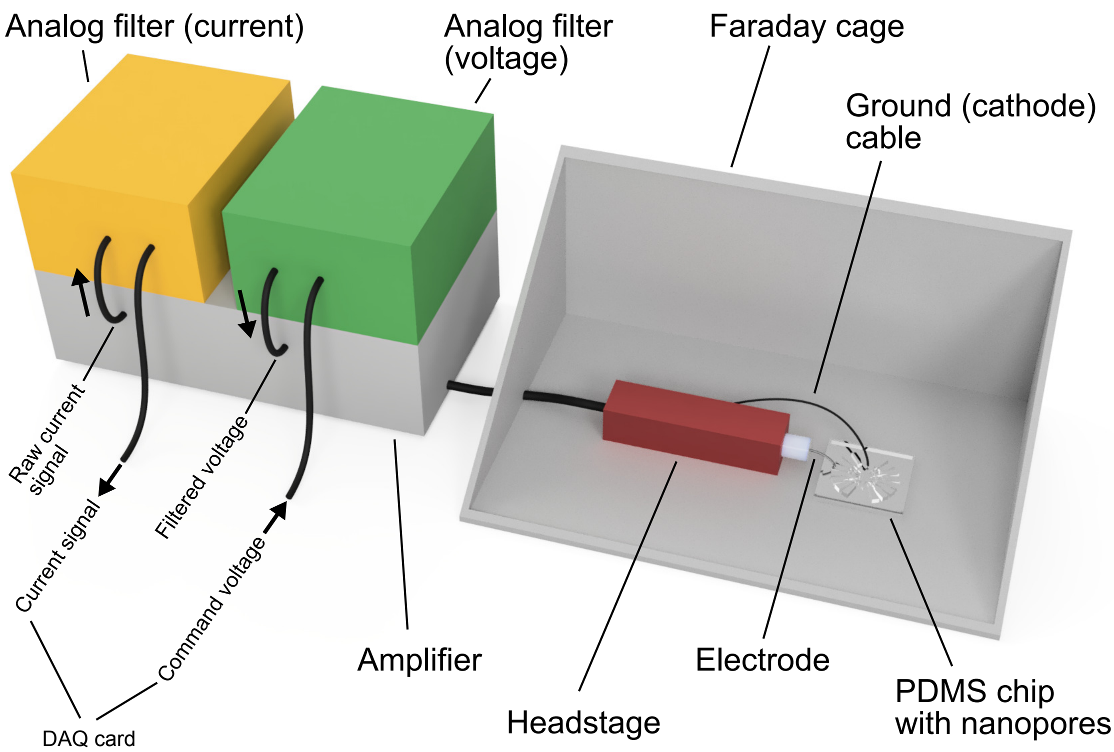


Fig. 2.3 Chip setup. A PDMS chip houses up to 16 nanopores as shown in Figure 2.2, each pore can be addressed separately by moving the electrode between the outer reservoirs. Analogue filters process both the command voltage and the current signal.

of measuring nA currents at a bandwidth > 10 kHz while applying mV potentials. For the chip setup we use an Axopatch 200B amplifier (Molecular Devices, CA, US) in whole-cell voltage clamp mode with the internal filter set to 100 kHz. The amplifier's output voltage is proportional to the current and passed through an additional analogue low-pass 8-pole Bessel filter (900CT, Frequency Devices, IL, US or 3382, Krohn-Hite, MA, US) with a 50 kHz cutoff frequency before being digitised by a data acquisition card (DAQ cards, PCIe-6251 or PCIe-6351, National Instruments, TX, US) at 250 kHz sampling frequency for recording in a computer. The DAQ card also sets the command potential across the nanopore, the signal is again passed through an 8-pole Bessel filter before reaching the amplifier. Since the command potential is kept constant during a measurement the filter is set to attenuate frequencies above 5 kHz. A Faraday cage shields the amplifier headstage from outside interference. Figure 2.3 shows a sketch of the experimental setup and signal pathways.

Data recording is handled by custom-written LabView software (National Instruments, TX, US). Using the recommended buffer setting, the program transfers chunks of 25 000 samples from the DAQ card to the computer 10 times a second, where they are displayed on screen and optionally saved to disk.

2.2.2 *Pressure setup*

The setup described in the previous section is well suited for standard experiments in which large amounts of data from many capillaries need to be collected on a limited number of samples. Once a sample has been added to the central reservoir, up to 16 nanopores can be measured sequentially simply by moving the anode between the outer chambers. A downside of embedding capillaries in a PDMS chip is the added overhead from the fabrication of the chip itself. Another drawback is that to prevent cross-contamination, each chip is used for the measurement of one sample. While the central reservoir can be flushed with buffer to remove most DNA, its large surface area makes it likely that molecules remain even after flushing. The one-chip-one-sample approach prevents using the same nanopore for the measurement of many molecules unless samples are mixed. As a consequence, translocation data from many pores is required to confidently compare samples. Lastly, the PDMS chip makes it difficult to access capillaries once they have been glued into place. To apply pressure, for example, the holes would have to be fitted with inlets stable for pressures up to 1.5 bar, adding another step to the fabrication process.

I found that inserting single capillaries into a commercially available holder takes about 2 min, comparing favourably to the chip approach given a pore fabrication success rate > 50 %. Since filling cannot be carried out on chip, capillaries are plasma-cleaned for 2 min and wholly submerged in buffer solution. In a sealed container they can be kept in the buffer for an extended period. I found storage up to a day to be unproblematic. To begin a measurement, a single capillary is taken from the buffer and threaded through a short piece of rubber tubing whose inner diameter matches the capillary size. The anode is made from 0.1 mm silver wire and cut to extend a short distance out of the holder. This allows threading the wire into the blunt end of the capillary to come into contact with the buffer solution inside. The piece of rubber tubing is then pushed into the capillary holder, forming a tight seal. This configuration holds the capillary in place for pressures up to 2 bar without having to screw the top part onto the holder.

The capillary holder slots directly into the amplifier headstage through a BNC connection. The headstage is attached to a piezoelectric linear stage (Q-545.240, Physik Instrumente, Germany) allowing vertical movement. The *cis* chamber can now be any reservoir with an open top that the pore can move into. I used the same PDMS chip as in the previous section but with the outer chambers sealed off and two 1 mm holes per chamber for capillary and electrode access. The small hole size limits evaporation of the buffer through contact with air. Each chip can be used for 17 different samples in the 16 outer chambers and the central reservoir. Because each capillary is used directly without further fabrication, this technique reduces the overhead in measurement preparation. The capillary can be moved between reservoirs to measure different samples by repositioning the PDMS chip. Since only the very tip enters the solution, the chance of cross-contamination is greatly reduced, particularly if the pore is lowered into fresh buffer between samples. Pressure can be applied without further modifications through the designated inlet in the capillary holder.

The voltage, pressure and capillary position are controlled with custom-written software. A multi-channel DAQ card (PCIe-6361, National Instruments, TX, US) outputs the command voltage to the amplifier and receives the current signal. It simultaneously sets the command voltage for the pressure regulator and measures the pressure signal. The regulator (QPV, Proportion-Air, IN, US) is connected to the building nitrogen supply as well as a vacuum pump (Duo 1.6, Pfeiffer Vacuum, Germany) and can output an ideal pressure between -1 bar and 5 bar where 0 bar corresponds to atmospheric pressure. Due to the pump's imperfect vacuum the lowest achievable pressure is about -0.88 bar, appendix C gives further detail on the calibration. The ability to apply vacuum to the *trans* reservoir allows pressure differentials across the nanopore in both directions. A servo controller (E-871, Physik Instrumente, Germany) connected to the computer via USB drives the piezoeletric stage. Figure 2.4 shows the experimental setup and signal pathways.

The control software is written in Rust, a systems-level programming language with an emphasis on performance and ease of parallel execution on several processors. The language's low-level approach similar to C or C++ means that the software can communicate directly with hardware driver libraries, which greatly reduces the memory footprint and potential for error in wrapper code. By replacing existing LabView programs it obviates the need for expensive licenses and avoids compatibility issues between different LabView versions. Modern code sharing and collaboration tools address the common criticism of software being tied to its creator, who is the only one able to understand and modify the code. Because the code is text-based, it can be shared and maintained much more readily than equivalent graphical LabView programs.

The current on the pressure setup is measured with a HEKA 800 USB amplifier (HEKA Elektronik, Germany) in voltage clamp mode without external analogue filters. The instrument's internal filter is set to a cutoff frequency of 30 kHz. Figure 2.5 compares the power spectral densities (PSDs) for the two setups in the relevant frequency range around 10 kHz, calculated by Welch's method from 1.8 s current chunks without molecule translocations. Each curve represents a measurement on a separate capillary at 600 mV in 4 M LiCl. The noise for the chip setup (blue curves) lies below the pressure setup (green curves) across the shown frequency range, with RMS noise averaged over 3 capillaries of 7 pA and 25 pA respectively. The vertical dashed lines show the filter cutoff frequencies, 50 kHz for the external filter on the chip setup (dashed blue line) and 30 kHz built into the HEKA amplifier

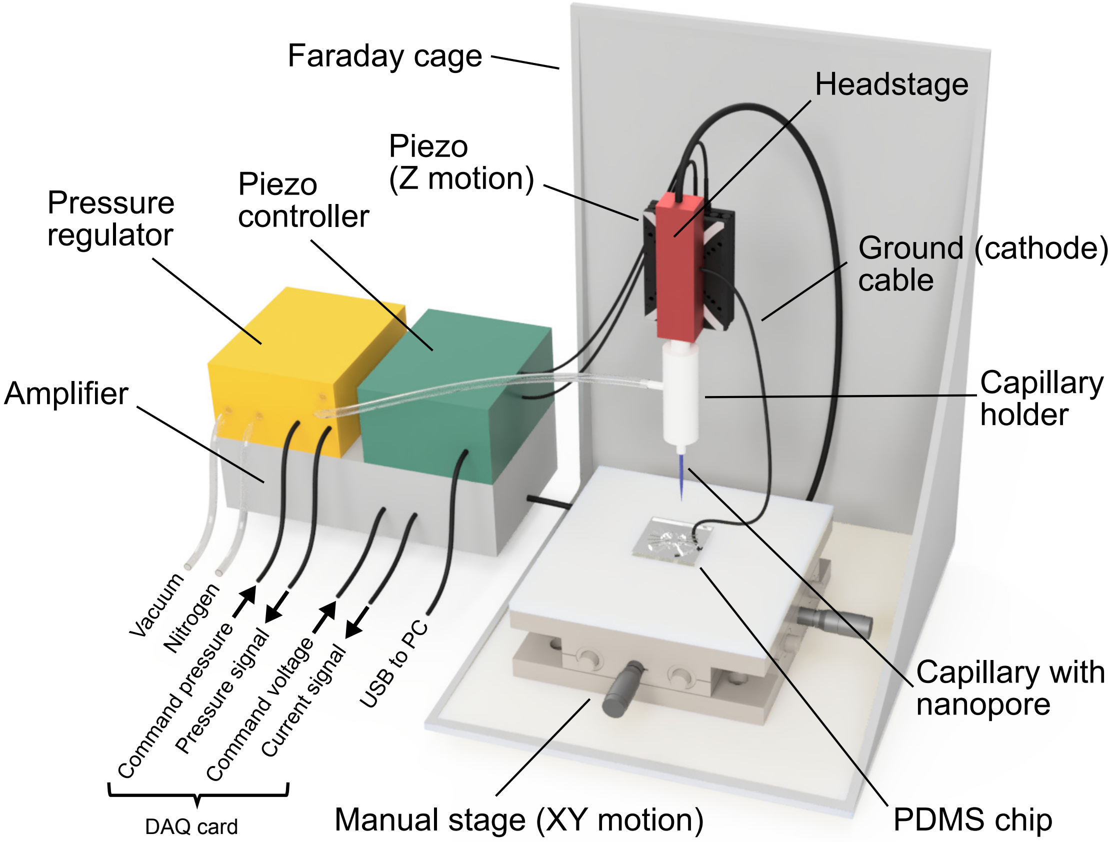


Fig. 2.4 Pressure setup. The nanopore sits at the tip of a glass capillary slotted into a holder. A piezoelectric stage allows vertical movement in and out of reservoirs in a PDMS chip. Pressure is applied through the inlet in the capillary holder with an electronically controlled regulator.

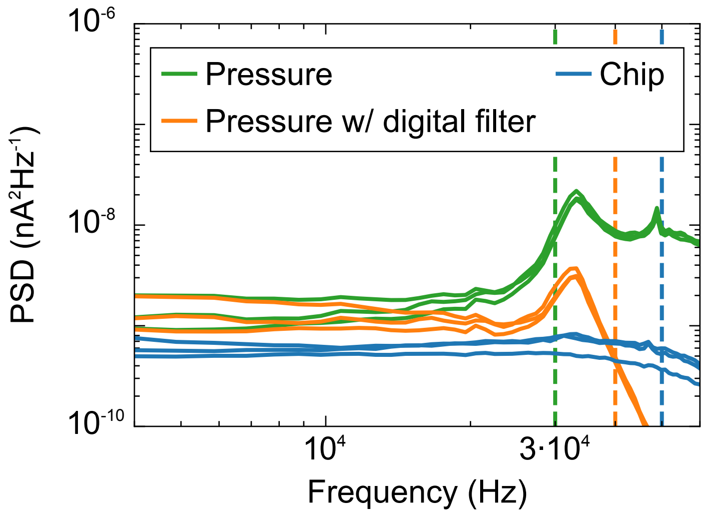


Fig. 2.5 Comparison of power spectral densities (PSDs), calculated with Welch's method from 1.8 s current chunks. Each curve represents a different capillary at 600 mV voltage bias in 4 M LiCl. The raw signal noise from the pressure setup (green) lies above the chip setup (blue) across the frequency range. The increase at 30 kHz is likely due to interference from other parts of the setup, see the main text for details. Processing with a digital 8-pole Bessel filter at 40 kHz cutoff significantly reduces the noise towards high frequencies (orange curve). Vertical dashed lines show the cutoff frequencies of the external analogue filter in the chip setup (blue), the filter built into the amplifier in the pressure setup (green) and the digital filter (orange).

on the pressure setup (dashed green line). Surprisingly, the PSD increases around the cutoff frequency of the built-in HEKA filter. An increase in noise at high frequencies has previously been attributed to capacitance and is well-described by applying Johnson's thermal noise model to an equivalent circuit of the nanopore system [54, 55]. However, the built-in filter should attenuate frequency components in the current signal above 30 kHz. Since the high-frequency noise is still present in the recorded data, it is likely that other parts of the setup affect the signal after it has left the amplifier. While the piezoelectric stage has been eliminated as the source, further tuning of the setup may improve the high-frequency noise characteristics.

Regardless of the increase in PSD at high frequencies, the pressure setup exhibits more noise across the range shown in Figure 2.5. To produce a current signal comparable to the chip setup, I implemented a digital 8-pole Bessel filter using the infinite impulse response (IIR) filter library available at [56]. A cutoff frequency of 40 kHz results in current data similar to the chip setup, varying definitions of the filter cutoff means that the values are not identical. The orange curve in Figure 2.5 shows that digital filtering lowers the pressure setup PSD in the relevant range > 10 kHz, causing an overall drop in the RMS noise to 8 pA. In particular the high-frequency components are effectively attenuated. While the noise is still slightly higher than in the chip setup, it is sufficiently low to detect substructure in translocation events. The distribution of RMS noise between pores becomes more significant than the difference between the two setups. Section 5.2 shows that the data can readily be analysed together.

2.3 Buffers & Samples

Potassium and lithium chloride (KCl and LiCl) salt solutions were prepared from their powdered form by mixing with deionised water. Unless otherwise specified, measurements were carried out at pH 8 by buffering with 10 mM Tris and 1 mM ethylenediaminetetraacetic acid (EDTA) (corresponding to 1xTE buffer). For measurements at other pH values I used 10 mM 4-Morpholineethanesulfonic acid (MES) for pH 5.5 and 10 mM 4-(2-Hydroxyethyl)piperazine-1-ethanesulfonic acid (HEPES) for pH 7. For each buffer the pH of the stock solution was adjusted to the desired value with hydrochloric acid (HCl) or potassium / lithium hydroxide (KHO / LiOH) and confirmed with a pH meter. In measurements with neutral polymers, 75 µM of polyethylene glycol (PEG) with a molecular weight of 8000 $\mathrm{g\,mol^{-1}}$ or 1 mM of dextran with a molecular weight of 6000 $\mathrm{g\,mol^{-1}}$ were added to the solution. All salts, polymers and buffer were purchased from Sigma-Aldrich.

DNA samples were added to the salt solution immediately before the measurement. For unmodified double-stranded nucleic acids, 8 kilo base pair (kbp) DNA (NoLimits DNA fragment) and lambda DNA were purchased from Thermo Fisher Scientific (MA, US) and New England Biolabs (MA, US) respectively. Designed, self-assembled DNA structures are described in detail in the next section, custom oligonucleotides were ordered from Integrated DNA Technologies (IA, US), circular M13mp18 scaffold from New England Biolabs (MA, US).

2.4 DNA structures

DNA carrier-based nanopore sensing relies on the ability to attach modifications at well-defined positions along a strand. While specific sequences in double strands can be labelled with streptavidin via biotin-transferring methyltransferases [39, 40] or dCas9 proteins [5, 41], DNA nanotechnology offers a bottom-up approach to rationally design and self-assemble barcoded carriers. The starting point is a single strand of DNA called the scaffold to which all modifications are attached. Because the translocation velocity in solid-state nanopores requires a certain length to reliably read out the current signal, viral DNA from the M13mp18 bacteriophage is used as a scaffold. It is available commercially as a circular strand of 7249 base pairs (bp) and must be linearised by cutting with a restriction enzyme. The cutting protocol has been published previously [42] and includes the following steps:

Hybridisation of marker oligonucleotide: A short (39 bp) strand is first attached to the circular DNA to mark the restriction site. This is done by mixing

- 20 µL M13mp18 DNA at a concentration of 250 ng µL^{-1}
- 4 µL 10x New England Biolabs CutSmart buffer
- 1 µL marker oligonucleotide at a concentration of 100 µM
- 14 µL deionised water

and heating the mixture to 65 °C, followed by cooling to 25 °C over 40 min in a thermocycler.

Scaffold cutting: Cutting at the restriction site is carried out by adding 0.5 µL BamHI-HF and 0.5 µL EcoRI-HF (New England Biolabs, MA, US) enzymes to the mixture and heating to 37 °C for 1 h, followed by purification with a Macherey-Nagel NucleoSpin Gel and PCR clean-up kit (Fisher Scientific, UK).

The cutting process removes a chunk of the circular strand, resulting in a linear single-stranded scaffold 7228 bp in length. Modifications are attached through Watson-Crick base pairing, the known scaffold sequence allows targeting arbitrary positions along its length. Different attachments produce distinct secondary peaks in the current signal, allowing the rational design of carrier libraries with unique barcodes. Figure 2.6 shows the modifications used in this work. In their initial barcoding scheme, Bell & Keyser [51] attached 11 adjacent dumbbell DNA structures to represent one bit as shown in blue. I used the same design for experiments with binary barcode DNA. Recent work in the group by Dr Jinbo Zhu extends the binary system to a quaternary one, where each position can be occupied by one of four modifications: dumbbells or 3 DNA junctions of different order [unpublished, personal communication]. 4-, 6- and 12-way junctions are shown in orange, green and purple respectively. Lastly I used an ‘anchor’ design for experiments investigating the translocation direction in which one end of the scaffold is folded into a 4-helix bundle. The right end in Figure 2.6 shows an example sketch of how this is accomplished with staple oligonucleotides. Across the scaffold, regions not occupied by modifications are covered by complementary oligonucleotides to make the scaffold double-stranded as indicated by the red strands. Appendix D provides sequence information for all structures.

Once a specific sequence of modifications has been designed, the structure is self-assembled by mixing the scaffold with the correct staples, ensuring that complementary

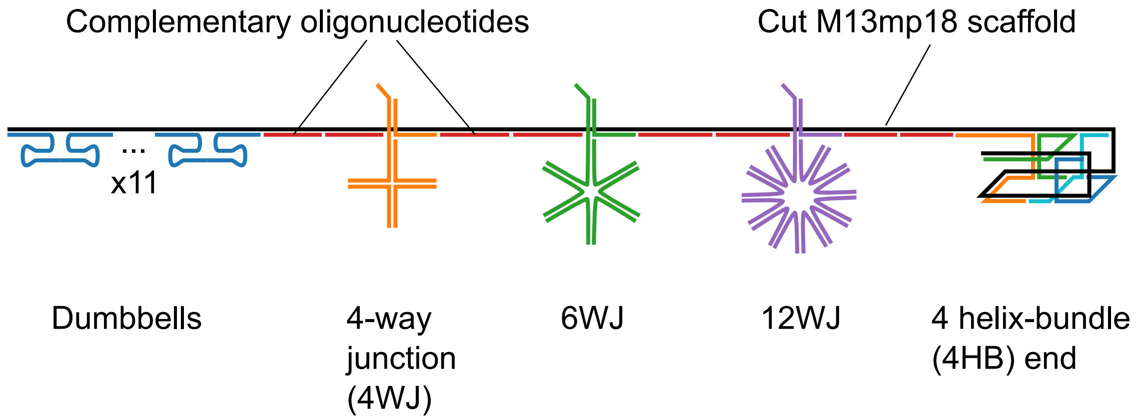


Fig. 2.6 DNA modifications on a M13mp18 scaffold. The blue strands on the left show a section of 11 dumbbell structures which were used by Bell & Keyser [51] to represent one bit in their binary barcoding system. Dr Jinbo Zhu recently extended this design to a quaternary system, with additional digits created from 4-, 6- and 12-way DNA junctions shown in orange, green and purple respectively [unpublished, personal communication]. The rightmost structure shows an 'anchor' design in which one end of the scaffold is folded into a 4-helix bundle (4HB). It should be noted that this is a simplified version for illustrative purposes, exact sequence information for all modifications can be found in appendix D.

oligonucleotides are only added for positions not occupied by attachments. The result is a double-stranded DNA carrier with modifications at the desired locations.

Synthesis for the dumbbells and anchor follows the protocol previously published in [51, SI] by first mixing

- 8 µL scaffold at a concentration of 100 nM
- 20 µL oligonucleotides (complementary and for modifications) at 200 nM
- 4 µL $MgCl_2$ solution at 100 mM
- 1.2 µL 100 mM Tris and 10 mM EDTA (pH 8)
- 6.8 µL of deionised water

followed by heating to 70 °C and subsequent cooling to 25 °C over 40 min in a thermocycler. The fivefold excess of oligonucleotides versus scaffold ensures that the scaffold is fully hybridised to complementary strands. Unbound oligonucleotides are subsequently removed by centrifugal filtering. 100k Amicon Ultra filters (Sigma-Aldrich) are pre-wetted with 500 µL of filtering solution containing 0.5 mM $MgCl_2$ and 10 mM Tris-HCl at pH 8 by centrifuging for 2 min at 1000G. Following pre-wetting the 40 µL DNA sample is added to 460 µL of filtering solution and centrifuged for 10 min at 9000G and 4 °C, this step is repeated 3 times. After each run the passed through liquid containing excess oligonucleotides is discarded while the DNA carriers remain in the filter. To retrieve the carriers after the 3 runs, the filter is placed upside down in a collection tube and spun at 1000G for 2 min at 4 °C. The final volume is 20 µL to 30 µL to which 10 v/v% post-filtering solution is added containing 100 mM Tris-HCl and 20 mM $MgCl_2$. The final DNA carrier concentration is determined from measuring with a UV-Vis spectrophotometer (NanoDrop 2000, Thermo Fisher Scientific, MA, US).

Synthesis for the 4-, 6-, and 12-way junctions follows an altered protocol [Dr Jinbo Zhu, personal communication]. The scaffold is synthesised following the protocol above only with the connecting strands for the junctions and the required complementary oligos. The junctions without connecting strands are annealed separately by mixing the component oligonucleotides with 10 mM Tris-HCl and 10 mM $MgCl_2$ at pH 7.5 to final concentrations of 10 µM (4-WJ) or 8.3 µM (6-WJ and 12-WJ). The solutions are then brought to 90 °C for 5 min and subsequently cooled to 65 °C for 15 min, 45 °C for 15 min, 37 °C for 20 min, 25 °C for 20 min and 4 °C for 20 min. To purify the assembled junctions they are run in native polyacrylamide gel electrophoresis (PAGE), 15 % PAGE for the 4-WJ and 5 % PAGE for the 6- and 12-WJ. Soaking the cut out target band in 200 µL of 10 mM Tris-HCl and 10 mM $MgCl_2$ for 12 h allows collecting the junction structures from the supernatant.

To attach the junctions to the carrier, they are added to the 40 µL sample of previously synthesised scaffold at a concentration of 40 pM. This occurs before the filtering step. The mix is then annealed at room temperature for 2 h. Lastly, filtering as described above removes excess oligonucleotides and junctions and results in roughly 30 µL of assembled DNA carrier.

Chapter 3

Controlling DNA conformation with fluid flows

Parts of this work have been published in:

Ermann, N., Hanikel, N., Wang, V., Chen, K., Weckman, N. E. & Keyser, U. F. Promoting Single-File DNA Translocations through Nanopores Using Electro-Osmotic Flow. *The Journal of Chemical Physics* **149,** 163311 (2018)

My thanks go to the following collaborators: Nikita Hanikel designed the asymmetric DNA strand and helped with measurements. Dr Nicole Weckman collected parts of the lambda DNA data. Vivian Wang carried out the measurements on dextran and in polyethylene glycol (PEG) solution and contributed lambda DNA and potassium chloride (KCl) data.

Chapter 1 introduced DNA barcodes as a strategy to encode information in the structure of single molecules. Modifications along a double strand of DNA produce distinct signals during translocation through a nanopore. Changing the position and size of the attachments defines unique signal sequences, allowing the creation of large barcode libraries.

Successful readout of a barcode fundamentally depends on the characteristics of the measurement device. The current noise must be low enough to distinguish peaks produced by modifications from the background. Similarly, the recording bandwidth must offer sufficient temporal resolution relative to the translocation speed to avoid overlap between closely spaced peaks. The experimental setups described in section 2.2 fulfils both these criteria when combined with the barcode-based encoding scheme. However, low-noise instrumentation is only one contributing factor to successful barcode reads. Another crucial aspect is the conformation of the DNA strand during translocation through the nanopore. The

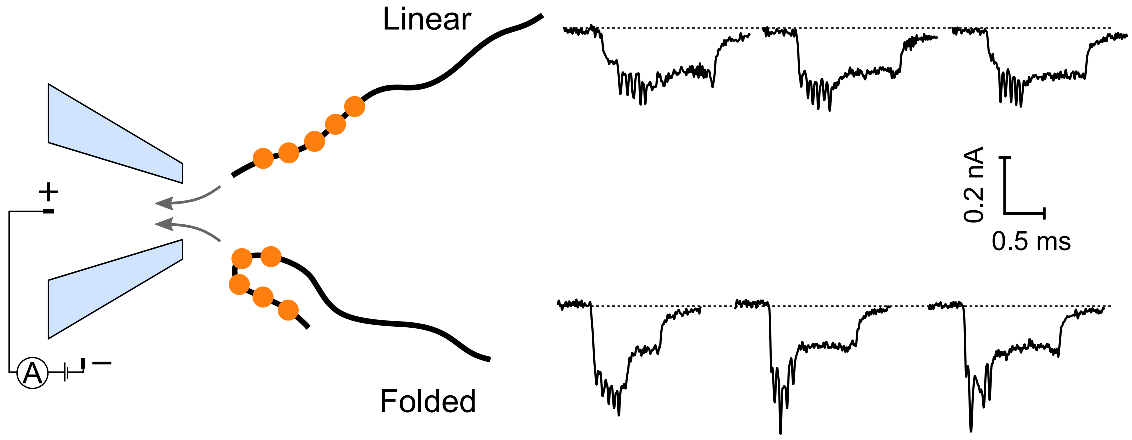


Fig. 3.1 Single-file versus folded translocations of a DNA strand through a nanopore. Folds occur when part of the strand folds back onto itself as it passes through the pore. The sample events in the right part of the figure are from a '11111' barcode consisting of 5 equal modifications in one half of the strand, measured in 4 M LiCl at 600 mV and pH 8.

strand can either pass through in a linear, single-file fashion when captured on one of its ends or translocate in a hairpin conformation where the strand folds back onto itself. Figure 3.1 shows single-file and folded translocations along with sample events of a barcode consisting of 5 equal modifications in one half of the strand. The current traces demonstrate that a linear conformation greatly facilitates distinguishing the peaks produced by each attachment. Folds, on the other hand, mean that modifications may be at similar positions in the pore at the same time. While some peaks are still visible, it is much more difficult to deduce the DNA structure from the current signal.

Folded translocations have been studied extensively in different pore types. Li *et al.* [57] investigated different DNA conformations in thin 2D-membrane pores. Storm *et al.* [58] extended the work to different DNA lengths, while Mihovilovic *et al.* [59] studied the position of the fold with a self-avoiding walk model. Steinbock *et al.* [60] were the first to investigate folding in glass nanocapillary pores. This chapter builds on these studies by showing that fluid flows, set up with different measurement parameters, influence the conformation of translocating DNA strands. Such conformational control offers a way to tune the translocation process towards an improved readout of modification sequences.

3.1 Electro-osmotic flow (EOF) in nanopores

Before investigating the effect of flows on DNA conformation it is instructive to discuss the hydrodynamics inherent in a nanopore system. Both glass and SiN surfaces acquire charge when in contact with aqueous solution [61, 62]. An electric field along the surface sets counterions screening the walls in motion, causing fluid flows even in the absence of external pressures. This effect is referred to as electro-osmotic flow (EOF) and gains particular relevance in nanofluidic systems given their high surface-to-volume ratios. We can obtain a quantitative description starting with the Boltzmann equation for the concentration of monovalent ions at a distance x from a charged surface

$$c_{\pm}(x) = c_0 N_a \exp\left(\mp\frac{e\Psi(x)}{k_B T}\right)$$

where $c_{\pm}$ is the molar ion concentration, c_0 the bulk concentration, N_a the Avogadro constant, e the elementary charge, $\Psi(x)$ the electric potential at a distance x, k_B the Boltzmann constant and T the temperature [63]. The net charge density in the solution follows as

$$\rho(x) = e\big(c_+(x) - c_-(x)\big) \tag{3.1}$$

$$= ec_0 N_a\left(\exp\left(-\frac{e\Psi(x)}{k_B T}\right) - \exp\left(+\frac{e\Psi(x)}{k_B T}\right)\right) \tag{3.2}$$

The anion concentration thus decays away from a surface with negative potential while cations accumulate in its proximity. To obtain an expression for the charge density as a function of the distance from the wall we require Poisson's equation

$$\nabla^2\Psi(x) = -\frac{\rho(x)}{\varepsilon_0\varepsilon_r}$$

which gives the spatial distribution of charges in an electric potential [63]. ε_0 and ε_r are the vacuum and relative permittivity respectively. Inserting equation 3.2 for the net charge density results in the Poisson-Boltzmann equation

$$\nabla^2\Psi(x) = -\frac{ec_0 N_a}{\varepsilon_0\varepsilon_r}\left(\exp\left(-\frac{e\Psi(x)}{k_B T}\right) - \exp\left(+\frac{e\Psi(x)}{k_B T}\right)\right)$$

We can linearise this equation assuming that the electric potential is much smaller than the thermal energy, i.e. $|\frac{e\Psi}{k_B T}| \ll 1$, which is valid for the glass surface charges encountered in this work. This leads to the Debye-Hückel equation [63]

$$\nabla^2\Psi(x) = \kappa^2\Psi(x) \quad \text{with} \quad \kappa = \sqrt{\frac{2e^2 c_0 N_a}{\varepsilon_0\varepsilon_r k_B T}}$$

To approximate the nanopore geometry we solve this equation in cylindrical coordinates to obtain a net charge density

$$\rho(r) = -\sigma\kappa\frac{I_0(\kappa r)}{I_1(\kappa R)}$$

where r is the radial distance from the centre, R the radius of the channel and I_n the nth-order modified Bessel function of the first kind. The surface charge density σ results from the charge neutrality condition $\int \sigma dA = -\int \rho dV$ where the integrals are over the surface area and the channel volume respectively [64]. The quantity $\frac{1}{\kappa}$ is called the Debye length and measures the characteristic screening distance away from the charged surface. The relative permittivity ε_r depends on the ionic species and for LiCl it can be approximated at 25 °C by [65, 66]

$$\varepsilon_{r,\mathrm{LiCl}} = 78.48 - 15.5c_0 + 1.96c_0^2 - 0.306c_0^{\frac{5}{2}} \tag{3.3}$$

This allows us to calculate the Debye length at varying ionic strength, where Figure 3.2 a) shows the dependence on the LiCl concentration. For values above 1 M LiCl, most of the surface charge is screened within 0.3 nm, demonstrating that electrostatic interactions of the DNA with the walls are unlikely for pore sizes of $\approx$ 15 nm.

We can define a dimensionless charge density $\tilde{\rho}$ to illustrate the distribution of ions in the nanopore channel, given by

$$\tilde{\rho}(\tilde{r}) = -\frac{\rho R}{\sigma} = -\tilde{\kappa}\frac{I_0(\tilde{\kappa}\tilde{r})}{I_1(\tilde{\kappa})} \quad \text{with} \quad \tilde{\kappa} = \kappa R \;\text{ and }\; \tilde{r} = \frac{r}{R}$$

where $\tilde{\kappa}$ is the inverse Debye length normalised by the channel radius which scales as the square root of the salt concentration. Figure 3.2 b) shows that at low salt, the screening regions overlap and maintain a net charge at the centre of the channel (blue curve). As the ionic strength increases the Debye length shrinks and the surface charge is screened within a short distance from the wall (green curve). EOF occurs as a result of the regions of net charge in the channel. An electric field applied along the cylinder axis sets in motion the two ion species in opposite directions. If a fluid volume contains imbalanced concentrations of the two ions, i.e. a net charge, their motion transfers net momentum to the fluid. We can model the resulting flows as a body force in the Stokes equation, given by

$$\mu\nabla^2 v = \overbrace{-\rho(r)E_z}^{\substack{\text{body force on volume with net}\\ \text{charge due to electric field}}} + \underbrace{\nabla p}_{\text{external pressure}} \tag{3.4}$$

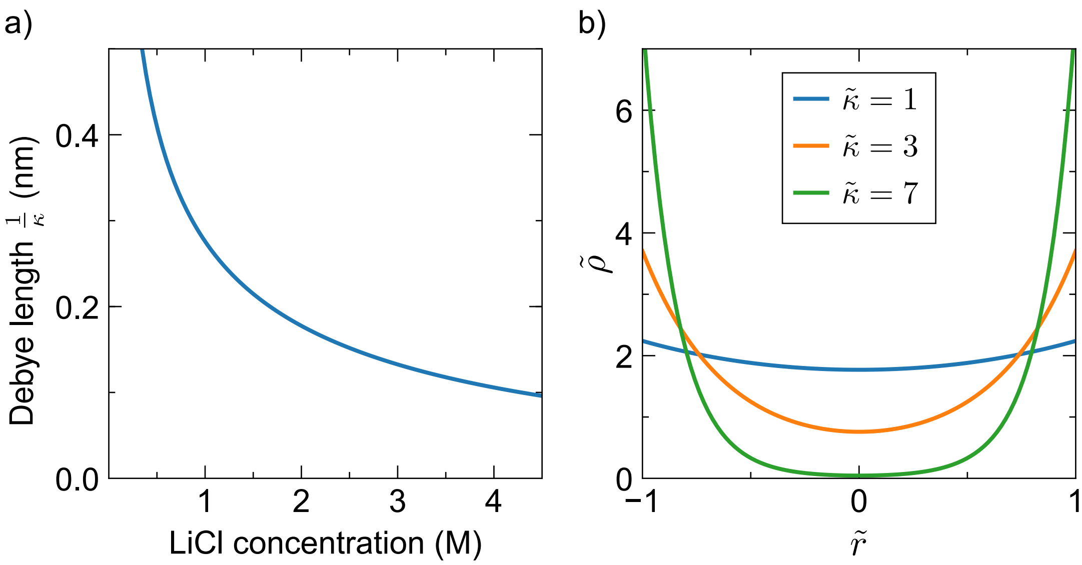


Fig. 3.2 Counterion screening of surfaces in a cylindrical channel. The Debye length $\frac{1}{\kappa}$ in **a)** measures the characteristic lengthscale within which ions screen the surface charge. It decreases with salt concentration and is smaller than 0.5 nm above 0.5 M LiCl. **b)** shows the dimensionless charge density along the cross-section of a cylindrical channel as a function of the normalised distance $\tilde{r}$ from the channel centre. The parameter $\tilde{\kappa}$ scales as the square root of the salt concentration, the channel walls are positioned at $\tilde{r} = \pm 1$. At high salt, counterions screen the surface charge within thin regions along the walls (green curve). At low salt, the charge imbalance spreads across the whole channel (blue curve).

where v is the flow velocity, μ the dynamic viscosity, E_z the electric field along the cylinder axis and p the pressure applied between the channel ends [67]. This shows that even without external pressure, fluid flows emerge when a voltage is applied across a nanopore with charged walls. Ignoring the pressure term with $p = 0$, we can solve this equation in cylindrical coordinates assuming no-slip boundary conditions at the walls to obtain the flow profile across the channel as [64]

$$v_{\mathrm{EOF}}(r) = -\frac{\sigma E_z}{\kappa\mu}\frac{I_0(\kappa R)}{I_1(\kappa R)}\left(1 - \frac{I_0(\kappa r)}{I_0(\kappa R)}\right) \tag{3.5}$$

The dimensionless fluid velocity

$$\tilde{v}_{\mathrm{EOF}}(\tilde{r}) = \frac{v_{\mathrm{EOF}}\mu}{\sigma E_z R} = -\frac{1}{\tilde{\kappa}}\frac{I_0(\tilde{\kappa})}{I_1(\tilde{\kappa})}\left(1 - \frac{I_0(\tilde{\kappa}\tilde{r})}{I_0(\tilde{\kappa})}\right) \quad \text{with} \quad \tilde{\kappa} = \kappa R \text{ and } \tilde{r} = \frac{r}{R} \tag{3.6}$$

is shown in Figure 3.3 a). At high salt, the net charge is confined to thin screening regions along the walls, meaning fluid at the centre only experiences a force due to laminar friction from the edges. The result is a plug-like flow profile where velocities are roughly constant across most of the channel (green curve). The more uniform charge density at low ionic strength means that body and laminar friction forces add up at the centre of the channel, resulting in a parabolic velocity profile (blue curve). Figure 3.3 b) shows the velocity v_{EOF} at $r = 0$ as a function of the LiCl concentration for typical parameter values of $\sigma = -0.02\,\mathrm{C/m^2}$ [61], $E_z = 1 \times 10^6\,\mathrm{V\,m^{-1}}$ [68, SI] and $R = 15\,\mathrm{nm}$. κ is calculated based on equation 3.3 as in Figure 3.2 a) while the concentration-dependence of the viscosity μ is shown in appendix E. The velocity at the centre of the channel drops with increasing LiCl concentration, demonstrating that EOF is more significant at low ionic strength.

While the simple model presented here allows rough predictions of fluid flows in a nanopore, a number of caveats must be kept in mind. Firstly, the Poisson-Boltzmann-Stokes formalism assumes ions to be point charges without a physical size and neglects interactions with water molecules. More detailed models include several layers of ions with different hydration states which can partially adsorb onto the surface [63]. The immobile ions in the immediate vicinity of the walls lead to the definition of a slip plane which separates a stationary layer from the moving fluid. The electric potential at this slip plane is called the zeta potential and influences EOF by determining an effective charge screened by movable ions. Ion interaction with the surface and solvent means that the zeta potential is not equal across monovalent salts but depends on the ionic species [69].

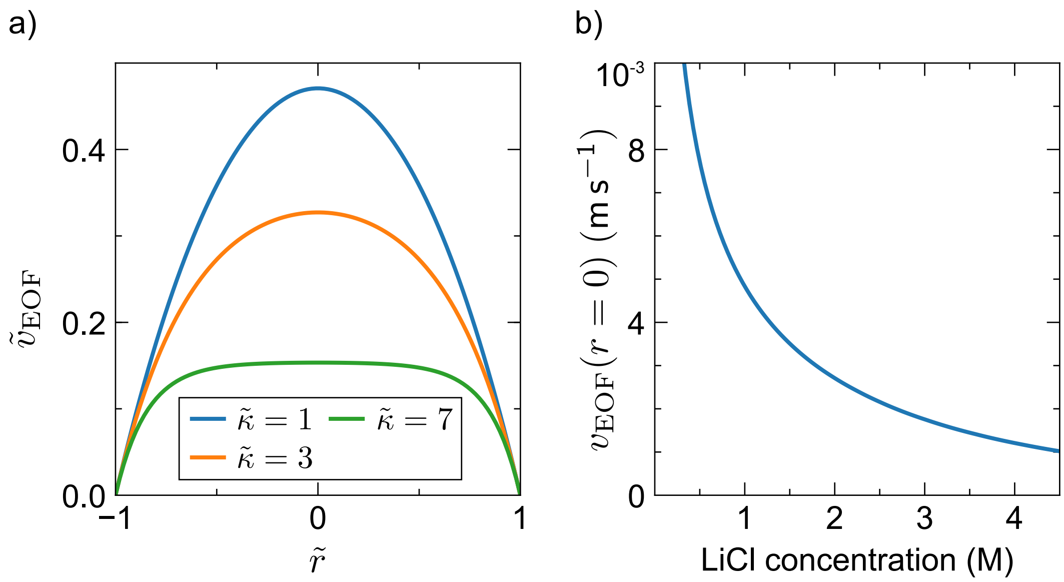


Fig. 3.3 Electro-osmotic flow (EOF) velocities in a cylindrical channel. **a)** shows the dimensionless velocity along the channel cross-section as a function of the normalised radius $\tilde{r}$ where the walls are positioned at $\tilde{r} = \pm 1$. $\tilde{\kappa}$ scales as the square root of the salt concentration. At high salt, charge screening within a thin region along the walls (c.f. Figure 3.2 b)) means that fluid in the centre of the channel is only set in motion through laminar friction, leading to a plug-shaped flow profile (green curve). At low salt, net charge across the channel experiences a body force resulting in a parabolic shape. **b)** shows the flow velocity at the centre of a 15 nm cylindrical channel as a function of the LiCl concentration for typical values of the electric field $E_z = 1 \times 10^6\,\mathrm{V\,m^{-1}}$ [68, SI] and surface charge $\sigma = -0.02\,\mathrm{C/m^2}$ [61]. Permittivity and viscosity values are calculated from equation 3.3 and appendix E respectively. The fluid velocity decreases with LiCl concentration and reaches values on the order of $1 \times 10^{-3}\,\mathrm{m\,s^{-1}}$ above 0.5 M LiCl.

A second caveat concerns the cylindrical channel geometry which allows analytical solutions for the charge density and flow fields. While a valid approximation for pores in 2D membranes, it fails to account for a number of effects in conical glass nanocapillaries. The asymmetry combined with negatively charged walls creates a dependence of the pore conductivity on the voltage polarity. As a result the absolute current at positive voltages is smaller than at negative voltage, a phenomenon termed current rectification [70]. EOF in the pore exhibits the opposite dependence, so that absolute flow rates are larger at positive bias leading to flow rectification [67, 71]. In addition to the conical inner shape, the outer walls of the nanocapillary interact with the bulk fluid to contribute to the overall flow. Laohakunakorn *et al.* [68] found that the electric field along the outer walls points in the opposite direction from that inside the pore. Since outside EOF is not limited by the pore confinement it continues to increase with lower salt concentration, eventually causing flows in front of the pore to point towards the pore mouth under positive voltages. This manifests itself as a surprising flow reversal as a function of ionic strength, however it should be noted that EOF inside the pore does not change direction.

The above studies measured flow velocities from the displacement of an optically trapped bead in front of the pore. The sensitivity of this method meant that EOF could only be measured up to salt concentrations of about 100 mM. Given the steep decrease of the flow velocity with ionic strength, previously reported nanopore experiments assumed that EOF would be negligible for high salt concentrations [72]. To detect flows even at high salt, we carried out measurements with 1 mM of 6000 $\mathrm{g\,mol^{-1}}$ dextran added to 1 M LiCl solution at pH 8. Dextran is a branched polysaccharide whose neutral charge means that it does not feel an electrophoretic force. Because of its branched structure it does, however, sufficiently reduce the ion current in the pore to cause discernible signals upon translocation (see appendix F for sample events). We added dextran only to the *cis* reservoir outside the capillary and applied a bias voltage of -600 mV. The plot in Figure 3.4 shows the frequency of dextran translocations over time. At a negative voltage regular events can be observed, consistent with EOF driving the uncharged molecules into the pore. After about 10 min the voltage was switched to 600 mV, inverting EOF to point out of the pore. We again observed translocations as outward flows carried dextran molecules back to the *cis* side. Since a limited number of molecules were initially transported through the pore, the event frequency decreases over time upon voltage reversal to a positive value. This behaviour of neutral molecules confirms that flows influence the translocation process even at the high salt concentrations commonly used in nanopore sensing. Despite their small magnitude compared to low ionic strength, EOF forces cannot be neglected in a description of DNA dynamics during passage

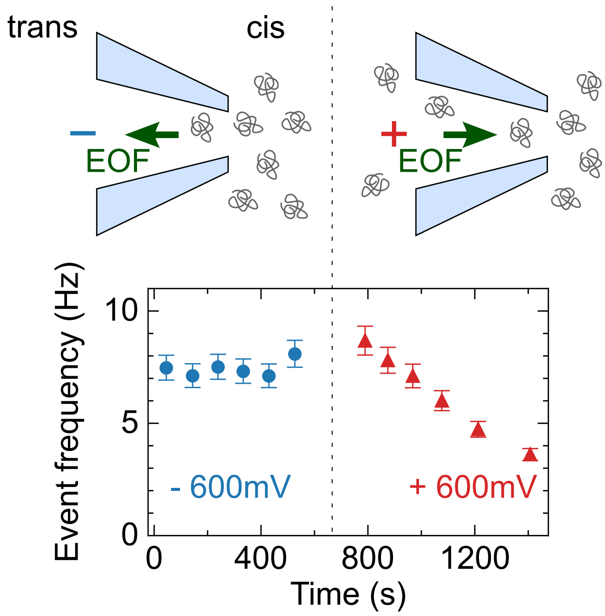


Fig. 3.4 EOF-driven translocation of neutral polymers. 1 mM of 6000 g mol^{-1} dextran was added to the *cis* reservoir outside the capillary in 1 M LiCl solution at pH 8. Translocations at a constant frequency could be observed under a negative bias voltage of −600 mV, indicating that EOF carried molecules through the pore. Error bars show 95 % confidence intervals for the rate parameter of the arrival time exponential distribution [1]. Upon voltage reversal to 600 mV the polymers travelled back out of the pore, however at a decreasing rate due to the limited number of molecules which initially reached the *trans* region inside the capillary. These results demonstrate that EOF influences the translocation process at salt concentrations in the molar range. Figure reproduced from [1] with the permission of AIP Publishing.

through a nanopore. The following section investigates the effect of fluid flows on the DNA conformation in greater detail.

3.2 *EOF and DNA folding*

The previous section showed that the fluid velocity in a channel due to EOF scales as

$$v \propto -\frac{\sigma E}{\kappa} \propto -\frac{\sigma E_z}{\sqrt{c}} \tag{3.7}$$

where σ is the surface charge, E_z the electric field, c the salt concentration and $\frac{1}{\kappa}$ the Debye length which measures the distance within which counterions screen the surface charge. More highly charged walls and a larger electric field thus increase the fluid velocity while a higher salt concentration reduces flows. These three parameters influence fluid motion in the pore, suggesting they may be suitable to control the DNA folding state.

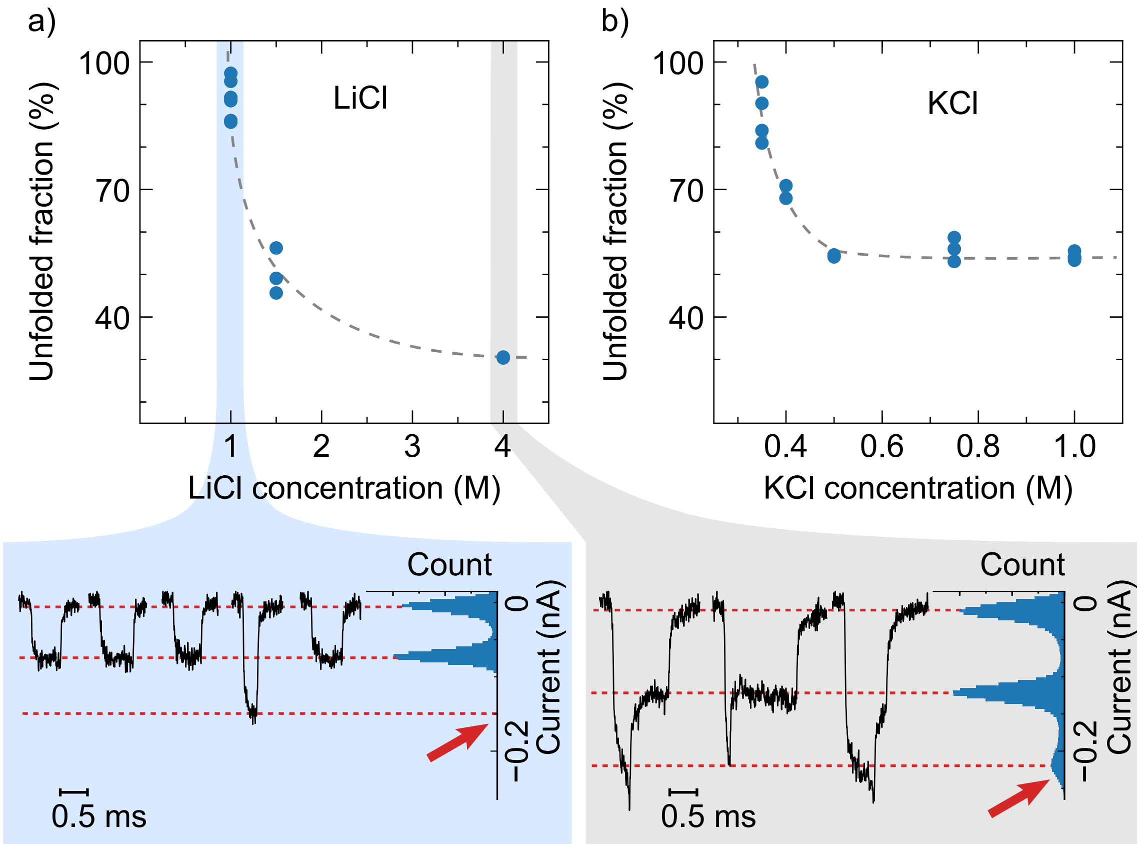


Fig. 3.5 The share of single-file, unfolded translocations at varying salt concentrations. Each marker represents a measurement of 8 kbp DNA on a separate pore with > 500 events, dashed lines serve as guides to the eye. In lithium chloride (LiCl), single-file passage through the pore increases below 2 M, reaching values > 80 % at 1 M (panel **a**)). The histograms in the lower part of the Figure show the distribution of the concatenated event current traces along with sample events. The peak corresponding to a hairpin fold, i.e. two double strands of DNA in the pore, disappears at low ionic strength. Panel **b)** shows that the effect also appears in potassium chloride (KCl), albeit with a lower level of folding at high salt and an onset of predominantly unfolded events at 0.4 M. Figure adapted from [1] with the permission of AIP Publishing.

Starting with the salt concentration, we expect a lower ionic strength to boost EOF. Figure 3.5 shows the fraction of unfolded 8 kbp events as a function of varying concentrations of lithium (LiCl) and potassium chloride (KCl). Each marker represents a measurement from a different nanopore with > 500 events, section 4.2 contains details on the analysis procedure[1]. For both salts, reducing the concentration greatly increases the share of single-file, unfolded translocations, eventually reaching > 80 %. The lower part of the figure shows sample translocations for LiCl as well as a histogram of the concatenated event current traces. The red arrows in the histograms confirm that the peak corresponding to the folded current level, i.e. the current measured with two double strands of DNA in the pore, disappears almost completely at a low salt concentration. Figure 3.6 confirms that the effect persists for longer 48.5 kbp DNA (lambda DNA), albeit starting from a higher overall level of folding as expected for longer molecules. These results suggest that EOF opposing the motion of DNA into the pore promotes single-file entry by increasing the counteracting force on high-drag conformations. The proposed mechanism is shown in Figure 3.7 along with sample events from 4 M and 1 M LiCl. Lowering the salt concentration increases flows out of the pore which leads to higher hydrodynamic drag on the DNA. Although the electrophoretic driving force increases with less effective screening of the DNA charge at low salt, outward EOF rises sufficiently to reduce the entry of folded conformations.

Interestingly, the unfolded fraction starts to increase at much lower salt concentrations for KCl compared to LiCl. In addition, the share of unfolded events plateaus between 50 % and 60 % for high KCl concentrations relative to < 40 % for LiCl. These effects are in line with the different behaviour of Li and K cations when screening DNA strands and glass surfaces. For DNA, Kowalczyk *et al.* [73] showed that Li cations bind much more strongly than K cations, reducing the effective driving force in the pore. This suggests that weaker EOF is sufficient to counteract the electrophoretic force for LiCl, resulting in an increase of unfolded translocations at higher salt concentrations. Similarly, the more efficient screening of the DNA charge by LiCl cations reduces the persistence length and facilitates folding, leading to the greater incidence of folded events at high LiCl concentrations compared to KCl. It should be noted that this refers to a difference in persistence length between ionic species, not different concentrations of the same salt. The changes in DNA persistence length between the high and low salt concentrations shown here are too small to explain the significant changes in folding [74].

Another contributing factor to the different behaviour of LiCl and KCl is their interaction with glass surfaces. It has been shown that at the same concentration, LiCl screening of a

[1]The slight differences in the unfolded fractions shown here compared to the values published in [1] are due to the updated, simplified data analysis outlined in section 4.2. They do not change the conclusions drawn from the data.

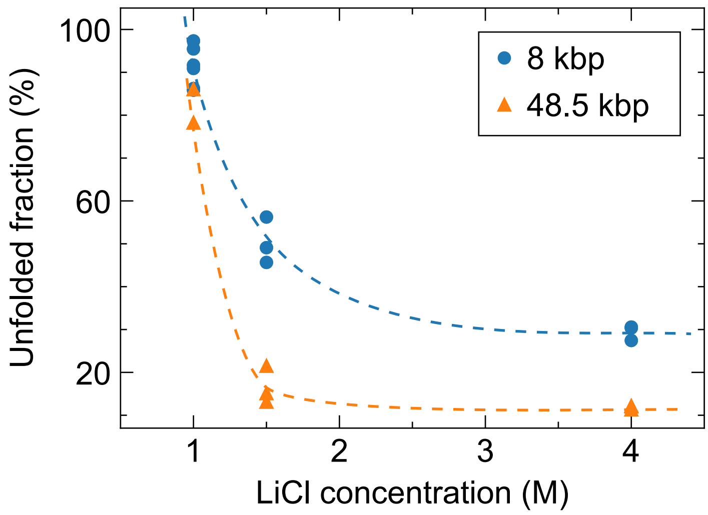


Fig. 3.6 Folding for different DNA lenghts in lithium chloride (LiCl). Blue markers reproduce the data in Figure 3.5 a), orange markers correspond to measurements on lambda DNA with a length of 48.5 kbp, dashed lines serve as guides to the eye. As expected from longer strands, lambda DNA experiences more folding at high salt concentrations. At low ionic strength, however, single-file translocations increase to values > 75 %, demonstrating that the salt concentration influences folding across different DNA lengths. Figure adapted from [1] with the permission of AIP Publishing.

quartz surface leads to a more negative zeta potential relative to KCl [69]. The zeta potential refers to the electric potential at the slip plane, i.e. the plane which separates immobile ions from the moving fluid layer. A higher zeta potential means a larger net charge to be set in motion by an applied voltage, leading to larger EOF. Mc Hugh *et al.* [75] confirmed increased outflows from glass nanocapillaries in LiCl compared to KCl solution. EOF is therefore expected to influence translocation behaviour at higher salt concentrations in LiCl relative to KCl as indicated by the data in Figure 3.5.

Having demonstrated that ionic strength affects DNA folding, we would expect the other parameters in equation 3.7 to also be relevant if EOF is indeed responsible for the effect. I therefore investigated folding at several driving voltages which linearly change the electric field E_z. Figure 3.8 shows the share of unfolded 8 kbp events as a function of voltage for different LiCl concentrations. At 4 M and 1.5 M, increasing the voltage leads to more folding. This can be explained by a larger electrophoretic force which outweighs any counteracting forces and makes it more likely that the molecule is pulled into the pore at the position which first enters the capture radius. Interestingly, at 1 M LiCl a higher voltage does not change the share of unfolded events. At this salt concentration, the previous paragraphs showed that EOF strongly influences the translocation process since drag forces become comparable to the electrophoretic driving force. Given that EOF scales linearly with the electric field, the

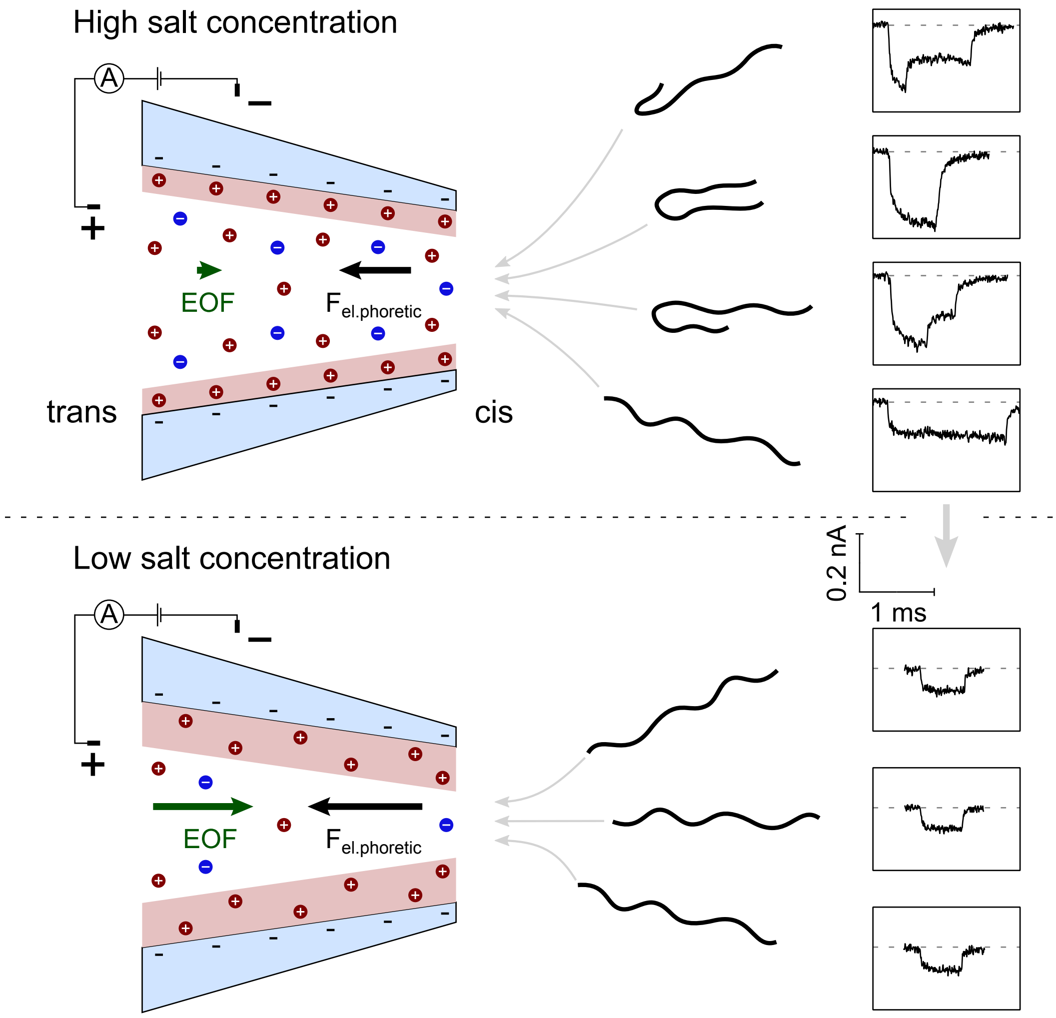


Fig. 3.7 Electro-osmotic flow (EOF) influences the conformation of DNA translocating through a nanopore. At high salt concentrations, screening of surface charges within a thin region along the walls leads to small EOF and the electrophoretic driving force dominates translocation behaviour. The DNA enters the pore at the position which first moves into the capture radius. At low salt, EOF grows and hydrodynamic drag forces become comparable to the electrophoretic force. This creates a preference for DNA entry in low-drag conformations, which promotes single-file, unfolded entry into the pore. Figure reproduced from [1] with the permission of AIP Publishing.

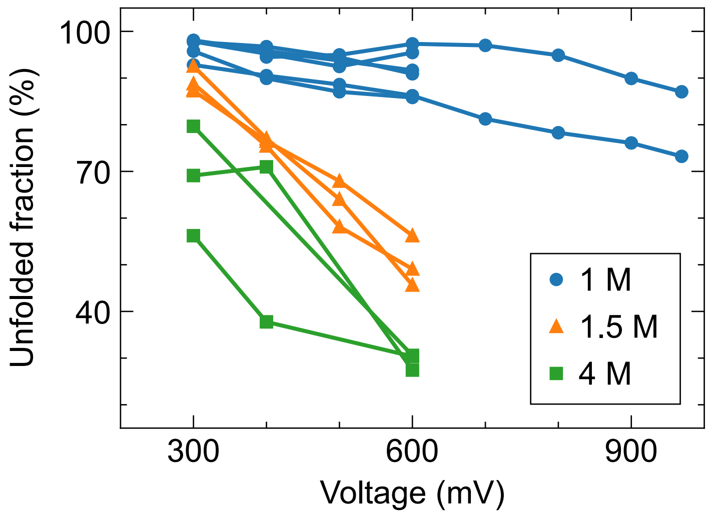


Fig. 3.8 Voltage dependence of 8 kbp DNA folding behaviour. At LiCl concentrations of 4 M and 1.5 M, the share of unfolded events decreases with voltage, as a growing electrophoretic force drives molecule into the pore as soon as they enter the capture radius. At 1 M LiCl, the linear scaling of electro-osmotic flow (EOF) with the voltage causes drag forces to partially counteract the electrophoretic force, discouraging entry of high-drag conformations. Figure adapted from [1] with the permission of AIP Publishing.

larger electrophoretic force at higher voltages may be compensated by a similar increase in counteracting drag.

Another parameter which influences EOF is the charge on the pore walls. The charge of silica surfaces depends strongly on the pH of the surrounding solution, becoming more negative in alkaline conditions as silanol groups deprotonate [61]. Reducing the pH below 8 therefore weakens EOF out of the pore, while the DNA charge remains unaffected in the pH range from 5 to 8 given the pK_a values of the bases and phosphate backbone [76]. Figure 3.9 shows the share of unfolded translocations for different pH values in 2 M and 0.5 M LiCl, each bar represents a measurement on a different nanopore. At 1 M LiCl, lowering the pH from 8 to 7 drops the unfolded fraction from > 80 % to around 50 %, in line with weaker EOF. The smaller decrease upon reducing the pH to 5.5 can be explained by the non-linear pH dependence of the surface charge, which exhibits a smaller gradient at low pH values [61]. Crucially, lowering the pH allows measurements at 0.5 M LiCl which are unfeasible in pH 8 because EOF out of the pore prevents DNA entry. A high share of unfolded events now appears at pH 7, while the large drop occurs between pH 7 and 5.5. Tuning the salt concentration and acidity of the measurement solution thus allows sensitive control of voltage-induced fluid flows in the pore and thereby the translocation process.

This concludes the discussion of the theoretically predicted EOF parameters from equation 3.7. The salt concentration, electric field and surface charge all influence DNA folding as

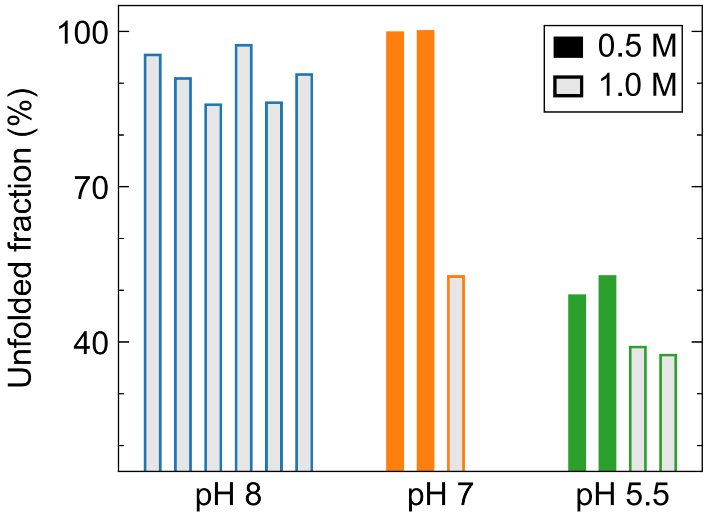


Fig. 3.9 Tuning electro-osmotic flow (EOF) and the DNA folding state with pH. Lower pH reduces the negative surface charge on the nanopore walls as silanol groups remain protonated [61]. This reduces EOF, causing folded translocations to re-emerge at pH 7 and 5.5 in 1 M LiCl. In pH 7, the increase in unfolded events now appears at a lower salt concentration of 0.5 M LiCl, while pH 5.5 inhibits EOF sufficiently for folding to persist down to 0.5 M salt. Figure adapted from [1] with the permission of AIP Publishing.

expected, with stronger EOF promoting single-file translocations. Another tuning parameter with a less obvious effect on EOF is the addition of polymers to the measurement solution. Both covalently bound and dynamically adsorbed wall coatings have been used in capillary electrophoresis to tune EOF [77]. One example is polyethylene glycol (PEG)[2], which has been shown to suppress EOF in 75 µm glass capillaries by adsorbing onto the surface [78]. The proposed mechanism states that the polymers shift the slip plane between immobile ions and the moving fluid layer away from the wall towards a lower zeta potential, reducing the fluid volume which is set in motion by an electric field [79, 80].

To test whether the same effect persists at much smaller nanopore length scales, we measured 8 kbp DNA in a solution containing 75 µM PEG 8000, corresponding to a molecular weight of 8000 $\mathrm{g\,mol^{-1}}$. PEG 8000 consists of about 180 ethylene oxide units with a contour length of roughly 50 nm. The polymer is uncharged and readily fits into a $\sim$ 15 nm pore given a hydrodynamic radius between 2.9 nm and 3.6 nm [81, 82]. The latter corresponds to an overlap concentration of $\sim$ 13 mM, orders of magnitude above the dilute 75 µM concentration used in the measurements.

Figure 3.10 shows current traces from a solution containing 5 nM 8 kbp in 1 M LiCl at pH 8 with and without PEG 8000. For the same DNA concentration, the addition of PEG

[2]Polyethylene glycol (PEG) and polyethylene oxide (PEO) refer to the same polymer consisting of ethylene oxide chains.

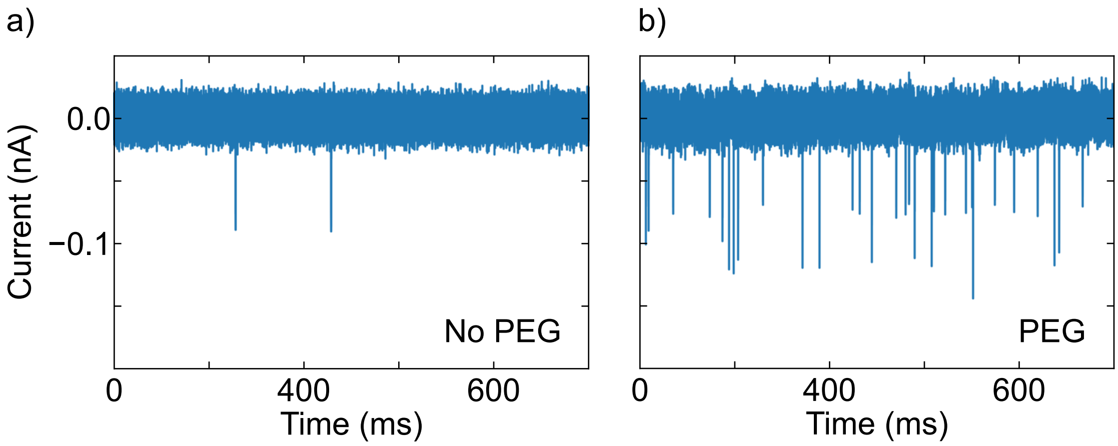


Fig. 3.10 Current traces for 5 nM 8 kbp DNA in 1 M LiCl at pH 8 with and without 75 µM PEG 8000 added to the solution. Neutral PEG polymers have been shown to quench EOF in glass channels [77, 78]. The greatly increased translocation frequency in panel **b)** compared to **a)** indicates that this effect also occurs in nanopores.

greatly increases the DNA translocation frequency. PEG molecules are too small and carry too little charge to produce discernable signals in the nanopore. Figure 3.11 a) confirms this result at different salt concentrations. In pH 8, lowering the salt concentration below 1.5 M LiCl decreases the event frequency as EOF drag becomes comparable to the electrophoretic driving force. Reduced capture rates due to EOF have been reported in Si_3N_4 pores and studied in simulations of a voltage-gate-modulated nanopore system [83–85]. The addition of PEG, however, inverts the dependence on ionic strength: the translocation frequency now rises with lower salt concentration. This confirms that adsorbed polymers quench flows in the pore, recovering the expected behaviour without EOF of a stronger electrophoretic force at low salt due to less efficient screening and a more negative effective DNA charge. Indeed, the addition of PEG leads to capture rate behaviour comparable to that at pH 5.5 (green markers), indicating that polymer adsorption reduces EOF in a similar manner as reducing the absolute surface charge.

The effect on DNA folding matches the expected result given lower EOF. Figure 3.11 b) compares the share of unfolded events with and without PEG. The addition of polymers shifts the onset of predominantly single-file translocations to lower salt concentrations while maintaining the shape of the curve. This demonstrates that PEG reduces EOF but does not completely block flows. As outflows grow with lower ionic strength the previously observed increase in single-file events reappears. This can be understood by considering that polymers move the slip plane by a distance related to the thickness of the adsorbed layer, causing EOF to once again become relevant when lower ionic strength increases the Debye length to a comparable distance.

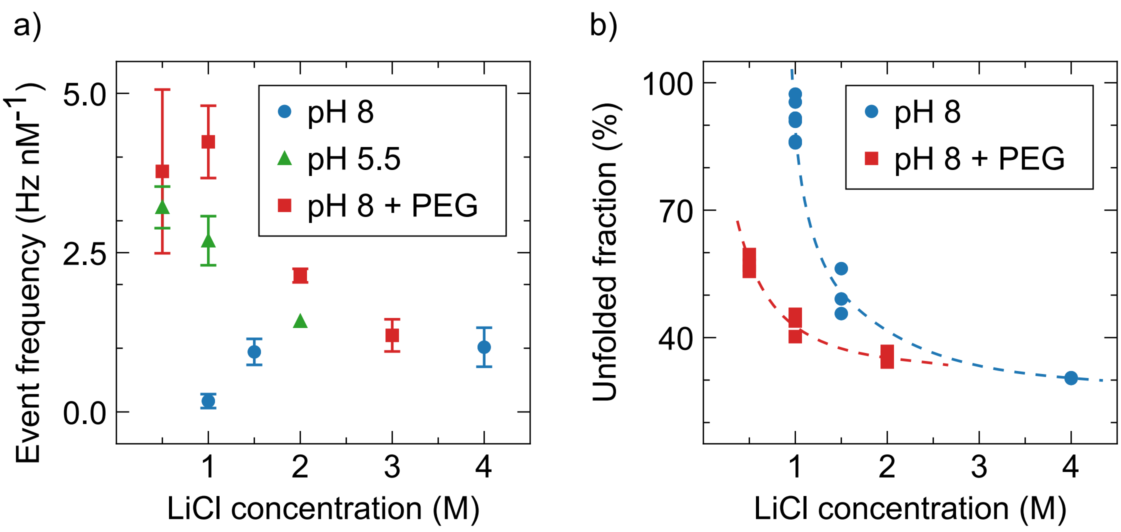


Fig. 3.11 Translocation frequency and folding behaviour with 75 µM PEG 8000. Adding PEG leads to an increase in the event frequency at lower LiCl concentration, the opposite of the case without PEG where translocations disappear almost completely below 1 M LiCl (panel **a)**). Comparison with pH 5.5 data indicates that PEG successfully quenches electro-osmotic flow (EOF). Panel **b)** shows that the addition of PEG reintroduces DNA folding at salt concentrations below 2 M LiCl, in line with reduced EOF. The share of unfolded events still increases with lower ionic strength, the onset is merely shifted to smaller salt concentrations. Dashed lines serve as guides to the eye. Figure adapted from [1] with the permission of AIP Publishing.

The above results show that the folding state of DNA translocating through a nanopore strongly depends on fluid flows. EOF out of the pore boosts the share of single-file, unfolded translocations to above 80 % as hydrodynamic drag counteracts the electrophoretic driving force to discourage high-drag conformations. I investigated the salt concentration, electric field and surface charge as the theoretically predicted parameters which influence EOF. As expected from the theory, lowering the salt concentration increases the unfolded fraction in both LiCl and KCl. The different behaviour between the two salts can be explained by differences in the DNA binding strength and zeta potential on glass surfaces. The electric field is inversely related to the unfolded fraction at high ionic strength but does not affect folding at low salt, where EOF's linear dependence on the voltage counteracts the increasing electrophoretic force. Modulating the surface charge with pH changes confirms that EOF promotes linear translocations, as does tuning fluid flows with EOF-quenching PEG molecules.

The precise control over DNA folding afforded by easily adjusted measurement parameters offers a way to promote single-file translocations, which facilitates the readout of modification sequences attached to the DNA. However, the complex interplay between electrophoretic and hydrodynamic forces means that EOF is tightly interlinked with other translocation parameters. Firstly, the high share of unfolded events at low salt comes with faster events as flows out of the pore reduce entry of folded conformations but the larger electrophoretic force outweighs drag to speed up single-file translocations. As a result, for successful decoding, the modification sequence must be sufficiently spaced to balance out the reduced temporal resolution. Secondly, the frequency of translocations drops considerably at the salt concentration where unfolded events become more likely. While tuning EOF with pH and PEG partially weakens this effect, it shows that for modification-decoding applications it may be preferable to simply exclude folded events from the analysis while benefiting from the high capture rate at high salt concentrations. Section 6.3 illustrates an adaptive measurement technique which can circumvent this issue.

The study of EOF in this section has demonstrated that the translocation process is sensitive to fluid flows in the pore. This raises further questions on the role of hydrodynamics in the system: firstly, can flow control the direction of DNA entry into the pore? Secondly, how does externally applied pressure change translocation behaviour? The following sections address these questions in greater detail.

3.3 EOF and the translocation direction

The previous section established that EOF out of a nanopore reduces the share of folded translocations where the DNA is captured along its contour. If the flow is sufficient to discourage the entry of high-drag conformations, it should be possible to break the inversion symmetry of equally likely DNA entry from both directions by modifying one end of the strand to experience higher drag. This would be useful for modification-decoding applications as it would obviate the need to include information on the intended readout direction in the modification sequence.

To test this hypothesis we designed an 'anchor' DNA strand whose one end has been shaped into a structure of 4 adjacent helices (4-helix bundle, 4HB). The design is based on the same single-stranded 7.2 kbp scaffold used for the barcoding scheme, details can be found in section 2.4. One end is folded into a 4HB anchor consisting of 4 double strands while complementary oligos cover the remainder of the scaffold to produce regular double-stranded DNA. The inset in Figure 3.12 shows the short 30 nm anchor at one end, followed by a much longer double-stranded region. The anchor serves two purposes in the following measurements: on the one hand, it makes the DNA ends asymmetric to test whether this influences the translocation direction. On the other hand, it conveniently acts as a marker to indicate which end has entered first. Similarly to the dumbbell structures in the barcoding

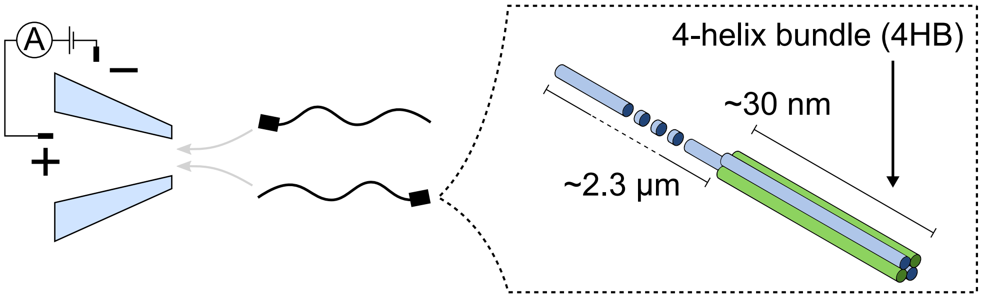


Fig. 3.12 An asymmetric 7.2 kbp DNA double strand with one multi-helix end. The inset shows the 30 nm 'anchor' end consisting of 4 double helixes attached to a much longer regular double strand. The DNA can enter the nanopore with either end first, a peak in the event current indicates the position of the multi-helix end. Figure adapted from [1] with the permission of AIP Publishing.

scheme, the anchor produces distinct peaks in the event current either at the start or end of the translocation, see appendix G for sample events.

Figure 3.13 a) shows the share of single-file, unfolded events for which the anchor end entered the pore last as a function of LiCl concentration at pH 8. At high salt, both translocation directions are equally likely. Moving to lower ionic strength, the share of events with the anchor at the tail end gradually increases to > 80 %. The curve closely resembles the rise in unfolded translocations with lower salt concentrations in Figure 3.5. EOF out of the pore thus has the same effect on the translocation direction as on the folding state: outflows increase the counteracting force on high-drag conformations, creating a preference for entry in the low-drag configuration. The anchor discourages capture at its end and makes it more likely for the unmodified end to enter first.

Figure 3.13 b) shows that the anchor DNA follows the same folding behaviour as 8 kbp DNA, with > 80 % single-file events at 1 M LiCl. While the complex interplay between electrophoretic and hydrodynamic forces in an asymmetric geometry make it difficult to arrive at a conclusive description of the translocation process, the combination of flow-dependent folding and directionality hints at a model in which the DNA makes multiple attempts to overcome an entry barrier, but is only permitted to do so in certain conformations. Such an entropic barrier has previously been described in glass nanocapillaries and stands in contrast to drift-dominated behaviour where the translocation begins as soon as the DNA enters the capture radius [72]. Future work could provide further insight into these models by combining current measurements with fluorescent imaging to study the residence time of DNA in front of the pore in different flow regimes, extending the work of Thacker *et al.* [86].

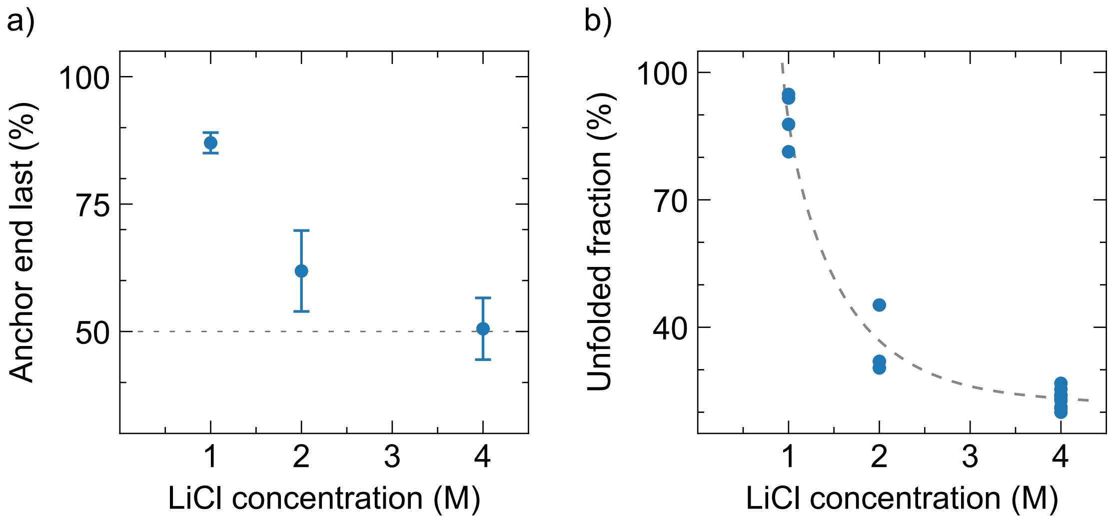


Fig. 3.13 Direction of entry and folding state of asymmetric DNA strands at pH 8. Lowering the LiCl concentration creates a preference for the multi-helix end to enter the pore last (panel **a**)), in line with preferred entry of low-drag conformations under electro-osmotic flow (EOF). The folding behaviour in panel **b**) matches that of an unmodified 8 kbp strand, c.f. Figure 3.5 a). Figure adapted from [1] with the permission of AIP Publishing.

3.4 Externally applied pressure

Up to this point all flows were produced by the nanopore system itself, due to electro-osmotic forces on charged fluid volumes along the pore walls. An alternative strategy is to induce fluid motion through an external pressure applied across the pore. This method has been successfully employed to detect translocations of short DNA strands as well as uncharged molecules [87]. Hoogerheide *et al.* [88] demonstrated that careful balancing of electrophoretic and hydrodynamic forces creates a pressure-voltage trap in front of the pore which can capture 3.27 kbp DNA for seconds. Rempfer *et al.* [89] showed similar trapping for glass nanocapillaries, while Yamazaki *et al.* [90] used external pressure for streaming potential measurements to study the surface charge of a SiN pore at different temperatures. In this section I extend the investigation into the effect of EOF on the DNA folding state to flows set up through external pressure. Pressure is applied to the inside of the capillary, positive values thus cause flows out of the pore (see Figure 3.14).

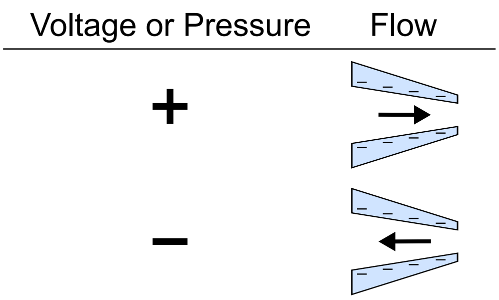


Fig. 3.14 Direction of fluid flows. Both voltage and pressure are applied to the *trans* reservoir inside the capillary, positive values thus cause flows out of the pore while negative values drive fluid into the pore.

The Stokes equation once again serves as a starting point for an approximate description of the hydrodynamic behaviour in the nanopore, given by

$$\mu\nabla^2 v = \overbrace{-\rho(r)E_z}^{\substack{\text{body force on volume with net}\\ \text{charge due to electric field}}} + \underbrace{\nabla p}_{\text{external pressure}}$$

where v is again the flow velocity, μ the dynamic viscosity and p the pressure [67]. In the description of EOF in section 3.1 we ignored the pressure term and obtained the flow profile in a cylindrical channel from the first term on the right hand side, which describes the body force on a fluid volume containing a net charge. To arrive at an analytical expression for the flow velocity, we now assume uncharged walls and therefore the absence of an electro-osmotic force. In a cylindrical geometry, solving the Stokes equation under no-slip boundary conditions without the body force term results in

$$v_{\text{pressure}} = \frac{\nabla p}{4\mu}(R^2 - r^2) = \frac{1}{4\mu}\frac{\Delta p}{L_{\text{eff}}}(R^2 - r^2)$$

where R is the channel radius, r the radial distance from the axis and we assume a uniform pressure gradient due to a drop Δp across an effective channel length L_{eff} [91]. This is Poiseuille's solution to pressure-driven, laminar flow in a pipe. We can define a velocity $\tilde{v}_{\text{pressure}}$ to visualise the flow profile, given by

$$\tilde{v}_{\text{pressure}} = \frac{v_{\text{pressure}}}{R^2} = \frac{1}{4\mu}\frac{\Delta p}{L_{\text{eff}}}(1 - \tilde{r}^2) \quad \text{with} \quad \tilde{r} = \frac{r}{R}$$

Figure 3.15 shows the parabolic profile across the channel due to the no-slip boundary conditions along the walls. The maximum velocity at $r = 0$ scales linearly with the pressure

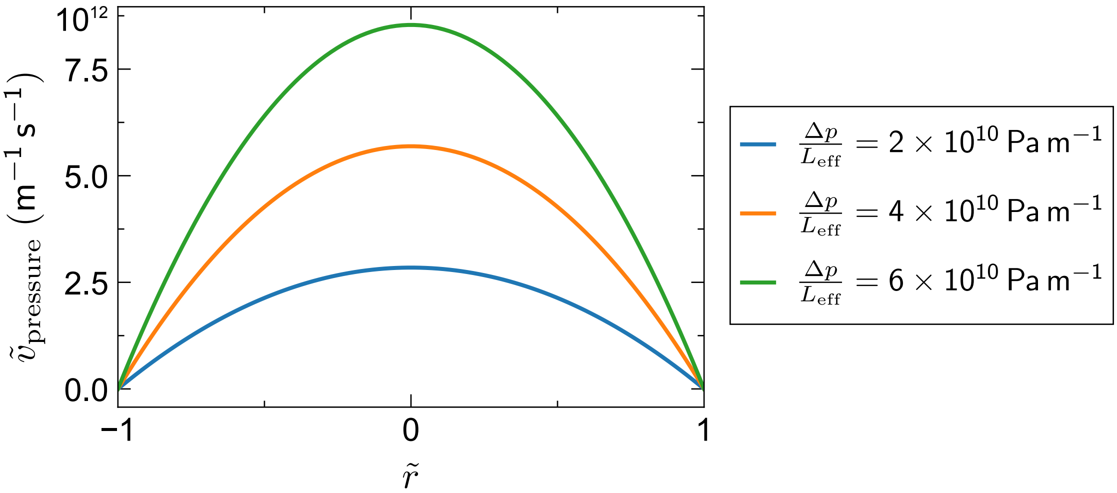


Fig. 3.15 Pressure-driven flow in a cylindrical channel. The plot shows the normalised flow velocity $\tilde{v}_{\text{pressure}}$ as a function of the normalised radius $\tilde{r}$ where the channel walls are positioned at $\tilde{r} = \pm 1$. No-slip boundary conditions at the surfaces result in a parabolic flow profile, the velocity scales linearly with the pressure gradient approximated by a pressure drop Δp over an effective channel length L_{eff}. The dynamic viscosity μ is calculated from appendix E.

drop across the channel Δp. While a cylindrical channel is only an approximation to the asymmetric geometry of a glass nanocapillary, it allows a rough comparison between the magnitudes of pressure-driven and electro-osmotic flows. Figure 3.16 a) reproduces the flow velocity due to EOF at the centre of a 15 nm channel ($r = 0$) as a function of LiCl concentration from section 3.1. Panel b) shows the pressure-driven maximum flow velocity as a function of the effective channel length L_{eff} for different pressure drops Δp. The total flow rate through a cross-section A of the channel

$$Q = \int v_{\text{pressure}} dA = \frac{\pi}{8\mu} \frac{\Delta p}{L_{\text{eff}}} R^4$$

must be constant along the capillary in an incompressible fluid [64]. Its 4th-power scaling with the channel radius R means it is dominated by the narrowest constriction at the tip of the capillary. As a result the majority of the pressure drop occurs across this length. Previous studies using scanning electron microscopy (SEM) images estimated lengths of the narrow constriction region on the order of 100 nm [42, 92]. These values informed the range of the effective channel length L_{eff} in Figure 3.16 b).

Comparing v_{EOF} and v_{pressure} shows that they are of similar magnitude, suggesting that externally applied pressure influences the DNA folding state during translocation. In section 3.2 it was found that the share of single-file, unfolded events begins to increase between 2 M and 1 M LiCl. Figure 3.16 illustrates that pressures on the order of 100 mbar produce

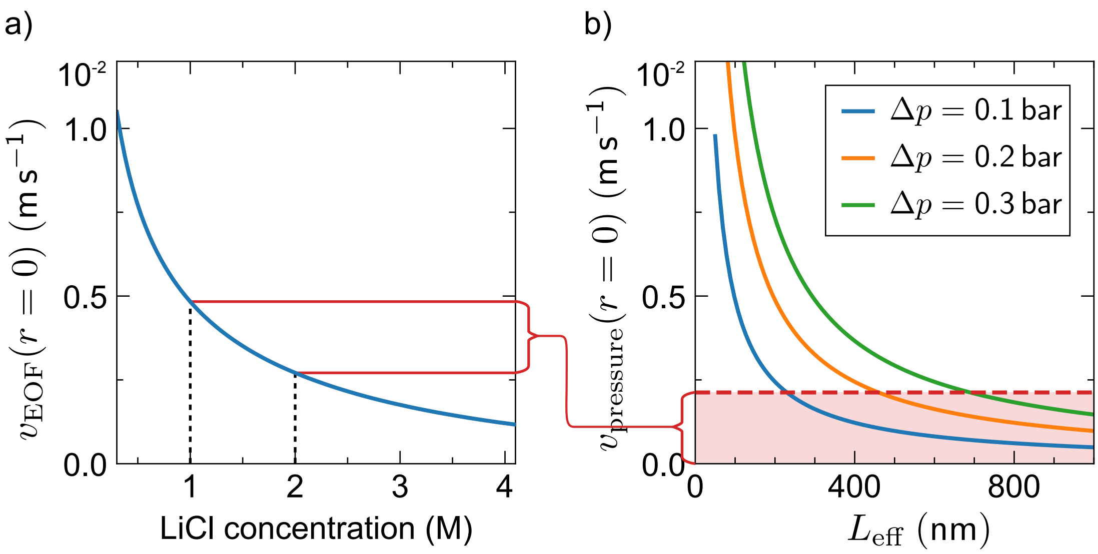


Fig. 3.16 Comparison of electro-osmotic (EOF) and pressure-driven flow velocities at the centre of a 15 nm cylindrical channel. Panel **a)** shows the decrease of v_{EOF} with LiCl concentration for typical values of $E_z = 1 \times 10^6\,\mathrm{V\,m^{-1}}$ for the electric field [68, SI] and $\sigma = -0.02\,\mathrm{C/m^2}$ for the surface charge [61], see section 3.1 for details. Panel **b)** shows the pressure-driven velocity v_{pressure} as a function of the effective channel length L_{eff} for varying pressure drops Δp. The magnitudes in both cases lie on the order of $1 \times 10^{-3}\,\mathrm{m\,s^{-1}}$. Red lines show that the difference in EOF velocity between 1 M and 2 M LiCl is comparable to flow due to millibar pressures, suggesting that such pressures affect DNA folding in a similar manner.

flow velocities comparable to the difference in EOF magnitude between these two salt concentrations. To test whether the effect on the DNA conformation is similar to EOF, I measured 8 kbp DNA in 2 M LiCl at pressures between -1 bar and 0.25 bar applied to the *trans* reservoir inside the capillary. 0 bar corresponds to atmospheric pressure and no difference between the reservoirs, negative values produce a vacuum on the *trans* side.

Figure 3.17 shows how translocation behaviour changes as a function of pressure in a 16.9 nm pore. Panel a) compares the event frequency with the duration of unfolded events. Negative pressures drive more molecules into the pore but speed up translocations, as expected from faster flows into the pore. Panel b) shows the share of unfolded events as a function of pressure. Above 0 bar the incidence of single-file translocations increases steeply, reaching close to 100 % at about 0.25 bar. The shape of the curve resembles the one obtained from tuning EOF by varying the salt concentration in Figure 3.5 a). Stronger outflows from the pore promote single-file entry in both cases. These results are an independent confirmation that fluid flows indeed influence the folding state of DNA molecules during translocation through a nanopore. Despite the different flow profiles, both EOF and pressure-driven flow reduces the entry of high-drag DNA conformations. Another important conclusion is that external pressure can serve as an additional parameter to control the translocation process. Tuning EOF requires changing the salt concentration, pH or solution content, all of which may be at odds with other experimental requirements. External pressure, on the other hand, is a directly controllable, independent parameter. Section 6.3 discusses a possible application in greater detail.

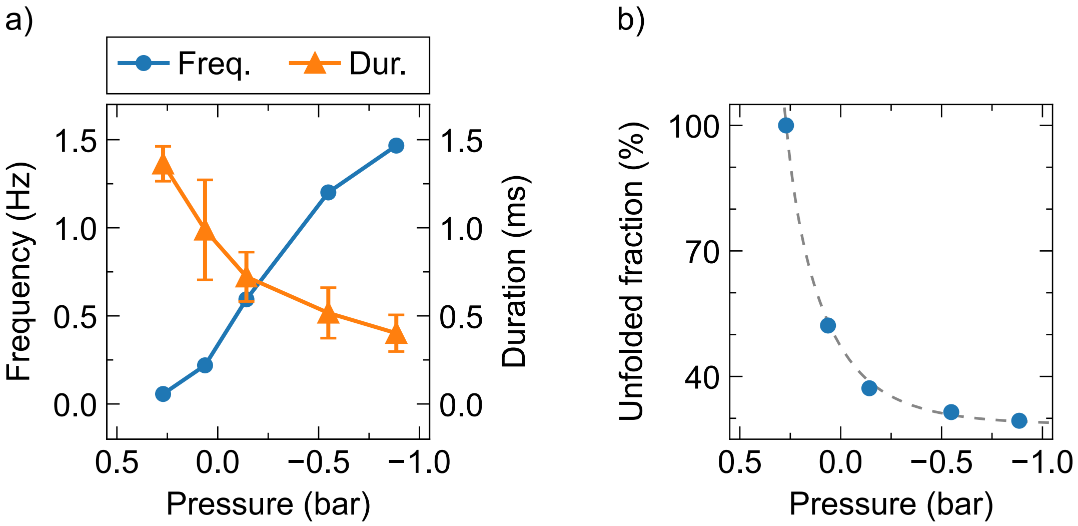


Fig. 3.17 The effect of external pressure on the event frequency, duration and folding state of 8 kbp DNA in a 16.9 nm pore in 2 M LiCl at pH 8. Pressure is applied to the *trans* reservoir inside the capillary, 0 bar corresponds to atmospheric pressure while negative values create a vacuum on the *trans* side. Panel **a)** shows that negative pressures increase the capture rate but speed up translocations, as expected from flow into the nanopore. Panel **b)** reveals an increase in the share of single-file, unfolded translocations with growing pressure. This is analogous to the effect of electro-osmotic flow (EOF) reported in section 3.2, confirming that fluid flows out of the pore discourage DNA entry in high-drag conformations.

3.5 Conclusion

This chapter discussed the influence of fluid flows on the translocation of DNA molecules through a nanopore. Electro-osmotic flows (EOF) are inherent in channel systems with charged walls when an electric field is applied along the channel axis. The electric field sets in motion fluid volumes which contain a net charge of screening counterions close to the surfaces. The resulting flow velocity decreases with salt concentration and in nanopores eludes direct detection at ionic strengths on the order of 1 M. The presence of flows in 1 M LiCl was revealed through translocations of neutral dextran molecules, which produced discernible signals when carried through the pore by the fluid. I then demonstrated that EOF influences the folding state of DNA during translocation. At high salt and negligible flow, a large share of molecules translocate in a folded hairpin conformation, where part of the strand folds back onto itself. Lowering the salt concentration greatly increases the fraction of unfolded events, reaching > 80 % single-file translocations at 1 M LiCl and 0.3 M KCl. The results revealed intriguing differences between the two salts, which can be explained by different interactions with the DNA and glass surface. Further evidence for the role of EOF came from experiments at different pH, which changes the surface charge of the glass pore. The observed folding matched the expected behaviour, with lower pH reducing flows due to a lower charge on the walls, resulting in a larger share of folded events. It was then demonstrated that PEG polymers added to the measurement solution quench EOF by dynamically adsorbing onto the nanopore surface, as previously reported for capillary electrophoresis [77, 78]. Folding once again increased for reduced EOF after the addition of PEG. These results suggest that fluid flows out of the pore create a preference for DNA entry in a low-drag conformation. To further test this hypothesis, we used DNA nanotechnology to design an asymmetric strand with a multi-helix bundle at one end. While entry is equally likely from both ends at high salt, lowering the salt concentration gradually reveals a preference for the multi-helix bundle to be at the tail end of the translocation, consistent with the preferred entry of low-drag conformations. Lastly, I demonstrated that flows due to externally applied pressure have the same effect on the DNA folding state as EOF, confirming the hydrodynamic origin of the observed phenomenon and validating pressure as an additional parameter to control the translocation process.

The above results are valuable for applications which aim to encode information in attachments along a double strand of DNA. Signals from folded translocations often need to be discarded because the peaks produced by modifications smear out in the folded section. Increasing the share of single-file, unfolded events through fluid flows directly increases the usable data and therefore shortens the time-to-result. It must be kept in mind that this is only the case if outflows do not significantly reduce the translocation frequency. The results indicate that this can be achieved through the addition of PEG to the measurement solution. In combination with externally applied pressure as an independent parameter, neutral polymer additives could offer precise control over the translocation process, making them a promising area of future research.

Chapter 4

Bayesian inference

Parts of this work have been published in:
Ermann, N., Chen, K. & Keyser, U. F. Bayesian Inference for Nanopore Data Analysis. arXiv: 1904.01040 (2019)
and
Chen, K., Kong, J., Zhu, J., Ermann, N., Predki, P. & Keyser, U. F. Digital Data Storage Using DNA Nanostructures and Solid-State Nanopores. *Nano Letters* **19,** 1210–1215 (2019)

The previous chapter discussed experimental methods to aid the nanopore-based readout of modifications along a DNA strand. A complementary factor is the analysis of current data to obtain the encoded information. A number of approaches have emerged for the analysis of solid-state nanopore data. Pedone *et al.* [93] and Balijepalli *et al.* [94] developed methods to discern short translocations affected by the finite response time of the electronic circuit. Raillon *et al.* [95] introduced a cumulative sum algorithm to track multiple current steps during a translocation. This approach was extended by Forstater *et al.* [96] for longer event durations as part of a larger analysis package targeted at nanopore data. Plesa & Dekker [97] similarly developed ready-to-use software, in this case with an emphasis on improved event detection in noisy current baselines as well as peak finding. In this chapter and the next I will outline Bayesian inference and deep learning as two powerful analysis methods to reveal information encoded in modifications along a DNA strand. Before illustrating the Bayesian approach with an example, the next section introduces the method's theoretical background.

4.1 Theory

This section has been adapted from Ermann et al. *[2].*

Bayesian techniques emerge from a fundamental statement about conditional probabilities called Bayes' rule [98, 99]:

$$\underbrace{P(\theta|D,M)}_{\text{posterior}} = \frac{\overbrace{P(D|\theta,M)}^{\text{likelihood}}\overbrace{P(\theta|M)}^{\text{prior}}}{\underbrace{P(D|M)}_{\text{evidence}}} \tag{4.1}$$

where M represents a model relating parameters to the experimental data, θ denotes the model parameters and D the experimental data. Bayes' rule states that the probability for the model parameters to take on a certain value given the data, called the posterior, is proportional to the probability of obtaining the data from the given choice of parameters, called the likelihood. While the two probabilities sound similar, this is in fact a powerful statement: in data analysis we are interested in the posterior, i.e. we want to know what the data can tell us about the parameters of our explanatory model. Bayes' rule allows us to obtain the posterior by relating it to a probability we can calculate, the likelihood. Using a nanopore signal as an example, the current trace exhibits Gaussian noise with variance σ_n^2. We can represent any model M describing the data with a function $h(\theta)$ which defines how the model relates its parameters θ to the observed signal. The likelihood for an individual data point d given θ and M is then simply proportional to the Gaussian noise probability distribution centred around d at the point predicted by $h(\theta)$, that is

$$P(d|\theta,M) = \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left(-\frac{(h(\theta)-d)^2}{2\cdot\sigma_n^2}\right) \tag{4.2}$$

Having obtained an expression for the likelihood, the remaining term with a dependence on the parameters θ is the prior $P(\theta|M)$. It encodes previous knowledge about the parameters, for example from previous experimental data [100]. A lack of initial information about θ is easily represented by a uniform probability distribution. The prior in combination with the likelihood lets us calculate the quantity of interest, namely the posterior probability over the parameter values given the data.

Up to this point, the discussion has focused on determining the parameter probability distribution for a given model. However, within the probabilistic framework we can equally ask which of several models is most likely given a set of experimental data. To answer this question we need to calculate $P(M_i|D)$ where i indexes several competing models. Bayes' rule tells us that

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \propto P(D|M)$$

In the last term of the equation we recognise the evidence $P(D|M)$ which appears in Equation 4.1. Assuming uniform priors $P(M_i)$ over different models M_i, the model with the highest probability of explaining the data is thus the one with the largest evidence [101]. To compute the evidence, we note that

$$P(D|M) = \int P(D,\theta|M)d\theta \tag{4.3}$$

$$= \int \underbrace{P(D|\theta,M)}_{\text{likelihood}} \underbrace{P(\theta|M)}_{\text{prior}} d\theta \tag{4.4}$$

The evidence is therefore an integral of likelihood and prior over the entire parameter space. Particularly for high-dimensional models where analytical optimisations are not possible, it is these integrals which make Bayesian techniques computationally intensive. However, the computational constraints have been partially overcome in recent years by the development of efficient Markov chain Monte-Carlo (MCMC) algorithms [102] and the continuous increase in hardware performance. Once the computation becomes feasible, the evidence value allows a probabilistic comparison of the explanatory power of different models. The following sections assess whether this approach can be useful for nanopore data.

4.2 *DNA folding*

One application for Bayesian inference is the analysis of DNA folding. As outlined in chapter 3, DNA strands translocating through a glass capillary nanopore can do so either single-file or in a folded conformation in which part of the strand folds back onto itself (see Figure 3.1). The classification of events as folded or unfolded has previously been based on simple current thresholds, e.g. in [51, SI]. While this is sufficient to filter out folded translocations, it relies on manually set, arbitrary thresholds which can skew the result when the folding state itself is of interest. Bayesian inference offers an alternative by providing a decision criterion whether the current data is better described by a folded or unfolded model. Figure 4.1 shows a sample event and the two models[1]. For single-file translocations, the current is

[1] The model fits result from setting the parameters to weighted means of the points sampling the posterior distribution, see section 3.3 in [103] for details.

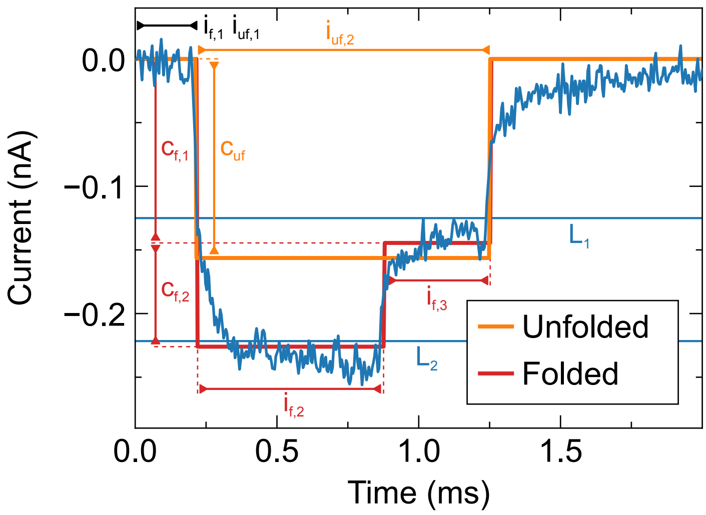


Fig. 4.1 Models for unfolded and folded events. Current levels are described as Heaviside step functions according to equations 4.5 and 4.6. As expected for this folded event the folded model produces the better fit, see the main text for how the fit is obtained.

modelled by a step after a time $i_{\mathrm{uf},1}$ followed by the same but inverted step after an additional increment $i_{\mathrm{uf},2}$, given by

$$h_{\mathrm{unfolded}}(t, i_{\mathrm{uf},1}, i_{\mathrm{uf},2}, c_{\mathrm{uf}}) = \Theta(t, i_{\mathrm{uf},1}, c_{\mathrm{uf}}) + \Theta(t, i_{\mathrm{uf},1} + i_{\mathrm{uf},2}, -c_{\mathrm{uf}}) \tag{4.5}$$

where t is the time, $i_{\mathrm{uf},1}$ and $i_{\mathrm{uf},2}$ the time increments, c_{uf} the current step and Θ the Heaviside step function

$$\Theta(t,i,c) = 0.5 \cdot c \cdot (\mathrm{sgn}(t-i)+1)$$
$$\text{with } \mathrm{sgn}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0 \end{cases}$$

The model for folded events contains on additional current step at the start of the event which is reversed after an increment $i_{\mathrm{f},2}$:

$$h_{\mathrm{folded}}(t, i_{\mathrm{f},1}, i_{\mathrm{f},2}, i_{\mathrm{f},3}, c_{\mathrm{f},1}, c_{\mathrm{f},2}) = \Theta(t, i_{\mathrm{f},1}, c_{\mathrm{f},1} + c_{\mathrm{f},2}) + \Theta(t, i_{\mathrm{f},1} + i_{\mathrm{f},2}, -c_{\mathrm{f},2}) \tag{4.6}$$
$$+ \Theta(t, i_{\mathrm{f},1} + i_{\mathrm{f},2} + i_{\mathrm{f},3}, -c_{\mathrm{f},1}) \tag{4.7}$$

While these models may seem crude and do not closely follow the current particularly at the end of the translocation, it must be kept in mind that the goal is to distinguish folded from unfolded events by comparing the model evidences. This means that their exact fit is not critical as long as they accurately represent the folding state. Having defined the model functions which relate parameters to expected measured data, we obtain the likelihoods for the folded (M_{f}) and unfolded (M_{uf}) models from equation 4.2 as

$$P(D|\theta, M_{\mathrm{f/uf}}) = \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left(-\sum_{k=1}^{L} \frac{(h_{\mathrm{f/uf}}(t_k,\theta) - d_k)^2}{2\cdot\sigma_n^2}\right) \tag{4.8}$$

where t_k is the k-th time point, d_k the k-th element of the current trace, L the length of the trace, σ_n the current noise and θ denotes the remaining model function parameters. The sum in the exponential results from the assumption of independent and identically distributed (i.i.d.) samples which allows expressing the likelihood as a product over Gaussian noise distributions for each data point. Following equation 4.4, the second requirement for the evidence calculation are the priors $P(\theta|M_{\mathrm{f/uf}})$. They specify any previous information we may have about the model parametrisation. In the case of DNA folding, one example of such information are the characteristic current levels produced by one or two strands occupying the pore. We can determine these levels for each pore by concatenating all the event currents and identifying the corresponding peaks in the histogram. The horizontal blue lines in Figure 4.1 indicate the levels L_1 and L_2 corresponding to one and two strands respectively. Comparison with the models shows that they directly inform our expectation of where current steps should lie: for folded events, we expect $c_{\mathrm{f},1}$ and $c_{\mathrm{f},2}$ to be centred around L_1 and $L_2 - L_1$ respectively. The priors could reflect the empirical distribution obtained from the current histogram, for simplicity I chose uniform priors with fixed bounds given by

$$P(c_{\mathrm{f},1}|M_{\mathrm{folded}}) = \begin{cases} (1.25\cdot L_1 - 0.75\cdot L_1)^{-1} & \text{for } 0.75\cdot L_1 < c_{\mathrm{f},1} < 1.25\cdot L_1 \\ 0 & \text{otherwise.} \end{cases}$$
$$P(c_{\mathrm{f},2}|M_{\mathrm{folded}}) = \begin{cases} (1.25\cdot \Delta L - 0.75\cdot \Delta L)^{-1} & \text{for } 0.75\cdot \Delta L < c_{\mathrm{f},2} < 1.25\cdot \Delta L \\ 0 & \text{otherwise.} \end{cases}$$
$$\text{where } \Delta L = L_2 - L_1$$

The prior for the unfolded current step is equal to the first folded step so that $P(c_{\mathrm{uf}}|M_{\mathrm{unfolded}}) = P(c_{\mathrm{f},1}|M_{\mathrm{folded}})$. The time increments i are constrained to positive values up to the length of the event. Having defined the priors, the evidences can now be calculated from

$$P(D|M_{\text{f/uf}}) = \int P(D|\theta, M_{\text{f/uf}})P(\theta|M_{\text{f/uf}})d\theta$$

This is a 3- and 5-dimensional integral for the unfolded and folded models respectively, determined by the model functions 4.5 and 4.6. I used the *MultiNest* tool which calculates the evidence with a nested sampling algorithm by progressively updating points in parameter space [104]. In addition to computing the evidence, the points obtained from the nested sampling procedure map out the posterior distribution, allowing more informed parameter estimates by showing the surrounding landscape and potentially multi-modal distributions. The algorithm outputs the natural logarithm of the evidence, $\ln P(D|M_{\text{f/uf}})$. To compare the two models, we calculate the difference between the log evidence values $\Delta\ln(Z)$, which corresponds to the log of a probability ratio via

$$\Delta\ln(Z) = \ln(P(D|M_{\text{unfolded}})) - \ln(P(D|M_{\text{folded}})) = \ln\frac{P(D|M_{\text{unfolded}})}{P(D|M_{\text{folded}})}$$

$\Delta\ln(Z)$ immediately gives a decision criterion for an event's folding state. A value > 0 means the data is better described by the unfolded model, while a value < 0 indicates a preference for the folded state. Figure 4.2 shows the outcome of analysing more than 800 events from a measurement of 8 kbp DNA in 4 M LiCl. The blue curve shows $\Delta\ln(Z)$, events are sorted in descending order. The vertical dashed line marks the point at which values become negative, for all events to the left the unfolded model is preferred, events to the right have higher evidence for the folded model. One notable feature is that positive values are of considerably smaller magnitude than negative ones. This is because the folded model can always assume the same shape as the unfolded one by setting $i_{\text{f},2}$ or $i_{\text{f},3}$ close to zero. Constraining the increments through their priors would mitigate the effect, but has no justification given that very small values are expected for a short fold at the start or the midpoint of a DNA strand respectively. A closer look at the Bayesian evidence reveals that such arbitrary thresholds are not necessary: if two models explain the data equally well, the one with fewer parameters will have the larger evidence, even if the difference may be small. This preference for a simpler explanation, referred to as Occam's razor, results from the evidence integral in equation 4.4. The integral can be interpreted as calculating the expected value of the likelihood function over a probability distribution defined by the prior [104, section 2]. Adding integration dimensions while keeping the likelihood, i.e. explanatory power, constant results in a lower evidence value as the parameter space expands.

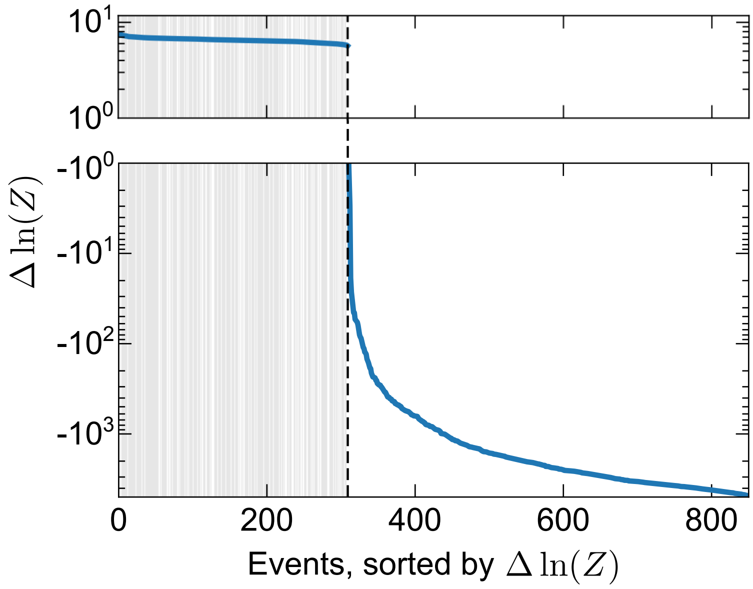


Fig. 4.2 Evidence difference $\Delta\ln(Z)$ between the unfolded and folded models. Events are sorted in descending order by their $\Delta\ln(Z)$, positive values indicate a preference for the unfolded model, negative ones mean a higher evidence for the folded model. The vertical dashed line marks the change in sign, negative values have larger magnitude because the folded model can also describe unfolded events as explained in the main text. The grey shading indicates events for which the current remains above half-way between the one and two strand current levels, suggesting the Bayesian approach correctly distinguishes folded from unfolded translocations.

As there is no source of ground truth to label the events as folded or unfolded, the accuracy of the Bayesian evidence cannot be assessed directly. To obtain an estimate, I compared the results to a simple thresholding method. The grey shading in Figure 4.2 marks events for which the current stayed above the midpoint between the levels for two and one strand in the pore, given by $L_1 + 0.5 * (L_2 - L_1)$. It is clear that all such translocations lie in the region of positive $\Delta\ln(Z)$, indicating that the comparison of Bayesian evidences accurately distinguishes folded from unfolded events. Translocations from the region of positive $\Delta\ln(Z)$ which lack the grey shading, i.e. events which were found to be folded by the simple threshold but unfolded by the Bayesian approach, were mostly due to spurious peaks in the current signal (see appendix H for sample events). Such peaks have previously been attributed to knots in the DNA [105]. Because the Bayesian models constrain the current steps to the ones expected for the respective folding state, the evidence comparison correctly identifies the events as unfolded.

It should be noted that a thresholding method can easily be made more robust to spurious features such as peaks, for example by requiring a certain area to fall below a threshold for folded events. However, the crucial advantage is that the Bayesian approach does away with such arbitrary parameter constraints and simply assesses the evidence that a certain model explains the observed data. This avoids skewing the result with manually extracted features. The individual evidences further act as quality metrics for the models themselves. While the difference indicates the preferred model, low evidence values highlight events which fail to be explained by either model due to more complicated folding or knotting patterns. Lastly, the nested sampling algorithm provides an estimate of the posterior parameter distribution as a side product of the evidence calculation. We thus obtain a wealth of information on which parameter values best describe the current trace beyond simple maximum-a-posteriori estimates.

The key drawback of the Bayesian approach are the computational demands of the evidence integral. While advances in algorithm design and computational power have made the calculation feasible, on a modern CPU (Intel Core i7-6700, 3.4 GHz) obtaining the evidence of the two folding models with *MultiNest* runs at roughly 0.5 events per second. Even with performance improvements through parallelisation and optimised code this will be significantly slower than a simple thresholding algorithm. It must also be noted that the two folding models are of low complexity with only 3- or 5-dimensional parameter spaces. The *MultiNest* algorithm scales exponentially with the number of parameters, alternative nested sampling tools still exhibit cubic power law scaling [106]. The computational time thus grows significantly with model complexity. This is particularly relevant for the analysis of modification peaks within a translocation signal as described in the next section.

4.3 Peak analysis

The previous section outlined the benefits of Bayesian inference when assessing the folding state of DNA passing through a nanopore. A natural follow-on question is whether the technique offers similar advantages when decoding peaks in the current signal due to modifications attached along a DNA strand. Similar to the folding case, the complex behaviour of DNA during translocation [107, 108] suggests the use of phenomenological models to describe the shape of the trace, extending the current steps with peak functions corresponding to the modifications. For full Bayesian model comparison, a separate model has to be defined for each of the modification sequences. It is clear that the evidence computation quickly becomes prohibitively expensive as each added modification doubles the number of models assuming two possible states. An alternative is to define a single model and incorporate parameters describing the presence or absence of a modification. The posterior distribution of the parameters then indicates the modification sequence. While this makes the computational requirements more feasible, it also negates one of the key advantages of the Bayesian approach by removing evidence ratios as a straightforward decision criterion. Given the exponential scaling of the *MultiNest* algorithm with the number of parameters, the computation would remain expensive as each added modification introduces another parameter. I found that deep learning is a promising alternative offering superior performance at high accuracy. The next chapter illustrates the use of neural networks for the analysis of nanopore data.

While the results in the next chapter suggest the use of deep learning to decode modification sequences, the technique depends on the availability of training data. Where such data cannot be measured or simulated, a modified Bayesian approach can still prove useful. Instead of modelling the whole translocation event, one can define models only to describe the modifications. This circumvents the exponential scaling of computational demands, since each modification merely contributes one additional calculation per model. The evidence is calculated for each model as before, providing a decision criterion for which modification is present with the highest probability.

An application of this approach is the analysis of data from a proof-of-concept DNA data storage scheme [3]. Figure 4.3 a) illustrates the design of a 7.2 kbp DNA scaffold decorated with overhangs of two different lengths to form a sequence of 56 bits. Panel b) shows an example current trace with each peak corresponding to either a short or long overhang. The measurement of low-noise translocations was slowed by non-specific interaction with the 5 nm nanopores as well as more difficult fabrication, the small diameter was required to resolve the closely spaced attachments. The dataset thus includes a total of 58 usable events

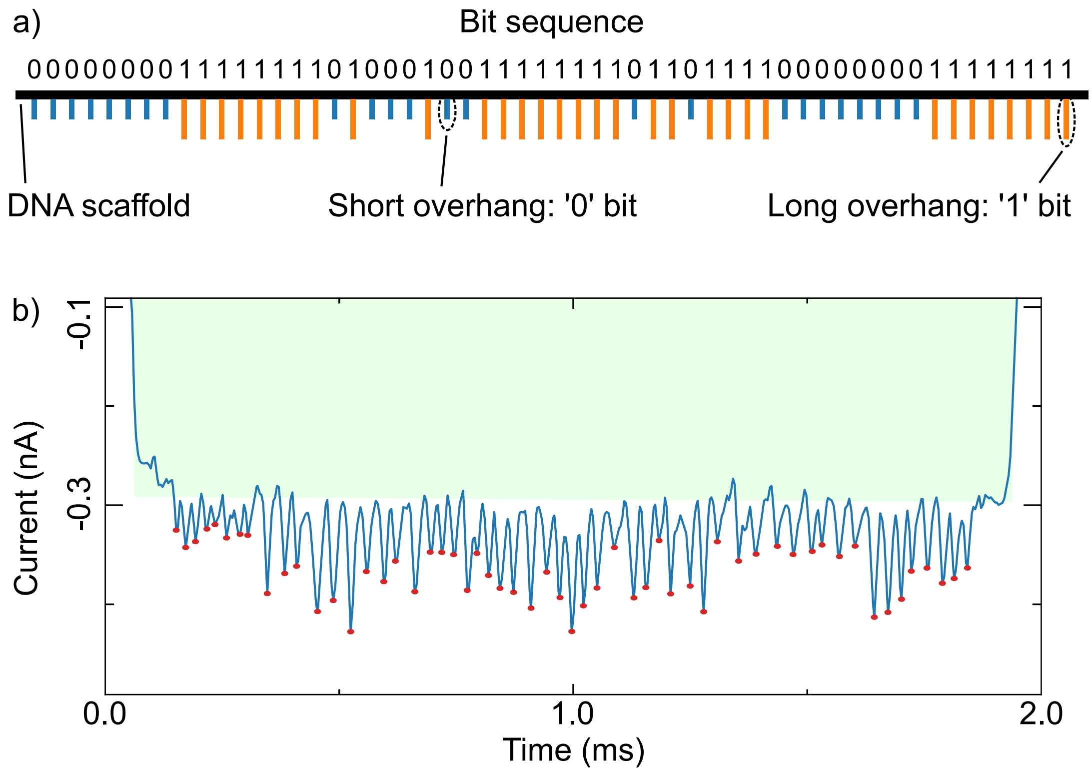


Fig. 4.3 DNA data storage scheme as published in [3]. **a)** Long and short overhangs along a DNA scaffold define a sequence of '1' and '0' bits respectively. Each scaffold contains 56 bits. **b)** Sample current trace from a translocation of the structure in a). The shaded green area shows an intra-event baseline relative to which peaks are measured, see appendix I for details. Red dots mark peaks detected by a peak-finding algorithm adapted from [97], each one corresponds to a single DNA overhang. Figure adapted from [2].

from DNA with the same modification sequence. The analysis task is to find a method which can read this sequence without prior knowledge of its bit content. To do so, an intra-event baseline is first identified relative to which peaks can be measured, shown by the shaded green region in Figure 4.3 b) and described in appendix I. The peaks are modelled by a Gaussian function added to the baseline, given by

$$h(t,\mu,a,\sigma) = a \cdot \exp\left(- \frac{(t-\mu)^2}{2 \cdot \sigma^2} \right) + b \tag{4.9}$$

where b is the baseline and μ, a and σ are the mean, amplitude and width respectively. The model only describes a single peak on top of the baseline current. This means that the priors must constrain the parameters sufficiently to model a single peak at the targeted position. To obtain prior information, the 56 most prominent peaks are detected with a classical algorithm adapted from [97] as shown by the red dots in Figure 4.3 b). The prior for the mean μ is then constrained closely around the position of the targeted peak, while the σ prior is informed by the distribution of peak widths across the dataset. The detected peaks also provide the necessary statistics to define priors on the amplitude a. The distribution of peak distances from the intra-event baseline across all events reveals two populations corresponding to the two overhang lengths, with mean $\bar{m}_1$ and sigma $\bar{w}_1$ for the long overhang and mean $\bar{m}_0$ and sigma $\bar{w}_0$ for the short one. This allows us to base the distinction between the models for long and short overhangs, i.e. '1' and '0' bits, on different prior ranges for the amplitude a, given by

$$P(a|M_j) = \begin{cases} (a_{j,\max} - a_{j,\min})^{-1} & \text{for } a_{j,\min} < a < a_{j,\max} \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{where}$$
$$a_{j,\min} = \bar{m}_j + s_{j,\min} \cdot \bar{w}_j$$
$$a_{j,\max} = \bar{m}_j + s_{j,\max} \cdot \bar{w}_j$$

where $P(a|M_j)$ is the amplitude prior for bit $j \in \{1,0\}$ and the scaling parameters are set to $s_{0,\min} = -0.5$, $s_{0,\max} = +0.5$, $s_{1,\min} = -2$ and $s_{1,\max} = 0$. The likelihood is defined with the assumption of Gaussian noise as in equation 4.8, but with the model function 4.9. For each event, we step along the 56 peaks and calculate the evidence for the '1' and '0' models. Figure 4.4 a) illustrates this process on the same example event as in Figure 4.3 b), the inset shows the two models for neighbouring peaks. The '1' model (dashed line) clearly describes

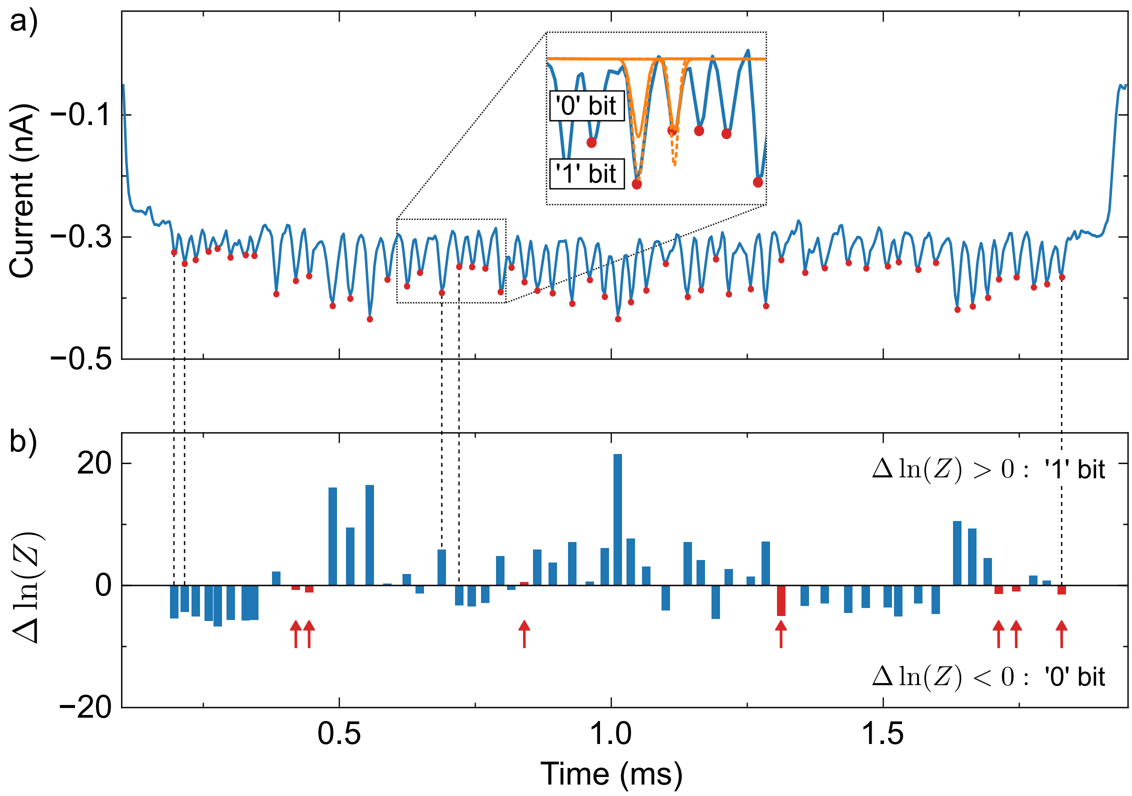


Fig. 4.4 Decoding of a 56-bit sequence. **a)** reproduces the sample event from Figure 4.3 a). The algorithm steps along each previously identified peak and calculates the evidence for a '1' bit and '0' bit model. Comparing the evidence values provides a decision criterion on which bit is present. The inset illustrates this procedure on two peaks, the left one is better described by the '1' model (dashed line), the right one has higher evidence for the '0' model (solid line). **b)** shows the evidence difference $\Delta\ln(Z)$ at each peak position, red bars and arrows mark wrongly decoded bits relative to the known bit sequence. Figure adapted from [2].

the left peak better, while the '0' model (solid line) has higher evidence for the right peak. As for the folding analysis in the previous section, the difference between the evidences given by

$$\Delta\ln(Z) = \ln(P(D|M_1)) - \ln(P(D|M_0)) = \ln\left(\frac{P(D|M_1)}{P(D|M_0)}\right)$$

serves as a decision criterion on which bit is present. Panel b) in Figure 4.4 shows $\Delta\ln(Z)$ for each of the peaks in the sample event. Blue bars correspond to correctly assigned bits, red bars and arrows indicate mismatches with the known bit sequence. For the particular event shown here, 49 out of the 56 peaks are correctly labelled.

There are two issues with the semi-Bayesian approach presented here. Firstly, due to the limited size of the dataset we are overfitting to the particular bit sequence in the sample events, meaning the method may not generalise to arbitrary encodings. While information on the sequence is not included explicitly in the priors, the scaling parameters are clearly set to achieve high accuracy on the specific dataset at hand. Secondly, distinguishing models merely through differently constrained priors can be considered 'un-Bayesian' in itself. The scaling parameters reintroduce arbitrary thresholds which the Bayesian approach aims to avoid through evidence calculation over the whole parameter space. More concretely, a simple threshold on the peak amplitudes would yield the same results as the Bayesian model comparison presented here.

Both of these issues would be solved by more refined models of the current peaks. Gaussians were chosen as the simplest functions to describe spikes in the current, but without physical justification from the underlying translocation process. As more is understood about how different DNA structures produce current features, it will be possible to move away from phenomenological models and towards descriptions motivated by physical behaviour. Even if peaks from differently sized overhangs are found to correspond to different parametrisations of the same model, a physical understanding will allow more informed parameter constraints to distinguish modifications.

Returning to the 56-bit DNA example and keeping in mind the drawbacks described above, the Bayesian approach still has the advantage of evidence values compared to distinguishing bits from a simple amplitude threshold. This means we can build up an aggregate estimate of the modification sequence if all events are expected to originate from the same DNA structure. If $\{D_i\}$ denotes the whole dataset with $i = [0, ..., 58]$ indexing the events, the cumulative evidence difference is given by

$$\begin{aligned}\Delta \ln(Z_N) &= \ln\left(\frac{P(\{D\}|M_1)}{P(\{D\}|M_0)}\right) = \ln\left(\prod_{i=1}^{N}\frac{P(D_i|M_1)}{P(D_i|M_0)}\right) \\ &= \sum_{i=1}^{N}\Big(\ln\big(P(D_i|M_1)\big) - \ln\big(P(D_i|M_0)\big)\Big)\end{aligned}$$

where the product over probability ratios treats each event as an independent sample. Because a DNA strand can enter the pore with either end at the start of a translocation, events are oriented based on the number of '0' bits in the 8 outermost positions before summing the evidence differences. Figure 4.5 shows the cumulative difference for each bit position after $N = 1$, $N = 29$ and $N = 58$ events, wrong assignments are again shown in red. Combining evidences into an aggregate estimate improves the accuracy, with only 2 wrongly identified bits when all 58 events are taken into consideration. The two mistakes occur at positions where neighbouring bits are of different value, indicating shift errors in which a wrongly identified or missed peak disrupts the localisation of all subsequent bits in the sequence. This suggests larger 'beacon' peaks or bit blocks to provide checks on the relative positions at

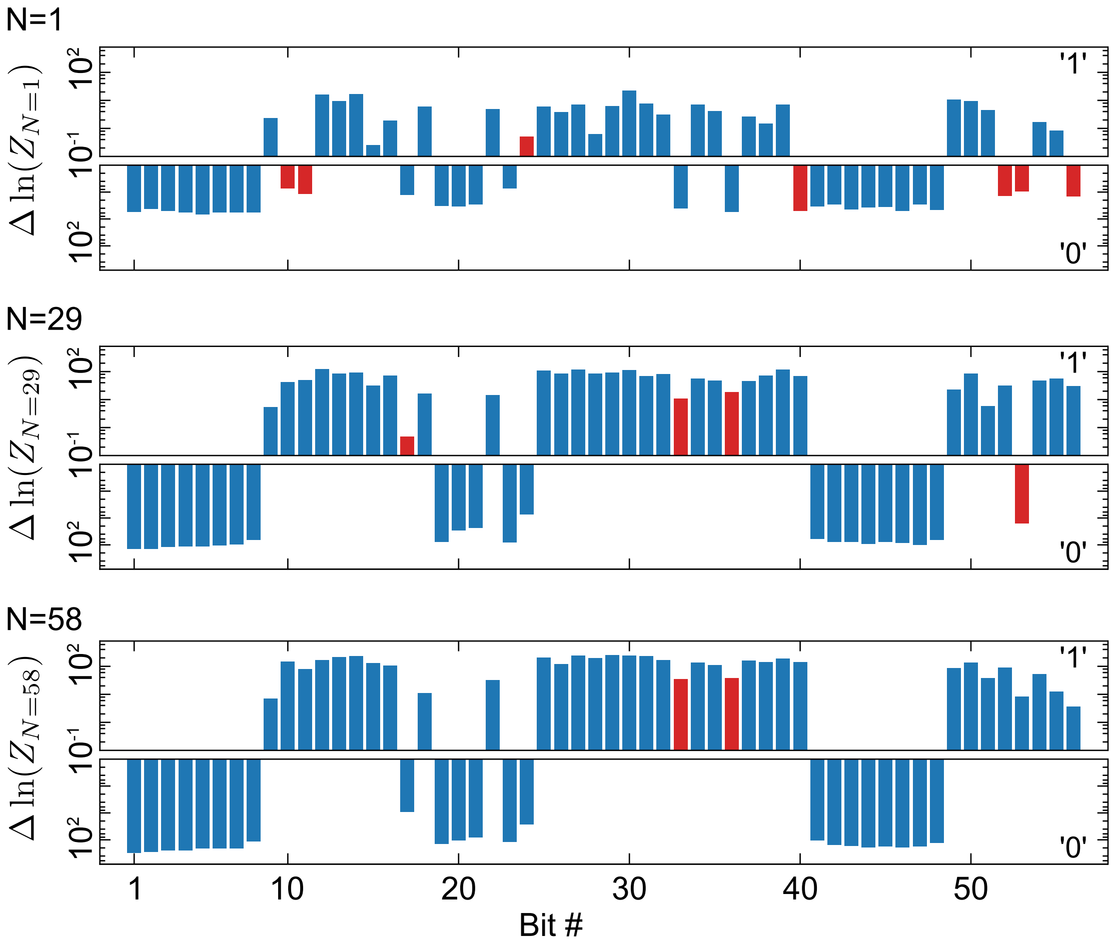


Fig. 4.5 Cumulative evidence across multiple events. The top row reproduces the evidence difference for the sample event from Figure 4.4 b), red bars again show wrongly assigned bits. Combining the values for multiple events improves the accuracy to 4 mistakes after 29 events (middle row) and 2 mistakes for the whole dataset (bottom row). Figure adapted from [2].

regular intervals. Particularly with more refined peak models, Bayesian evidence values can thus inform the design of improved modification sequences.

4.4 Conclusion

This chapter investigated the use of Bayesian inference for the analysis of nanopore translocations. Recently developed nested sampling algorithms allow the calculation of evidence values for arbitrary models, quantifying how well the model describes the experimental data. Comparing the evidences of different models provides a direct decision criterion which one should be preferred. The distinction of folded from unfolded translocations is shown to benefit from the Bayesian approach. Compared to threshold-based algorithms, the relative evidence for a folded and unfolded model avoids skewing the result with arbitrary thresholds. Comparison with an alternative method indicates that the Bayesian approach correctly identifies the DNA folding state. The main drawback is the much slower performance relative to classical algorithms at roughly 0.5 events per second. The high computational demands prevent the use of full model comparison for the readout of modification sequences on a DNA strand. An alternative approach is to compare models only for each individual peak associated with a modification. This technique is employed to successfully decode a 56-bit sequence designed for a proof-of-concept DNA storage scheme. However, unless an improved physical understanding of the translocation process can inform more refined models to describe the current peak produced by a modification, the Bayesian approach loses most of its advantages relative to algorithms based on manual thresholds. In particular where a large amount of training data is available, deep learning offers superior performance at higher accuracy. The next chapter discusses neural networks as an alternative analysis tool for nanopore data.

Chapter 5

Deep learning

An alternative approach to the exact probabilistic calculations in the previous chapter is to train a deep neural network to approximate a desired probability, for example the probability $P(m|C)$ for the presence of a certain molecule m given a current trace C. Initially developed as models of brain function in the 1940s to 1960s, neural networks have experienced a renaissance over the last decade, enabled by the emergence of large datasets in combination with powerful computing hardware [109]. These factors have made deep learning the method of choice for many machine learning tasks, outperforming other techniques in a range of problems from image classification [110] to natural language processing [111]. Many of the recent breakthroughs are based on convolutional neural networks (CNNs), which replace layers of fully connected nodes with nodes taking the result of convolutional operations on the previous layer as their input. This allows CNNs to learn and localise features in the input data at different length scales, with deeper layers (those closer to the output) corresponding to higher-level features.

A key insight for the use of deep learning in single molecule science is that while experiments often suffer from low signal-to-noise ratios and a lack of physical models fully explaining the data, they can be regarded as 'self-labelling' dataset generators. Supervised deep learning requires a set of input data annotated with labels, for example the class an entry belongs to or a continuous value it is associated with. A measurement of a sample containing a single type of molecule directly provides these labels if contamination can be kept negligible. Using nanopore sensing of DNA carriers as an example, we can collect separate translocation data for each of the barcodes we are interested in detecting. We then train the neural network to classify each current trace as one of the distinctly barcoded molecules, i.e. the network learns to estimate $P(m|C)$. This is a shift in philosophy from the Bayesian approach in the previous chapter, where models of the translocation process were assessed as to their probability of having produced a certain current trace. With deep learning, we do not make assumptions about how the data emerged from the underlying process or which features are particularly important. The neural network is presented only with labelled current traces and trained to output labels given a trace. This ‘black box’ approach avoids bias from manual feature extraction and model definition and can be advantageous for sensing applications which only require an accurate mapping between a signal and a molecular species.

The usefulness of CNNs has been demonstrated for simulated single-molecule data [112], but applications focusing on experimental data are slow to emerge [113–115]. In the following I show that the identification of molecular species from nanopore current signals benefits greatly from a deep learning approach. CNNs are introduced in greater detail in the next section, followed by a description of two network implementations for nanopore event analysis.

5.1 Convolutional Neural Networks

At the fundamental level, a deep feedforward network defines a mapping

$$\mathbf{o} = f(\mathbf{x};\theta)$$

between an input vector **x** and an output vector **o**, parametrised by θ [109]. The goal of supervised learning is to find the parameters θ^* so that $f(\mathbf{x};\theta^*)$ approximates the underlying true mapping $\mathbf{y} = f^*(\mathbf{x})$ as closely as possible, where **y** is a vector of targets (or labels) associated with the input **x**. This approximation should be based on all the samples in our dataset.

To train a network we require a metric which represents how far the mapping with a given parametrisation differs from the true mapping, usually called a **loss or cost function** $L(\mathbf{x},\mathbf{y},\theta)$. While the loss can be chosen freely, certain functions are more suitable than others because their gradients remain informative for optimisation even for extreme input values. A common choice of loss function is based on the maximum likelihood principle, that is we find the parametrisation θ^* which maximises the likelihood $P(\mathbf{y}|\mathbf{x};\theta)$ of producing the targets **y** given the inputs **x** for all samples in the dataset [109]. This loss function, called cross-entropy loss, is given by

$$L(\mathbf{x},\mathbf{y},\theta) = -\log P(\mathbf{y}|\mathbf{x};\theta) \tag{5.1}$$

Before discussing the minimisation of the loss function it is instructive to take a closer look at how $f(\mathbf{x};\theta)$ maps inputs **x** to outputs **o**. The standard component in a neural network is

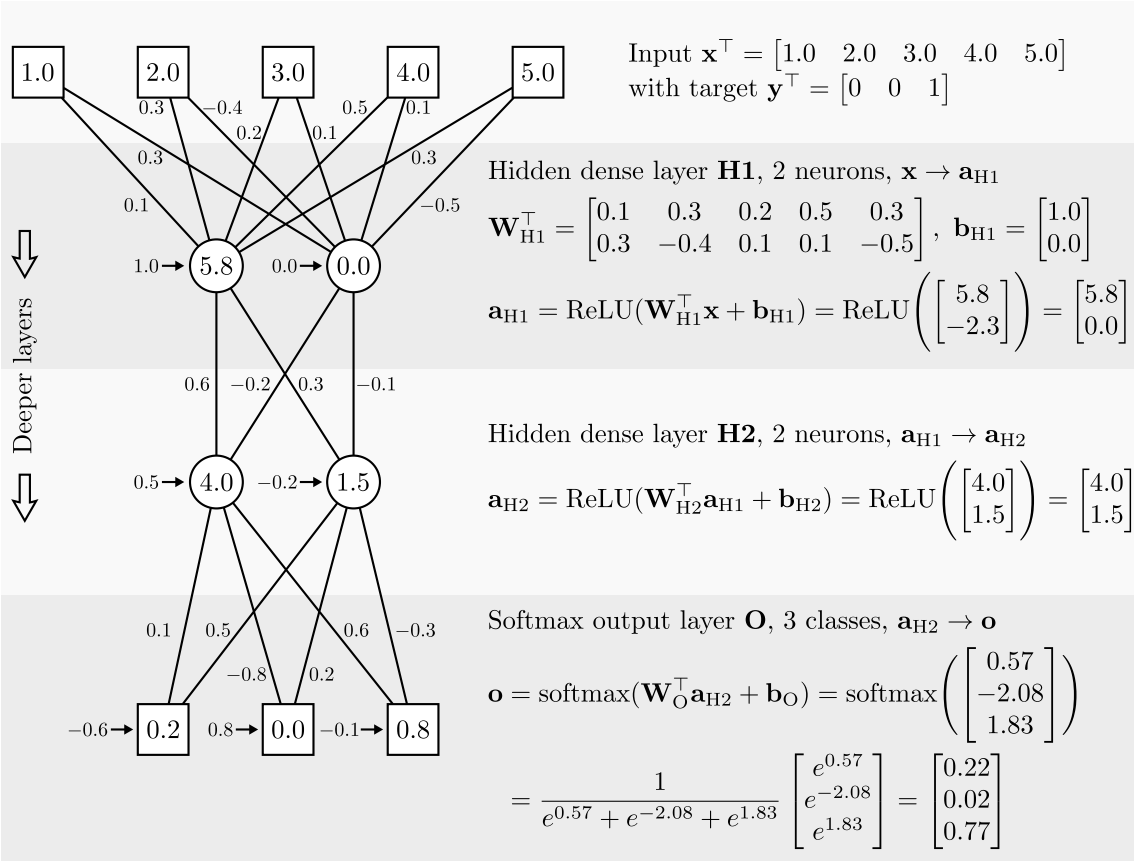


Fig. 5.1 An example neural network consisting of two fully connected hidden layers. The network maps a vector input of size 5 to an output vector whose 3 elements sum to 1 and estimate the probability of the input belonging to one of 3 classes.

the **layer** which consists of several nodes or neurons. Much like the whole network, each layer takes inputs and produces outputs, however depending on the type of layer these can now be of higher dimension and may therefore be represented by tensors. Figure 5.1 shows a simple neural network consisting of input and output layers as well as two fully connected (or dense) layers. The layers between the input and output are referred to as hidden layers, in this case layers H1 and H2. Each neuron in a fully connected layer takes their input from all the outputs of the previous layer, sums up the the inputs multiplied by its weights, adds a bias and applies an activation function. This means that a dense layer turns a vector input into a vector output with the operation

$$\mathbf{a} = g(\mathbf{W}^\top \mathbf{x} + \mathbf{b})$$

where $\mathbf{a}$ is the output vector, $\mathbf{x}$ the input vector, $\mathbf{W}$ a matrix whose i-th column contains the weights for the layer's i-th neuron, $\mathbf{b}$ are the biases and g is the activation function. A

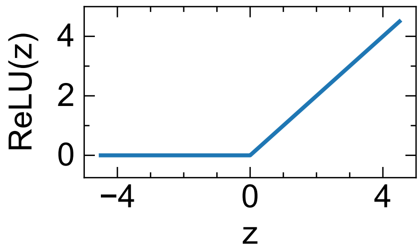


Fig. 5.2 The ReLU activation function applied to layer outputs allows neural networks to learn non-linear mappings between inputs and outputs.

standard choice for the activation function is the **rectified linear unit (ReLU)**, given by

$$\mathrm{ReLU}(z) = \max\{0, z\}$$

shown in Figure 5.2. The activation function's non-linearity ensures that the network is able to map non-linear relationships between inputs and outputs. A forward pass of the network consists of propagating the inputs through all layers to calculate the outputs. Starting with the input shown in Figure 5.1, we can calculate the output of the first hidden layer H1 as

$$\begin{aligned}
\mathbf{a}_{\mathrm{H1}} &= \mathrm{ReLU}(\mathbf{W}_{\mathrm{H1}}^{\top}\mathbf{x} + \mathbf{b}_{\mathrm{H1}}) \\
&= \mathrm{ReLU}\left(\begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.5 & 0.3 \\ 0.3 & -0.4 & 0.1 & 0.1 & -0.5 \end{bmatrix} \cdot \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \\ 5.0 \end{bmatrix} + \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix} \right) \\
&= \mathrm{ReLU}\left(\begin{bmatrix} 5.8 \\ -2.3 \end{bmatrix} \right) = \begin{bmatrix} 5.8 \\ 0.0 \end{bmatrix}
\end{aligned}$$

If $|x|$ is the number of elements in the input vector and n is the number of nodes, a dense layer contains $|x| \cdot n$ weights plus n biases, i.e. a total of $(|x| + 1) \cdot n$ trainable parameters. Following the above procedure we propagate the calculation through the second hidden layer until we reach the output layer O. The activation function in this layer depends on the learning problem we are trying to solve. For a classification task with n classes, we want the output vector to contain n elements each modelling the likelihood of the input belonging to the respective class. Since we are modelling probabilities the entries should lie between 0 and 1 and sum to 1. This can be achieved by using the **softmax** function as the activation

[109], given by

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

With the weights and biases in Figure 5.1 we obtain the elements of the output vector **o** through

$$o_i = \text{softmax}(\mathbf{d})_i = \frac{\exp(d_i)}{\sum_{j=1}^{|\mathbf{d}|} \exp(d_j)}$$
$$\text{where } \mathbf{d} = \mathbf{W}_\text{O}^\top \mathbf{a}_\text{H2} + \mathbf{b}_\text{O}$$
$$\text{for example in Figure 5.1:} \quad o_1 = \frac{e^{0.57}}{e^{0.57} + e^{-2.08} + e^{1.83}} = 0.22$$

Based on the softmax output **o** and the target vector **y**, we now need to calculate the likelihood $P\big(\mathbf{y}|\mathbf{x};\theta\big)$ to determine the loss according to equation 5.1. In classification tasks, the target vector **y** encodes labels in a 'one-hot' fashion, meaning it is a vector of zeroes with a single one entry indicating the class. This lets us model the likelihood with a multinoulli distribution [116, p. 35], i.e. the probability of the model matching an input **x** to a target vector **y** is

$$P\big(\mathbf{y}|\mathbf{x};\theta\big) = \prod_{i=1}^{|\mathbf{y}|} o_i^{y_i}$$
$$\text{so that } L\big(\mathbf{x},\mathbf{y},\theta\big) = -\log \prod_{i=1}^{|\mathbf{y}|} o_i^{y_i} = -\sum_{i=1}^{|\mathbf{y}|} y_i \log o_i$$

where θ contains all the trainable weights and biases which produce the model output **o**. In the example in Figure 5.1, the loss for the given parametrisation is therefore

$$L\big(\mathbf{x},\mathbf{y},\theta\big) = -\sum_{i=1}^{|\mathbf{y}|} y_i \log \frac{\exp(d_i)}{\sum_{j=1}^{|\mathbf{d}|}\exp(d_j)} = -\log \frac{\exp(d_3)}{\sum_{j=1}^{|\mathbf{d}|}\exp(d_j)} \tag{5.2}$$
$$= -\log \frac{e^{1.83}}{e^{0.57} + e^{-2.08} + e^{1.83}} = -\Big(1.83 - \log\big(e^{0.57} + e^{-2.08} + e^{1.83}\big)\Big) \tag{5.3}$$
$$= 0.27 \tag{5.4}$$

where we used the fact that all entries in the target vector **y** are zero except for the third one. The actual value of the loss is arbitrary, its usefulness lies in the fact that a reduction in the loss indicates a more accurate mapping between the input **x** and the target **y**. The goal of network training is to minimise the loss based on the sample dataset. This means exploring the high-dimensional space parametrised by θ for a minimum.

Deep learning solves this optimisation problem with a technique called **stochastic gradient descent** (SGD) [109]. We initially calculate a forward pass through the whole network, arriving at a loss value as shown in equation 5.4. We then compute the gradient of this loss with respect to each trainable parameter by stepping through the network in reverse. The gradient directly informs the change that should be applied to the associated parameter to reduce the loss. A simple updating rule is

$$\theta \leftarrow \theta - \varepsilon \frac{1}{m'} \nabla_\theta \sum_{i=1}^{m'} L\big(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \theta\big) \tag{5.5}$$

where ε is called the learning rate and m' is the size of a subset of training samples called a **minibatch** (or batch) [109]. $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ denote the i-th input and i-th target in the total dataset. This means that each gradient step is only based on a limited number of samples. This 'stochastic' selection makes it possible to train networks on large amounts of data, because each step only sums over the loss from a fixed number of samples and does not depend on the size of the whole dataset. The implementations in the following sections rely on the widely used Adam optimisation algorithm, which aims to improve the simple updating rule in equation 5.5 with an adaptive learning rate [117].

Regardless of the exact updating rule, the SGD approach fundamentally requires knowledge of the loss gradients with respect to the trainable parameters. We calculate the gradients with the **back-propagation** algorithm [118], which takes its name from the fact that we step through the network 'backwards'. Given a parameter somewhere in the network, the chain rule of calculus allows us to derive the gradient of the loss with respect to this parameter. For example in Figure 5.1, the gradient with respect to the leftmost weight in the second hidden layer H2 can be calculated as

$$\frac{\partial L(\mathbf{x},\mathbf{y},\theta)}{\partial W_{\mathrm{H2};1,1}} = \frac{\partial L}{\partial o_1}\frac{\partial o_1}{\partial a_{\mathrm{H2};1}}\frac{\partial a_{\mathrm{H2};1}}{\partial W_{\mathrm{H2};1,1}} + \frac{\partial L}{\partial o_2}\frac{\partial o_2}{\partial a_{\mathrm{H2};1}}\frac{\partial a_{\mathrm{H2};1}}{\partial W_{\mathrm{H2};1,1}} + \frac{\partial L}{\partial o_3}\frac{\partial o_3}{\partial a_{\mathrm{H2};1}}\frac{\partial a_{\mathrm{H2};1}}{\partial W_{\mathrm{H2};1,1}}$$

where

$$\frac{\partial L}{\partial o_i} = -y_i\frac{1}{o_i}$$

$$\begin{aligned}\frac{\partial o_i}{\partial a_{\mathrm{H2};1}} &= \frac{\partial\ \mathrm{softmax}(\mathbf{d})_i}{\partial a_{\mathrm{H2};1}} = o_i\left(\frac{\partial d_i}{\partial a_{\mathrm{H2};1}} - \sum_{j=1}^{|\mathbf{d}|} o_j \frac{\partial d_j}{\partial a_{\mathrm{H2};1}}\right)\\ &= o_i\left(W_{\mathrm{O};1,i} - \sum_{j=1}^{|\mathbf{d}|} o_j W_{\mathrm{O};1,j}\right) \quad \text{with } \mathbf{d} = \mathbf{W}_{\mathrm{O}}^{\top}\mathbf{a}_{\mathrm{H2}} + \mathbf{b}_{\mathrm{O}}\end{aligned}$$

$$\frac{\partial a_{\mathrm{H2};1}}{\partial W_{\mathrm{H2};1,1}} = \begin{cases} a_{\mathrm{H1};1}, & \text{if } (\mathbf{W}_{\mathrm{H2}}^{\top}\mathbf{a}_{\mathrm{H1}} + \mathbf{b}_{\mathrm{H2}})_1 > 0\\ 0, & \text{otherwise}\end{cases}$$

so that

$$\begin{aligned}\frac{\partial L(\mathbf{x},\mathbf{y},\theta)}{\partial W_{\mathrm{H2};1,1}} &= -y_3\frac{1}{\cancel{o_3}}\cancel{o_3}\left(W_{\mathrm{O};1,3} - \sum_{j=1}^{|\mathbf{d}|} o_j W_{\mathrm{O};1,j}\right)a_{\mathrm{H1};1}\\ &= -y_3\left(W_{\mathrm{O};1,3} - o_1 W_{\mathrm{O};1,1} - o_2 W_{\mathrm{O};1,2} - o_3 W_{\mathrm{O};1,3}\right)a_{\mathrm{H1};1}\\ &= -\left(0.60 - 0.22\cdot 0.10 - 0.02\cdot(-0.80) - 0.77\cdot 0.60\right)\cdot 5.80 = -0.76\end{aligned}$$

The back-propagation algorithm recognises that many derivatives appear multiple times and can be stored without having to be recalculated, making it scalable to large networks [109]. If we used this single example to train the network with a learning rate of $\varepsilon = 0.1$, the update rule 5.5 would tell us to change the weight according to

$$W_{\mathrm{H2};1,1} \leftarrow W_{\mathrm{H2};1,1} - \varepsilon\frac{\partial L(\mathbf{x},\mathbf{y},\theta)}{\partial W_{\mathrm{H2};1,1}} = 0.60 - 0.1\cdot(-0.76) = 0.67$$

resulting in a reduced loss of $L(\mathbf{x},\mathbf{y},\theta) = 0.21$. To complete one training step we carry out this procedure for every trainable parameter in the model while assuming all other parameters remain the same.

The training goal is to arrive at a set of parameters θ^* which correspond to a minimum in the loss. Convergence towards a minimum, however, is not guaranteed and an area of active research. One successful strategy to improve convergence is **batch normalisation** [119]. For batch normalisation in a layer we collect the inputs into the activation function

$\mathbf{d} = \mathbf{W}^{\top}\mathbf{h} + \mathbf{b}$ across all samples in a minibatch and calculate their mean μ and standard deviation σ. We then normalise the inputs according to $\frac{d_i - \mu_i}{\sigma_i}$ where d_i is the i-th input. This has the effect of stabilising the learning procedure and improving convergence.

A key requirement for a neural network is its ability to generalise, that is to perform equally well on previously unseen data based on the features learned during training. We check this by training a model on a random chunk of the total dataset (the **training set**) and subsequently measuring its performance on the remainder (the **test set**). A failure to generalise, called overfitting, manifests itself as a higher loss on the test relative to the training set. **Dropout** is an important strategy to improve generalisation [120]. It involves randomly setting a number of layer outputs to zero during training. This can be thought of as training a slightly different model on each iteration, which makes the underlying base model less prone to overfitting.

In addition to dense layers, convolutional neural networks contain **convolutional layers** [121]. As opposed to connecting every input with each output of the previous layer, convolutional layers calculate their output by convolving their inputs with a kernel of a given size. In a 1-dimensional convolution, the elements of the output tensor are given by

$$\mathbf{A}_{i,j} = \sum_{m=1}^{M} \sum_{l=1}^{L} \mathcal{W}_{l,m,i} \mathbf{H}_{l,m+j-1} + \mathbf{b}_i$$

where M is the kernel size, L is the number of rows in the input, $\mathbf{H}$ is the input tensor, $\mathcal{W}$ is a tensor containing the kernel weights and $\mathbf{b}$ contains one bias per kernel. Figure 5.3 illustrates this operation with two consecutive convolutional layers. The first layer C1 applies 3 kernels of size 3 to a vector input $\mathbf{x}$. The weights of the kernels are learned during network training, resulting in $3 \cdot 3 = 9$ trainable kernel parameters. To obtain the output, we step each kernel along the input and sum the element-wise multiplication of the kernel with the input elements at every position. Padding the input with zeroes on either side ensures that the output has the same number of columns as the input. The number of rows in the output is equal to the freely choosable number of kernels. The second layer C2 again shows a 1-dimensional convolution, but operates on a 2D-tensor input. This means that each kernel is now also a 2D-tensor, resulting in a larger number of trainable kernel weights $2 \cdot 3 \cdot 3 = 18$. It should be noted that the example in Figure 5.3 omits the activations functions in both layers for clarity, these would usually be applied to the results of the convolutional operation $\mathbf{A}_{\mathrm{C1}}$ and $\mathbf{A}_{\mathrm{C2}}$.

A concept closely related to convolutional layers is **pooling** [109]. This refers to reducing the dimensionality of the convolution results by combining regions into a single value. A **max pooling** layer, for example, tiles the input with a mask of a certain size and takes only

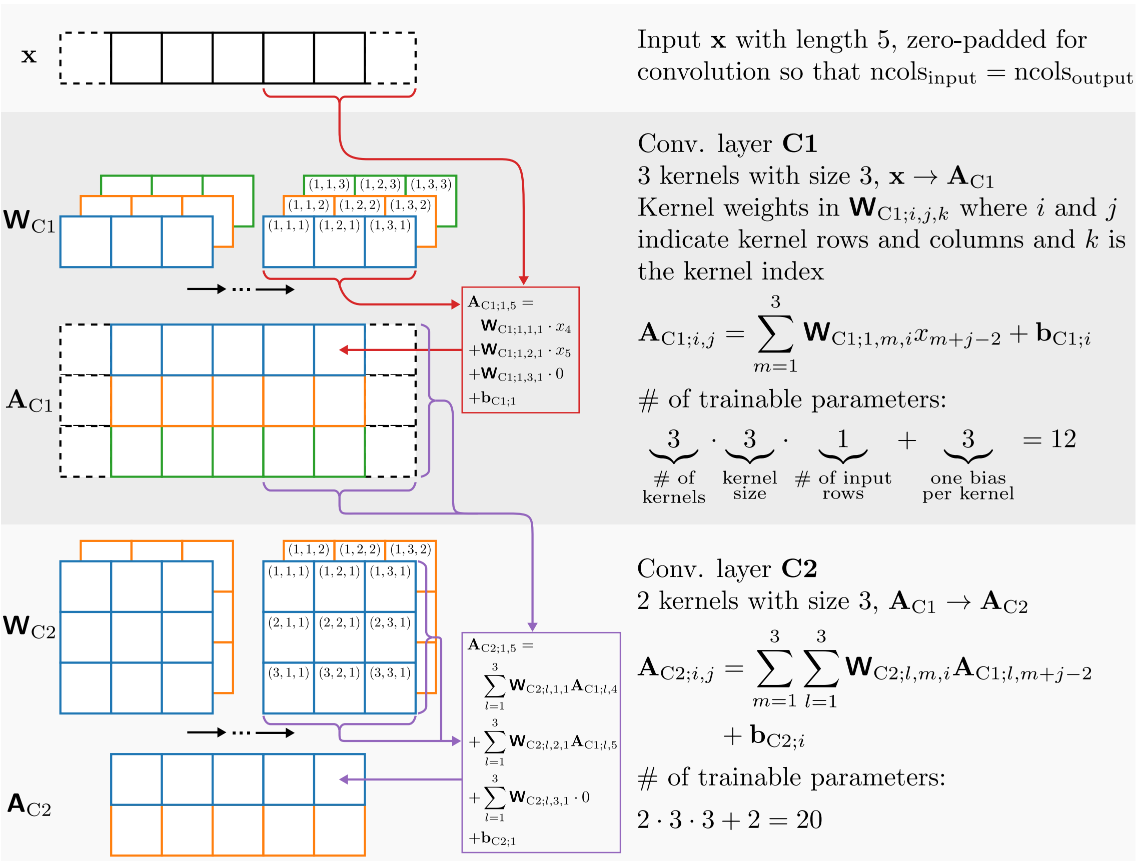


Fig. 5.3 An example of 1-D convolutional layers. Such layers shift a kernel along their input in a convolutional manner to produce an output. The 3 kernels in the first layer C1 transform a 5-element input into a $(3,5)$ tensor, which the 2 kernels in layer C2 subsequently reduce to a $(2,5)$ tensor. Kernel weights are learned during training and remain the same at each input position. This allows convolutional neural networks to identify features across the input. Note that activation functions have been omitted for clarity, they would usually be applied to the layer outputs $\mathbf{A}_{\mathrm{C1}}$ and $\mathbf{A}_{\mathrm{C2}}$.

the maximum values within each mask while discarding the rest. The effect is translational invariance to small shifts, so that the output remains the same when features in the input move.

Convolutional layers can be thought of as a constraint on the network architecture which improves learning for certain tasks. Because each kernel uses the same weights across the input, it can learn to identify features at any position in the input. In nanopore translocation data, such a feature could for example be the peaks produced by modifications along a DNA strand. An added benefit of the convolutional approach is the reduced number of trainable parameters compared to dense layers, which improves training efficiency. Convolutional layers have proved highly successful for a variety of applications and will be used in both the architectures introduced in the following.

5.2 *Event classification*

Parts of this work have been published in:
Misiunas, K., Ermann, N. & Keyser, U. F. QuipuNet: Convolutional Neural Network for Single-Molecule Nanopore Sensing. *Nano Letters* **18,** 4040–4045 (2018)

Dr Karolis Misiunas designed the network and tested it on the Bell & Keyser dataset [51].

As mentioned in the introduction to this chapter, single-molecule techniques are a promising area of application for deep learning because measurements of individual molecular species generate 'self-labelling' datasets. Nanopore translocation experiments are particularly suitable because they create a large number of samples in a relatively short time: in the right conditions event frequencies of around 1 Hz can be maintained for hours. One ideal dataset to test deep learning performance on nanopore data is Bell & Keyser's barcoding scheme which encoded information in modifications along a DNA strand [51]. Dumbbell structures at 5 defined positions along the contour serve as 'bits', with the 2 outer attachments acting as a 'frame' and the presence or absence of the inner three encoding $2^3 = 8$ distinct bit sequences (Figure 5.4). As part of this work the authors separately measured at least 1000 events for each of the 8 molecular barcodes, providing a large dataset of fully labelled translocations. For some of the barcodes, a protein is bound in a different section of the DNA strand to demonstrate the sensing scheme. The resulting extra peak will be ignored here to focus on the barcode classification problem.

Columns 2 and 4 in table 5.1 show the number of events for each barcode along with the number of nanopores they were measured on. In addition to Bell & Keyser's data, I carried

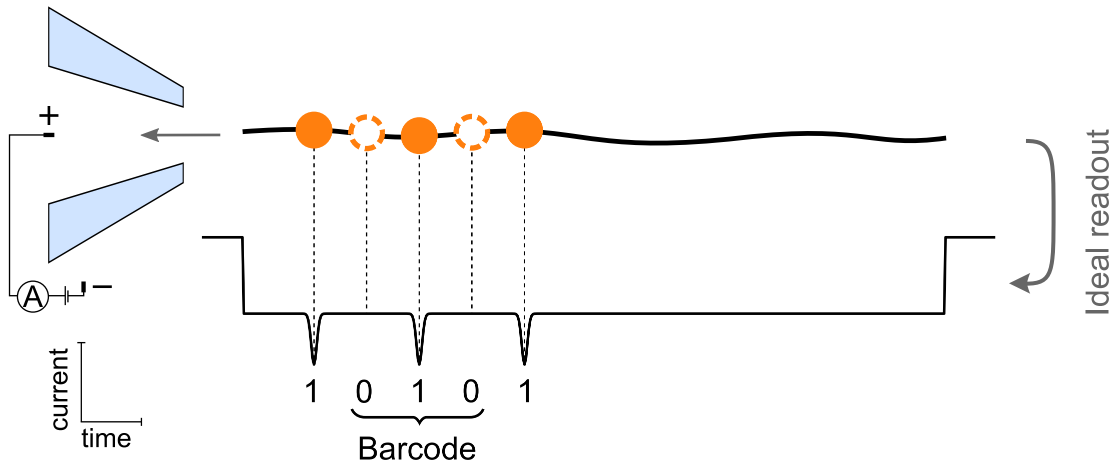


Fig. 5.4 DNA barcode design by Bell & Keyser [51]. 5 modifications in one half of the DNA strand produce secondary peaks in already reduced current during translocation. The 2 outer modifications serve as boundary markers while the presence or absence of the 3 inner ones represent '1' and '0' bits.

	Bell & Keyser [51]			**Additional data**		
Barcode	total	filtered	# pores	total	filtered	# pores
000	8329	6084	5	3149	2277	4
001	14006	9005	7	-	-	-
010	3420	2519	4	-	-	-
011	24239	16609	14	2664	1740	5
100	1407	1002	3	-	-	-
101	11061	7978	6	1779	1244	2
110	10185	7337	5	-	-	-
111	11784	7644	7	674	460	3

Table 5.1 Dataset of barcode DNA used to train and test the classification network. A filtering step excludes events based on their electronic charge deficit (ECD) and minimum current. The Bell & Keyser columns correspond to the data published in [51], I measured additional data on the pressure setup described in section 2.2.2.

out additional measurements for the '101', '011', '000' and '111' barcodes on the pressure setup described in section 2.2.2. Their numbers are shown in columns 5 and 7. Each event in the dataset contains the translocation current trace, a barcode label, a quality metric and the value of the unfolded current level corresponding to one double strand of DNA in the pore. The unfolded level per nanopore is determined from the peak in the histogram of all concatenated event currents. The quality metric indicates whether the event has successfully passed two filtration steps as outlined in [4]: first, we exclude translocations whose electronic charge deficit (ECD) lies more than two standard deviations from the mean. This removes signals due to fragments or sticking on the pore surface. Second, we remove events whose minimal current lies more than $3.2\times$ below the unfolded level to clear translocations with abnormally large peaks. Columns 3 and 6 in table 5.1 show how many events remain after the filtration steps.

The task for the neural network presented in this section is to classify each current trace by assigning it to one of the 8 possible barcodes. More precisely, we want to approximate the probability $P(b|C)$ that the molecule b passed through the pore given a current trace C, where $b \in B$ and B is the set of the 8 barcodes. We chose a network architecture of 6 convolutional layers followed by two fully connected layers as shown in figure 5.5 [4]. Each convolutional layer performs batch normalisation and has a ReLU activation function. The number of convolutional kernels increases from 64 to 128 to 256 every two layers, while the kernel size drops from 7 to 5 to 3. Deeper layers thus capture higher level features in the input. After every two convolutional layers a max pooling layer of size 3 samples down the temporal dimension, introducing invariance to small shifts in peak positions. During training, dropout on 25 % of nodes is carried out after each pooling layer to improve network generalisation. Two fully connected layers with 512 nodes and ReLU activation are attached to the convolutional base network, followed by an output layer with softmax activation for the 8 barcode classes.

The grey boxes in figure 5.5 show how the layers modify the input tensor during the forward pass through the network. The input is a tensor of shape $(32,700,1)$, where 700 corresponds to the length of the event current trace and 32 is the size of the minibatch of events processed per gradient update. We chose 700 as the length of the temporal dimension to fit all events in the dataset. Shorter events are padded with average baseline noise levels (Gaussian noise with $\mu = 0$ and $\sigma = 0.06$). Different pore sizes and shapes directly influence the current drop produced by an unfolded double strand of DNA. Because each pore in the dataset was used to measure a single barcode, this can cause the network to learn identifying pores instead of barcodes. In addition to training on data from several pores, we reduced such overfitting by normalising the current trace for each event by dividing by the unfolded

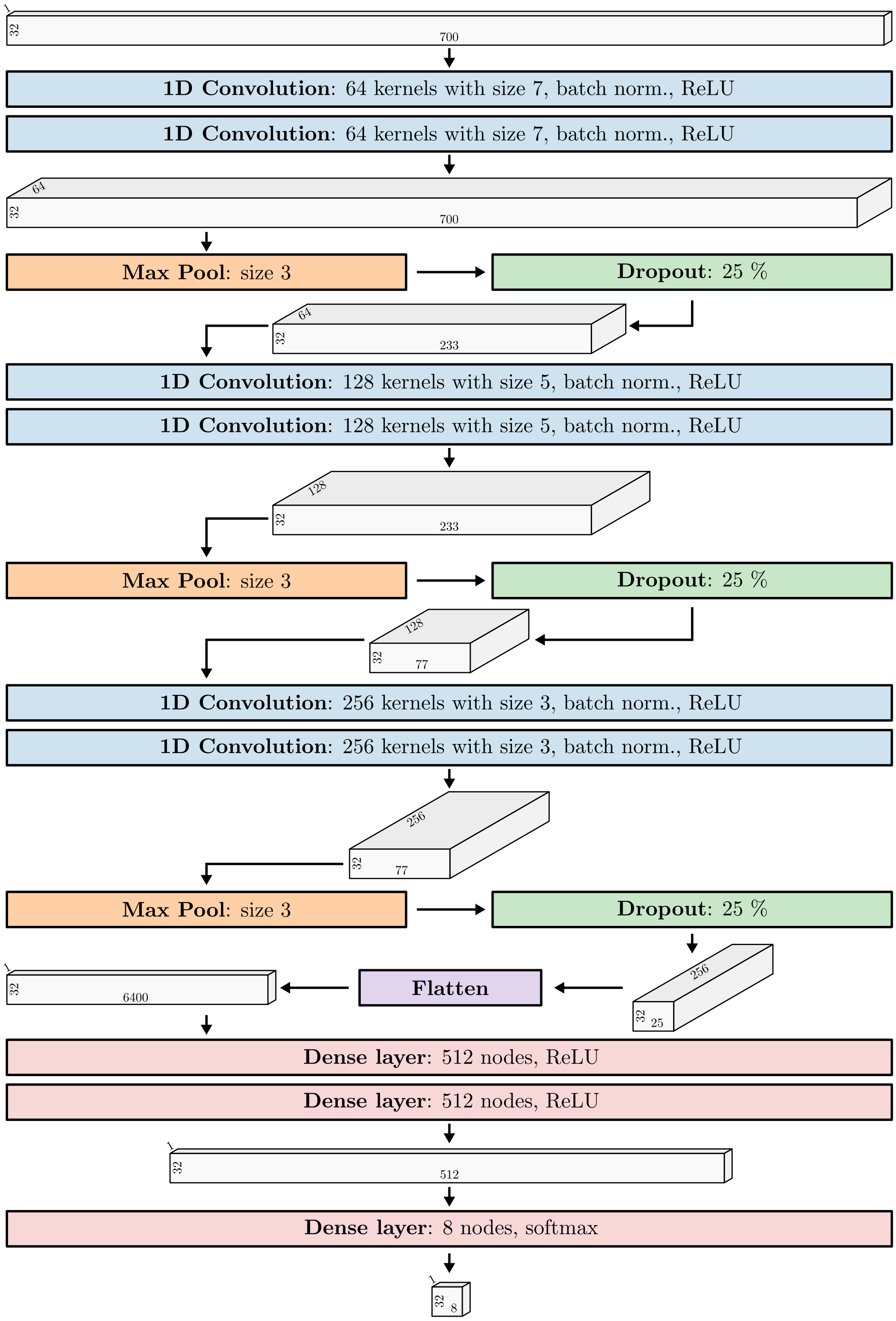


Fig. 5.5 Classification network architecture from [4]. The base network consists of 3 sets of 2 convolutional layers followed by dropout and max pooling. Two dense (or fully connected) layers pass their input to a final dense output layer with softmax activation. The network maps a normalised translocation current trace to a vector whose 8 elements correspond to likelihoods of belonging to the 8 barcodes.

	Bell & Keyser data [51]	
	precision (%)	data utilised (%)
Bell & Keyser method [51]	93.7	19.4
Human classification	97.8	45.0
Conv. network (all data)	94.2	100.0
Conv. network (best 80%)	98.8	80.0

Table 5.2 Comparison of network performance with the thresholding algorithm used by Bell & Keyser [51] and manual labelling by Dr. Karolis Misiunas. Precision refers to the share of correctly assigned events. The classification network achieves higher precision than the previous approach while discarding less data.

level for the pore. This results in a range from 0 to -1 between the baseline and the unfolded DNA current. As the input tensor moves through the network, each kernel in a convolutional layer increases the last dimension by one, while max pooling layers reduce the temporal dimension. After the base network of 6 convolutional and 3 max pooling layers, the input has been transformed into a tensor of shape $(32, 25, 256)$. Following the 2 fully connected layers the final network output is a tensor of shape $(32, 8, 1)$, where the 8 values in each row sum to 1 and estimate the probability of the event belonging to the respective class.

We train the network for 200 epochs, where each epoch corresponds to a run through the training set [4]. We use the Adam optimisation algorithm [117] with a learning rate of 0.001, reduced by 3 % after every epoch and operating on a cross-entropy loss function. A full training run on all 54 714 samples on an Nvidia GeForce 1080 GTX Ti graphics card takes roughly 2.3 h. Classification runs at 1600 events per second.

On the Bell & Keyser dataset, the convolutional network achieves a precision of 94.2 % for the barcode classification task[1]. This compares to 93.7 % for the threshold-based approach used by Bell & Keyser and 97.8 % for manual labelling (by Dr. Karolis Misiunas) as shown in table 5.2. Precision refers to the share of correctly classified events, showing that our deep learning approach slightly outperforms the Bell & Keyser thresholding algorithm [51] and comes within range of classification by a human expert.

The true strength of the neural network becomes apparent when considering the fraction of data which can be reliably analysed. To calculate the 'data utilised' metric, we use the maximum value in the network output as a confidence indicator for the respective classification. We then sort all events from highest to lowest confidence, which allows us

[1]The small difference to the precision of 94.6 % reported in the published version ([4]) is due to the random initial weight settings. This means that subsequent training runs of the same model on the same data may produce slightly different results.

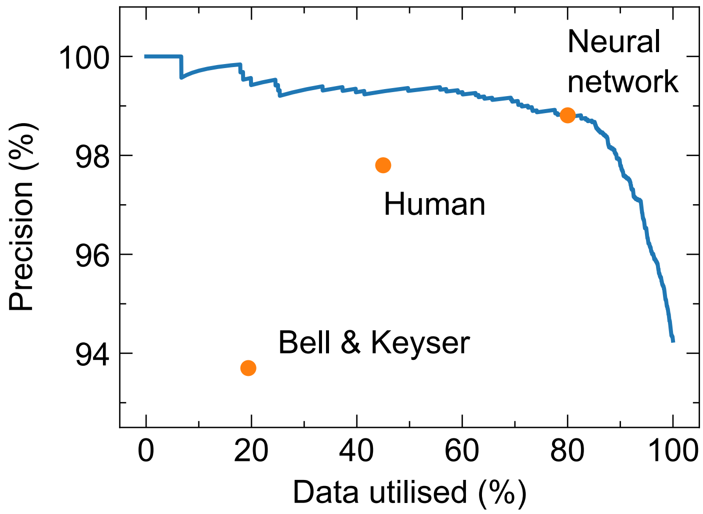


Fig. 5.6 Classification precision versus data utilised for the neural network, the previously used algorithm and manual labelling. The blue curve shows the network performance as more low-confidence events are included, up to 100 % of events. At 80 % data utilised the network outperforms human classification while including 5 times more events than the algorithm used in Bell & Keyser [51]. Adapted with permission from [4], copyright 2018 American Chemical Society.

to define a cut-off below which classifications are considered unreliable. Including only the 80 % most confident predictions boosts precision to 98.8 %, above the performance of a human expert. Crucially, both the Bell & Keyser algorithm and human labelling only use 19.4 % and 45.0 % of the data respectively. Figure 5.6 illustrates the network precision as a function of the utilised data as well as the performance of the other methods. It is clear that at comparable precision, the neural network increases the amount of usable data by a factor of 5 compared to the classical algorithm and by a factor of 2 relative to classification by a human expert.

The main reason for this striking difference is the network's ability to classify the barcode of folded events. Figure 5.7 shows a number of sample translocations. Out of the 12 events, only those shaded in blue could be analysed by the Bell & Keyser algorithm, while the neural network correctly labelled all of them. Detecting peaks in the folded section of a DNA strand is difficult for algorithms relying on manual feature extraction and thresholding. A neural network, on the other hand, is able to learn a possibly complicated combination of features, such as where peaks may appear in a folded section and how they affect the ECD even when not directly visible. While this is highly beneficial for the classification problem such as the one presented here, the black-box nature of deep learning means that it is not directly possible to infer which features are indeed relevant.

In addition to the overall precision, it is also instructive to look at the individual precision for each barcode. The confusion matrix in Figure 5.8 a) shows the fraction of events which

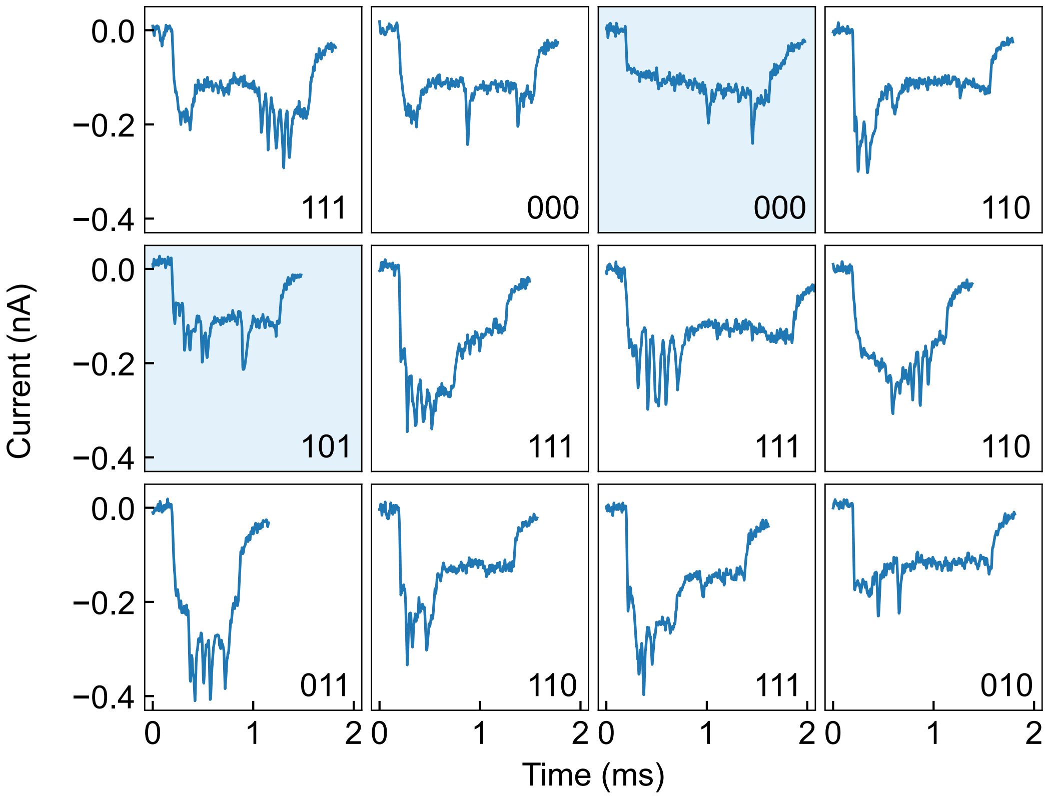


Fig. 5.7 Sample events and the barcode assigned by the neural network in the bottom right corner of each subplot. The Bell & Keyser algorithm [51] does not work on folded events and could only classify the traces shaded in blue. The neural network assigned the correct barcode to all the events shown here. Adapted with permission from [4], copyright 2018 American Chemical Society.

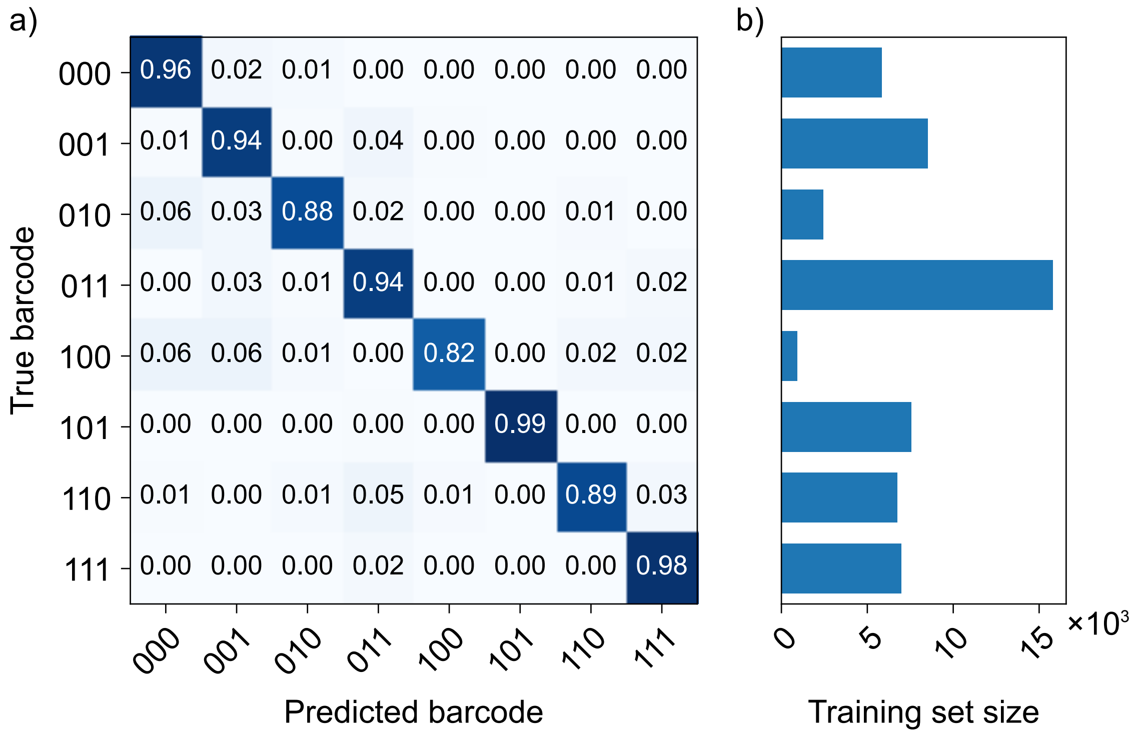


Fig. 5.8 Confusion matrix and training set size for each barcode. Each row in **a)** shows how the network classified events of the respective true barcode. Comparison with the training set sizes in **b)** illustrates that smaller training sets correlate with lower precision. Adapted with permission from [4], copyright 2018 American Chemical Society.

the network assigned to the 8 classes for each of the true barcodes. Comparing the values to part b) of the Figure shows that larger training datasets for a given class increase the precision. The 919 events for the '100' barcode result in a correct fraction of 82 % significantly below the other classes. For barcodes with a training set size of at least 5000, the data further indicates a higher precision for the symmetric barcodes '000', '101' and '111'. These barcodes appear the same irrespective of the entry direction of the DNA strand, suggesting the learning task is simpler without having to learn invariance to temporal inversion.

To further investigate the influence of training set size on precision, the network was trained on varying numbers of samples for the '110' barcode. Figure 5.9 shows that the share of correct '110' classifications increases sharply to $\sim 80\,\%$ for 2000 training samples, followed by a slower rise to $\sim 90\%$ between 5000 and 8000 samples. The diminishing improvements suggest any further increase in dataset size will have a marginal effect. These results permit a rough recommendation of about 5000 training samples per class for high-precision predictions in this particular classification task.

In an earlier paragraph I mentioned generalisation as a key goal of network training. For the Bell & Keyser dataset, this meant ensuring that the network learns to identify

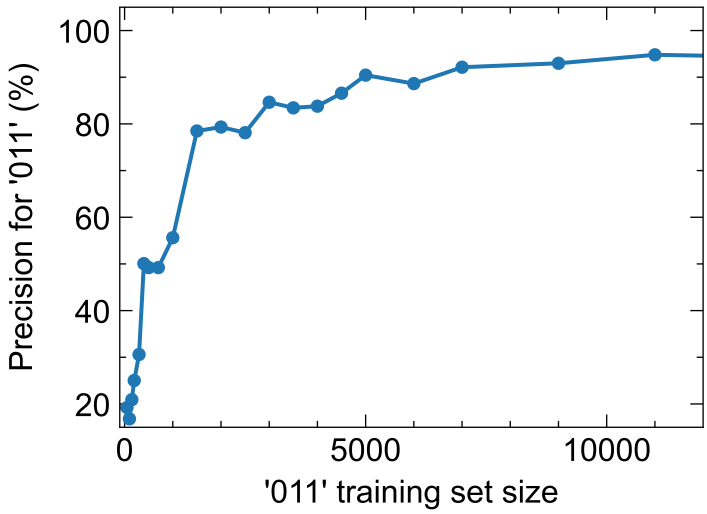


Fig. 5.9 Precision for the '011' barcode versus '011' training set size. The precision increases steeply up to 2000 training events, followed by a slower rise to $> 90\,\%$ for more than 5000 samples. Adapted with permission from [4], copyright 2018 American Chemical Society.

molecular barcodes instead of the nanopore they are measured with. Moving one level up from nanopores, a similar generalisation goal can be formulated about the measurement device: ideally, a trained network should be able to correctly classify events not only from different nanopores, but also different experimental setups. To assess how well the given network performs on data from a different setup, I collected translocation data on the pressure setup described in section 2.2.2 for the '000', '011', '101' and '111' barcodes (see table 5.1). Part a) in Figure 5.10 shows the size of the Bell & Keyser and additional datasets, as well as their division into training and test sets. The whole dataset consists of 63 899 events, different colors mark the 8 distinct barcodes. Blocks are sized corresponding to the number of events they contain while their hatching indicates whether they were used in the training or test stage. The table in part b) of Figure 5.10 shows how the neural network performs when trained on part of the data and tested on a different subset. The top left entry of 94.2 % corresponds to the precision reported in table 5.2 when training and testing on the Bell & Keyser dataset. When classifying events from the additional test dataset with the Bell & Keyser-trained model, however, the precision drops to only 64.5 %. This shows that the network does not fully generalise to the current features of a barcoded molecule but includes information on the measurement device, such as different noise characteristics. As as result, a model trained on data from one experimental setup may not be directly transferable to classify data collected from a second one, even when measuring the same molecules.

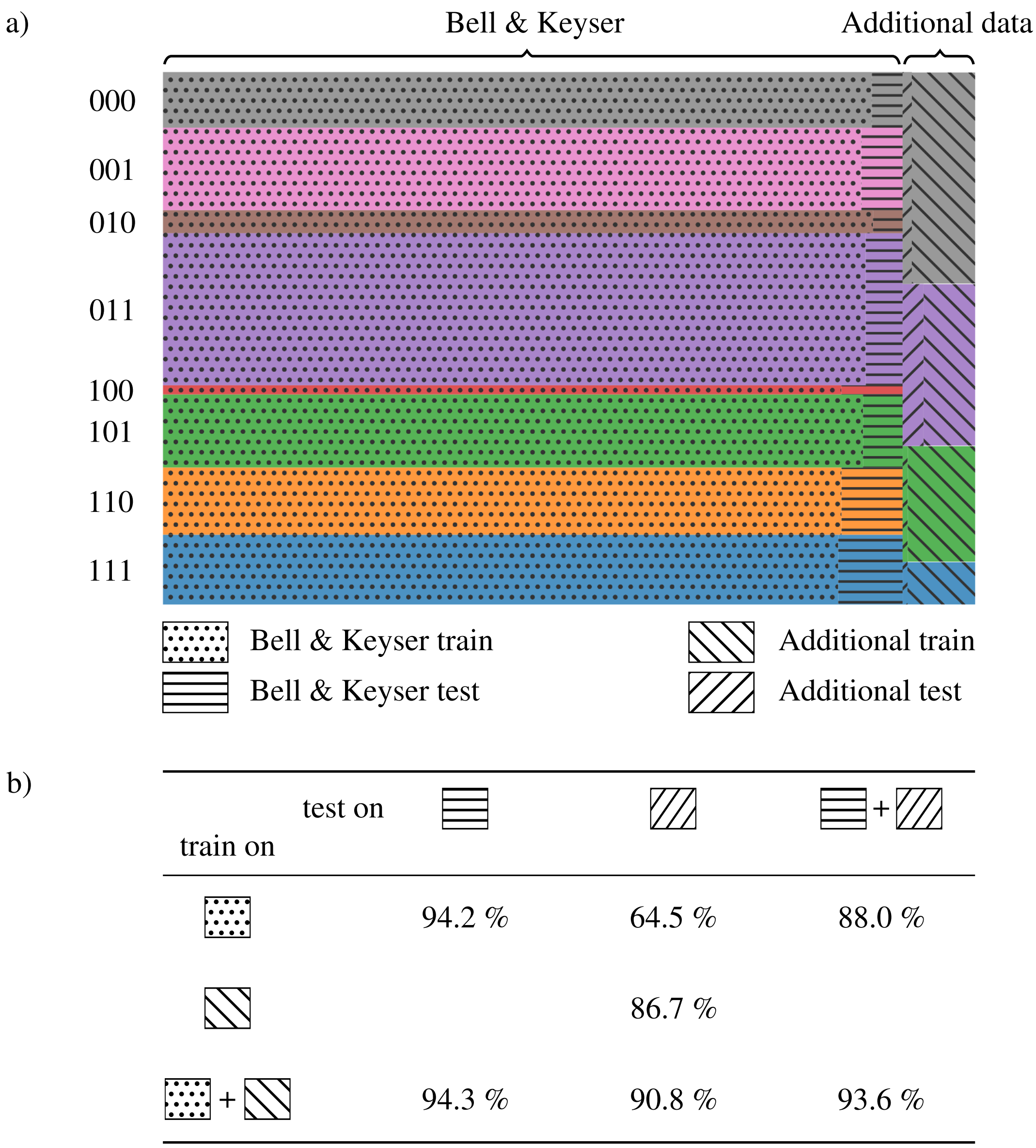


Fig. 5.10 Dataset breakdown and network performance on data from different experimental setups. **a)** illustrates the number of training and test events from Bell & Keyser [51] and additional data measured on a separate instrument, including a total of 63 899 samples. Box sizes reflect the number of events, colours indicate the barcode and hatching shows assignment to training or testing. The top left entry in the table in **b)** corresponds to the previously reported precision of 94.2 % on the Bell & Keyser dataset. Testing the same network on the additional data reduces the precision to only 64.5 %, which can be recovered to 90.8 % by including additional training data from the separate instrument.

Although the network fails to avoid overfitting to the measurement device, this can be mitigated by training on datasets from different setups. Much like events for the same barcode but from different nanopores reduce overfitting to individual pores, samples from different setups improve generalisation across instruments. The third row in the table in Figure 5.10 b) shows precision values for a model trained on both the Bell & Keyser and additional training data. Introducing training samples from the additional dataset improves classification performance on both test sets. While the increase in precision for the Bell & Keyser test set is marginal, the value for the additional test set rises to 90.8 % compared to 86.7 % when trained only on the additional data. This demonstrates the importance of a large collection of samples to train a 'base' model, which can then be adapted to data from a different measurement device through further training. In particular, it should be noted that the training set for the additional data (4811 events) is much smaller than that of the Bell & Keyser samples (54 714 events). Nevertheless, combining the two achieves a precision $> 90\,\%$ for both test sets. This suggests a protocol in which enhancing a model 'pretrained' on the Bell & Keyser dataset with much smaller amounts of data from a different instrument creates a network capable of generalised high-precision classification.

The above paragraphs outline a deep learning approach to the molecular classification of nanopore current signals. While not informed by a physical understanding of the underlying translocation process, this 'black-box' technique is nevertheless extremely successful at solving the task at hand. In addition to being considerably faster than the Bayesian approach in the previous chapter, neural network-based deep learning avoids the manual extraction of features. Once a suitable network architecture has been identified, it merely requires datasets of sample events labelled with the associated molecule. Nanopore measurements are capable of producing such datasets within hours. A related advantage is the immediate relationship between precision and training set size. Although performance improvements diminish with increasing size, a larger number of training samples directly leads to higher precision, which is not necessarily the case for manual algorithm design. Lastly, the automatic feature extraction built into neural networks can greatly increase the fraction of analysable data. For the nanopore classification task presented here the improvement is striking: the ability to reliably assign barcodes to current signals from folded DNA strands boosts the fraction of usable events by a factor of 5 compared to the algorithm previously used by Bell & Keyser [51].

While powerful for the analysis of current signals from known molecules, deep learning-based classification relies on the availability of large amounts of labelled data. The above results suggest a training set size of around 5000 events per distinct barcode to achieve a precision $> 90\,\%$. Such datasets can be readily obtained with nanopore measurements, however they may prove prohibitive for other single-molecule techniques. More importantly, even for nanopores this only applies to measurements of a limited number of known molecules. One of the key advantages of the barcoding strategy introduced by Bell & Keyser is the scalability of the encoding scheme [51]: by using modifications on a DNA strand as 'bits' at predefined positions, each additional bit doubles the number of distinct barcodes. The 2^n scaling with the number of modifications n quickly leads to enormous library sizes. However, the classification network presented in this section relies on thousands of training samples *per barcode*, doubling the measurement effort for each added bit. In addition, peak positions may not be known beforehand for some sensing schemes, such as the detection of dCas9 probes bound to a DNA strand [5]. These issues highlight the limitations of neural networks which directly learn molecular classification from sample data. The following section outlines an alternative deep learning approach which addresses some of these drawbacks.

5.3 Peak detection

A neural network which learns to classify current traces based on sample events for each molecule does not scale to systems which encode large amounts of information in sequences of modifications along a DNA strand. For such systems, a more suitable strategy is to locate and classify individual modifications as opposed to modification sequences. This is the approach taken by Bell & Keyser's original thresholding algorithm [51] and the Bayesian method in section 4.3. In a deep learning context, detecting individual peaks avoids the exponential scaling of the training set requirements with the number of modifications. However, peak labelling is a more challenging problem than labelling of translocation events: in the previous section, we could assign all events from a measurement of a single molecule to the same class. How do we reliably identify regions in a current trace as a peak and compile a training dataset?

The key insight is that the detection of peaks for labelling does not have the same constraints as detection for barcode classification. In the labelling task, we know which molecule produced the current trace and therefore which features to expect. This allows straightforward checks to verify labels, for example ensuring that the number of labelled peaks is equal to the number of '1' bits in the barcode and that the peaks appear within a certain distance of each other. Put more generally, peak labelling lets us trade recall for precision. **Recall** refers to the share of correctly labelled events out of the whole dataset, while **precision** gives the correct fraction out of all labelling attempts. By only accepting labels which we believe to be highly reliable, we can make sure that our overall labelling is mostly correct (high precision) at the expense of not including all events (low recall).

	Barcode	filtered (see table 5.1)	unfolded & unbound	labelled
Bell & Keyser [51]	10001	6084	1071	980
	10011	9005	464	411
	10101	2519	465	423
	10111	16609	2780	2616
	11001	1002	174	165
	11011	7978	238	216
	11101	7337	1489	1389
	11111	7644	1406	1316
Dr Jinbo Zhu, unpublished	1234	-	922	751

Table 5.3 Dataset for the peak detector network. Bell & Keyser barcodes now include the bounding bits as they represent valid peaks. Peak labelling is only carried out on unfolded translocations for DNA carriers without a bound protein. The labelled column includes events whose peak labels have passed a number of verification checks. See Figure 5.16 for details on the ‘1234’ barcode.

Table 5.3 shows the outcome of this labelling strategy. I again use the Bell & Keyser barcode dataset, in this case each ‘bit’ is defined as a ‘1’ peak. The barcodes include the boundary bits which were previously ignored because they were not part of the information-encoding sequence. Starting with the filtered sets of events from table 5.1, I exclude folded translocations as originally defined by Bell & Keyser as well as DNA strands which have a target bound along their contour. As discussed previously, folded events smear out peak patterns while peaks from bound targets affect distance-based verification checks. Each event in the ‘unfolded & unbound’ column runs through a peak-labelling procedure multiple times with progressively less strict parameters, i.e. more peaks are found on each iteration. Breaking the loop when the number of labelled peaks equals the number of bits in the barcode constitutes a first correctness check. In addition, the labelling only returns peaks which are within a threshold distance from their nearest neighbour, if multiple such clusters exist the one with the most peaks is returned. This clustering check makes it more likely that only bits have been labelled and not spurious peaks further along the DNA strand. The peak detection itself relies on the threshold-based ‘find_peaks’ function, available in the signal processing module of the SciPy Python package [122]. For each labelled peak I define a bounding region between the maximum current values within 10 data points on either side of the peak. These ‘ground truth’ regions in combination with the ‘1’ class constitute the output of the labelling step, the final event count is shown in the ‘labelled’ column in table 5.3.

The learning task for peak detection differs from the classification problem outlined in the previous section. Whereas before we assigned a barcode to each translocation current trace, the aim here is to detect bounding regions around peaks and identify the modification which produced each peak. The neural network must learn a combined regression (bounding regions) and classification (peak class) task. This is directly analogous to object detection in computer vision, where a model outputs bounding boxes around objects as well as the category they belong to. Convolutional neural networks have proved highly successful at this class of problems [123], suggesting they also perform well at current trace peak detection. The model introduced here is a 1-dimensional implementation of the the Single Shot MultiBox Detector (SSD) by Liu *et al.* [124]. This architecture combines the bounding region regression and classification into a single neural network. SSD is not a standalone model but builds on top of a base network, which can be any design suitable for the task at hand. I chose the convolutional layers of the classification network from the previous section as the base, without dropout and the last max pooling layer. In the SSD approach, the input space is tiled with suggestions for bounding regions, called anchors. The network then predicts offsets for each anchor to match it to a ground truth region, as well as a softmax output to indicate the class of the object contained in the region.

A key idea is to address input features of different sizes with progressively shrinking convolutional layers. Figure 5.11 shows the 1D-SSD architecture, the topmost tensor with shape (32, 77, 256) corresponds to the output of the base network. The subsequent layers receiving this tensor as an input should learn how to predict the location and class for regions at 77 anchor positions in the model input. Similarly the convolutional layer after the next max pooling outputs a temporal dimension of size 38, associating it with 38 positions in the input. Figure 5.12 a) illustrates the tiling with anchor regions on a sample event. Vertical black lines at the bottom of the plot show the anchor positions associated with the three convolutional feature layers. At each position I suggest 3 bounding regions, each scaled slightly differently relative to a base size (scalings of 1, $\frac{7}{8}$ and $\frac{9}{8}$). These anchor regions are shown as horizontal black lines at the leftmost positions in the plot. Based on tests of network performance, I chose anchor region base sizes of $\frac{12}{700}$, $\frac{16}{700}$ and $\frac{20}{700}$ for the 77, 38 and 19 feature layers respectively, so that more condensed feature layers cover larger regions. The built-in ability to handle differently sized objects is one of the main advantages of the SSD approach. The implementation outlined here does not make full use of this advantage since the peak widths and therefore region base sizes are quite similar. It may however prove useful for future modification schemes which define collections of closely spaced peaks as one ‘object’ to be detected. For example, one could envision a combination of the classification and SSD

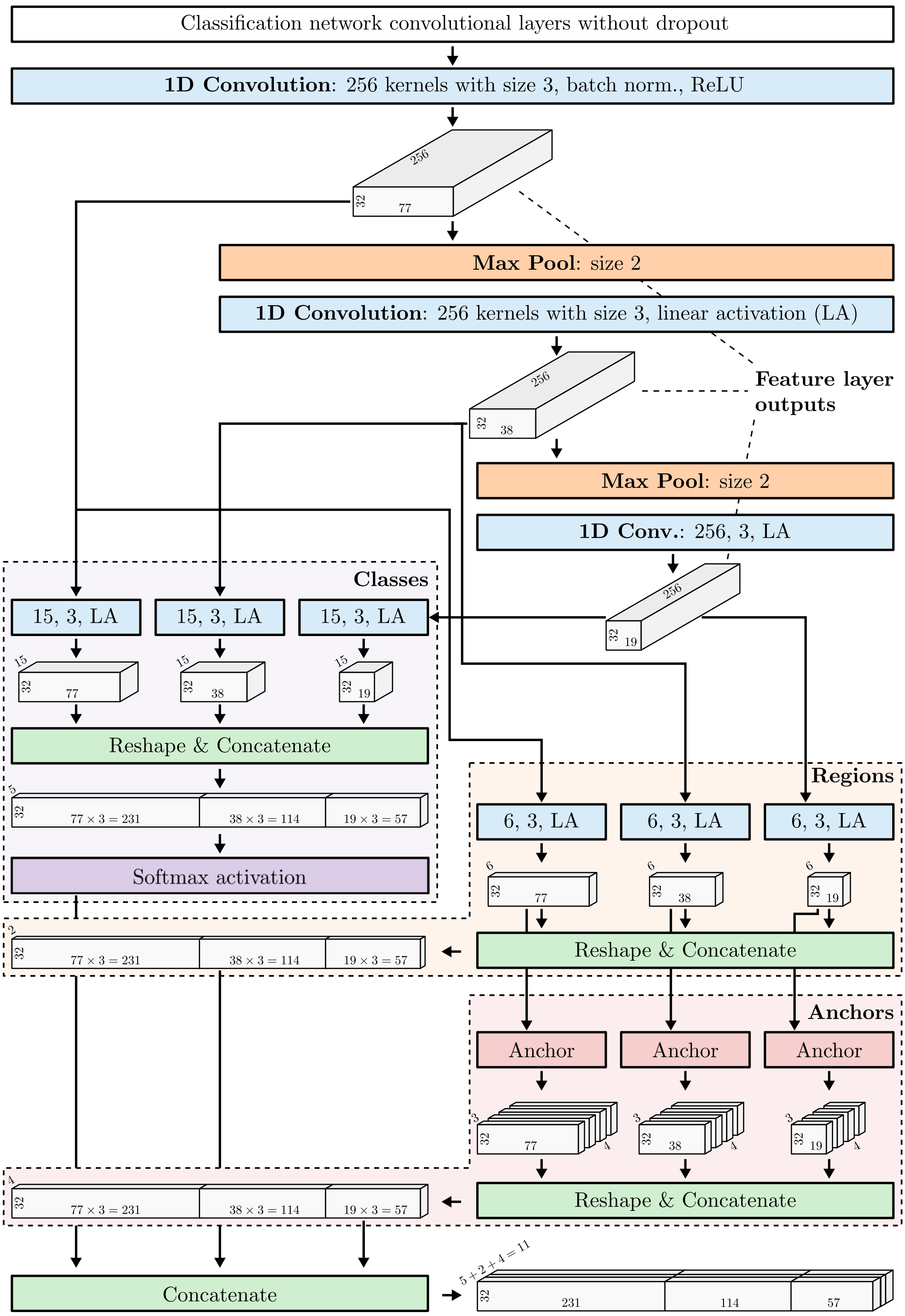


Fig. 5.11 Peak detector network architecture. The convolutional layers of the classification network in the previous section serve as the base, extended by a 1-D implementation of the SSD architecture [124]. Each feature layer is connected to a 'Regions' unit which predicts offsets to region anchors tiling the input and a 'Classes' unit which predicts the peak class contained in the region. The network maps a normalised event current trace to predictions of peak positions and classes, including a confidence value.

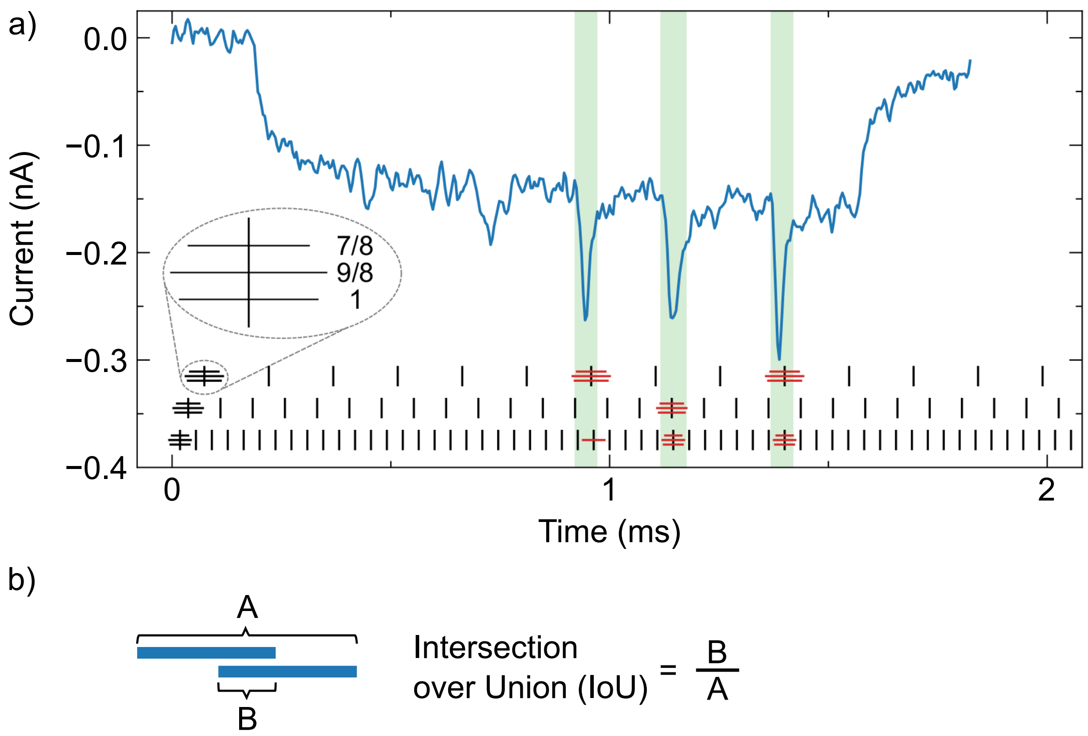


Fig. 5.12 Anchor region tiling and matching to ground truths. **a)** shows a sample event with 3 peaks, ground truth labels are represented by green shaded areas spanning the y-axis. The neural network identifies peak positions by predicting offsets to predefined anchor regions which tile the input, their centre positions are shown by short vertical lines at the bottom of the plot. Each position contains 3 regions of different scales, illustrated by horizontal lines at the leftmost anchor positions. Red horizontal lines show anchors matched to ground truth regions for model training, see the main text for the matching procedure. **b)** illustrates the intersection over union (IoU) metric to measure region overlap.

architectures presented in this chapter, where the region covering all the peaks in a barcode is identified together but other peaks constitute distinct classes.

To learn the combined regression and classification task, each feature layer is connected to two downstream convolutional layers. The first of these, in the purple ‘Classes’ box in Figure 5.11, learns to assign a class to each anchor region. The particular network shown here can detect 4 distinct peaks plus one background class (no peak) for 3 regions at each anchor position, resulting in $(4+1)*3 = 15$ kernels. After reshaping and concatenating the class layer outputs from all feature layers, a softmax activation function produces probability estimates across the 5 classes. The second downstream layer, in the orange ‘Regions’ box, predicts offsets to the proposed anchor regions to match peaks exactly. For each of the 3 anchor regions this requires an offset to the start and end of the region, resulting in $2*3 = 6$ kernels. I again reshape and concatenate the region layer outputs from all feature layers so that the second dimension in the resulting tensor, now of size $231+114+57=402$, runs across all anchor regions. The last SSD component, in the red 'Anchors' box, does not contain any trainable parameters. Its purpose is to introduce the tiled anchor regions into the model, so that the relative offsets from the 'Regions' box can be converted back into absolute region predictions. As described in the previous paragraph, for each anchor position in a feature layer I propose the 3 regions defined by a start and end. I also include variances to normalise the relative offsets, leading to 4 values per region. Concatenating the outputs from the 'Classes', 'Regions' and 'Anchors' boxes produces the final model output shown in the bottom right corner, with the second dimension corresponding to anchor regions and the third dimension including softmax class predictions, anchor offsets and absolute anchor coordinates.

To train the network, we need to convert the labelled ground truth regions into the same format as the model output tensor. This means determining for each anchor region whether it corresponds to a ground truth peak. The overlap of two regions is measured with the intersection over union (IoU) metric, shown in part b) of Figure 5.12. A value of 0 indicates no overlap, while an IoU of 1 means the two regions are identical. Following the assignment procedure in Liu *et al.* [124], I first calculate the IoU of all anchor regions with all ground truth regions. I then match each ground truth to the anchor with the highest IoU. The remaining anchor regions are assigned to the highest IoU ground truth if the IoU is > 0.5. I classify anchors with a maximum IoU of < 0.3 as background and those > 0.3 and < 0.5 as neutral, meaning they are ignored in the loss calculation. For all matched anchor regions I then calculate the offset from the associated ground truth region. Figure 5.12 a) shows the outcome of this assignment procedure: green shaded areas correspond to ground truth regions resulting from the peak labelling, while red lines indicate all anchor regions matched to a ground truth.

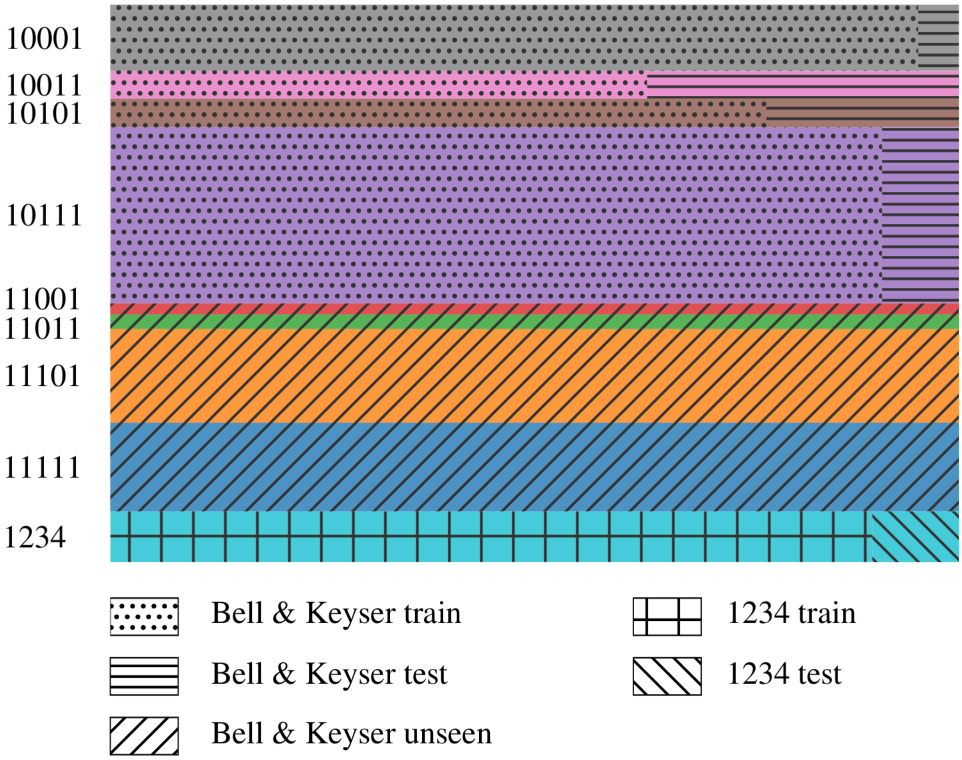
The network loss is the sum of the localisation loss of predicted versus ground truth regions and the classification loss. The latter calculates cross-entropy on a softmax activation, while the localisation uses smooth L1 loss [125]. The network presented here is an adaptation of the SSD Keras [126, 127] implementation by Ferrari [128] to 1-dimensional inputs. Training uses the Adam optimisation algorithm [117]. I determine the number of epochs by continuously monitoring the loss on a validation dataset not used for training and stopping once there has been no improvement for 10 epochs. This adaptive training strategy, called early stopping, avoids overfitting to the training dataset [109]. Similarly I reduce the learning rate by 80 % from a starting value of 0.001 whenever the validation loss has not decreased for 8 epochs to improve model convergence. One epoch of training on the Bell & Keyser dataset in table 5.3 with an Nvidia GeForce 1080 GTX Ti graphics card takes roughly 8 s, resulting in overall training times below 10 min for early stopping after 50 to 70 epochs.

For the model to produce useful predictions the output needs to pass through several post-processing steps. The red regions in Figure 5.12 a) show that the raw model output assigns many regions to a single ground truth peak. Multiple regions allow the network to learn localisation without having to select a single anchor region [124]. To filter out multiples, I initially remove all anchor regions with a classification confidence < 0.25. I then carry out non-maximum suppression on the remaining regions by iteratively picking the highest confidence region and removing all other regions with an IoU overlap > 0.3. On average this results in one high-confidence region prediction per ground truth.

The motivation for individual peak detection rather than the classification of whole events is the adaptability to changing modification sequences. The requirement of thousands of training events for each additional permutation quickly becomes unfeasible for long sequences. To illustrate how a peak detection network handles different modification schemes, I trained the model described above on 4 out of the 8 Bell & Keyser barcodes and tested its precision on the remaining 4 'unseen' barcodes. Figure 5.13 a) shows the relative sizes of the training, test and unseen datasets. It should be emphasized that a classification network trained with sample events for each barcode cannot generalise to produce meaningful predictions for barcodes it has never encountered.

The peak detector, on the other hand, achieves a mean average precision (mAP) of 89.8 % on the unseen barcodes. The mAP metric refers to the prediction precision averaged over several recall values. I follow the calculation in [129] by averaging the maximum precision values with a recall greater than 11 equally spaced thresholds. The blue curve in Figure 5.14 represents the precision versus recall for peak detection on the unseen set, while the orange circles indicate the 11 values which combine into the mAP based on the recall thresholds shown as dashed vertical lines. The curve is obtained from sorting all region predictions by descending confidence and calculating precision as the share of correct regions (IoU > 0.5 and correct class) out of attempted predictions and recall as the fraction of correct regions out of all ground truth regions in the dataset. On the unseen data, > 97 % of predicted regions correctly detect a peak and almost all of the labelled peaks are found. This is reflected in the sample events in Figure 5.15. Ground truth regions labelling peaks are shown as shaded areas spanning the y-axes while darker blocks indicate region predictions. For all shown events the network correctly detects all the peaks making up a barcode. The crucial aspect is the model's ability to generalise to the generic identification of peaks, even through it has not encountered any of the particular peak sequences in the examples.

a)



b)

test on / train on				+
	90.7 %		89.8 %	
		90.5 %		
+	90.5 %	86.8 %	89.3 %	90.3 %

Fig. 5.13 Dataset breakdown and performance of the peak detector network. **a)** shows the division of all 8267 events into training, test and unseen sets, with colours indicating barcodes. **b)** shows mAP (mean average precision, see Figure 5.14) values for different combinations of training and test sets. The peak detector network achieves an mAP approaching 90 % on barcodes it has never encountered.

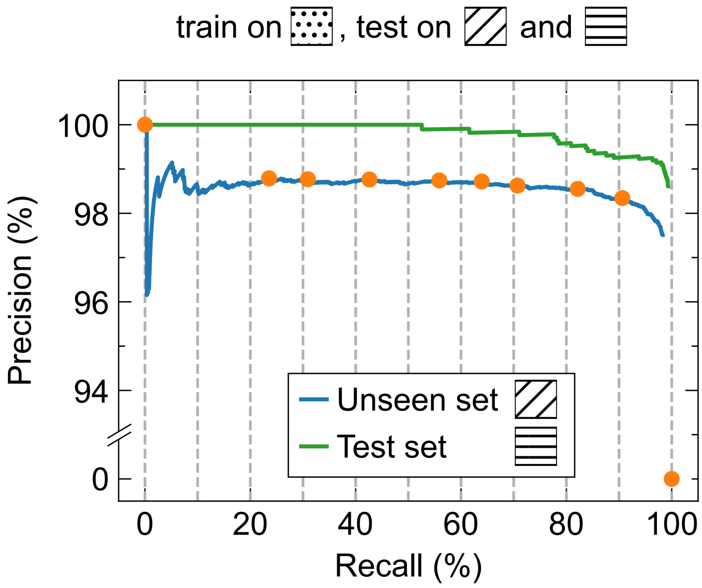


Fig. 5.14 Precision versus recall curves for the unseen (blue) and test (green) sets and mAP calculation. The precision for the unseen set shows that $> 97\,\%$ of predictions are correct, while the maximum recall approaching $100\,\%$ indicates that the network finds almost all ground truth peaks, even for barcodes it has never encountered. The mAP score is the average of the orange dots, obtained by taking the maximum precision value with a recall greater than 11 equally spaced thresholds (dashed lines) [129].

While the peak detector performs well at this task, the network has merely learned to emulate the thresholding-based algorithm used to label the peaks. One could argue that deep learning does not add any benefit if the labelling algorithm would perform similarly well. While a valid criticism for the Bell & Keyser dataset, it fails to take into account the neural network's extensibility to the detection of several object classes. Figure 5.16 shows a novel barcode design by Dr Jinbo Zhu based on 4 different modifications, see section 2.4 for details. It greatly increases the information content for each position in a sequence from 2 in the binary to 4 in the quaternary system. A classical algorithm relying on manual feature extraction and thresholding would have to take into account subtle differences in peak height and shape. The neural network detector, on the other hand, automatically learns the relevant features from the training data. Crucially, the labelling procedure does not change significantly from the previous case. Comparing the median position of identified peaks with the event length indicates the translocation direction. Peak classes can then be assigned simply by labelling peaks in the known order of modifications, where '1' corresponds to the same modification as in the Bell & Keyser barcodes and '2' through '4' classify the new structures. The thresholding algorithm merely needs to find peaks, not classify them. In addition to clustering, the labelling strategy includes checking that the '4'-labelled peak

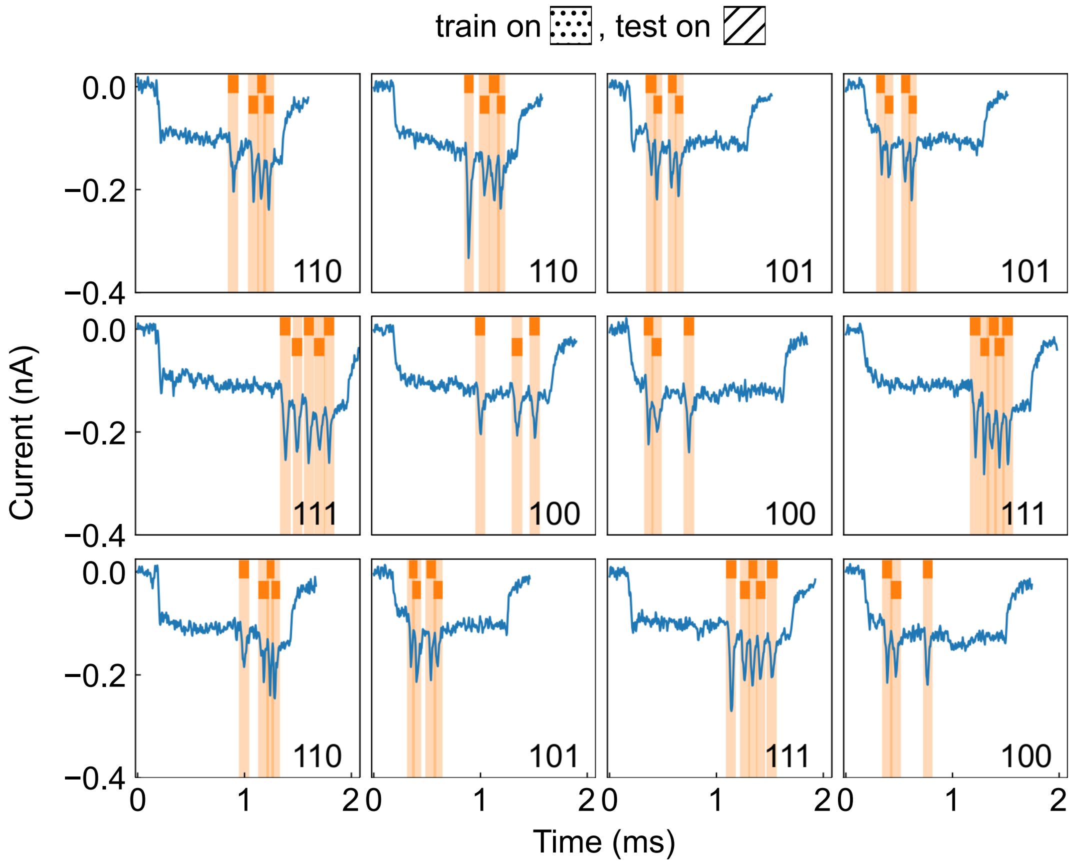


Fig. 5.15 Peak detection on unseen sample events from the Bell & Keyser dataset. Shaded areas spanning the y-axes show peak ground truth labels, darker blocks indicate the predicted peak positions. The network correctly finds all peaks for events from 4 previously unseen barcodes.

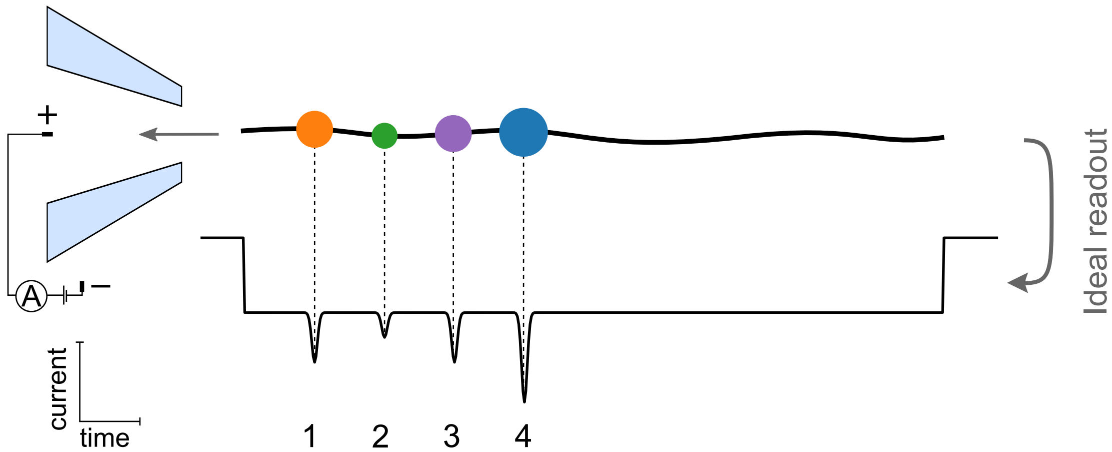


Fig. 5.16 Multilevel barcode design by Dr. Jinbo Zhu [unpublished, personal communication]. Different modifications produce distinct peaks, allowing each position in the sequence to encode 4 values as opposed to 2. In the dataset used here, the first modification is identical to that used for '1' bits in the Bell & Keyser scheme, while the 3 additional ones are labelled '2' through '4'.

reaches at least 1.5 times the unfolded current level to ensure that it indeed corresponds to the largest peak. By using information on the known modification sequences in the training data, we can again trade recall for precision and produce a dataset of correctly labelled peaks of different classes.

The sample events in Figure 5.17 show the network performance on the '1234' dataset. Shaded areas spanning the y-axes again show ground truth regions, orange, green, purple and blue shading corresponds to the 4 modifications from '1' through '4'. Solid blocks at the top of each subplot indicate the predicted regions and classes. The network correctly localises and classifies all but two peaks. The strength of the deep learning approach becomes apparent when focusing on the '1' (orange) and '3' (purple) predictions. The '1' modification sometimes produces larger peaks than the '3' attachment, sometimes smaller peaks. Manual feature extraction would require further criteria on top of simple height thresholding to deal with such overlap. The neural network, on the other hand, can automatically take into account relevant features such as peak shape or the presence of neighbouring modifications.

The central plot in Figure 5.17 marked by a red bounding box shows an example of correct localisation but false classification. A comparatively large peak produced by the '1' modification leads to classification as '4', while the '2' ground truth region is misclassified as '3'. A more quantitative look at such prediction errors can shed light on the 'black-box' behaviour of the neural network. To do so I trained the predictor model on the training sets from both the Bell & Keyser and '1234' data and assessed its performance on their combined test sets. Figure 5.18 gives an overview of the most common error categories, following the approach in [130]. Each row includes the model predictions for one of the 4 classes. The

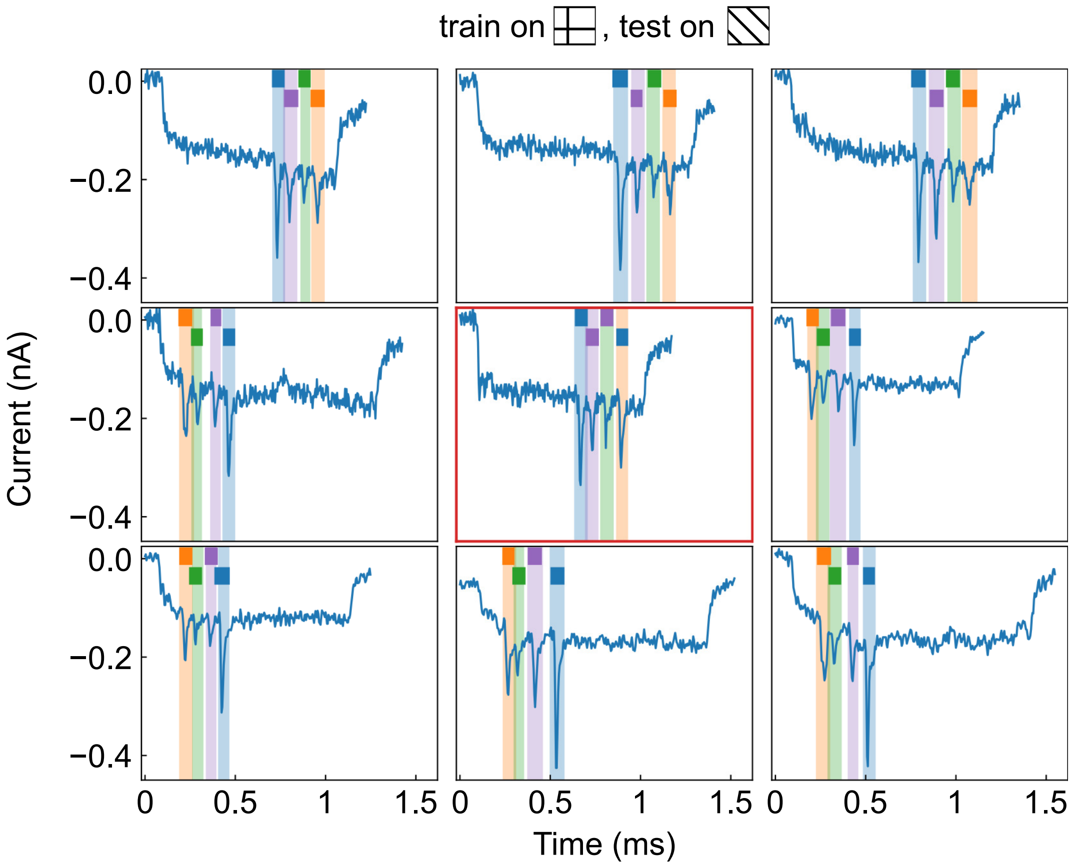


Fig. 5.17 Prediction on sample events from the '1234' dataset. Ground truths are shown as shaded regions spanning the y-axes, dark blocks indicate predictions while colours correspond to the 4 peak classes. The network correctly localises and classifies all but 2 peaks in the examples. It automatically extracts relevant features, allowing it to deal with overlapping peak heights for example for the '1' (orange) and '3' (purple) modifications. The central subplot with a red bounding box illustrates classification errors for unusually large '1' and '2' peaks.

outcome of a prediction is counted as correct if the region has the largest IoU with a ground truth region, the value is > 0.5 and the classes match. For $0.1 < \text{IoU} < 0.5$ or regions which do not have the highest IoU with a ground truth region but correctly predict the class, I count a localisation error. This means that there is a peak of the matching class nearby, but the positioning lacks accuracy. IoUs < 0.1 count as background errors, i.e. the network predicts a peak where the ground truth does not include one. Lastly, wrong class errors occur when the network correctly localises a region but misclassifies the contained peak. The central plot in Figure 5.17, for example, shows a 'wrong class 1' error in which a '4' modification was incorrectly predicted for a '1' peak.

The first column in Figure 5.18 illustrates the cumulative share of these error categories for all predicted regions sorted by descending confidence. Columns 2 through 4 give example events for which these mistakes occurred. The first row indicates that incorrect '1' predictions are most commonly positioned over background, i.e. the ground truth suggests there is no peak. The sample events, however, show that many of these errors are in fact due to false labelling. The network correctly identifies a peak missed during ground truth assignment. This can be seen as proof of the network's generalisation ability: it has learned to detect modifications irrespective of their position along the DNA strand or relative to other modifications. In addition, these background errors allow us to estimate an upper bound of the labelling procedure's error rate. Assuming the neural network reliably detects peaks outside the barcode cluster, less than 2 % of all 1869 '1' predictions correspond to peaks missed during ground truth labelling.

For '2' and '4' predictions, the most common error is to assign the respective label to regions which really contain a '1' modification. The example events show that this is due to '1' peaks which lie at the tails of their height distribution. Relatively small '1' peaks are wrongly assigned to the '2' class, while larger ones end up in the '4' category. This is particularly apparent in the Bell & Keyser barcode events, which only contain '1' peaks and therefore illustrate the variation in peak heights produced by the same modification. The performance of the '2', '3' and '4' classes may be improved by more balanced training datasets. Figure 5.13 a) shows that the Bell & Keyser set contains considerably more events than the '1234' set, which is exacerbated by the fact that each event in the former provides at least two '1' peaks compared to one peak per class in the latter. However, overlap between peaks cannot be avoided completely for modifications of similar size and shape. Analysis of prediction errors helps to quantify this overlap and can inform the design of modifications which produce distinct peaks.

This section has demonstrated that a deep learning-based peak detector solves the scalability issues of direct classification networks which rely on sample data per molecule. Through

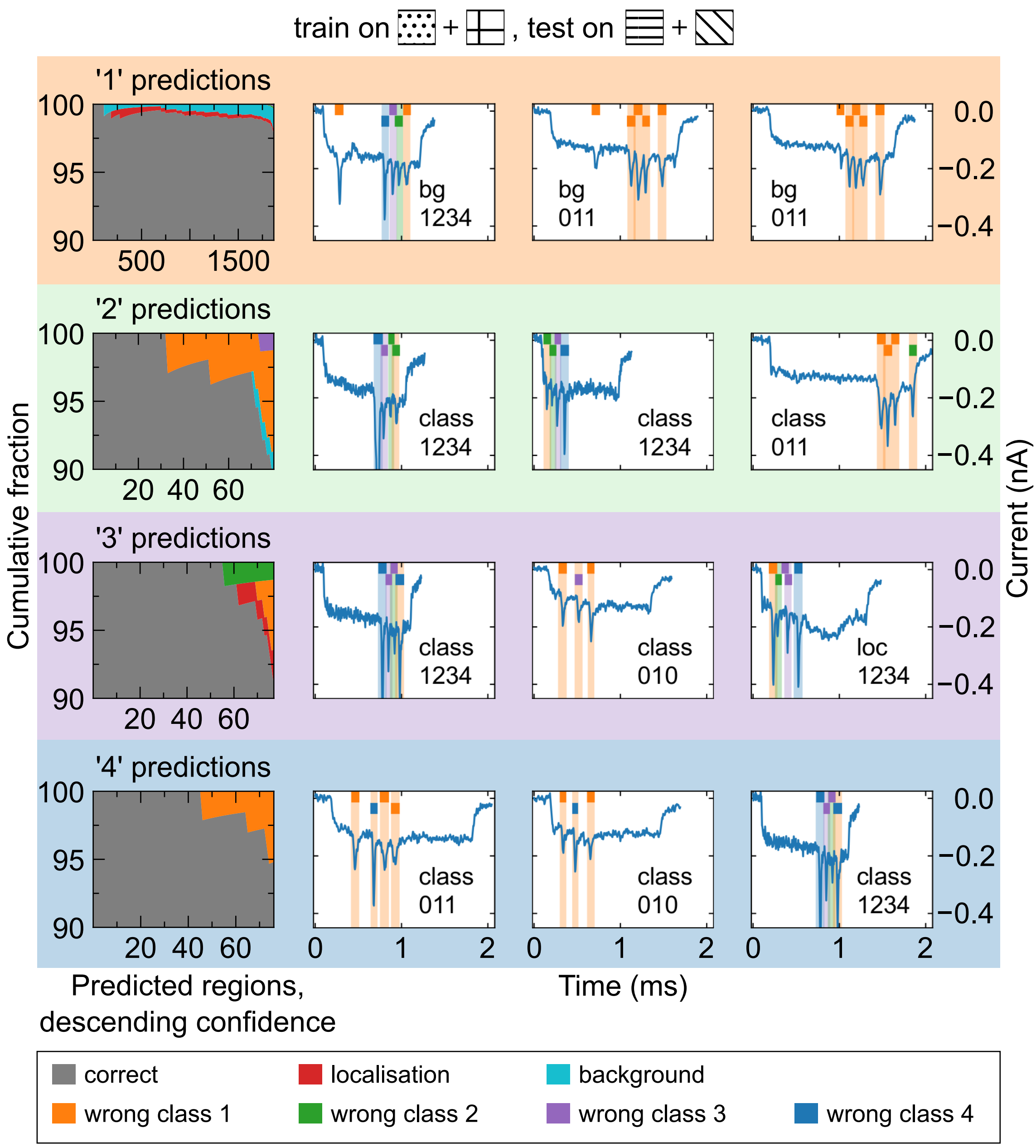


Fig. 5.18 Error analysis for the combined Bell & Keyser and '1234' dataset following the method in [130]. The first column shows the assignment of each prediction to one of 7 categories: correct, localisation error (correct class but inaccurate bounding region), background error (false positive peak prediction) and a wrong class error for each of the 4 classes (correct localisation but wrongly assigned to the ground truth mentioned in the error). The first row, for example, illustrates that the most common error for '1' predictions is of the background type, meaning the predicted region does not overlap with a ground truth. The sample events in columns 2 through 4 show that this is likely due to wrong labelling, i.e. the model in fact correctly finds a peak. '2' and '4' predictions most commonly fail by wrongly assigning their class to a '1' peak. While the unbalanced training set sizes may play a role, the sample events suggest that variations in peak height make some of these errors unavoidable.

a labelling strategy that uses information on the expected number and position of peaks, it becomes possible to reliably label regions in the current trace with the associated modification. A 1-dimensional adaption of the SSD architecture [124] trained on this labelled data achieves mAP test scores $> 90\,\%$ on sequences consisting of a single modification (Bell & Keyser) as well as those encoding a quaternary system with differently sized attachments ('1234'). The network reaches an mAP of 89.8 % for modification sequences it has not encountered during training, demonstrating its ability to generalise to arbitrary peak positions. Fewer trainable parameters due to a lack of dense layers mean that training and inference are faster than for the classification network. A crucial advantage compared to manually designed algorithms is again the automatic extraction of features. The results on '1234' data show that the network performs well even for cases where considerable overlap of peak heights renders simple thresholding impossible, indicating that it learns features which may not be obvious to the human eye.

The obvious drawback to peak detection is that it requires unfolded translocation events. The difficult-to-predict shape of peaks in the folded section of a DNA strand precludes reliable labelling. This is a disadvantage compared to the direct classification network, whose ability to classify folded translocations results in a fivefold increase in the number of usable events. Whether or not the exclusion of a potentially large fraction of events is problematic depends on the sensing task at hand. Sections 3.2 and 6.3 discuss approaches to promote unfolded translocations and their advantage over simply ignoring folded events. Another challenge with peak detection is the need to interpret the predicted peaks to estimate the modification sequence which produced them. This adds another layer of complexity to the analysis pipeline compared to the direct assignment of a barcode with the classification network. Lastly, the peak labelling strategy relies on the exclusion of any event which does not pass a number of verification checks. I therefore train the detector network on low noise events with distinct peak clusters. When processing arbitrary translocation data, this means that the detector network requires a pre-filter to select events with similarly high quality. The performance of the classification network on a binary yes/no output in [4] suggests that another neural network before the peak detector would be well-suited to assess the readability of peak structure. However, this would again come at the cost of higher complexity compared to the simple ECD and current threshold for direct classification.

5.4 Choosing the best approach

Having tested and discussed different analysis methods to decode information in nanopore signals of modified DNA strands, it should be clear that the choice of algorithm must

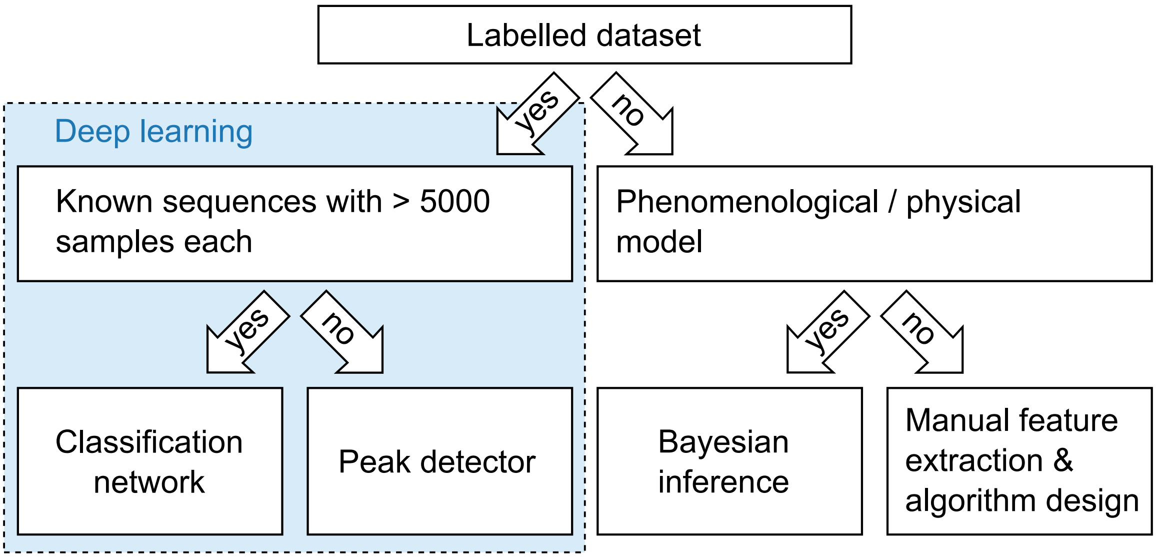


Fig. 5.19 Decision tree for nanopore event analysis. Deep learning can offer superior performance without manual feature extraction, but relies on large labelled datasets.

depend primarily on the question to be answered. Manually designed algorithms can be applied quickly to target obvious features and are transparent in their outcome. Bayesian inference probabilistically assesses models of the translocation process and fully maps out their parameter space. Both of these techniques rely on low noise signals and manual feature extraction, but can deal with small datasets. A deep learning-based classification network achieves unprecedented precision even on folded translocations, but requires training sets in excess of 5000 events for each class. A neural network peak detector performs well on much smaller training sets and scales well to long modification sequences or attachments in arbitrary positions. However, it requires data filtering and low noise events.

While general recommendations are difficult, it is possible to outline the prerequisites for using each technique. Figure 5.19 shows that deep learning fundamentally relies on the availability of labelled training datasets. Training on simulated current traces could reduce the experimental effort in the future, however we currently lack the physical understanding of the translocation process to produce event traces indistinguishable from measured ones. Within the deep learning branch, if the modification sequences to be measured are known a priori with thousands of sample events for each molecule, one can benefit from the precision and straightforward analysis pipeline of a classification network. For peaks at arbitrary positions or long modification sequences, a peak detection network offers high precision with the automatic feature extraction inherent in deep learning. If large training datasets are not available, having a phenomenological or physical model of the translocation process suggests the use of Bayesian inference. The alternative are algorithms based on the manual extraction of features, such as the one originally used by Bell & Keyser [51]. These decision criteria should provide a first guidance on the choice of analysis algorithm, which can be refined according to the specific sensing task.

Chapter 6

Adaptive real-time decoding

The previous chapter showed that deep learning approaches allow fast analysis of translocation events at high accuracy. A natural follow-on question is whether this analysis can be carried out in real time. Fast and direct measurement of single molecules is commonly presented as one of the advantages of nanopore sensors, however the standard approach today is to record the current for processing at a later stage. This prolongs the time-to-result as it prevents stopping the measurement when a conclusive result has been reached. This chapter outlines a real-time decoding system for DNA barcodes. I first show that under usual measurement conditions, current hardware allows real-time molecule identification with a neural network classifier. I then derive a probability calculation which reveals how many barcodes of each class must be detected to determine a mixture with high confidence. The last section investigates multiple translocations of the same molecule as a technique to adaptively re-measure noisy events.

6.1 Real-time classification

One obvious constraint to real-time processing is an upper bound on the available computation time. At a commonly used sampling frequency of 250 kHz, 250 000 data points need to be handled every second, including filtering as well as event detection and classification. While the feasibility depends on the processing power and event frequency, the following demonstrates that with a modern CPU (Intel Core i7-7700K, 4.2 GHz), computation time does not limit real-time neural network classification at a 250 kHz bandwidth and event frequencies on the order of 1 Hz. Figure 6.1 outlines the analysis scheme which produces a list of classified translocation events from a raw current trace input. The first step is to filter out high-frequency noise in the current, which was previously done by feeding the current through 8-pole analogue low-pass Bessel filters (900CT, Frequency Devices, IL, USA or 3382, Krohn-Hite, MA, USA) before digitising. I found that digital filtering after the data has been acquired by a computer produces equivalent output without the expensive equipment and additional connections. Importantly, the events obtained from different filtering methods can be mixed for neural network training, as illustrated by the additional dataset in section 5.2. The digital filter was implemented as an 8-pole Bessel filter with a cut-off frequency of 40 kHz closely following the equivalent analogue devices, section 2.2.2 provides further details.

The second step in the analysis scheme assesses the noise in the current trace to determine whether it is low enough to allow event detection. An ideal noise metric allows setting a simple threshold beyond which event detection is suspended and corrective steps may be taken, such as temporarily inverting the voltage to drive out a molecule stuck in the pore (kick-out). Usually, the standard deviation of the current trace serves as a measure of the noise. However, it is well known to be sensitive to outliers. As a result, translocation events in a current trace considerably affect the standard deviation even when the underlying noise has not changed. The left column in Figure 6.2 a) shows a clean current trace with a standard deviation of 8 pA. Part b) of the Figure includes a translocation event in a trace of the same length, causing a jump in the standard deviation to 19 pA. In part c) the pore has deteriorated, as evidenced by the increased low-frequency noise and the unstable baseline. Ideally, we would like to be able to detect and respond to this change. However, the standard deviation of 12 pA is *lower* than for a clean baseline with a translocation event in part b). As a result, there is no straightforward way to set a threshold on the standard deviation to identify the onset of increased noise. The issue has previously been addressed by Plesa & Dekker [97], who propose an algorithm which detects translocations and iteratively replaces the points during events with the current at the start of the event. While this is an effective solution to estimate both the moving average and the standard deviation, it is unnecessarily complex for event frequencies up to about 10 Hz. A simpler alternative is the use of quantiles as noise estimators. The difference between the 0.5th (the median) and 0.841th quantile, for example, is a straightforward but powerful noise metric. The exact q-values can be chosen arbitrarily, in this case the first value gives the data's median while the second one corresponds to the mean + 1σ for a normal distribution. The difference between these quantile values approximates the standard deviation of the baseline current without taking into account the lower current points during translocations. This can be seen from the distributions of current values in the second column of Figure 6.2. The red ranges overlaid on the histograms show the mean $\pm$ 1σ, while orange ranges indicate the median $\pm$ the quantile difference. The presence of a translocation event in part b) greatly increases the standard deviation, while the quantile

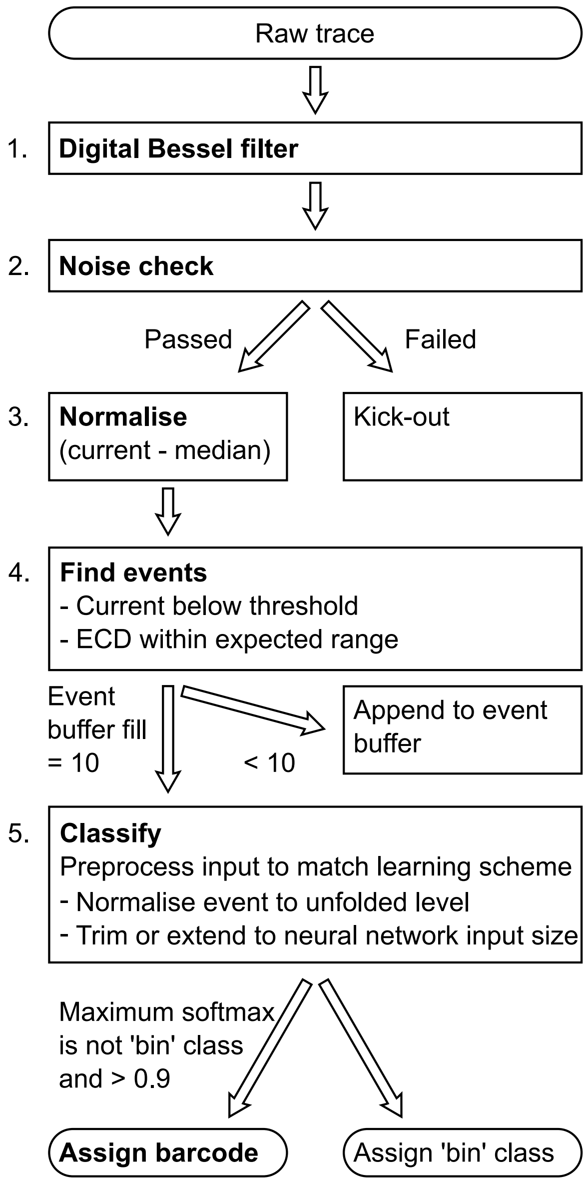


Fig. 6.1 Processing pipeline for real-time classification. The raw current trace is first passed through a digital Bessel filter, followed by a quantile-based noise check. After normalisation, events are detected if the current drops below a threshold. The unfolded current level is calculated from a FIFO buffer of the 10 most recent events. In the last stage a neural network assigns each event to a class, the ‘bin’ class contains translocations which fail the filtering step or have a prediction confidence $< 90\,\%$.

difference remains unchanged. The latter only increases in part c) following an increase in the underlying nanopore noise. This makes the quantile difference an ideal metric on which to base a threshold which determines the state of the pore: in the example, we can exclude any current chunks with a difference $> 10\,\text{pA}$, which correctly identifies a noisy pore but is not affected by translocations.

If the current chunk has passed the noise check it moves to the next step for event detection. The detection algorithm identifies translocations as deviations from the median current below a certain threshold. I set the threshold as a multiple of the quantile difference mentioned in the previous paragraph, which adapts it to the noise of the specific experiment. I exclude events if their electronic charge deficit (ECD) falls outside a specified range, usually the mean ECD $\pm 1\sigma$ of previous events from the same molecule. Compared to analysing current data once a measurement has been recorded, real-time processing requires estimating population statistics from the small sample of events which have been measured up to this point. For example, one of the crucial preprocessing steps for neural network classification in section 5.2 is the normalisation of events to the unfolded current level. However, this unfolded level is calculated from the events themselves, which means it is not available at the start of a measurement. To obtain an estimate I keep an event buffer which contains the 10 most recent events. The unfolded level is then approximated by the median of the concatenated intra-event current traces of all events in the buffer.

In the last step of the analysis scheme the neural network classifies the events according to their barcode. To match the network input shape I trim or extend events following the same steps as in section 5.2. Unlike in the original classification network, the softmax output now consists of 9 class values for the 8 barcodes and a 'bin' class. During training the 'bin' class includes all events which do not pass the initial filtering steps, i.e. the difference between the 'total' and 'filtered' columns in table 5.1. This integrates the filtering into the network and means all detected events can be passed directly into the classifier. The final, aggregate count includes all events identified as one of the barcodes with a $> 90\%$ probability, those below the threshold are assigned to the 'bin' class. Figure 6.3 shows the aggregate barcode count in a real-time measurement containing an equimolar mix of four barcodes. The counts increase roughly in parallel, reflecting the equal concentrations. We can conclude that real-time neural network classification is feasible and can greatly speed up the time-to-result. However, two questions remain: Firstly, given the characteristics of the neural network as a classifier, can we take into account any bias the network may introduce into the barcode counts? This would allow separating classifier bias from true concentration differences and could help explain unexpected counts, such as the twofold difference between the '101' and '000' barcodes in

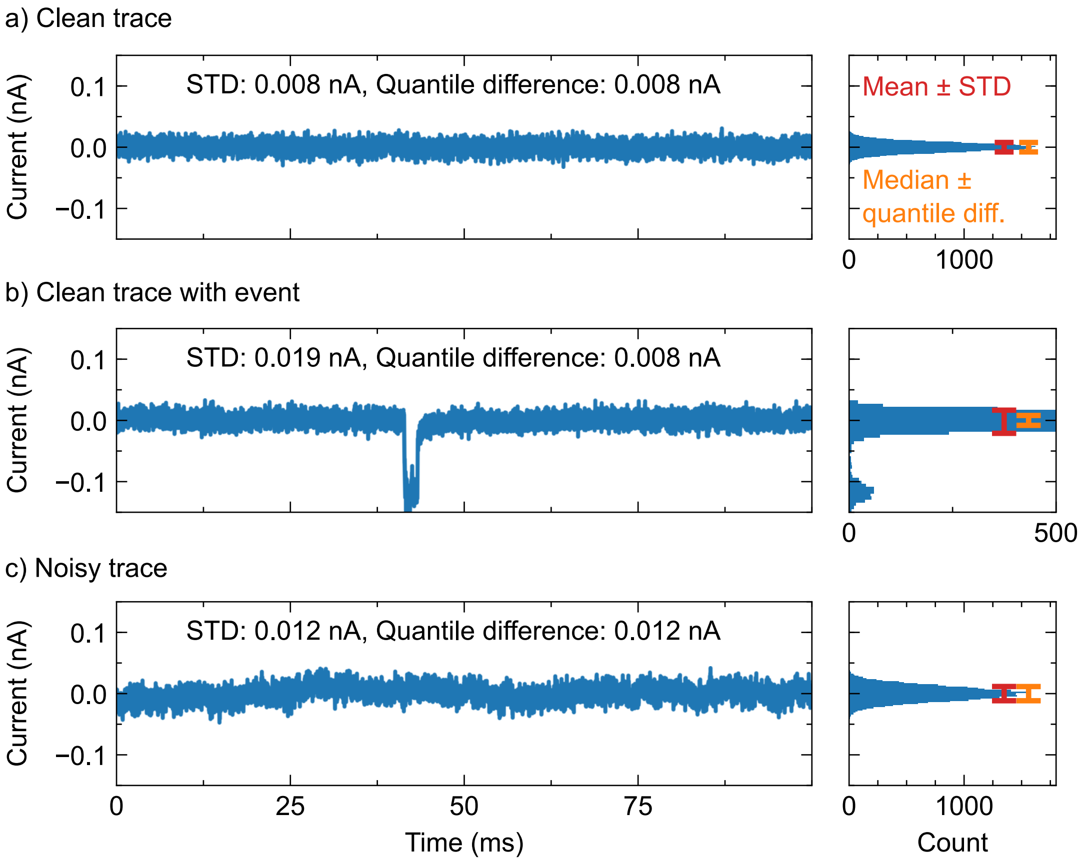


Fig. 6.2 A quantile-based metric is more suitable than the standard deviation (STD) to assess noise in a nanopore. The left panels show raw current traces under different noise conditions, the right panels contain histograms of all current values on the left. Red and orange overlays show the mean $\pm\sigma$ and the median $\pm$ one quantile difference respectively. The quantile difference (QD) is the distance between the 0.841th and 0.5th quantile, corresponding to one σ in a normal distribution. In the low-noise trace in panel **a)**, STD and QD are equal at 8 pA. The translocation event in panel **b)** increases the STD to 19 pA although the baseline noise remains unchanged. For the true increase in noise in panel **c)** both metrics rise to 12 pA. Compared to the STD, a simple QD-threshold of 10 pA correctly identifies the noisy trace while remaining unaffected by the translocation event.

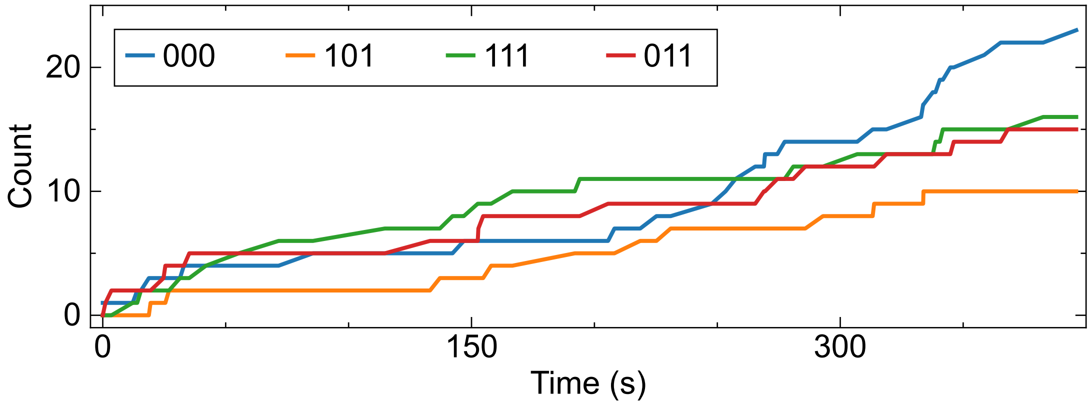


Fig. 6.3 Real-time barcode classification with the neural network from section 5.2. The solution contained an equimolar concentration of the '000', '101', '111' and '011' barcodes in 4 M LiCl at pH 8.

Figure 6.3. Secondly, which count is required to conclude the presence of a specific barcode with high certainty? The following section addresses these issues.

6.2 *How many events are required to infer the presence of a barcode?*

The ability to classify translocations in real time raises the question how long a measurement must run to allow conclusions on the solution content. While the translocation frequency is a key factor, the answer critically depends on how many events are required in each class to indicate the presence of the respective barcode with high certainty. In probabilistic terms, we are looking for the probability of the barcode mixture m being present given we have counted N_b events of class b. If M is the set of possible mixtures and B are the available barcodes, we want to calculate

$$P\big(m|\{N_b\}\big) \quad \text{where} \quad m \in M, b \in B$$

In the following, the abstract calculation will be presented alongside a concrete example of the measurement shown in the previous section to illustrate how it can be applied to real-time nanopore sensing. In the example a mixture can consist of up to four distinct barcodes, resulting in 16 possibilities, that is

$$\begin{aligned} B =& (000, 101, 111, 011) \\ M =& \{NNNN, NNNY, NNYN, NNYY, NYNN, NYNY, NYYN, NYYY, \\ & YNNN, YNNY, YNYN, YNYY, YYNN, YYNY, YYYN, YYYY\} \end{aligned}$$

where Y indicates the presence of the barcode at its respective position while N signifies its absence. $YNYN$, for example, represents a mixture which contains only the ‘000’ and ‘111’ barcodes. The left columns in the following equations demonstrate the generic derivation while in the right columns I will calculate the probability of the $YNYY$ mix given counts of 3, 0, 2, and 2 for the four barcodes, that is $P\big(YNYY|N_{000} = 3, N_{101} = 0, N_{111} = 2, N_{011} = 2\big)$.

Starting with Bayes’ Rule (equation 4.1), the desired probability can be written as

Derivation:

$$P\big(m|\{N_b\}\big) = \frac{P\big(\{N_b\}|m\big)P\big(m\big)}{P\big(\{N_b\}\big)}$$

Example:

$$P\big(YNYY|N_{000} = 3, N_{101} = 0,$$

$$N_{111} = 2, N_{011} = 2\big) =$$

$$\frac{P\big(N_{000} = 3, \ldots|YNYY\big)P\big(YNYY\big)}{P\big(N_{000} = 3, \ldots\big)}$$

We rewrite the evidence $P\big(\{N_b\}\big)$ as a sum over mixtures to obtain an expression which only depends on two variables

$$\begin{aligned} P\big(\{N_b\}\big) &= \sum_{k \in M} P\big(\{N_b\}, k\big) \\ &= \sum_{k \in M} P\big(\{N_b\}|k\big)P\big(k\big) \\ P\big(m|\{N_b\}\big) &= \frac{P\big(\{N_b\}|m\big)P\big(m\big)}{\sum_{k \in M} P\big(\{N_b\}|k\big)P\big(k\big)} \end{aligned}$$

$$P\big(YNYY|N_{000} = 3, \ldots\big) =$$

$$\frac{P\big(N_{000} = 3, \ldots|YNYY\big)P\big(YNYY\big)}{\sum_{k \in M} P\big(N_{000} = 3, \ldots|k\big)P\big(k\big)}$$

$P\big(m|\{N_b\}\big)$ can thus be calculated based on the mixture priors $P\big(k\big)$ and the likelihoods $P\big(\{N_b\}|k\big)$ where $k \in M$. In a sensing context, the priors will be based on the expected incidence of each barcode depending on the sensing targets. If nothing is known about the mixture probabilities we can simply set the prior to $\frac{1}{|M|}$ where $|M|$ is the number of mixtures, i.e. each mixture is equally likely. This leaves the likelihood $P\big(\{N_b\}|k\big)$, i.e. the probability to obtain the barcode counts $\{N_b\}$ in a measurement of the mixture k. The aggregate counts for each barcode are related to the individual likelihoods $P\big(b|k\big)$, $b \in B$ through the multinomial distribution:

$$P(\{N_b\}|k) = \frac{N!}{\prod_{n\in\{N_b\}} n!} \prod_{b\in B} P(b|k)^{N_b}$$

where $N = \sum_{b\in B} N_b$

$$P(N_{000} = 3, \dots |YNYY)$$
$$= \frac{7!}{3!0!2!2!} P(000|YNYY)^3$$
$$P(101|YNYY)^0 P(111|YNYY)^2$$
$$P(011|YNYY)^2$$

This relies on the individual events that make up the aggregate count to be independent, an assumption met by the sequential translocations of molecules in nanopore sensing. The crucial information for the desired probability are the likelihoods $P(b|k)$, i.e. the probabilities to measure a barcode in a given mixture. We can determine these likelihoods experimentally by calculating the fraction of each barcode relative to the total event count for every mixture. However this approach quickly becomes unfeasible due to the large number of required measurements: only 4 distinct barcodes in the example already entail 16 possible mixtures, a number which doubles for every added barcode. To deal with this exponential growth an alternative approach is to base the likelihood calculations on measurements of individual barcodes, corresponding to the mixtures $NNNY$, $NNYN$, $NYNN$ and $YNNN$ in the example. For each of these mixtures we determine the barcode counts, obtaining a square confusion matrix **C** analogous to that in Figure 5.8 whose rows contain the true barcodes while the columns indicate the predicted barcodes:

$$\mathbf{C} = \begin{bmatrix} C_{b_1,b_1} & C_{b_1,b_2} & \cdots & C_{b_1,b_{|B|}} \\ C_{b_2,b_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ C_{b_{|B|},b_1} & \cdots & \cdots & C_{b_{|B|},b_{|B|}} \end{bmatrix}$$

where $S_j = S_i \; \forall i, j \in B$ and $S_b = \sum_{j=1}^{|B|} C_{b,b_j}$

$$\mathbf{C} = \begin{bmatrix} C_{000,000} & C_{000,101} & \cdots & C_{000,011} \\ C_{101,000} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ C_{011,000} & \cdots & \cdots & C_{011,011} \end{bmatrix}$$
$$= \begin{bmatrix} 98 & 1 & 0 & 1 \\ 2 & 90 & 1 & 7 \\ 0 & 0 & 95 & 5 \\ 2 & 1 & 1 & 96 \end{bmatrix}$$
$$S_{000} = S_{101} = S_{111} = S_{011} = 100$$

$C_{101,000} = 2$, for example, indicates that in a measurement only containing the '101' barcode, in two instances events were wrongly classified as the '000' barcode. For the following the total number of events must be normalised to the same level for each measurement so that the sums across matrix columns are equal, i.e. $S_j = S_i \; \forall i, j \in B$. In the example this has been done so that each measurement appears to have resulted in 100 events. Based on the confusion matrix we can approximate the likelihoods $P\big(b|k\big)$ for any mixture by summing the instances in which events were classified as the b barcode across the individual measurements contained in the k mix, that is

$$P\big(b|k\big) = \frac{\sum_{a \in A} C_{a,b}}{\sum_{a \in A} S_a}$$

where $A = \{a \in B | a \text{ contained in mix } k\}$

$$\begin{aligned} &P\big(000|YNYY\big) \\ &= \frac{C_{000,000} + C_{111,000} + C_{011,000}}{S_{000} + S_{111} + S_{011}} \\ &= \frac{98+0+2}{100+100+100} = \frac{1}{3} \\ &P\big(101|YNYY\big) = \frac{1+0+1}{300} = \frac{1}{150} \\ &P\big(111|YNYY\big) = \frac{0+95+1}{300} = \frac{8}{25} \\ &P\big(011|YNYY\big) = \frac{1+5+96}{300} = \frac{17}{50} \end{aligned}$$

The assumption made here is that a measurement of an equal mixture of barcodes is equivalent to measuring each barcode separately and adding up the counts. This assumption breaks down if the underlying translocation process contains interactions between different barcodes in the same sample. For example, if there was a preference for the '000' barcode relative to '111', measuring an equal mixture of the two would not result in equal counts while the above additive approach would. At high salt concentrations we have not observed such a barcode preference, suggesting the assumption is valid. Greater care needs to be taken at lower salt concentrations, where section 3.3 demonstrated that modifications along the DNA strand can influence translocation behaviour. Similarly the capture rate has been shown to depend on DNA length, with a higher translocation frequency for longer strands [72]. It should be noted that while the additive approach does not model hierarchies in the physical translocation process, it successfully takes into account biases in the classifier. The second row in the confusion matrix **C** shows that the neural network is the least accurate for the '101' barcode, frequently mislabelling it as '011'. By adding up the counts we take into account

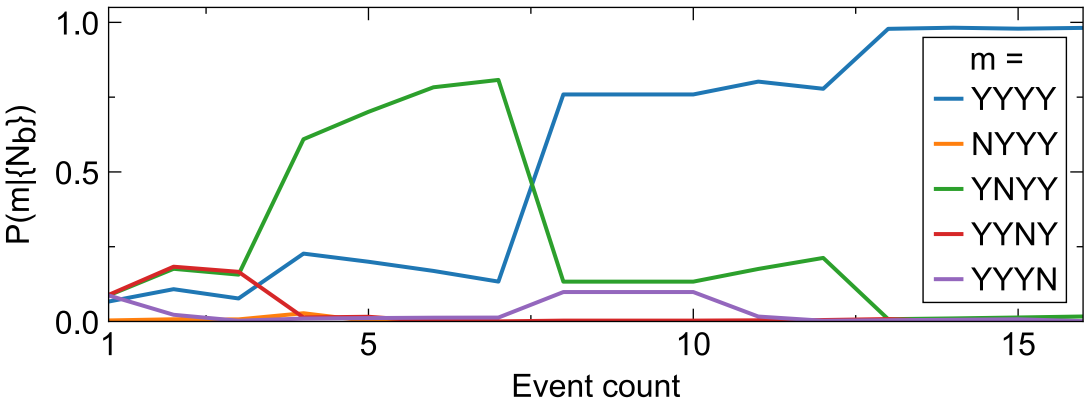


Fig. 6.4 Probabilities for the measured solution containing the mixture m given the barcode counts $\{N_b\}$ from Figure 6.3. Mixtures indicate the presence and absence of barcodes at each position with Y and N respectively, for example $YNYY$ contains all but the '101' barcode. For the first few events the $YNYY$ mixture (green line) appears most likely, with a probability reaching 80 %. The detection of the first '101' barcode in the 8th event greatly boosts the $YYYY$ (blue line) probability. After 13 events we can conclude that the measured solution contains all 4 barcodes with a confidence > 97 %.

the fact that an equal mixture containing the '101' barcode likely has a lower '101' count than the other contained barcodes because of the bias in the classifier.

We now possess all the information to calculate $P\left(m|\{N_b\}\right)$, i.e. the probability for the mixture m being measured given the barcode counts $\{N_b\}$. In the example, the full calculation for one mixture gives

$$
\begin{aligned}
&P(m) = \frac{1}{|M|} = \frac{1}{16} \quad \forall \quad m \in M \\
&P\left(YNYY|N_{000} = 3, N_{101} = 0, N_{111} = 2, N_{011} = 2\right) \\
&\quad = \frac{P\left(N_{000} = 3, \ldots |YNYY\right) P\left(YNYY\right)}{P\left(N_{000} = 3, \ldots\right)} = \frac{P\left(N_{000} = 3, \ldots |YNYY\right) \cancel{P\left(YNYY\right)}}{\sum_{k \in M} P\left(N_{000} = 3, \ldots |k\right) \cancel{P(k)}} \\
&\quad = \frac{\frac{7!}{3!0!2!2!} \frac{1}{3}^3 \frac{1}{150}^0 \frac{8}{25}^2 \frac{17}{50}^2}{\sum_{k \in M} P\left(N_{000} = 3, \ldots |k\right)} = \mathbf{0.808}
\end{aligned}
$$

As the count changes over the course of a measurement we can continuously update the probabilities for all mixtures. By setting a confidence threshold this allows us to run a measurement until we have identified the barcode mixture with a desired certainty, for example $> 95\%$. Figure 6.4 shows the probabilities for the top 5 out of the 16 possible mixtures at the start of the measurement in Figure 6.3. Over the first few events the $YNYY$ mixture appears most likely, growing in probability to $\sim 80\%$. This changes rapidly as soon as the first '101' barcode is detected in the 8th event, which greatly increases the probability for the $YYYY$ mixture. After 13 events the $YYYY$ probability has reached $> 97\%$, correctly identifying the sample as containing all four barcodes. The count at this point is $N_{000} = 3$, $N_{101} = 2$, $N_{111} = 2$ and $N_{011} = 4$. We can therefore conclude that for the given confusion matrix of 4 barcodes, it only takes on the order of 10 events to predict the correct barcode mixture with a certainty $> 95\%$. A key assumption in the above calculation is that all mixtures contain equimolar concentrations. Unequal barcode mixtures require more detailed modelling of the likelihoods $P(b|k)$ with $b \in B$ and $k \in M$. Nevertheless, the results show that given the right conditions, the content of a solution can be determined from few events.

6.3 Multiple translocations of the same molecule

The previous section demonstrated that given a high-precision classifier, 10s of events can be sufficient to determine which barcodes are present in a solution. Of course, this only refers to events with the right attributes to be passed to the classifier, for example an ECD within a certain range. For some of the analysis methods outlined in sections 4.3 and 5.3 this also means unfolded translocations, which can quickly exclude more than 60 % of events. A further share of events drops out because of the confidence threshold on the classifier predictions themselves. Overall only a small fraction of recorded translocations contributes to the final analysis result.

Chapter 3 outlined how fluid flows can increase the share of unfolded events and thereby usable data. However, flows out of the pore significantly reduce the translocation frequency. Generally it appears difficult to achieve the three simultaneous goals of unfolded and slow events at a high frequency solely by setting constant parameters. An alternative is to selectively adapt these parameters to control translocation behaviour. While both voltage and pressure switching are too slow to be applied on pore entry of a DNA strand, a proven technique is to invert the voltage to drive a molecule back out of the pore, followed by another inversion for recapture. This 'ping-pong' technique was pioneered by Gershow & Golovchenko [131], who showed recapture of a DNA molecule up to 22 times for a nanopore in a 2D membrane. Plesa *et al.* [132] studied the relaxation dynamics of long DNA strands with the same technique, achieving up to 1000 recaptures by lowering the sample concentration to pM. In glass nanocapillaries, Bell *et al.* [133] employed ping-pong to illustrate direction-dependent translocation dynamics for asymmetric nanopores.

While Gershow & Golovchenko [131] recognised the promise of measuring the same molecule multiple times to improve the signal-to-noise ratio, their work and subsequent investigations focused on DNA dynamics and the translocation process. To prove useful for modification decoding applications, the ping-pong technique needs to fulfil several requirements: firstly, it should recapture the same molecule multiple times even at sample concentrations in the nM range, without interference from other molecules. Secondly, the current signals from recaptured molecules should have noise characteristics similar to regular translocations to allow re-reading until the desired information has been decoded. Thirdly, recapture should be fast enough to speed up the overall frequency of readable events.

I investigated the ping-pong technique on an equimolar mixture of the ‘000’, ‘011’, ‘101’ and ‘111’ barcodes summing to a total concentration of 0.5 nM in 4 M LiCl solution buffered to pH 8. This corresponds to the measurement conditions commonly used for nanopore sensing. Figure 6.5 shows the stages in a typical ping-pong recapture. A molecule translocating into the nanopore at 600 mV constitutes the ‘ping’ event which triggers a voltage inversion. t_{delay} is the time between the end of the ping event and the switch, I varied the ‘pong’ voltage between -500 mV and -200 mV. The negative voltage drives the DNA strand back out of the pore, which triggers a second inversion back to 600 mV. t_{pong} and $t_{\mathrm{pong\ delay}}$ refer to the times from the ‘pong’ event to the first and second switch respectively. If recapture is successful, another translocation into the pore is detected $t_{\mathrm{recapture}}$ after the second voltage inversion.

Pressure across the pore is another parameter which influences recapture behaviour. Figure 6.6 shows that pressure values refer to the *trans* reservoir, 0 bar indicates atmospheric pressure while negative values correspond to vacuum inside the capillary. It should be noted that because of the slow response time of a pressure change on the order of 100 ms, I set constant values of 0.0 bar, -0.25 bar or -0.5 bar without responding to translocation events. The reason for negative pressures relates to the asymmetric geometry of glass nanocapillaries: the confinement on the *trans* side means that a molecule can be driven back out of the pore successfully, however recapturing it from the *cis* side is more challenging because it re-enters the bulk. Negative pressures produce fluid flows into the pore, which create additional drag on DNA movement away from the pore.

Previous studies relied on different DNA lengths or extremely low sample concentrations to show that the same molecule re-enters the pore [131, 132]. DNA barcodes offer a more flexible method of confirming recapture. Figure 6.7 shows 4 pairs of example traces, the two columns correspond to the ‘ping’ and recaptured translocations. Given the equimolar mixture of 4 barcodes, the matching barcodes strongly suggest that the same molecule indeed re-enters the pore.

I tested the effect of different negative pressures combined with a varying pong voltage on several ping-pong characteristics. The first key metric is the pong success rate, i.e. the fraction of attempts in which the DNA is successfully driven back out of the pore. I found

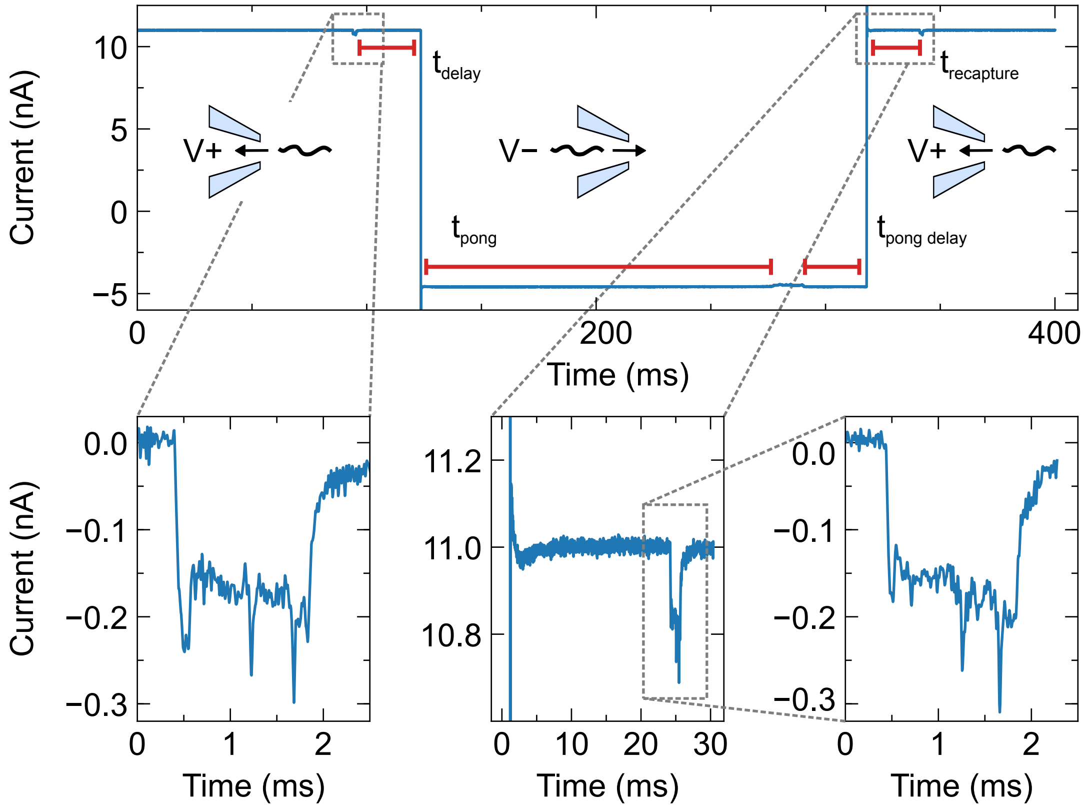


Fig. 6.5 The ping-pong procedure of measuring a molecule multiple times. Initially, a positive voltage of 600 mV drives a DNA strand into the pore, constituting the 'ping' event. After a time t_{delay} the voltage is switched to a negative value, driving the molecule back out of the pore. Once the outward 'pong' translocation has been detected after t_{pong}, another voltage inversion recaptures the molecule for a second translocation into the pore. I varied the negative ('pong') voltage between −500 mV and −200 mV, combined with pressures applied on the inside of the nanocapillary between 0.0 bar and −0.5 bar.

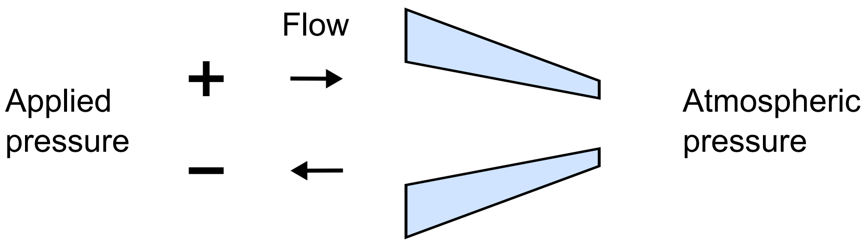


Fig. 6.6 Pressure is applied to the inside of the capillary relative to atmospheric pressure outside. Positive values produce fluid flows out of the pore, negative values refer to vacuum inside the capillary and cause inflows.

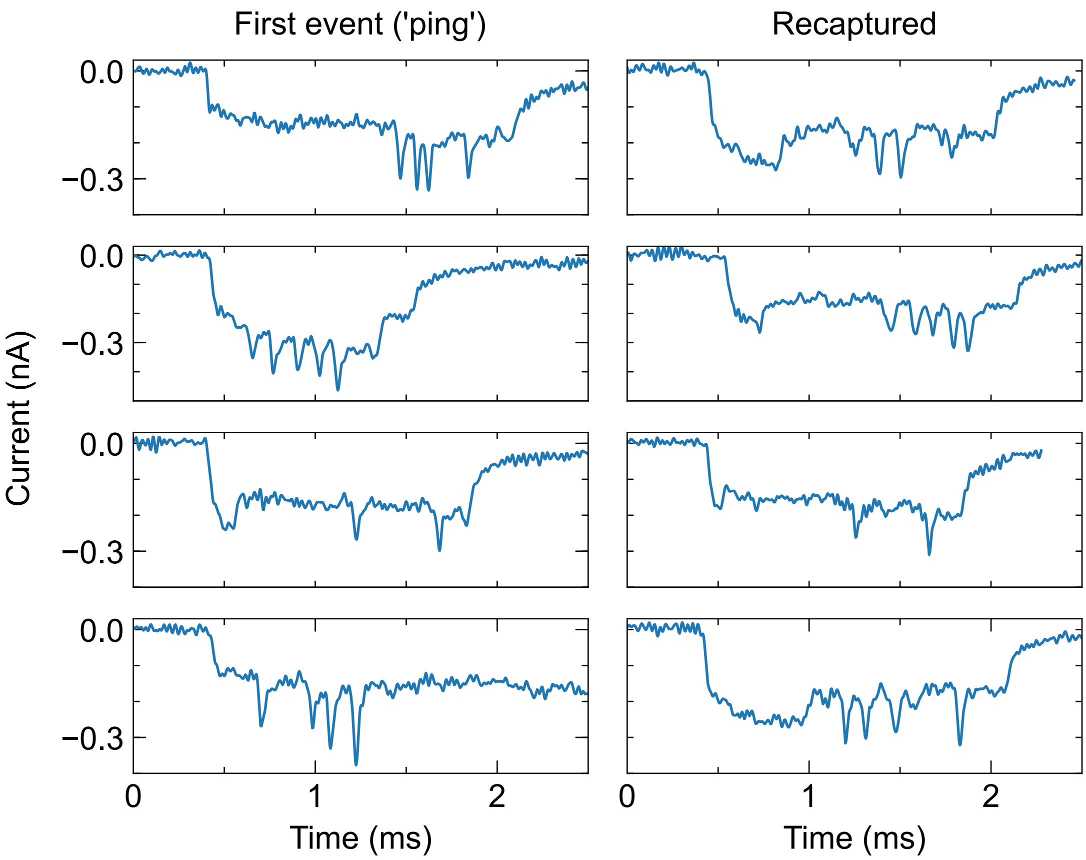


Fig. 6.7 Comparison of ‘ping’ and recaptured events reveals matching barcodes, indicating that the same molecule is indeed recaptured during the ping-pong procedure.

that a pong event could be achieved reliably for all tested parameters if a negative voltage is applied long enough. This is a key difference from recapture in symmetric 2D membranes, where the forward translocation can drive the molecule outside of the capture region on the *trans* side. Glass nanocapillaries, on the other hand, confine the DNA following the initial translocation, facilitating the passage back out of the pore under negative voltage.

The time t_{pong} illustrates the effect of different voltages and pressures on the pong translocation. Figure 6.8 a) shows that the mean wait until translocation back out of the pore increases for smaller absolute voltages, in line with the expectation of a lower driving force. t_{pong} also rises with more negative pressures applied to the *trans* side of the capillary. These pressures set up fluid flows into the pore, which counteract the outward-facing electrophoretic force on the molecule.

Figure 6.8 b) shows the success rate of recapturing a molecule within roughly 80 ms of switching back to a positive voltage. High pong voltages of −500 mV and −400 mV drive the DNA too far out of the pore, leading to low recapture rates below 30 %. At lower absolute voltages the success rate increases, particularly for negative pressures. At $-200\,\text{mV}$ and $-0.5\,\text{bar}$ all molecules could be recaptured successfully.

While this suggests low absolute pong voltages and high negative pressures for reliable ping-pong, another critical parameter is the time $t_{\text{recapture}}$ between the switch back to a positive voltage and the arrival of the recaptured molecule. When this time drops to very small values, parts of the translocation signal disappear in the transient current spikes of the circuit's RC-response, making it difficult to deconvolve and classify the DNA's current trace. Figure 6.8 c) shows that $t_{\text{recapture}}$ tends to decrease for smaller absolute pong voltages and more negative pressures. The parameter range which improves the recapture success rate thus also worsens the signal-to-noise ratio of recaptured molecules.

The above results identify an important trade-off for the ping-pong technique in modification-decoding applications. A delay between the return to positive voltage and the arrival of the recaptured molecule ensures a low-noise translocation signal, but makes recapture itself less likely. For the glass nanocapillaries used here, Figure 6.8 suggests a pong voltage between $-300\,\text{mV}$ and $-200\,\text{mV}$ with a pressure as negative as possible while maintaining a high enough $t_{\text{recapture}}$. Ideally this would drive the DNA out of the pore but create a trap in close proximity to the entrance, similar to the one reported in [88], allowing efficient recapture after voltage switching. A parameter which merits further investigation is the time $t_{\text{pong delay}}$ between the detection of a pong event and the second voltage switch. This delay took around 50 ms in the data presented here due to the event detection operating on fixed-size chunks of current. Faster voltage switching could boost recapture rates particularly for high absolute pong voltages.

Despite the scope for further investigation, these results show that driving DNA strands in and out of the pore is a feasible strategy to improve analysis. A crucial question is under which conditions adaptively re-analysing molecules reduces the total measurement time required to determine the content of a solution. Figure 6.9 compares the time between the arrival of usable events for a measurement without (red line) and with ping-pong (blue lines), a detailed derivation can be found in appendix J. Y-axes show the expected value of the wait normalised by the mean time $\overline{\Delta t}$ between all recorded translocations during regular positive voltage measurements, i.e. not just usable ones. $\overline{\Delta t}$ is simply the inverse of the mean translocation frequency $\bar{f}$. The normalised value $\mathbb{E}[\Delta t]/\overline{\Delta t}$ is always greater than one due to events which cannot be decoded successfully and therefore require a wait for the next translocation. Figure 6.9 a) shows the expected wait as a function of the ratio between the time added for each set of ping-pong iterations Δt_{pp} and the mean overall time between events $\overline{\Delta t}$. This ratio is a metric for how long ping-pong takes relative to the wait until the next translocation. Each ping-pong set is assumed to consist of 4 iterations after which

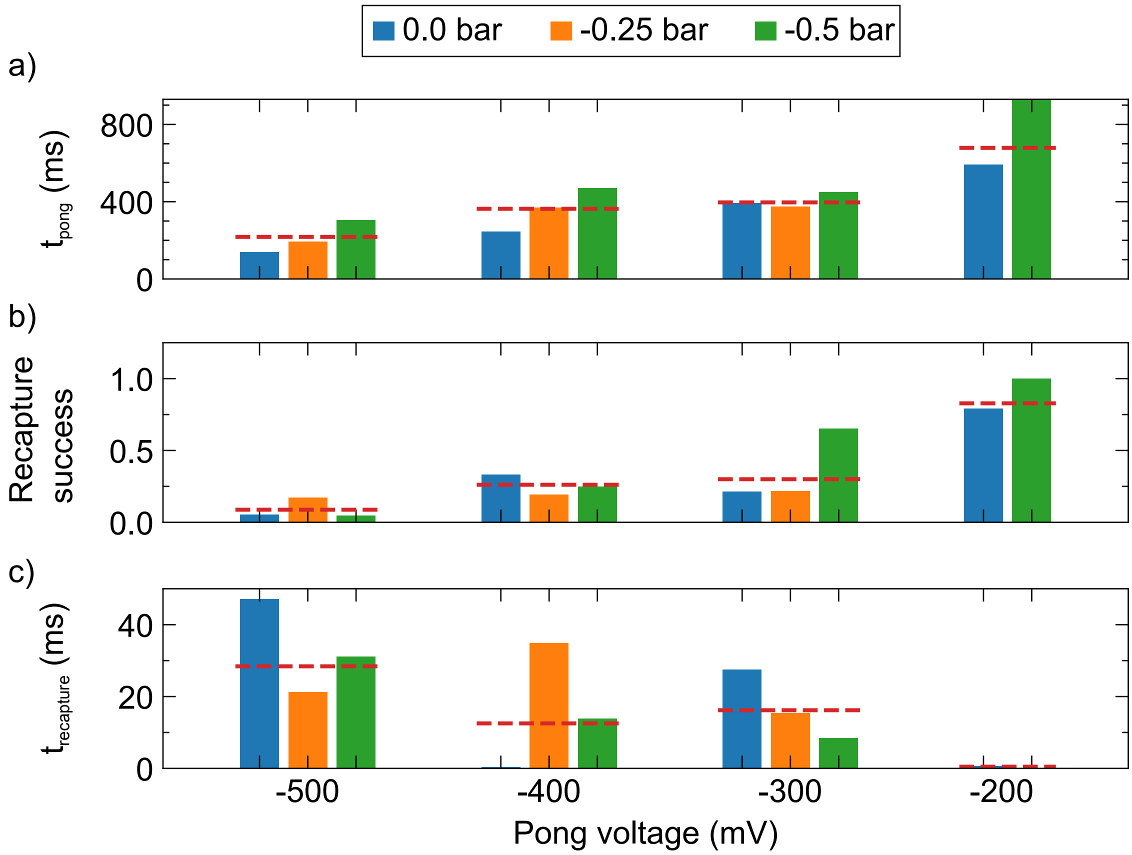


Fig. 6.8 Statistics from a total of 452 ping-pong operations at different pong voltages and pressures. Measurements were carried out with an equal mixture of the '000', '101', '111' and '011' barcodes adding up to a total concentration of 0.5 nM in 4 M LiCl at pH 8. Red dashed lines correspond to means of all ping-pong operations across the three pressures. Panel **a)** shows the mean times t_{pong} between the first voltage switch and the arrival of the pong translocation. The wait increases for smaller absolute pong voltages as expected from a lower driving force out of the pore, as well as higher negative pressures which create fluid flows into the pore. Panel **b)** shows the share of attempts in which a molecule is successfully recaptured following exit out of the pore. Lower absolute pong voltages and negative pressures increase the success rate by preventing molecules to be driven too far into the bulk. However, panel **c)** demonstrates that the same parameters reduce the time $t_{\text{recapture}}$ between the second voltage switch and the arrival of the recaptured event. Very low times $t_{\text{recapture}}$ such as those at -200 mV make it difficult to discern clear events because the signal is drowned out by the current spikes in the circuit's RC response. This suggests a pong voltage between -300 mV and -200 mV for successful ping-pong.

improvements become marginal (see Figure J.3 in appendix J). The three subplots in Figure 6.9 a) correspond to increasing values of P_{dec}, the likelihood that a translocation produces an analysable signal. Solid, dashed and dotted blue lines show increasing recapture success rates P_{rc} which refer to the probability of recapturing a molecule during one ping-pong iteration (cf. Figure 6.8 b)). In all plots ping-pong reduces the overall measurement time if the blue curves lie below the red line. This is always the case at low decoding probabilities P_{dec}, where re-analysing a molecule is preferable to waiting for the next one. For more successful decoding, the curves cross over at high ratios $\Delta t_{\mathrm{pp}}/\overline{\Delta t}$. This can be understood by considering that if a set of ping-pong iterations take too long to re-measure a molecule, it becomes more efficient to instead wait until one of the next ones produces a readable signal. For example, assuming 60 % of translocations produce an analysable signal, each ping-pong set takes $\Delta t_{\mathrm{pp}} \approx 4 \cdot 500\,\mathrm{ms} = 2\,\mathrm{s}$ (see Figure 6.8 a)) and one ping-pong iteration has a 50 % chance of recapturing the molecule, a base event frequency of $\bar{f} = 0.75\,\mathrm{Hz}$ makes a regular measurement preferable to ping-pong, as shown by the orange dot versus the red line in the bottom panel in Figure 6.9 a)[1].

Figure 6.9 b) illustrates the wait time dependence on the decoding probability P_{dec}. The advantage of the ping-pong technique shrinks for increasing P_{dec} and eventually turns into a longer wait as most translocations can be decoded on the first attempt. The crossover occurs earlier for lower recapture success rates P_{rc} and higher ratios $\Delta t_{\mathrm{pp}}/\overline{\Delta t}$ where ping-pong comes with a higher time cost and loses molecules more frequently. These results indicate that translocating molecules multiple times does not always speed up the overall measurement time. Particularly at high event frequencies it may be more beneficial to simply discard noisy signals and focus on analysable translocations. It should be noted, however, that for unequal mixtures of molecules the event frequency of the lowest-concentration target determines how long a measurement must be run to obtain a reliable result. In that case, ping-pong can greatly reduce the overall time even if it would slow down detection of molecules present at a higher concentration.

Under the right conditions, selectively repeating translocations for folded or otherwise difficult-to-read strands is thus a useful technique to enhance their signal-to-noise ratio. Fast switching ensures that the same molecule can be recaptured multiple times even at nM concentrations. With the right combination of voltage and pressure, recaptured translocations produce low-noise signals comparable to those at a constant positive voltage. The speedup in overall measurement time due to selective re-measuring of molecules can be assessed from an analysis of the waiting time between usable events. Ping-pong integrates directly into real-

[1] The calculation presented here assumes a constant number of ping-pong iterations, ignoring that the recapture cycle can be interrupted on successful decoding. While Δt_{pp} may be shorter on average, this does not change the general result.

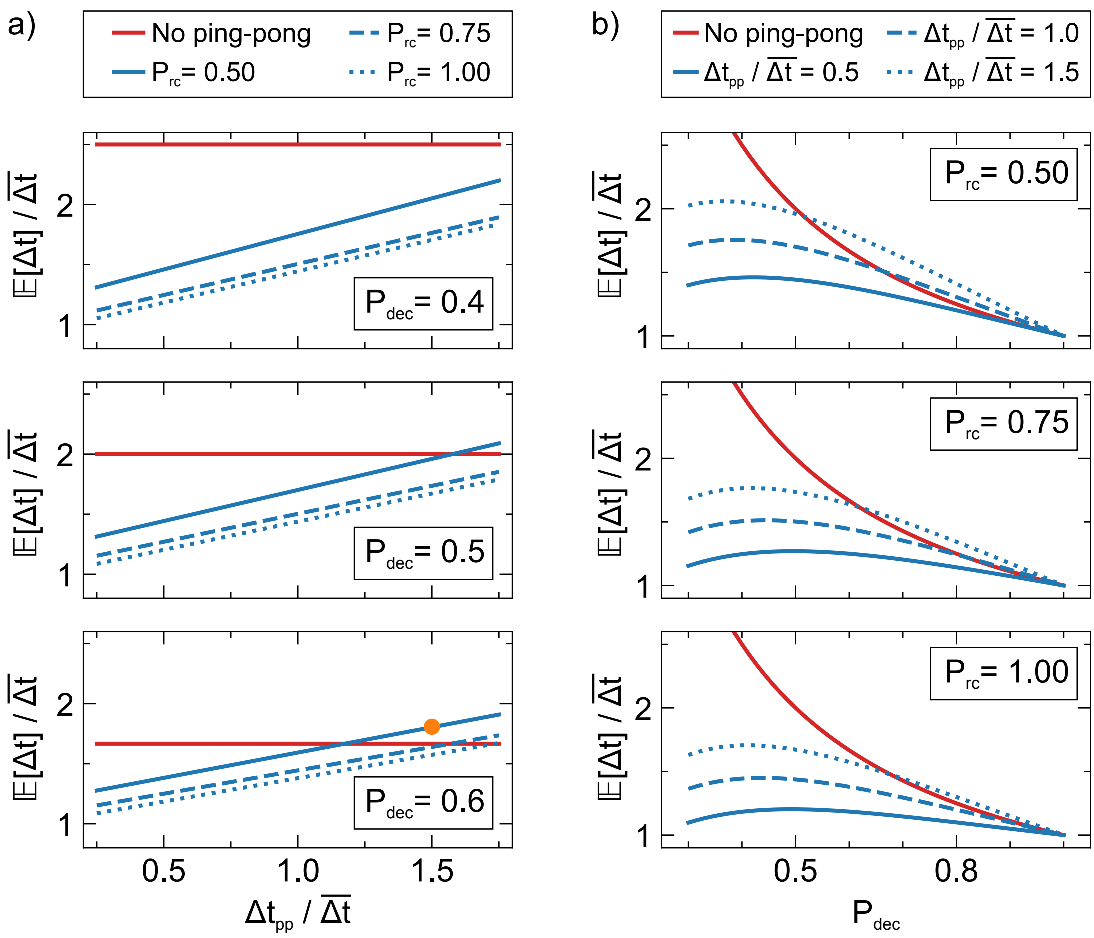


Fig. 6.9 Expected waiting times between **usable** events with (blue curves) and without ping-pong (red curve). Ping-pong reduces the overall measurement time where the blue curves lie below the red line. $\overline{\Delta t}$ is the mean time between **all** events without ping-pong, equivalent to $^1/_{\bar{f}}$ where $\bar{f}$ is the mean event frequency at a positive voltage. Δt_{pp} corresponds to the time added by a set of ping-pong iterations, in this case 4 of the recaptures shown in Figure 6.5. P_{dec} is the probability of a translocation producing an analysable signal, P_{rc} refers to the recapture success rate of a single ping-pong operation (cf. Figure 6.8 b)). Y-axes show normalised expected wait times between usable events, all values lie above 1 because usable events are a subset of all detected translocations. The crossover in the bottom subplot in panel **a)** shows that when a high share of translocations produces an analysable signal, it can be more efficient to simply wait for the next usable event instead of re-reading molecules with ping-pong. Panel **b)** confirms that particularly for low recapture success rates P_{rc} ping-pong can lead to longer measurement times, demonstrating that the experimental conditions determine whether repeated translocations are in fact beneficial. Appendix J contains further detail on the derivation.

time detection by triggering recapture whenever a DNA strand's modification sequence has not been decoded successfully. It complements deep learning-based molecular classification by adapting measurement parameters in response to the output of a neural network, making it an interesting building block for a real-time molecular decoding system.

6.4 Conclusion

In nanopore experiments, raw current data is usually recorded for 'off-line' processing at a later stage. This negates one of the key advantages of the technique, namely that the sequential signals allow stopping the measurement as soon as a clear picture has emerged. This chapter outlined components of a real-time detection system for DNA barcodes. I found that current hardware is fast enough to identify events and carry out neural network-based classification for a current bandwidth of 250 kHz and event frequencies on the order of 1 Hz. A quantile-based threshold constitutes a straightforward method to continuously monitor the pore noise. Given a classifier with accuracies $> 90\,\%$, 10s of readable events are sufficient to determine which mixture of 4 barcodes is present in the measurement solution. This is based on a probability calculation which gives a theoretical foundation to nanopore classification. The share of events which pass the filtering step before the classifier can drop to below 40 % due to noise and folded translocations. A way of boosting the readable fraction is the ping-pong technique, which refers to adaptively switching the voltage to re-measure molecules multiple times by driving them out of and back into the pore. I found that with the right voltage and pressure, DNA strands can be recaptured with success rates $> 90\,\%$. The expected wait between analysable events revealed that ping-pong reduces the measurement time if the time taken for the recapture procedure is short relative to the mean time between translocations. The underlying calculation can inform future measurements by highlighting which conditions make recapture advantageous.

The work presented in this chapter is a first effort to push the nanopore-based readout of DNA modifications towards sensing applications outside of a research setting. Classifying events in real time not only allows adapting parameters to optimally measure single molecules, but also provides the information to continuously update results including their confidence values. Combined with the theoretical considerations outlined in section 6.2, this gives a clear indication when a conclusive result has been reached and the measurement can be stopped. Future work should extend the probability calculations to unequal barcode mixtures as well as narrow down the ideal parameters for recapturing molecules with the ping-pong technique.

Chapter 7

Conclusion & Outlook

Wallace H. Coulter's patent does not reveal whether he was aware of his technique's extraordinary dynamic range [6]. As long as analytes modulate the ion current as they pass through a constriction, resistive pulse sensing can be used for their detection. As a result, the size of the ions themselves constitutes the lower bound on the smallest detectable molecule, for typical sub-nanometre radii this bound lies far below the diffraction limit for optical imaging. While the size of the constriction must be adapted to the sensing target, the same underlying principle can equally measure red blood cells and single DNA bases.

Recent work has built on resistive pulse sensing to encode information in the position and type of attachments along a double strand of DNA. Passing the designed molecules through a nanopore allows readout of the modification sequence within minutes. The technique has been employed in proof-of-concept studies for highly scalable sensing at the single-molecule level [45, 51] as well as long-term data storage with an information density on the order of tens of nanometres [3, 52]. All such applications critically rely on the accurate decoding of the nanopore current signal to reveal the modification sequence. In this work, I investigated experimental methods and analysis techniques to improve the readout of such structures.

Experimentally, I found that fluid flows influence the conformation and entry direction of strands translocating through a nanopore. For the conical glass pores studied here, a double strand of DNA can enter linearly in a single-file fashion or in a folded state in which part of the strand folds back onto itself in a hairpin-like manner. Fluid flows opposing the movement of DNA strongly promote single-file translocations, which is attributed to the higher drag of folded conformations compared to linear ones. Further evidence for this hypothesis comes from the preferred entry of an asymmetric strand with the high-drag end at the tail of the translocation. The emergence of predominantly unfolded events can be induced both with flows due to externally applied pressure and electro-osmotic flows (EOF) along the pore walls. These results demonstrate the impact of EOF at higher salt concentrations than expected from previous studies.

Folded translocations make the readout of attachment sequences more difficult, as secondary current peaks smear out in the folded region. Promoting single-file entry is therefore advantageous for modification-decoding applications. The ideal case would provide unfolded translocations at a high frequency and slow velocity, offering a short time-to-result and high temporal resolution. While outflows from the nanopore reduce folding, they also decrease the capture rate of DNA carriers. pH changes or the addition of neutral polymers such as polyethylene glycol (PEG) appear to maintain a higher translocation frequency while discouraging folded entry. This suggests that the flow fields can be fine-tuned through the various experimental parameters to obtain advantageous translocation behaviour. Particularly in combination with externally applied pressure, an improved understanding of the interplay between the different parameters will be necessary to realise the technique's full potential.

A promising approach to uncover DNA behaviour in and around the pore is fluorescent imaging of the translocation process. Thacker *et al.* [86] successfully visualised DNA translocations, however under specific salt conditions. Extending this work to a wider parameter range would shed light on the effect of changing flow regimes and additives such as PEG. Crucially, it would also answer open questions on the different stages of the translocation process. Although outflows from the pore discourage folding and reduce the event frequency, lowering the salt concentration still leads to a speed-up in velocity. This suggests that molecular capture is dominated by fluid flows while the translocation itself mainly depends on the electrophoretic force. Fluorescent imaging would reveal whether the DNA passes in a single-file fashion only after making several entry attempts while positioned in front of the pore, as expected from a barrier-limited process described previously [64, 134].

In addition to experimental methods, I investigated analysis techniques for the readout of the underlying modification sequences from the current signal. I found that Bayesian inference based on recently developed nested sampling algorithms is useful to analyse the folding state of DNA, but suffers from slow performance for high-dimensional models due to the exponential or power law scaling of computational demands with the number of model parameters. This makes nested sampling-based techniques unsuitable for direct analysis of modification sequences through evidence calculation of comprehensive models describing the current trace. While a partially Bayesian approach was shown to work for small datasets, the true strength of the technique will lie in comparing the validity of physical models of the translocation process once they become more refined. An interesting alternative for the decoding of modification sequences may be to describe the signal with a Hidden Markov

Model (HMM), where each DNA structure along the strand corresponds to a different state with defined transitions between them. This approach has been successfully employed to determine the opening states of single ion channels [135] and may allow segmenting translocation current traces into regions corresponding to the different attachments.

As a separate analysis strategy, I investigated the readout of modification sequences using convolutional neural networks (CNNs). Deep learning has achieved impressive results in a range of fields from computer vision to speech recognition over the last years, suggesting it may similarly benefit the processing of nanopore data. Two network architectures with different objectives were presented. The first one assigns a barcode to a given current trace out of a known set of possible barcodes, based on the data from Bell & Keyser's [51] encoding scheme. The network achieves a precision > 94 %, superior to the algorithm used in the original publication and approaching the accuracy of human labelling. Crucially, it is able to correctly process folded events, leading to a fivefold increase in the share of usable data. A key drawback is the need for thousands of labelled sample events per barcode. As a result, the architecture does not scale to long or variable modification sequences whose number of permutations doubles with every added bit.

The second presented architecture mitigates this issue by directly identifying the individual modification which produced a secondary current peak. Because the network learns to detect and assign peaks as opposed to classifying the whole current trace, it readily extends to previously unseen modification sequences. For a selection of unseen barcodes from the Bell & Keyser [51] dataset, it achieves an mAP score of 89.8 % with > 97 % correct predictions. The network is further shown to perform well on a novel scheme in which four different modifications produce distinct peaks, extending the binary encoding system to a quaternary one. The automatic feature extraction inherent in supervised learning means that the network can take into account peak characteristics which may not be obvious to the human eye. The disadvantage of the peak detection approach is that it cannot analyse folded events and requires post-processing of the output to match it to an underlying modification sequence.

Future work on deep learning-based approaches for nanopore data should investigate the feasibility of network training based on simulated data. This would reduce the considerable measurement effort currently required to compile training datasets. Preliminary work by Dr Karolis Misiunas on Generative Adversarial Networks (GANs) [136] suggests that simulated data could be successfully produced by a neural network itself. Another promising idea is the use of recurrent neural networks (RNNs) to process the current signal. This architecture is aimed at the analysis of time-series data and may allow taking into account longer-range dependencies between parts of the current trace than a convolutional kernel. Inspiration may be taken from the state-of-the-art base calling tools for Oxford Nanopore Technologies'

MinION sequencer, which use RNNs to infer DNA base sequences from the current signal [21, 137].

The last chapter in this thesis laid some of the groundwork for incorporating the readout of modification sequences into a real-time nanopore sensing platform. I found that CNN performance is sufficiently fast to classify events in real time. With a high-precision classifier, a probability calculation revealed that for 4 distinct barcodes from the Bell & Keyser [51] dataset, tens of events are enough to determine the content of a mixture with high confidence. The calculation can easily be extended to larger library sizes and provides a sound foundation to estimate the time-to-result for a given measurement. I further investigated the optimal parameters for re-measuring molecules by driving them out of and back into the pore. This ping-pong technique can selectively increase the signal-to-noise ratio for individual molecules. While further work is required to fine-tune the parameters, I found that given the right conditions ping-pong can be an effective method to reduce the time-to-result.

This work paves the way for the use of nanopore-based detection of structural DNA modifications in sensing and data storage applications. More than 60 years after Wallace H. Coulter's patent, his idea allows us to not only observe single DNA molecules, but directly visualise nanometre-sized attachments along the strand. It is extraordinary that this is possible with readily available benchtop instrumentation and nothing more than a very small hole which can be fabricated within minutes. Undoubtedly some issues must be solved before the technique can fulfil its promise as a fast and flexible nanoscale sensor. Extending the pore lifetime in complex samples, optimising the experimental conditions and scaling up fabrication are merely a selection of the challenges that lie ahead. Nevertheless, the ingenious ideas and tireless efforts of the last six decades which have transformed a cell counter into a DNA sequencer leave no doubt that these hurdles will be overcome.

References

1. Ermann, N. *et al.* Promoting Single-File DNA Translocations through Nanopores Using Electro-Osmotic Flow. *The Journal of Chemical Physics* **149,** 163311 (2018).
2. Ermann, N., Chen, K. & Keyser, U. F. Bayesian Inference for Nanopore Data Analysis. arXiv: 1904.01040 (2019).
3. Chen, K. *et al.* Digital Data Storage Using DNA Nanostructures and Solid-State Nanopores. *Nano Letters* **19,** 1210–1215 (2019).
4. Misiunas, K., Ermann, N. & Keyser, U. F. QuipuNet: Convolutional Neural Network for Single-Molecule Nanopore Sensing. *Nano Letters* **18,** 4040–4045 (2018).
5. Weckman, N. E. *et al.* Multiplexed DNA Identification Using Site Specific dCas9 Barcodes and Nanopore Sensing. *ACS Sensors* **4,** 2065–2072 (2019).
6. Coulter, W. H. *pat.* 2.656.508 (United States of America) (1953).
7. Lines, R. W. *The Electrical Sensing Zone Method (The Coulter Principle)* in *Particle Size Analysis* (eds Stanley-Wood, N. G. & Lines, R. W.) 102 (Royal Soc. of Chemistry, Cambridge, 1992).
8. Hodgkin, A. L., Huxley, A. F. & Katz, B. Measurement of Current-Voltage Relations in the Membrane of the Giant Axon of Loligo. *The Journal of Physiology* **116,** 424–448 (1952).
9. Hodgkin, A. L. & Huxley, A. F. Currents Carried by Sodium and Potassium Ions through the Membrane of the Giant Axon of Loligo. *The Journal of Physiology* **116,** 449–472 (1952).
10. Hodgkin, A. L. & Huxley, A. F. Propagation of Electrical Signals Along Giant Nerve Fibres. *Proceedings of the Royal Society of London. Series B, Biological Sciences* **140,** 177–183 (1952).
11. Hladky, S. B. & Haydon, D. A. Discreteness of Conductance Change in Bimolecular Lipid Membranes in the Presence of Certain Antibiotics. *Nature* **225,** 451–453 (1970).
12. Neher, E. & Sakmann, B. Single-Channel Currents Recorded from Membrane of Denervated Frog Muscle Fibres. *Nature* **260,** 799–802 (1976).
13. Kasianowicz, J. J., Brandin, E., Branton, D. & Deamer, D. W. Characterization of Individual Polynucleotide Molecules Using a Membrane Channel. *Proceedings of the National Academy of Sciences* **93,** 13770–13773 (1996).
14. Krasilnikov, O., Sabirov, R., Ternovsky, V., Merzliak, P. & Muratkhodjaev, J. A Simple Method for the Determination of the Pore Radius of Ion Channels in Planar Lipid Bilayer Membranes. *FEMS Microbiology Letters* **105,** 93–100 (1992).

15. Vodyanoy, I. & Bezrukov, S. M. Sizing of an Ion Pore by Access Resistance Measurements. *Biophysical Journal* **62,** 10–11 (1992).

16. Bezrukov, S. M. & Vodyanoy, I. Probing Alamethicin Channels with Water-Soluble Polymers. Effect on Conductance of Channel States. *Biophysical Journal* **64,** 16–25 (1993).

17. Patel, A. *et al.* MinION Rapid Sequencing: Review of Potential Applications in Neurosurgery. *Surgical Neurology International* **9,** 157 (2018).

18. Tyler, A. D. *et al.* Evaluation of Oxford Nanopore's MinION Sequencing Device for Microbial Whole Genome Sequencing Applications. *Scientific Reports* **8,** 10931 (2018).

19. Kono, N. & Arakawa, K. Nanopore Sequencing: Review of Potential Applications in Functional Genomics. *Development, Growth & Differentiation* **61,** 316–326 (2019).

20. Minervini, C. F. *et al.* Nanopore Sequencing in Blood Diseases: A Wide Range of Opportunities. *Frontiers in Genetics* **11,** 76 (2020).

21. Magi, A., Semeraro, R., Mingrino, A., Giusti, B. & D'Aurizio, R. Nanopore Sequencing Data Analysis: State of the Art, Applications and Challenges. *Briefings in Bioinformatics* **9,** 1256–1272 (2017).

22. Kobayashi, Y. & Martin, C. R. Toward a Molecular Coulter® Counter Type Device. *Journal of Electroanalytical Chemistry* **431,** 29–33 (1997).

23. Kobayashi, Y. & Martin, C. R. Highly Sensitive Methods for Electroanalytical Chemistry Based on Nanotubule Membranes. *Analytical Chemistry* **71,** 3665–3672 (1999).

24. Li, J. *et al.* Ion-Beam Sculpting at Nanometre Length Scales. *Nature* **412,** 166–169 (2001).

25. Storm, A. J., Chen, J. H., Ling, X. S., Zandbergen, H. W. & Dekker, C. Fabrication of Solid-State Nanopores with Single-Nanometre Precision. *Nature Materials* **2,** 537–540 (2003).

26. Kwok, H., Briggs, K. & Tabard-Cossa, V. Nanopore Fabrication by Controlled Dielectric Breakdown. *PLoS ONE* **9,** e92880 (2014).

27. Briggs, K., Kwok, H. & Tabard-Cossa, V. Automated Fabrication of 2-Nm Solid-State Nanopores for Nucleic Acid Analysis. *Small* **10,** 2077–2086 (2014).

28. Briggs, K. *et al.* Kinetics of Nanopore Fabrication during Controlled Breakdown of Dielectric Membranes in Solution. *Nanotechnology* **26,** 084004 (2015).

29. Steinbock, L., Stober, G. & Keyser, U. Sensing DNA-Coatings of Microparticles Using Micropipettes. *Biosensors and Bioelectronics* **24,** 2423–2427 (2009).

30. Bell, N. A. W. *et al.* DNA Origami Nanopores. *Nano Letters* **12,** 512–517 (2012).

31. Wei, R., Martin, T. G., Rant, U. & Dietz, H. DNA Origami Gatekeepers for Solid-State Nanopores. *Angewandte Chemie International Edition* **51,** 4864–4867 (2012).

32. Hernández-Ainsa, S. *et al.* DNA Origami Nanopores for Controlling DNA Translocation. *ACS Nano* **7,** 6024–6030 (2013).

33. Han, A. *et al.* Sensing Protein Molecules Using Nanofabricated Pores. *Applied Physics Letters* **88,** 093901 (2006).

34. Plesa, C. *et al.* Fast Translocation of Proteins through Solid State Nanopores. *Nano Letters* **13,** 658–663 (2013).

35. Li, W. *et al.* Single Protein Molecule Detection by Glass Nanopores. *ACS Nano* **7,** 4129–4134 (2013).

36. Kowalczyk, S. W., Hall, A. R. & Dekker, C. Detection of Local Protein Structures along DNA Using Solid-State Nanopores. *Nano Letters* **10,** 324–328 (2010).

37. Smeets, R. M. M., Kowalczyk, S. W., Hall, A. R., Dekker, N. H. & Dekker, C. Translocation of RecA-Coated Double-Stranded DNA through Solid-State Nanopores. *Nano Letters* **9,** 3089–3095 (2009).

38. Singer, A. *et al.* Nanopore Based Sequence Specific Detection of Duplex DNA for Genomic Profiling. *Nano Letters* **10,** 738–742 (2010).

39. Hanz, G. M., Jung, B., Giesbertz, A., Juhasz, M. & Weinhold, E. Sequence-Specific Labeling of Nucleic Acids and Proteins with Methyltransferases and Cofactor Analogues. *Journal of Visualized Experiments* **93,** e52014 (2014).

40. Chen, K. *et al.* Ionic Current-Based Mapping of Short Sequence Motifs in Single DNA Molecules Using Solid-State Nanopores. *Nano Letters* **17,** 5199–5205 (2017).

41. Yang, W. *et al.* Detection of CRISPR-dCas9 on DNA with Solid-State Nanopores. *Nano Letters* **18,** 6469–6474 (2018).

42. Bell, N. A. W. & Keyser, U. F. Specific Protein Detection Using Designed DNA Carriers and Nanopores. *Journal of the American Chemical Society* **137,** 2035–2041 (2015).

43. Plesa, C., Ruitenberg, J. W., Witteveen, M. J. & Dekker, C. Detection of Individual Proteins Bound along DNA Using Solid-State Nanopores. *Nano Letters* **15,** 3153–3158 (2015).

44. Kong, J., Bell, N. A. W. & Keyser, U. F. Quantifying Nanomolar Protein Concentrations Using Designed DNA Carriers and Solid-State Nanopores. *Nano Letters* **16,** 3557–3562 (2016).

45. Sze, J. Y. Y., Ivanov, A. P., Cass, A. E. G. & Edel, J. B. Single Molecule Multiplexed Nanopore Protein Screening in Human Serum Using Aptamer Modified DNA Carriers. *Nature Communications* **8,** 1552 (2017).

46. Loh, A. Y. Y. *et al.* Electric Single-Molecule Hybridization Detector for Short DNA Fragments. *Analytical Chemistry* **90,** 14063–14071 (2018).

47. Rothemund, P. W. K. Folding DNA to Create Nanoscale Shapes and Patterns. *Nature* **440,** 297–302 (2006).

48. Castro, C. E. *et al.* A Primer to Scaffolded DNA Origami. *Nature Methods* **8,** 221–229 (2011).

49. Plesa, C., van Loo, N., Ketterer, P., Dietz, H. & Dekker, C. Velocity of DNA during Translocation through a Solid-State Nanopore. *Nano Letters* **15,** 732–737 (2015).

50. Wanunu, M. Nanopores: A Journey towards DNA Sequencing. *Physics of Life Reviews* **9,** 125–158 (2012).

51. Bell, N. A. W. & Keyser, U. F. Digitally Encoded DNA Nanostructures for Multiplexed, Single-Molecule Protein Sensing with Nanopores. *Nature Nanotechnology* **11,** 645–651 (2016).

52. Chen, K., Zhu, J., Bošković, F. & Keyser, U. F. Nanopore-Based DNA Hard Drives for Rewritable and Secure Data Storage. *Nano Letters.* Published online. doi:10.1021/acs.nanolett.0c00755 (2020).

53. Bell, N. A. W. *et al.* Multiplexed Ionic Current Sensing with Glass Nanopores. *Lab on a Chip* **13,** 1859 (2013).

54. Smeets, R. M. M., Keyser, U. F., Dekker, N. H. & Dekker, C. Noise in Solid-State Nanopores. *Proceedings of the National Academy of Sciences* **105,** 417–421 (2008).

55. Knowles, S. F., Keyser, U. F. & Thorneywork, A. L. Noise Properties of Rectifying and Non-Rectifying Nanopores. *Nanotechnology* **31,** 10LT01 (2019).

56. Porr, B. *Iir1* 2020. https://github.com/berndporr/iir1 (Feb. 7, 2020).

57. Li, J., Gershow, M., Stein, D., Brandin, E. & Golovchenko, J. A. DNA Molecules and Configurations in a Solid-State Nanopore Microscope. *Nature Materials* **2,** 611–615 (2003).

58. Storm, A. J., Chen, J. H., Zandbergen, H. W. & Dekker, C. Translocation of Double-Strand DNA through a Silicon Oxide Nanopore. *Physical Review E* **71,** 051903 (2005).

59. Mihovilovic, M., Hagerty, N. & Stein, D. Statistics of DNA Capture by a Solid-State Nanopore. *Physical Review Letters* **110,** 028102 (2013).

60. Steinbock, L. J., Otto, O., Chimerel, C., Gornall, J. & Keyser, U. F. Detecting DNA Folding with Nanocapillaries. *Nano Letters* **10,** 2493–2497 (2010).

61. Behrens, S. H. & Grier, D. G. The Charge of Glass and Silica Surfaces. *The Journal of Chemical Physics* **115,** 6716–6721 (2001).

62. Sonnefeld, J. Determination of Surface Charge Density Parameters of Silicon Nitride. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **108,** 27–31 (1996).

63. Schoch, R. B., Han, J. & Renaud, P. Transport Phenomena in Nanofluidics. *Reviews of Modern Physics* **80,** 839–883 (2008).

64. Muthukumar, M. *Polymer Translocation* (CRC Press, 2016).

65. Tikanen, A. C. & Fawcett, W. R. Application of the Mean Spherical Approximation and Ion Association to Describe the Activity Coefficients of Aqueous 1:1 Electrolytes. *Journal of Electroanalytical Chemistry* **439,** 107–113 (1997).

66. Wei, Y.-Z. & Sridhar, S. Dielectric Spectroscopy up to 20 GHz of LiCl/H2O Solutions. *The Journal of Chemical Physics* **92,** 923–928 (1990).

67. Laohakunakorn, N. & Keyser, U. F. Electroosmotic Flow Rectification in Conical Nanopores. *Nanotechnology* **26,** 275202 (2015).

68. Laohakunakorn, N., Thacker, V. V., Muthukumar, M. & Keyser, U. F. Electroosmotic Flow Reversal Outside Glass Nanopores. *Nano Letters* **15,** 695–702 (2015).

69. Ma, X. & Pawlik, M. Effect of Alkali Metal Cations on Adsorption of Guar Gum onto Quartz. *Journal of Colloid and Interface Science* **289,** 48–55 (2005).

70. Siwy, Z. S. Ion-Current Rectification in Nanopores and Nanotubes with Broken Symmetry. *Advanced Functional Materials* **16,** 735–746 (2006).

71. Laohakunakorn, N. *et al.* A Landau–Squire Nanojet. *Nano Letters* **13,** 5141–5146 (2013).

72. Bell, N. A. W., Muthukumar, M. & Keyser, U. F. Translocation Frequency of Double-Stranded DNA through a Solid-State Nanopore. *Physical Review E* **93,** 022401 (2016).

73. Kowalczyk, S. W., Grosberg, A. Y., Rabin, Y. & Dekker, C. Modeling the Conductance and DNA Blockade of Solid-State Nanopores. *Nanotechnology* **22,** 315101 (2011).

74. Baumann, C. G., Smith, S. B., Bloomfield, V. A. & Bustamante, C. Ionic Effects on the Elasticity of Single DNA Molecules. *Proceedings of the National Academy of Sciences* **94,** 6185–6190 (1997).

75. Mc Hugh, J., Andresen, K. & Keyser, U. F. Cation Dependent Electroosmotic Flow in Glass Nanopores. *Applied Physics Letters* **115,** 113702 (2019).

76. Liu, J. Adsorption of DNA onto Gold Nanoparticles and Graphene Oxide: Surface Science and Applications. *Physical Chemistry Chemical Physics* **14,** 10485–10496 (2012).

77. Doherty, E. A. S., Meagher, R. J., Albarghouthi, M. N. & Barron, A. E. Microchannel Wall Coatings for Protein Separations by Capillary and Chip Electrophoresis. *ELECTROPHORESIS* **24,** 34–54 (2003).

78. Preisler, J. & Yeung, E. S. Characterization of Nonbonded Poly(Ethylene Oxide) Coating for Capillary Electrophoresis via Continuous Monitoring of Electroosmotic Flow. *Analytical Chemistry* **68,** 2885–2889 (1996).

79. Doherty, E. A. S. *et al.* Critical Factors for High-Performance Physically Adsorbed (Dynamic) Polymeric Wall Coatings for Capillary Electrophoresis of DNA. *ELECTROPHORESIS* **23,** 2766–2776 (2002).

80. Hickey, O. A., Harden, J. L. & Slater, G. W. Molecular Dynamics Simulations of Optimal Dynamic Uncharged Polymer Coatings for Quenching Electro-Osmotic Flow. *Physical Review Letters* **102,** 108304 (2009).

81. Rempp, P. Contribution à l'étude des solutions de molécules en chaîne à squelette oxygéné - I. Préparation de substances-modèle ; détermination des masses moléculaires et étude de la polydispersité des polyoxyéthylèneglycols. *Journal de Chimie Physique* **54,** 421–431 (1957).

82. Scherrer, R. & Gerhardt, P. Molecular Sieving by the Bacillus Megaterium Cell Wall and Protoplast. *Journal of Bacteriology* **107,** 718–735 (1971).

83. Chen, P. *et al.* Atomic Layer Deposition to Fine-Tune the Surface Properties and Diameters of Fabricated Nanopores. *Nano Letters* **4,** 1333–1337 (2004).

84. He, Y., Tsutsui, M., Fan, C., Taniguchi, M. & Kawai, T. Gate Manipulation of DNA Capture into Nanopores. *ACS Nano* **5,** 8391–8397 (2011).

85. He, Y., Tsutsui, M., Fan, C., Taniguchi, M. & Kawai, T. Controlling DNA Translocation through Gate Modulation of Nanopore Wall Surface Charges. *ACS Nano* **5,** 5509–5518 (2011).

86. Thacker, V. V., Ghosal, S., Hernández-Ainsa, S., Bell, N. A. W. & Keyser, U. F. Studying DNA Translocation in Nanocapillaries Using Single Molecule Fluorescence. *Applied Physics Letters* **101,** 223704 (2012).

87. Lu, B. *et al.* Pressure-Controlled Motion of Single Polymers through Solid-State Nanopores. *Nano Letters* **13,** 3048–3052 (2013).
88. Hoogerheide, D. P., Lu, B. & Golovchenko, J. A. Pressure–Voltage Trap for DNA near a Solid-State Nanopore. *ACS Nano* **8,** 7384–7391 (2014).
89. Rempfer, G. *et al.* Selective Trapping of DNA Using Glass Microcapillaries. *Langmuir* **32,** 8525–8532 (2016).
90. Yamazaki, H. *et al.* Label-Free Single-Molecule Thermoscopy Using a Laser-Heated Nanopore. *Nano Letters* **17,** 7067–7074 (2017).
91. Faber, T. E. *Fluid Dynamics for Physicists* (Cambridge University Press, Cambridge, 1995).
92. Steinbock, L. J., Lucas, A., Otto, O. & Keyser, U. F. Voltage-Driven Transport of Ions and DNA through Nanocapillaries. *ELECTROPHORESIS* **33,** 3480–3487 (2012).
93. Pedone, D., Firnkes, M. & Rant, U. Data Analysis of Translocation Events in Nanopore Experiments. *Analytical Chemistry* **81,** 9689–9694 (2009).
94. Balijepalli, A. *et al.* Quantifying Short-Lived Events in Multistate Ionic Current Measurements. *ACS Nano* **8,** 1547–1553 (2014).
95. Raillon, C., Granjon, P., Graf, M., Steinbock, L. J. & Radenovic, A. Fast and Automatic Processing of Multi-Level Events in Nanopore Translocation Experiments. *Nanoscale* **4,** 4916 (2012).
96. Forstater, J. H. *et al.* MOSAIC: A Modular Single-Molecule Analysis Interface for Decoding Multistate Nanopore Data. *Analytical Chemistry* **88,** 11900–11907 (2016).
97. Plesa, C. & Dekker, C. Data Analysis Methods for Solid-State Nanopores. *Nanotechnology* **26,** 084003 (2015).
98. LII. An essay towards solving a Problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London* **53,** 370–418 (1763).
99. *Bayesian Data Analysis* (ed Gelman, A.) Reprinted, 1. ed. 1995 (Chapman & Hall, London, 1998).
100. Sivia, D. S. & Skillings, J. H. *Data Analysis: A Bayesian Tutorial* (Oxford Univ. Press, Oxford, 2012).
101. MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms* (Cambridge University Press, Cambridge, UK ; New York, 2003).
102. Hines, K. E. A Primer on Bayesian Inference for Biophysical Systems. *Biophysical Journal* **108,** 2103–2113 (2015).
103. Feroz, F. & Hobson, M. P. Multimodal Nested Sampling: An Efficient and Robust Alternative to Markov Chain Monte Carlo Methods for Astronomical Data Analyses. *Monthly Notices of the Royal Astronomical Society* **384,** 449–463 (2008).
104. Feroz, F., Hobson, M. P. & Bridges, M. MultiNest: An Efficient and Robust Bayesian Inference Tool for Cosmology and Particle Physics. *Monthly Notices of the Royal Astronomical Society* **398,** 1601–1614 (2009).
105. Plesa, C. *et al.* Direct Observation of DNA Knots Using a Solid-State Nanopore. *Nature Nanotechnology* **11,** 1093–1097 (2016).

106. Handley, W. J., Hobson, M. P. & Lasenby, A. N. Polychord: Next-Generation Nested Sampling. *Monthly Notices of the Royal Astronomical Society* **453,** 4385–4399 (2015).
107. Kesselheim, S., Müller, W. & Holm, C. Origin of Current Blockades in Nanopore Translocation Experiments. *Physical Review Letters* **112,** 018101 (2014).
108. Wang, V., Ermann, N. & Keyser, U. F. Current Enhancement in Solid-State Nanopores Depends on Three-Dimensional DNA Structure. *Nano Letters* **19,** 5661–5666 (2019).
109. Goodfellow, I. J., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
110. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *ImageNet Classification with Deep Convolutional Neural Networks* in *Advances in Neural Information Processing Systems 25* (eds Pereira, F., Burges, C. J. C., Bottou, L. & Weinberger, K. Q.) 1097–1105 (Curran Associates, Inc., 2012).
111. Hinton, G. *et al.* Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine* **29,** 82–97 (2012).
112. Albrecht, T., Slabaugh, G., Alonso, E. & Al-Arif, S. M. R. Deep Learning for Single-Molecule Science. *Nanotechnology* **28,** 423001 (2017).
113. Celik, N. *et al.* Deep-Channel Uses Deep Neural Networks to Detect Single-Molecule Events from Patch-Clamp Data. *Communications Biology* **3,** 3 (2020).
114. Zhang, P. *et al.* Analyzing Complex Single-Molecule Emission Patterns with Deep Learning. *Nature Methods* **15,** 913–916 (2018).
115. Nehme, E., Weiss, L. E., Michaeli, T. & Shechtman, Y. Deep-STORM: Super-Resolution Single-Molecule Microscopy by Deep Learning. *Optica* **5,** 458–464 (2018).
116. Murphy, K. P. *Machine Learning: A Probabilistic Perspective* (MIT Press, Cambridge, MA, 2012).
117. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. arXiv: 1412.6980 (2017).
118. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning Representations by Back-Propagating Errors. *Nature* **323,** 533–536 (1986).
119. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv: 1502.03167 (2015).
120. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15,** 1929–1958 (2014).
121. LeCun, Y. *Generalization and Network Design Strategies* Technical Report CRG-TR-98-4 (University of Toronto Connectionist Research Group, 1989). http://yann.lecun.com/exdb/publis/pdf/lecun-89.pdf.
122. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17,** 261–272 (2020).
123. Razavian, A. S., Azizpour, H., Sullivan, J. & Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. arXiv: 1403.6382 (2014).
124. Liu, W. *et al. SSD: Single Shot MultiBox Detector* in *Computer Vision – ECCV 2016* (eds Leibe, B., Matas, J., Sebe, N. & Welling, M.) 21–37 (Springer International Publishing, Cham, 2016).

125. Girshick, R. Fast R-CNN. arXiv: 1504.08083 (2015).

126. Chollet, F. *et al. Keras* 2015. https://keras.io (Feb. 4, 2020).

127. Abadi, M. *et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* 2015. https://www.tensorflow.org/.

128. Ferrari, P. *A Keras Port of Single Shot MultiBox Detector* 2018. https://github.com/pierluigiferrari/ssd_keras (Feb. 4, 2020).

129. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* **88,** 303–338 (2010).

130. Hoiem, D., Chodpathumwan, Y. & Dai, Q. *Diagnosing Error in Object Detectors* in *Computer Vision – ECCV 2012* (eds Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y. & Schmid, C.) 340–353 (Springer, Berlin, Heidelberg, 2012).

131. Gershow, M. & Golovchenko, J. A. Recapturing and Trapping Single Molecules with a Solid-State Nanopore. *Nature Nanotechnology* **2,** 775–779 (2007).

132. Plesa, C., Cornelissen, L., Tuijtel, M. W. & Dekker, C. Non-Equilibrium Folding of Individual DNA Molecules Recaptured up to 1000 Times in a Solid State Nanopore. *Nanotechnology* **24,** 475101 (2013).

133. Bell, N. A. W., Chen, K., Ghosal, S., Ricci, M. & Keyser, U. F. Asymmetric Dynamics of DNA Entering and Exiting a Strongly Confining Nanopore. *Nature Communications* **8,** 380 (2017).

134. Grosberg, A. Y. & Rabin, Y. DNA Capture into a Nanopore: Interplay of Diffusion and Electrohydrodynamics. *The Journal of Chemical Physics* **133,** 165102 (2010).

135. Hines, K. E., Bankston, J. R. & Aldrich, R. W. Analyzing Single-Molecule Time Series via Nonparametric Bayesian Inference. *Biophysical Journal* **108,** 540–556 (2015).

136. Goodfellow, I. J. *et al.* Generative Adversarial Networks. arXiv: 1406.2661 (2014).

137. Wick, R. R., Judd, L. M. & Holt, K. E. Performance of Neural Network Basecalling Tools for Oxford Nanopore Sequencing. *Genome Biology* **20,** 129 (2019).

138. *CRC Handbook of Chemistry and Physics* (eds Rumble, J. R., Lide, D. R. & Bruno, T. J.) 100th ed. (CRC Press, Boca Raton, FL, 2019).

Appendix A

Nanopore size estimation

Nanopores fabricated by laser-pulling of glass capillaries sit at the tip of a conical geometry. Scanning electron microscopy (SEM) images have revealed that the cone tapers at two angles as shown in Figure A.1 [42]. Bell & Keyser [42] introduced a model to relate the measured pore resistance to its diameter, given by

$$R = \frac{1}{\sigma}\left(\frac{4L}{\pi d(d+2L\tan(\theta_1))} + \frac{2}{\pi(d+2L\tan(\theta_1))\tan(\theta_2)} + \frac{1}{2d}\right)$$

where R is the resistance and σ the solution conductivity (see appendix B). I used this model to estimate pore sizes in combination with taper angle values of $\theta_1 = 0.092\,\text{rad}$ and $\theta_2 = 0.046\,\text{rad}$ as given in [42].

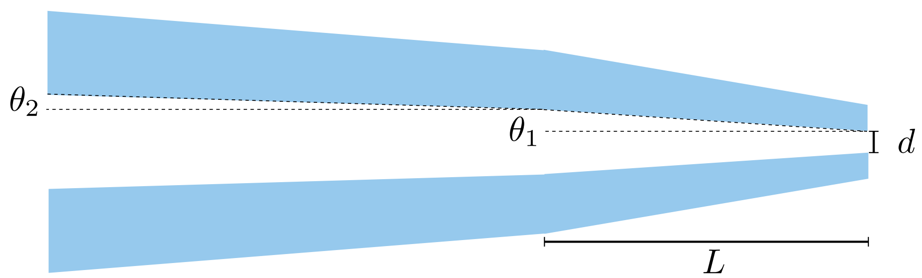


Fig. A.1 Double-taper geometry of nanopores fabricated from laser-pulling of glass capillaries.

Appendix B

Solution conductivities

Values taken from [138], lines are cubic spline interpolations.

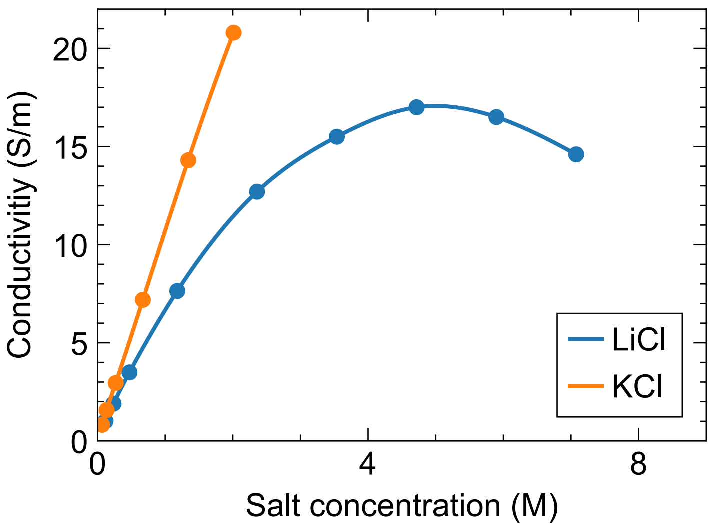


Fig. B.1

Appendix C

Pressure calibration

The electronic regulator adjusts the pressure P in the range -1 bar to 5 bar proportionally to an input voltage V between 0 V and 10 V, defining a mapping

$$P = (V \cdot 0.6) - 1.0$$

shown by the blue curve in Figure C.1. The orange curve illustrates the actual pressure measured with an external meter, indicating a considerable deviation from the expected value. To correct for this error, a target pressure $\tilde{P}$ is associated with a voltage via

$$V = \frac{\tilde{P} + 1}{0.6} \cdot C$$

with a correction factor $C = 1.175$ resulting in the green curve. It should be noted that due to the imperfect vacuum, the maximum achievable negative pressure is about -0.75 bar as shown by the orange line at 0 V. The correction presented here does not take into account this effect, where this was relevant in section 3.4 it was corrected through interpolation in the data analysis.

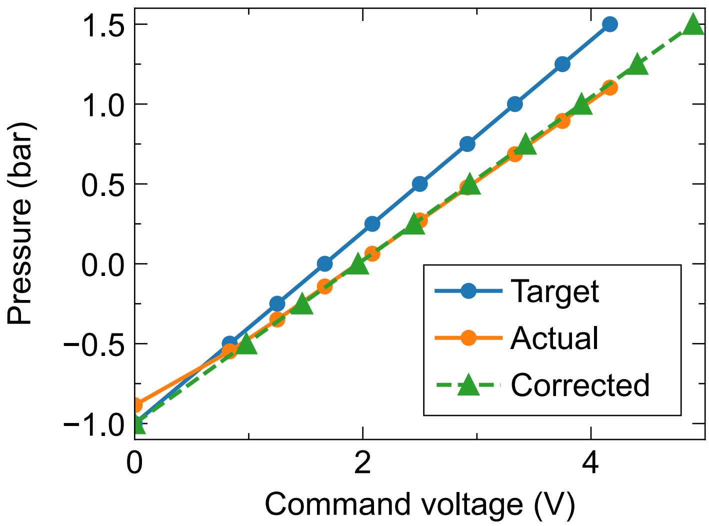

1.5
1.0
0.5
0.0
−0.5
−1.0
Pressure (bar)
0
2
4
Command voltage (V)
Target
Actual
Corrected

Fig. C.1

Appendix D

DNA sequences

D.1 Carrier oligos

Complementary oligonucleotides for the m13 DNA carrier as described in [51].

1	TTTTCGTAATCATGGTCATAGCTGTTTCCTGTGTGAAATTGTTATC
2	CGCTCACAATTCCACACAACATACGAGCCGGAAGCATA
3	AAGTGTAAAGCCTGGGGTGCCTAATGAGTGAGCTAACT
4	CACATTAATTGCGTTGCGCTCACTGCCCGCTTTCCAGT
5	CGGGAAACCTGTCGTGCCAGCTGCATTAATGAATCGGC
6	CAACGCGCGGGGAGAGGCGGTTTGCGTATTGGGCGCCA
7	GGGTGGTTTTTCTTTTCACCAGTGAGACGGGCAACAGC
8	TGATTGCCCTTCACCGCCTGGCCCTGAGAGAGTTGCAG
9	CAAGCGGTCCACGCTGGTTTGCCCCAGCAGGCGAAAAT
10	CCTGTTTGATGGTGGTTCCGAAATCGGCAAAATCCCTT
11	ATAAATCAAAAGAATAGCCCGAGATAGGGTTGAGTGTT
12	GTTCCAGTTTGGAACAAGAGTCCACTATTAAAGAACGT
13	GGACTCCAACGTCAAAGGGCGAAAAACCGTCTATCAGG
14	GCGATGGCCCACTACGTGAACCATCACCCAAATCAAGT
15	TTTTTGGGGTCGAGGTGCCGTAAAGCACTAAATCGGAA
16	CCCTAAAGGGAGCCCCCGATTTAGAGCTTGACGGGGAA
17	AGCCGGCGAACGTGGCGAGAAAGGAAGGGAAGAAAGCG
18	AAAGGAGCGGGCGCTAGGGCGCTGGCAAGTGTAGCGGT
19	CACGCTGCGCGTAACCACCACACCCGCCGCGCTTAATG
20	CGCCGCTACAGGGCGCGTACTATGGTTGCTTTGACGAG
21	CACGTATAACGTGCTTTCCTCGTTAGAATCAGAGCGGG
22	AGCTAAACAGGAGGCCGATTAAAGGGATTTTAGACAGG
23	AACGGTACGCCAGAATCCTGAGAAGTGTTTTTATAATC
24	AGTGAGGCCACCGAGTAAAAGAGTCTGTCCATCACGCA
25	AATTAACCGTTGTAGCAATACTTCTTTGATTAGTAATA

26	ACATCACTTGCCTGAGTAGAAGAACTCAAACTATCGGC
27	CTTGCTGGTAATATCCAGAACAATATTACCGCCAGCCA
28	TTGCAACAGGAAAAACGCTCATGGAAATACCTACATTT
29	TGACGCTCAATCGTCTGAAATGGATTATTTACATTGGC
30	AGATTCACCAGTCACACGACCAGTAATAAAAGGGACAT
31	TCTGGCCAACAGAGATAGAACCCTTCTGACCTGAAAGC
32	GTAAGAATACGTGGCACAGACAATATTTTTGAATGGCT
33	ATTAGTCTTTAATGCGCGAACTGATAGCCCTAAAACAT
34	CGCCATTAAAAATACCGAACGAACCACCAGCAGAAGAT
35	AAAACAGAGGTGAGGCGGTCAGTATTAACACCGCCTGC
36	AACAGTGCCACGCTGAGAGCCAGCAGCAAATGAAAAAT
37	CTAAAGCATCACCTTGCTGAACCTCAAATATCAAACCC
38	TCAATCAATATCTGGTCAGTTGGCAAATCAACAGTTGA
39	AAGGAATTGAGGAAGGTTATCTAAAATATCTTTAGGAG
40	CACTAACAACTAATAGATTAGAGCCGTCAATAGATAAT
41	ACATTTGAGGATTTAGAAGTATTAGACTTTACAAACAA
42	TTCGACAACTCGTATTAAATCCTTTGCCCGAACGTTAT
43	TAATTTTAAAAGTTTGAGTAACATTATCATTTTGCGGA
44	ACAAAGAAACCACCAGAAGGAGCGGAATTATCATCATA
45	TTCCTGATTATCAGATGATGGCAATTCATCAATATAAT
46	CCTGATTGTTTGGATTATACTTCTGAATAATGGAAGGG
47	TTAGAACCTACCATATCAAAATTATTTGCACGTAAAAC
48	AGAAATAAAGAAATTGCGTAGATTTTCAGGTTTAACGT
49	CAGATGAATATACAGTAACAGTACCTTTTACATCGGGA
50	GAAACAATAACGGATTCGCCTGATTGCTTTGAATACCA
51	AGTTACAAAATCGCGCAGAGGCGAATTATTCATTTCAA
52	TTACCTGAGCAAAAGAAGATGATGAAACAAACATCAAG
53	AAAACAAAATTAATTACATTTAACAATTTCATTTGAAT
54	TACCTTTTTTAATGGAAACAGTACATAAATCAATATAT
55	GTGAGTGAATAACCTTGCTTCTGTAAATCGTCGCTATT
56	AATTAATTTTCCCTTAGAATCCTTGAAAACATAGCGAT
57	AGCTTAGATTAAGACGCTGAGAAGAGTCAATAGTGAAT
58	TTATCAAAATCATAGGTCTGAGAGACTACCTTTTTAAC
59	CTCCGGCTTAGGTTGGGTTATATAACTATATGTAAATG
60	CTGATGCAAATCCAATCGCAAGACAAAGAACGCGAGAA
61	AACTTTTTCAAATATATTTTAGTTAATTTCATCTTCTG
62	ACCTAAATTTAATGGTTTGAAATACCGACCGTGTGATA
63	AATAAGGCGTTAAATAAGAATAAACACCGGAATCATAA
64	TTACTAGAAAAAGCCTGTTTAGTATCATATGCGTTATA
65	CAAATTCTTACCAGTATAAAGCCAACGCTCAACAGTAG
66	GGCTTAATTGAGAATCGCCATATTTAACAACGCCAACA
67	TGTAATTTAGGCAGAGGCATTTTCGAGCCAGTAATAAG
68	AGAATATAAAGTACCGACAAAAGGTAAAGTAATTCTGT
69	CCAGACGACGACAATAAACAACATGTTCAGCTAATGCA

70	GAACGCGCCTGTTTATCAACAATAGATAAGTCCTGAAC
71	AAGAAAAATAATATCCCATCCTAATTTACGAGCATGTA
72	GAAACCAATCAATAATCGGCTGTCTTTCCTTATCATTC
73	CAAGAACGGGTATTAAACCAAGTACCGCACTCATCGAG
74	AACAAGCAAGCCGTTTTTATTTTCATCGTAGGAATCAT
75	TACCGCGCCCAATAGCAAGCAAATCAGATATAGAAGGC
76	TTATCCGGTATTCTAAGAACGCGAGGCGTTTTAGCGAA
77	CCTCCCGACTTGCGGGAGGTTTTGAAGCCTTAAATCAA
78	GATTAGTTGCTATTTTGCACCCAGCTACAATTTTATCC
79	TGAATCTTACCAACGCTAACGAGCGTCTTTCCAGAGCC
80	TAATTTGCCAGTTACAAAATAAACAGCCATATTATTTA
81	TCCCAATCCAAATAAGAAACGATTTTTTGTTTAACGTC
82	AAAAATGAAAATAGCAGCCTTTACAGAGAGAATAACAT
83	AAAAACAGGGAAGCGCATTAGACGGGAGAATTAACTGA
84	ACACCCTGAACAAAGTCAGAGGGTAATTGAGCGCTAAT
85	ATCAGAGAGATAACCCACAAGAATTGAGTTAAGCCCAA
86	TAATAAGAGCAAGAAACAATGAAATAGCAATAGCTATC
87	TTACCGAAGCCCTTTTTAAGAAAAGTAAGCAGATAGCC
88	GAACAAAGTTACCAGAAGGAAACCGAGGAAACGCAATA
89	ATAACGGAATACCCAAAAGAACTGGCATGATTAAGACT
90	CCTTATTACGCAGTATGTTAGCAAACGTAGAAAATACA
91	TACATAAAGGTGGCAACATATAAAAGAAACGCAAAGAC
92	ACCACGGAATAAGTTTATTTTGTCACAATCAATAGAAA
93	ATTCATATGGTTTACCAGCGCCAAAGACAAAAGGGCGA
94	CATTCAACCGATTGAGGGAGGGAAGGTAAATATTGACG
95	GAAATTATTCATTAAAGGTGAATTATCACCGTCACCGA
96	CTTGAGCCATTTGGGAATTAGAGCCAGCAAAATCACCA
97	GTAGCACCATTACCATTAGCAAGGCCGGAAACGTCACC
98	AATGAAACCATCGATAGCAGCACCGTAATCAGTAGCGA
99	CAGAATCAAGTTTGCCTTTAGCGTCAGACTGTAGCGCG
100	TTTTCATCGGCATTTTCGGTCATAGCCCCCTTATTAGC
101	GTTTGCCATCTTTTCATAATCAAAATCACCGGAACCAG
102	AGCCACCACCGGAACCGCCTCCCTCAGAGCCGCCACCC
103	TCAGAACCGCCACCCTCAGAGCCACCACCCTCAGAGCC
104	GCCACCAGAACCACCACCAGAGCCGCCGCCAGCATTGA
105	CAGGAGGTTGAGGCAGGTCAGACGATTGGCCTTGATAT
106	TCACAAACAAATAAATCCTCATTAAAGCCAGAATGGAA
107	AGCGCAGTCTCTGAATTTACCGTTCCAGTAAGCGTCAT
108	ACATGGCTTTTGATGATACAGGAGTGTACTGGTAATAA
109	GTTTTAACGGGGTCAGTGCCTTGAGTAACAGTGCCCGT
110	ATAAACAGTTAATGCCCCCTGCCTATTTCGGAACCTAT
111	TATTCTGAAACATGAAAGTATTAAGAGGCTGAGACTCC
112	TCAAGAGAAGGATTAGGATTAGCGGGGTTTTGCTCAGT
113	ACCAGGCGGATAAGTGCCGTCGAGAGGGTTGATATAAG

114	TATAGCCCGGAATAGGTGTATCACCGTACTCAGGAGGT
115	TTAGTACCGCCACCCTCAGAACCGCCACCCTCAGAACC
116	GCCACCCTCAGAGCCACCACCCTCATTTTCAGGGATAG
117	CAAGCCCAATAGGAACCCATGTACCGTAACACTGAGTT
118	TCGTCACCAGTACAAACTACAACGCCTGTAGCATTCCA
119	CAGACAGCCCTCATAGTTAGCGTAACGATCTAAAGTTT
120	TGTCGTCTTTCCAGACGTTAGTAAATGAATTTTCTGTA
121	TGGGATTTTGCTAAACAACTTTCAACAGTTTCAGCGGA
122	GTGAGAATAGAAAGGAACAACTAAAGGAATTGCGAATA
123	ATAATTTTTTCACGTTGAAAATCTCCAAAAAAAAGGCT
124	CCAAAAGGAGCCTTTAATTGTATCGGTTTATCAGCTTG
125	CTTTCGAGGTGAATTTCTTAAACAGCTTGATACCGATA
126	GTTGCGCCGACAATGACAACAACCATCGCCCACGCATA
127	ACCGATATATTCGGTCGCTGAGGCTTGCAGGGAGTTAA
128	AGGCCGCTTTTGCGGGATCGTCACCCTCAGCAGCGAAA
129	GACAGCATCGGAACGAGGGTAGCAACGGCTACAGAGGC
130	TTTGAGGACTAAAGACTTTTTCATGAGGAAGTTTCCAT
131	TAAACGGGTAAAATACGTAATGCCACTACGAAGGCACC
132	AACCTAAAACGAAAGAGGCAAAAGAATACACTAAAACA
133	CTCATCTTTGACCCCCAGCGATTATACCAAGCGCGAAA
134	CAAAGTACAACGGAGATTTGTATCATCGCCTGATAAAT
135	TGTGTCGAAATCCGCGACCTGCTCCATGTTACTTAGCC
136	GGAACGAGGCGCAGACGGTCAATCATAAGGGAACCGAA
137	CTGACCAACTTTGAAAGAGGACAGATGAACGGTGTACA
138	GACCAGGCGCATAGGCTGGCTGACCTTCATCAAGAGTA
139	ATCTTGACAAGAACCGGATATTCATTACCCAAATCAAC
140	GTAACAAAGCTGCTCATTCAGTGAATAAGGCTTGCCCT
141	GACGAGAAACACCAGAACGAGTAGTAAATTGGGCTTGA
142	GATGGTTTAATTTCAACTTTAATCATTGTGAATTACCT
143	TATGCGATTTTAAGAACTGGCTCATTATACCAGTCAGG
144	ACGTTGGGAAGAAAAATCTACGTTAATAAAACGAACTA
145	ACGGAACAACATTATTACAGGTAGAAAGATTCATCAGT
146	TGAGATTTAGGAATACCACATTCAACTAATGCAGATAC
147	ATAACGCCAAAAGGAATTACGAGGCATAGTAAGAGCAA
148	CACTATCATAACCCTCGTTTACCAGACGACGATAAAAA
149	CCAAAATAGCGAGAGGCTTTTGCAAAAGAAGTTTTGCC
150	AGAGGGGGTAATAGTAAAATGTTTAGACTGGATAGCGT
151	CCAATACTGCGGAATCGTCATAAATATTCATTGAATCC
152	CCCTCAAATGCTTTAAACAGTTCAGAAAACGAGAATGA
153	CCATAAATCAAAAATCAGGTCTTTACCCTGACTATTAT
154	AGTCAGAAGCAAAGCGGATTGCATCAAAAAGATTAAGA
155	GGAAGCCCGAAAGACTTCAAATATCGCGTTTTAATTCG
156	AGCTTCAAAGCGAACCAGACCGGAAGCAAACTCCAACA
157	GGTCAGGATTAGAGAGTACCTTTAATTGCTCCTTTTGA

158	TAAGAGGTCATTTTTGCGGATGGCTTAGAGCTTAATTG
159	CTGAATATAATGCTGTAGCTCAACATGTTTTAAATATG
160	CAACTAAAGTACGGTGTCTGGAAGTTTCATTCCATATA
161	ACAGTTGATTCCCAATTCTGCGAACGAGTAGATTTAGT
162	TTGACCATTAGATACATTTCGCAAATGGTCAATAACCT
163	GTTTAGCTATATTTTCATTTGGGGCGCGAGCTGAAAAG
164	GTGGCATCAATTCTACTAATAGTAGTAGCATTAACATC
165	CAATAAATCATACAGGCAAGGCAAAGAATTAGCAAAAT
166	TAAGCAATAAAGCCTCAGAGCATAAAGCTAAATCGGTT
167	GTACCAAAAACATTATGACCCTGTAATACTTTTGCGGG
168	AGAAGCCTTTATTTCAACGCAAGGATAAAAATTTTTAG
169	AACCCTCATATATTTTAAATGCAATGCCTGAGTAATGT
170	GTAGGTAAAGATTCAAAAGGGTGAGAAAGGCCGGAGAC
171	AGTCAAATCACCATCAATATGATATTCAACCGTTCTAG
172	CTGATAAATTAATGCCGGAGAGGGTAGCTATTTTTGAG
173	AGATCTACAAAGGCTATCAGGTCATTGCCTGAGAGTCT
174	GGAGCAAACAAGAGAATCGATGAACGGTAATCGTAAAA
175	CTAGCATGTCAATCATATGTACCCCGGTTGATAATCAG
176	AAAAGCCCCAAAAACAGGAAGATTGTATAAGCAAATAT
177	TTAAATTGTAAACGTTAATATTTTGTTAAAATTCGCAT
178	TAAATTTTTGTTAAATCAGCTCATTTTTTAACCAATAG
179	GAACGCCATCAAAAATAATTCGCGTCTGGCCTTCCTGT
180	AGCCAGCTTTCATCAACATTAAATGTGAGCGAGTAACA
181	ACCCGTCGGATTCTCCGTGGGAACAAACGGCGGATTGA
182	CCGTAATGGGATAGGTCACGTTGGTGTAGATGGGCGCA
183	TCGTAACCGTGCATCTGCCAGTTTGAGGGGACGACGAC
184	AGTATCGGCCTCAGGAAGATCGCACTCCAGCCAGCTTT
185	CCGGCACCGCTTCTGGTGCCGGAAACCAGGCAAAGCGC
186	CATTCGCCATTCAGGCTGCGCAACTGTTGGGAAGGGCG
187	ATCGGTGCGGGCCTCTTCGCTATTACGCCAGCTGGCGA
188	AAGGGGGATGTGCTGCAAGGCGATTAAGTTGGGTAACG
189	CCAGGGTTTTCCCAGTCACGACGTTGTAAAACGACGGC
190	CAGTGCCAAGCTTGCATGCCTGCAGGTCGACTCTAGAGGATCTTTT

D.2 Dumbbell bits

Dumbbell attachments as described in [51].

Bit 1: Replace oligos 26 - 32 in table D.1 with the following:

1	ACATCACTTGTCCTCTTTTGAGGAACAAGTTTTCTTGTCCTGAGTAGA
2	AGAACTCAAATCCTCTTTTGAGGAACAAGTTTTCTTGTCTATCGGCCT
3	TGCTGGTAATTCCTCTTTTGAGGAACAAGTTTTCTTGTATCCAGAACA

4	ATATTACCGCTCCTCTTTTGAGGAACAAGTTTTCTTGTCAGCCATTGC
5	AACAGGAAAATCCTCTTTTGAGGAACAAGTTTTCTTGTACGCTCATGG
6	AAATACCTACTCCTCTTTTGAGGAACAAGTTTTCTTGTATTTTGACGC
7	TCAATCGTCTTCCTCTTTTGAGGAACAAGTTTTCTTGTGAAATGGATT
8	ATTTACATTGTCCTCTTTTGAGGAACAAGTTTTCTTGTGCAGATTCAC
9	CAGTCACACGTCCTCTTTTGAGGAACAAGTTTTCTTGTACCAGTAATA
10	AAAGGGACATTCCTCTTTTGAGGAACAAGTTTTCTTGTTCTGGCCAAC
11	AGAGATAGAATCCTCTTTTGAGGAACAAGTTTTCTTGTCCCTTCTGAC
12	CTGAAAGCGTAAGAATACGTGGCACAGACAATATTTTGAATGGCT

Bit 2: Replace oligos 40 - 46 in table D.1 with the following:

1	CACTAACAACTCCTCTTTTGAGGAACAAGTTTTCTTGTTAATAGATTA
2	GAGCCGTCAATCCTCTTTTGAGGAACAAGTTTTCTTGTTAGATAATAC
3	ATTTGAGGATTCCTCTTTTGAGGAACAAGTTTTCTTGTTTAGAAGTAT
4	TAGACTTTACTCCTCTTTTGAGGAACAAGTTTTCTTGTAAACAATTCG
5	ACAACTCGTATCCTCTTTTGAGGAACAAGTTTTCTTGTTTAAATCCTT
6	TGCCCGAACGTCCTCTTTTGAGGAACAAGTTTTCTTGTTTATTAATTT
7	TAAAAGTTTGTCCTCTTTTGAGGAACAAGTTTTCTTGTAGTAACATTA
8	TCATTTTGCGTCCTCTTTTGAGGAACAAGTTTTCTTGTGAACAAAGAA
9	ACCACCAGAATCCTCTTTTGAGGAACAAGTTTTCTTGTGGAGCGGAAT
10	TATCATCATATCCTCTTTTGAGGAACAAGTTTTCTTGTTTCCTGATTA
11	TCAGATGATGTCCTCTTTTGAGGAACAAGTTTTCTTGTGCAATTCATC
12	AATATAATCCTGATTGTTTGGATTATACTTCTGAATAATGGAAGGG

Bit 3: Replace oligos 54 - 60 in table D.1 with the following:

1	TACCTTTTTTTCCTCTTTTGAGGAACAAGTTTTCTTGTAATGGAAACA
2	GTACATAAATTCCTCTTTTGAGGAACAAGTTTTCTTGTCAATATATGT
3	GAGTGAATAATCCTCTTTTGAGGAACAAGTTTTCTTGTCCTTGCTTCT
4	GTAAATCGTCTCCTCTTTTGAGGAACAAGTTTTCTTGTGCTATTAATT
5	AATTTTCCCTTCCTCTTTTGAGGAACAAGTTTTCTTGTTAGAATCCTT
6	GAAAACATAGTCCTCTTTTGAGGAACAAGTTTTCTTGTCGATAGCTTA
7	GATTAAGACGTCCTCTTTTGAGGAACAAGTTTTCTTGTCTGAGAAGAG
8	TCAATAGTGATCCTCTTTTGAGGAACAAGTTTTCTTGTATTTATCAAA
9	ATCATAGGTCTCCTCTTTTGAGGAACAAGTTTTCTTGTTGAGAGACTA
10	CCTTTTTAACTCCTCTTTTGAGGAACAAGTTTTCTTGTCTCCGGCTTA
11	GGTTGGGTTATCCTCTTTTGAGGAACAAGTTTTCTTGTTATAACTATA
12	TGTAAATGCTGATGCAAATCCAATCGCAAGACAAAGAACGCGAGAA

Bit 4: Replace oligos 68 - 74 in table D.1 with the following:

1	AGAATATAAATCCTCTTTTGAGGAACAAGTTTTCTTGTGTACCGACAA
2	AAGGTAAAGTTCCTCTTTTGAGGAACAAGTTTTCTTGTAATTCTGTCC
3	AGACGACGACTCCTCTTTTGAGGAACAAGTTTTCTTGTAATAAACAAC
4	ATGTTCAGCTTCCTCTTTTGAGGAACAAGTTTTCTTGTAATGCAGAAC
5	GCGCCTGTTTTCCTCTTTTGAGGAACAAGTTTTCTTGTATCAACAATA
6	GATAAGTCCTTCCTCTTTTGAGGAACAAGTTTTCTTGTGAACAAGAAA
7	AATAATATCCTCCTCTTTTGAGGAACAAGTTTTCTTGTCATCCTAATT
8	TACGAGCATGTCCTCTTTTGAGGAACAAGTTTTCTTGTTAGAAACCAA
9	TCAATAATCGTCCTCTTTTGAGGAACAAGTTTTCTTGTGCTGTCTTTC
10	CTTATCATTCTCCTCTTTTGAGGAACAAGTTTTCTTGTCAAGAACGGG
11	TATTAAACCATCCTCTTTTGAGGAACAAGTTTTCTTGTAGTACCGCAC
12	TCATCGAGAACAAGCAAGCCGTTTTTATTTTCATCGTAGGAATCAT

Bit 5: Replace oligos 82 - 88 in table D.1 with the following:

1	AAAAATGAAATCCTCTTTTGAGGAACAAGTTTTCTTGTATAGCAGCCT
2	TTACAGAGAGTCCTCTTTTGAGGAACAAGTTTTCTTGTAATAACATAA
3	AAACAGGGAATCCTCTTTTGAGGAACAAGTTTTCTTGTGCGCATTAGA
4	CGGGAGAATTTCCTCTTTTGAGGAACAAGTTTTCTTGTAACTGAACAC
5	CCTGAACAAATCCTCTTTTGAGGAACAAGTTTTCTTGTGTCAGAGGGT
6	AATTGAGCGCTCCTCTTTTGAGGAACAAGTTTTCTTGTTAATATCAGA
7	GAGATAACCCTCCTCTTTTGAGGAACAAGTTTTCTTGTACAAGAATTG
8	AGTTAAGCCCTCCTCTTTTGAGGAACAAGTTTTCTTGTAATAATAAGA
9	GCAAGAAACATCCTCTTTTGAGGAACAAGTTTTCTTGTATGAAATAGC
10	AATAGCTATCTCCTCTTTTGAGGAACAAGTTTTCTTGTTTACCGAAGC
11	CCTTTTTAAGTCCTCTTTTGAGGAACAAGTTTTCTTGTAAAAGTAAGC
12	AGATAGCCGAACAAAGTTACCAGAAGGAAACCGAGGAAACGCAATA

D.3 4HB anchor end

Replace oligos 181-190 in table D.1 with the following:

1	ACCCGTCGGATTCTCCGTGGGAACAAA
2	CCAAGCTTGCATTCTTCGCTAACCAGGCAAAG
3	GTTGGGTTTTTTTTTTTAACGCCAGGGTTTTC
4	CGCCATTCGCCACGTAATGGGTCGACTCTAGA
5	TCGGTGCGGGCCGCCTGCAGGATAGGTCACGT
6	GGCGAAAGGGGGTTGTAAAATAACCGTGCATC
7	TGGTGTAGATGGGTGCCGGAATTACGCCAGCT
8	TGCCAGTTTGAGGCACTCCACAAGGCGATTAA
9	GCACCGCTTCTGGCGCATCGCGACGGCCAGTG

10	TTTTTCGGCCTCAGGAAGATCGGGACGACGACAGTATTTTTT
11	CGGCGGATTGACTTCAGGCTTGGGAAGGGCGA
12	CCAGTCACGACGATGTGCTGGCCAGCTTTCCG

D.4 1234 barcode

Unpublished design by Dr Jinbo Zhu, personal communication.

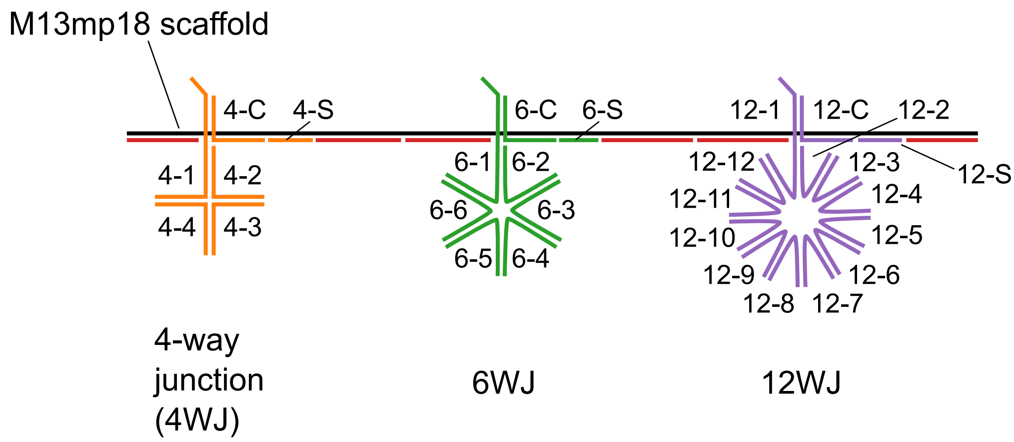


Fig. D.1

'1' modification: Consists of 11 dumbbell structures, identical to bit 1 in the Bell & Keyser barcode system [51], see section D.2 for strand sequences.

'2' modification: a 4-way junction is separately assembled from the following strands, see Figure D.1:

4-1	GGATGTGTGCCATAGTGGATTGCGGCTGAACCTTCGAGTGCTGTATGAGAGGTGAGTGATG
4-2	GGTTCAGCCGCAATCCTCGCCTGCACTCTACC
4-3	GGTAGAGTGCAGGCGATGAGCACGAGTCTTGC
4-4	GCAAGACTCGTGCTCAACTATGGCACACATCC

The assembled structure is then attached to the carrier by replacing oligo 43 in table D.1 with the following:

4-C	CACCTCTCATACAGCACTCGTTTAATTTTAAAAGTTTGAGTA
4-S	ACATTATCATTTTGCGGA

'3' modification: a 6-way junction is separately assembled from the following strands, see Figure D.1:

6-1	GCTAAGATGGTTCAGCCGCAATTCAGACAGCCACTCTT CCATCATACGTTTGAGGATGTGTAGGTTGAGGTGTAGTG
6-2	CGTATGATGGAAGAGTGGCTGTCTTGAGCACGAGTCTTGCTATTCTCC
6-3	GGAGAATAGCAAGACTCGTGCTCACCGAATGCCACCACGCTCCTGACC
6-4	GGTCAGGAGCGTGGTGGCATTCGGCGTCCAGCTCTGATCCGATACTGC
6-5	GCAGTATCGGATCAGAGCTGGACGACTATGGCACACATCCCTACAACG
6-6	CGTTGTAGGGATGTGTGCCATAGTGAATTGCGGCTGAACCATCTTAGC

The assembled structure is then attached to the carrier by replacing oligo 57 in table D.1 with the following:

6-C	CCTCAACCTACACATCCTCATTAGCTTAGATTAAGACGCTGA
6-S	GAAGAGTCAATAGTGAAT

'4' modification: a 12-way junction is separately assembled from the following strands, see Figure D.1:

12-1	GGCACAGCTATAATAACGCAATCCTCTCCGGCCTCAAA CTACTTTACCTTGTAGGATAGGAAGATATGAGGAGTGAG
12-2	GGTAAAGTAGTTTGAGGCCGGAGACCGAATGGAGTCTGTTCTCGACGC
12-3	GCGTCGAGAACAGACTCCATTCGGACAATTACGAACCAACTTAGGACC
12-4	GGTCCTAAGTTGGTTCGTAATTGTGGTCATCGTGGCGTACCATATACC
12-5	GGTATATGGTACGCCACGATGACCTCTTCGATCTACCCGATAGGCTCC
12-6	GGAGCCTATCGGGTAGATCGAAGACGTACAGGTGTGACTTGAATTTGC
12-7	GCAAATTCAAGTCACACCTGTACGAGTGTTAGAATACAACAAGCGACC
12-8	GGTCGCTTGTTGTATTCTAACACTGCATCTCATACGGCAGTATCCGCC
12-9	GGCGGATACTGCCGTATGAGATGCTGAGCACGGAACTGTCAACCTTGC
12-10	GCAAGGTTGACAGTTCCGTGCTCACGTTCATTAAGATAAATCTGATCC
12-11	GGATCAGATTTATCTTAATGAACGACTATGCCTGCTACATGCACTTCC
12-12	GGAAGTGCATGTAGCAGGCATAGTGGATTGCGTTATTATAGCTGTGCC

The assembled structure is then attached to the carrier by replacing oligo 71 in table D.1 with the following:

12-C	CTCATATCTTCCTATCCTACTTAAGAAAAATAATATCCCATC
12-S	CTAATTTACGAGCATGTA

Appendix E

Solution viscosities

Values taken from [138].

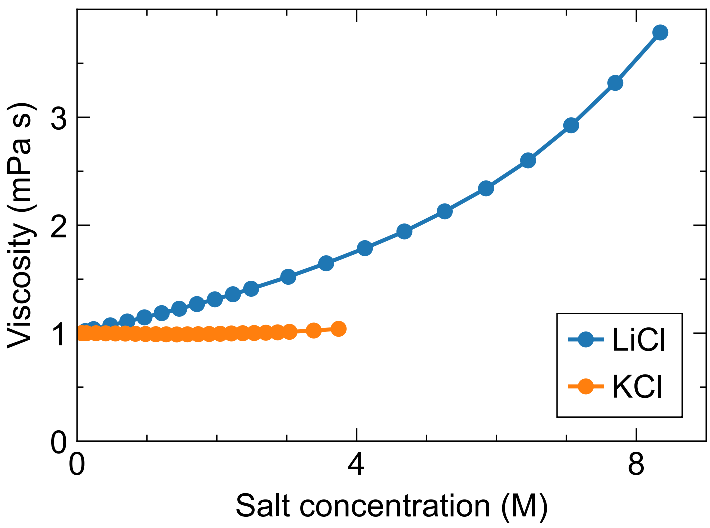


Fig. E.1

Appendix F

Dextran sample events

Figure reproduced from [1, SI] with the permission of AIP Publishing. At negative voltage, electro-osmotic flow (EOF) drives uncharged dextran molecules into the nanopore, producing drops in the current. Switching the voltage to 600 mV inverts the flow, carrying dextran back out of the pore.

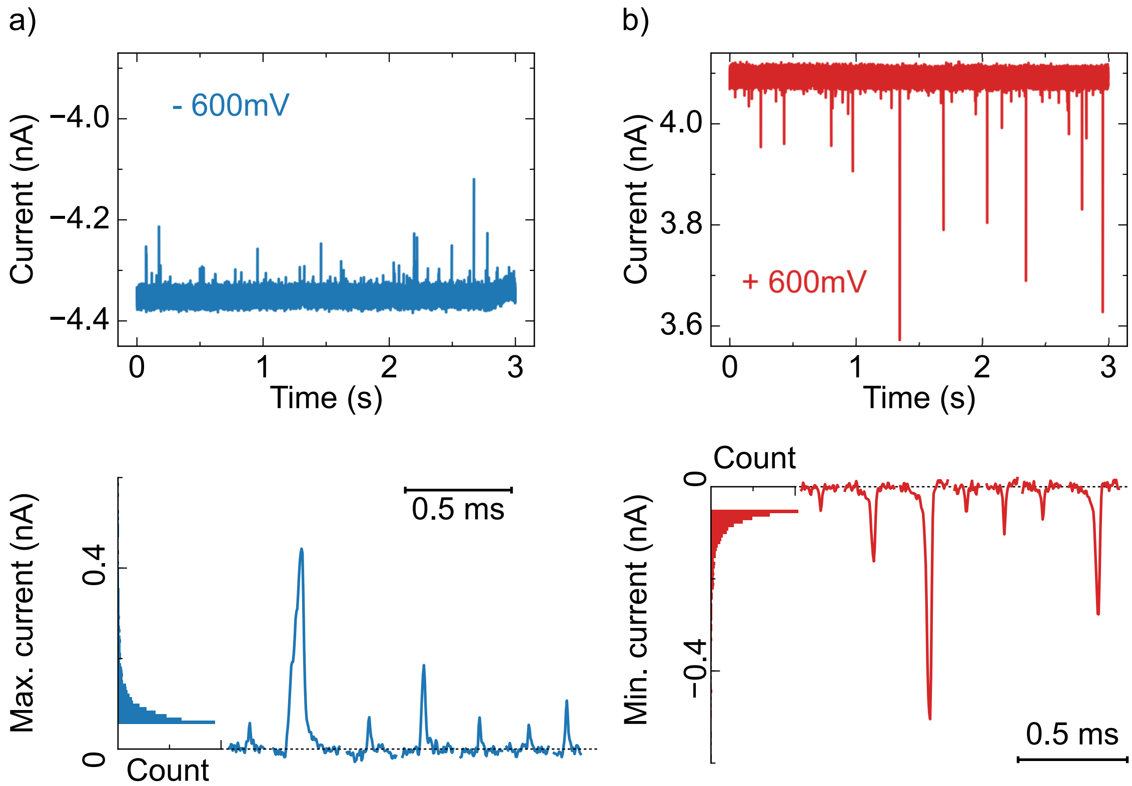


Fig. F.1

Appendix G

Anchor DNA sample events

Sample current traces from translocations of anchor DNA structures in 4 M, 2 M and 1 M LiCl at pH 8. Peaks at either end of the translocation indicate the direction of entry. Figure adapted from [1, SI] with the permission of AIP Publishing.

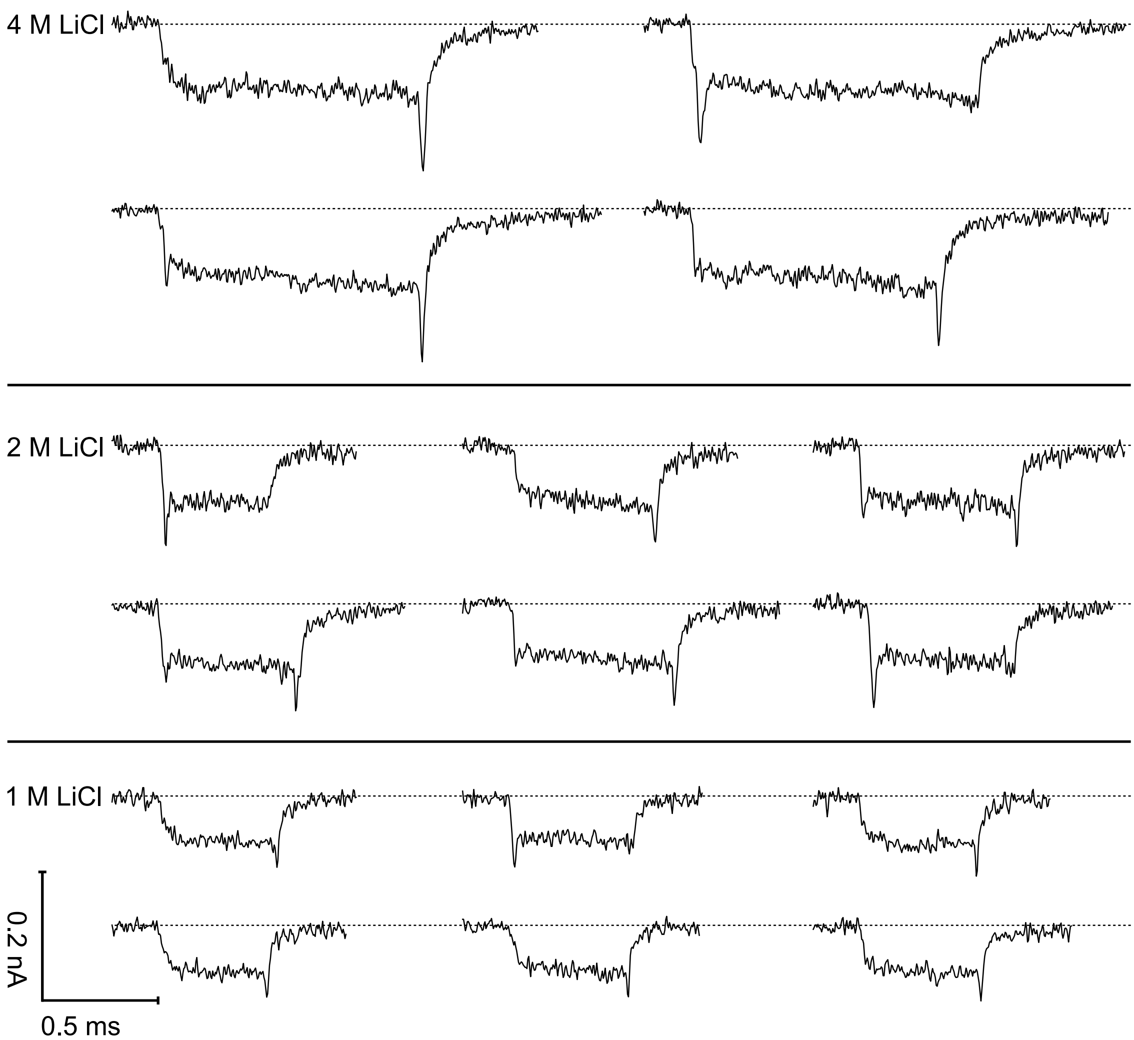


Fig. G.1

Appendix H

Bayesian folding analysis

Sample events which reach the threshold of $L_1 + 0.5 * (L_2 - L_1)$ (horizontal blue line) between the current levels for one and two DNA strands in the pore but which were correctly classified as unfolded by Bayesian model comparison.

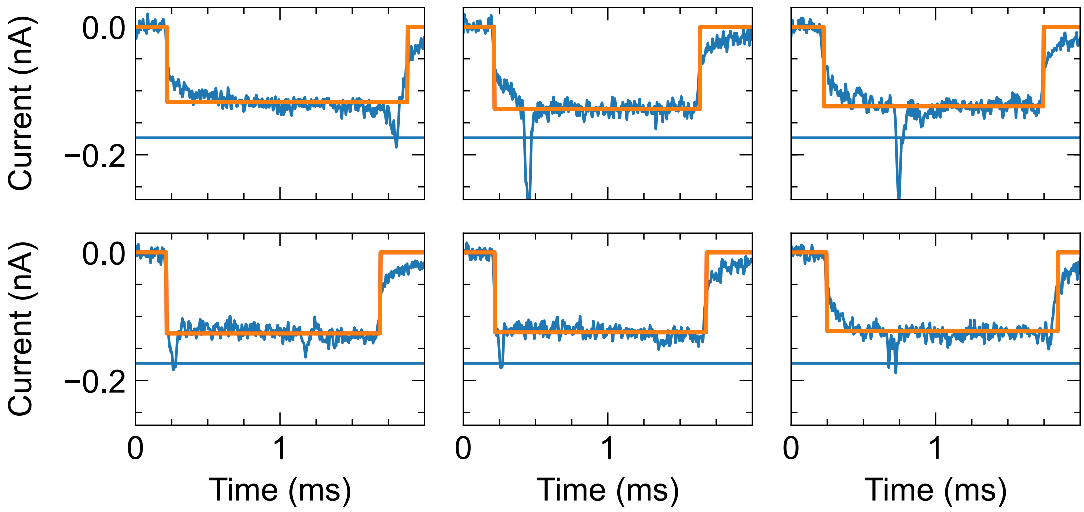


Fig. H.1

Appendix I

Baseline identification for 56-bit design

This section has been adapted from Ermann et al. *[2].*

In order to analyse peaks in the intra-event current we need to first define a baseline relative to which peaks will be measured. Figure I.1 shows the same example event as in the main text. The green shaded region indicates the baseline, the following outlines how it is defined. Programmatic notation is used where $\bar{v}[i:j]$ refers to the slice of vector $\bar{v}$ between indices i and j. Assignments of the form $A = B$ work from right to left, i.e. assign the value B to the variable A. C refers to a cut-off current level given by $L_0 - 0.75 * (L_0 - L_1)$ where L_0 is the unimpeded pore current and L_1 the current with one double strand of DNA in the pore. y denotes the vector of current values, t the time points.

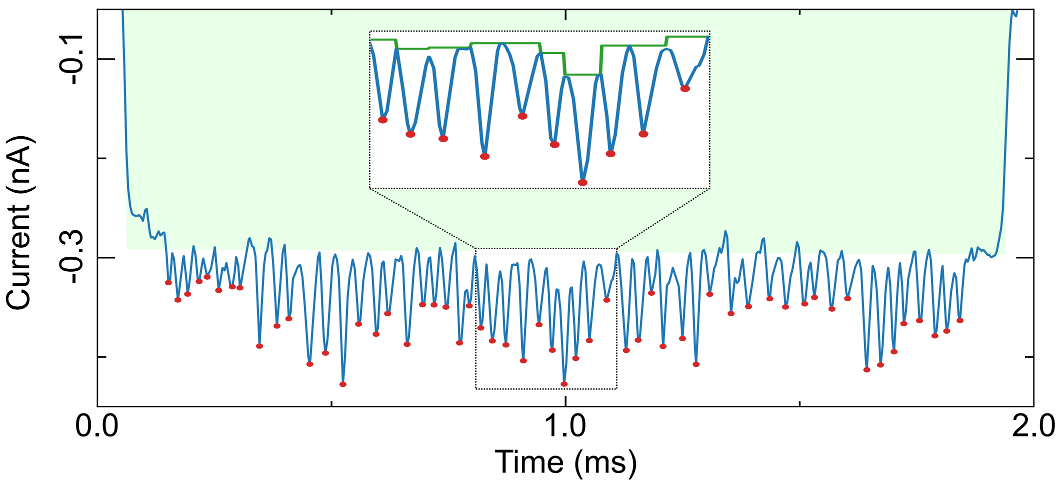


Fig. I.1

1. We first remove the previously identified peaks marked with red dots by setting the current values in each subregion found to be part of a peak to the most positive value

within that subregion. This results in a new current trace $\hat{y}$ with the peak section replaced by piecewise constant values as shown by the green line in the inset in Figure I.1.

2. On the modified current trace $\hat{y}$ we determine two index values a and b which define the first and last positions at which $\hat{y} < C$, where C is the cut-off value calculated initially. We then fit a linear function $f(t,s,i) = i + s * t$ to the region $\hat{y}[a:b]$, obtaining least-squares fit values $\tilde{i}$ and $\tilde{s}$.
3. On the original current trace y, we again determine two index values a and b corresponding to the first and last positions at which $y < C$. We then replace the slice between these indices with values calculated from the function $f(t,\tilde{s},\tilde{i})$ from the previous step, i.e. if the new current trace $\tilde{y}$ is initially the same as y, we assign $\tilde{y}[a:b] = f(t[a:b],\tilde{i},\tilde{s})$ to obtain a new current trace $\tilde{y}$ that contains a linear intra-event baseline between the start and end of the event without taking into account peaks. This trace, shown by the shaded green region in Figure I.1, serves as the baseline from which we calculate peak amplitudes.

Appendix J

Time between events with and without ping-pong

The following geometric series will be used in this section:

$$\text{For } x \neq 1, \quad \sum_{n=0}^{N-1} x^n = \frac{1-x^N}{1-x} \tag{J.1}$$

$$\text{For } |x| < 1, \quad \sum_{n=0}^{\infty} x^n = \frac{1}{1-x} \tag{J.2}$$

$$\text{and} \quad \sum_{n=1}^{\infty} n x^{n-1} = \frac{1}{(1-x)^2} \tag{J.3}$$

Ping-pong iterations increase the fraction of usable events by selectively re-measuring difficult-to-read translocations, however they delay the measurement by repeatedly inverting the voltage for a total time Δt_{pp}. This raises the question whether ping-pong in fact improves the overall frequency of usable events, i.e. those which can be decoded successfully. Equivalently to the frequency we can calculate the expected value of the arrival time difference between consecutive usable events, $\mathbb{E}[\Delta t]$, which makes the derivation more tractable. Figure J.1 a) shows the paths which an event can follow during translocation without ping-pong. With a probability P_{dec} it is successfully decoded on the first attempt. This outcome is associated with a time $\overline{\Delta t} = {}^{1}/{\bar{f}}$ where $\bar{f}$ is the mean frequency of all forward translocations into the pore, not only usable ones. If decoding fails with probability $(1 - P_{\mathrm{dec}})$, another molecule arrives after $\overline{\Delta t}$, bringing the total wait to $2 \cdot \overline{\Delta t}$. We can construct the expected value of the total wait by summing up the probability of each successful outcome multiplied by the associated waiting time, given by

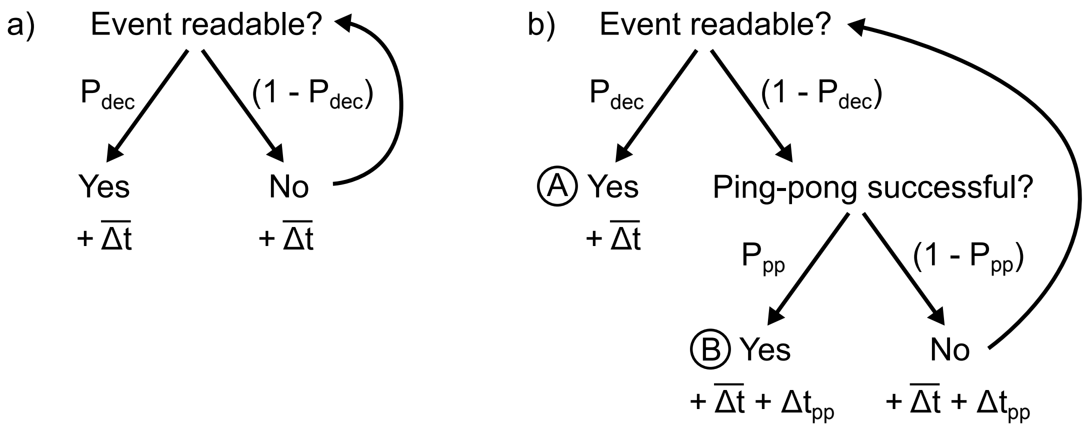


Fig. J.1 Possible paths to a successful signal read. **a)** shows the case without ping-pong, where an event has a chance P_{dec} of being analysed successfully. Each failed read adds a wait $\overline{\Delta t}$ for the next translocation. With ping-pong in part **b)**, failed analysis triggers re-measuring of the molecule. Ping-pong itself has a chance P_{pp} of producing a successful read, if it fails the cycle restarts with a wait for the next event but having already invested a time $\overline{\Delta t} + \Delta t_{\text{pp}}$.

$$
\begin{aligned}
\underset{\text{no ping-pong}}{\mathbb{E}}[\Delta t] &= \overline{\Delta t} \cdot P_{\text{dec}} \\
&+ 2 \cdot \overline{\Delta t} \cdot (1 - P_{\text{dec}}) \cdot P_{\text{dec}} \\
&+ 3 \cdot \overline{\Delta t} \cdot (1 - P_{\text{dec}})^2 \cdot P_{\text{dec}} + \dots \\
&= \overline{\Delta t} \cdot P_{\text{dec}} \cdot \sum_{n=1}^{\infty} n(1 - P_{\text{dec}})^{n-1} \\
&\overset{\text{using J.3}}{=} \frac{\overline{\Delta t}}{P_{\text{dec}}}
\end{aligned}
$$

which recovers the intuitive result of dividing the mean time between events by the probability of a successful read, i.e. the less easily signals can be decoded the longer we have to wait for an analysable translocation. Ping-pong changes the possible outcomes as shown in Figure Figure J.1 b). The successful decoding branch remains the same, a failure to read now leads to remeasuring the molecule. Before deriving the expected waiting time in the ping-pong case we need to calculate the probability P_{pp} of multiple recaptures leading to a successful read. Figure J.2 shows the possible outcomes: each ping-pong iteration has a chance P_{rc} to recapture the molecule, followed by the decoding probability P_{dec}. After N iterations, P_{pp} is given by

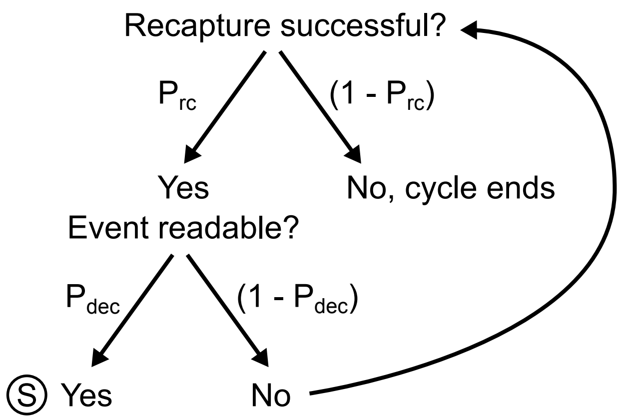


Fig. J.2 Possible outcomes of multiple ping-pong iterations. A molecule is recaptured with a probability P_{rc}, followed by a chance P_{dec} of obtaining a successful read. If the read fails the procedure can be repeated, 'losing' the molecule in the bulk ends the cycle. The probability of a successful read is the sum of the outcome 'S' across all iterations.

$$P_{\mathrm{pp}}(N, P_{\mathrm{rc}}, P_{\mathrm{dec}}) = P_{\mathrm{rc}} \cdot P_{\mathrm{dec}} \cdot \Big(1 + P_{\mathrm{rc}}(1-P_{\mathrm{dec}}) + P_{\mathrm{rc}}^2(1-P_{\mathrm{dec}})^2 + \cdots + P_{\mathrm{rc}}^{N-1}(1-P_{\mathrm{dec}})^{N-1}\Big) \tag{J.4}$$

$$= P_{\mathrm{rc}} \cdot P_{\mathrm{dec}} \cdot \sum_{n=0}^{N-1} (P_{\mathrm{rc}}(1-P_{\mathrm{dec}}))^n \overset{\text{using J.1}}{=} P_{\mathrm{rc}} \cdot P_{\mathrm{dec}} \cdot \frac{1-(P_{\mathrm{rc}}(1-P_{\mathrm{dec}}))^N}{1-P_{\mathrm{rc}}+P_{\mathrm{rc}}P_{\mathrm{dec}}} \tag{J.5}$$

Figure J.3 shows the asymptotic behaviour of the ping-pong success rate towards a maximum value. The number of iterations N should therefore be set to a number at which P_{pp} begins to saturate and further improvements are marginal.

Returning to the waiting time between readable events with ping-pong, the expected value is again derived by considering the successful outcomes and summing up their probabilities multiplied by the waits. This results in the following, where lines J.7 and J.8 correspond to successful outcomes 'A' and 'B' in Figure J.1 b) respectively:

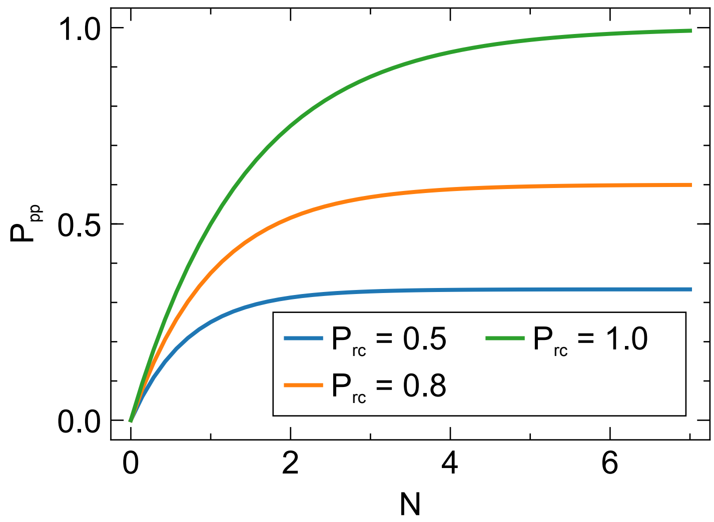


Fig. J.3 Probability P_{pp} of N ping-pong iterations leading to a successful molecule read for a decoding probability $P_{\text{dec}} = 0.5$. Higher recapture success rates P_{rc} increase P_{pp} at all N values. The asymptotic behaviour towards a maximum given by equation J.2 suggests setting the number of iterations to a value beyond which improvements become marginal, here $N \approx 4$.

$$\text{With } F = (1-P_{\text{pp}})(1-P_{\text{dec}}) \tag{J.6}$$

$$\underset{\text{ping-pong}}{\mathbb{E}}[\Delta t] = \overline{\Delta t}\cdot P_{\text{dec}} + (2\overline{\Delta t}+\Delta t_{\text{pp}})\cdot P_{\text{dec}}\cdot F + (3\overline{\Delta t}+2\Delta t_{\text{pp}})\cdot P_{\text{dec}}\cdot F^2 + \dots \tag{J.7}$$

$$+ (\overline{\Delta t}+\Delta t_{\text{pp}})\cdot(1-P_{\text{dec}})P_{\text{pp}} + 2(\overline{\Delta t}+\Delta t_{\text{pp}})\cdot(1-P_{\text{dec}})P_{\text{pp}}\cdot F + \dots \tag{J.8}$$

$$= \underbrace{\overline{\Delta t}\cdot P_{\text{dec}} + 2\cdot\overline{\Delta t}\cdot P_{\text{dec}}\cdot F + 3\cdot\overline{\Delta t}\cdot P_{\text{dec}}\cdot F^2 + \dots}_{\overline{\Delta t}\cdot P_{\text{dec}}\cdot\sum_{n=1}^{\infty} nF^{n-1}} \tag{J.9}$$

$$+ \underbrace{\Delta t_{\text{pp}}\cdot P_{\text{dec}}\cdot F + 2\cdot\Delta t_{\text{pp}}\cdot P_{\text{dec}}\cdot F^2 + \dots}_{\Delta t_{\text{pp}}\cdot P_{\text{dec}}\cdot F\cdot\sum_{n=1}^{\infty} nF^{n-1}} \tag{J.10}$$

$$+ \underbrace{(\overline{\Delta t}+\Delta t_{\text{pp}})\cdot(1-P_{\text{dec}})P_{\text{pp}} + 2(\overline{\Delta t}+\Delta t_{\text{pp}})\cdot(1-P_{\text{dec}})P_{\text{pp}}\cdot F + \dots}_{(\overline{\Delta t}+\Delta t_{\text{pp}})\cdot(1-P_{\text{dec}})P_{\text{pp}}\sum_{n=1}^{\infty} nF^{n-1}} \tag{J.11}$$

$$\overset{\text{using J.3}}{=} \frac{\overline{\Delta t}\cdot P_{\text{dec}} + \Delta t_{\text{pp}}\cdot P_{\text{dec}}\cdot F + (\overline{\Delta t}+\Delta t_{\text{pp}})\cdot(1-P_{\text{dec}})P_{\text{pp}}}{(1-F)^2} \tag{J.12}$$

$$= \frac{\overline{\Delta t}\left(P_{\text{dec}} + P_{\text{pp}} - P_{\text{dec}}P_{\text{pp}}\right) + \Delta t_{\text{pp}}\left(P_{\text{pp}} - P_{\text{dec}}P_{\text{pp}} + P_{\text{dec}}\cdot F\right)}{(1-F)^2} \tag{J.13}$$

The expected time between events in the ping-pong case therefore depends on the mean waiting time $\overline{\Delta t}$, the decoding probability P_{dec}, the time added for each set of ping-pong iterations Δt_{pp} and the likelihood of successfully reading an event during these iterations P_{pp}.

Inserting the result from J.5 replaces P_{pp} with a dependency on the number of ping-pong iterations N and the recapture success rate of a single ping-pong operation P_{rc}. Figure 6.9 in the main text compares the expected waits with and without ping-pong to assess under which conditions selectively re-measuring molecules reduces the overall measurement time.